Learning Infinite-Horizon Average-Reward Restless Multi-Action Bandits via Index Awareness

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the online restless bandits with average-reward and multiple actions, where the state of each arm evolves according to a Markov chain, and the reward of pulling an arm depends on both the current state of the corresponding Markov chain and the action taken. Since finding the optimal control is typically intractable for restless bandits, existing learning algorithms are often computationally expensive or with a regret bound that is exponential in the number of arms and states. In this paper, we advocate *index-aware reinforcement learning* (RL) solutions to design RL algorithms operating on a much smaller dimensional subspace by exploiting the inherent structure in restless bandits. Specifically, we first propose novel index policies to address dimensionality concerns, which are provably optimal. We then leverage the indices to develop two low-complexity index-aware RL algorithms, namely, (i) GM-R2MAB, which has access to a generative model; and (ii) UC-R2MAB, which learns the model using an upper confidence style online exploitation method. We prove that both algorithms achieve a sub-linear regret that is only polynomial in the number of arms and states. A key differentiator between our algorithms and existing ones stems from the fact that our RL algorithms contain a novel exploitation that leverages our proposed provably optimal index policies for decision-makings.

1 Introduction

Restless multi-armed bandits (RMAB) [59] have been used to model a variety of sequential decision making problems such as congestion control [7, 6], job scheduling [47, 31, 12, 61], wireless communication [52, 19, 9, 18, 64], healthcare [21, 13, 40, 43, 44, 36], queueing systems [25, 29, 5, 15, 38, 37], and cloud computing [16, 48]. In RMAB, there is a collection of N arms, each of which is endowed with a state that evolves independently accordingly to a two-action Markov Decision Process (MDP) [51]. If the arm is "pulled" or "activated" at any moment in time, then it advances stochastically according to one transition kernel, and if not, then it advances according to a different kernel. Rewards are generated with each transition, and the goal is to maximize the expected total reward over an infinite horizon, subject to a constraint on the number of arms activated at any moment in time.

A critical limitation of classical RMAB frameworks is that only two actions, either pulled or not pulled, are allowed for each arm. This is restrictive since the decision maker in many applications often has access to multiple actions for each arm. To this end, we consider an under-examined generalization of RMAB that allows for multiple actions per arm with different degrees of costs, which we call *the restless multi-action bandits*, dubbed as R2MAB, in the infinite-horizon average-reward setting. Our objective is to develop simple reinforcement learning (RL) algorithms to solve R2MAB without knowing the underlying MDPs associated with each arm. Although RMAB or R2MAB has found its success in many applications as aforementioned, the fundamental theoretical understanding of how to develop *low computation-complexity* and *order-of-optimal regret* RL algorithms, two of the most

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

important performance metrics for online restless bandits, remains in its infancy so far. In particular, three important aspects of RL algorithms for RMAB or R2MAB deserve special attentions:

▷ First, most upper confidence bound (UCB) based policies [41, 55, 42, 54] for RMAB often leverage *a heuristic policy* in the exploitation phase, e.g., constantly pulling one arm, which does not have any performance guarantee. Hence, these policies *may not perform close to the offline optimum*. This is partially due to the fact that finding an optimal solution for RMAB is PSPACE-hard [50], hence infeasible. The fundamental challenge lies in the explosion of state space and the curse of dimensionality prevents computing optimal policies. Though Whittle index policy [59] is a powerful tool to address the state space explosion, finding the Whittle index is typically intractable [47].

▷ Second, existing RL algorithms with theoretical guarantee of a sub-linear regret upper bound, e.g., colored-UCRL2 [49], suffers from *an exponential computation complexity*, and the regret bound is *exponential in the number of states and arms*. This is due to the fact that it needs to solve Bellman equations with an exponentially large space set. Likewise, the class of Thompson sampling based algorithms [34, 33] provide theoretical guarantees in the Bayesian setting, where the updates can be computationally expensive, especially when the likelihood functions are complex.

▷ Third, although the design of low computation-complexity solutions for online RMAB has been gaining attentions, e.g., [23, 8, 14, 35, 57, 60], many challenges remain unsolved. For instance, [23, 8, 14, 35] lacked finite-time performance analysis and these multi-timescale stochastic approximation algorithms often suffer from slow convergence. [60] focused on the finite-horizon setting, which makes their approach not directly applicable to ours since the average reward setting studied in this paper is more challenging to analyze compared to the finite-horizon setting, which necessitates different proof techniques. [57] achieved low-complexity policy but is constrained to a specific birth-death Markovian model and not easy to be directly generalized.

The lack of a fundamental understanding on how to design efficient RL algorithms for R2MAB in the infinite-horizon average-reward setting that consider the above three aspects in terms of provably optimal policy, computation complexity, exponential pre-factor in the regret bound, and sub-linear regret performance guarantees, motivates us to fill the gap by advocating **index-aware RL solutions** in this paper. Specifically, our index-aware RL solutions operate on a much smaller dimensional subspace by exploiting the inherent structure encoded in R2MAB problems. This requires us to first design low-complexity provably optimal index policies for R2MAB, and then RL algorithms that leverage the structure of provably optimal index policies in the exploitation phase for decision-makings so as to reduce the high computational complexity and exponential factor in regret analysis. Our main contributions in this paper are summarized as follows:

• To address the dimensionality concerns in R2MAB, we first develop low-complexity index policies for R2MAB when the underlying MDPs are known. In contrast to the celebrated Whittle index policy, which requires the notoriously difficult-to-verify indexability condition, we bypass the task of deriving index policies by taking a more general linear programming (LP) approach, and hence is computationally efficient. We show that our proposed index policy is asymptotically optimal.

• We develop two index-aware RL algorithms for infinite-horizon average-reward R2MAB, namely, (i) GM-R2MAB, a generative model based approach that obtains samples initially then creates the model; and (ii) UC-R2MAB, an online approach wherein the model is updated as samples are obtained. Both algorithms follow a two-stage pattern of model construction and an index policy based solution. The algorithms solve an extended linear programming (ELP) problem from which we construct the provably optimal index policies to be executed during the exploitation phase in both algorithms. The key differentiator between our index-aware RL solutions and aforementioned state of the arts stems from two perspectives: (a) our index-aware RL solutions are computationally appealing since our index based solutions are merely based on solving an ELP, which is exponentially better than state-of-the-art methods such as colored-UCRL2; (b) our index-aware RL solutions contain a novel exploitation phase by leveraging our proposed provably optimal index policy for decision-makings, rather than using a heuristic one or black-box oracle in aforementioned existing algorithms.

• We provide the first-ever regret analysis for infinite-horizon average-reward R2MAB. We show that the above key differentiators in the design of GM-R2MAB and UC-R2MAB, contribute to their $\tilde{\mathcal{O}}(\sqrt{T})$ regret, which is only polynomial in the number of arms and states. It is worth noting that GM-R2MAB and UC-R2MAB achieve low computation complexity and a sub-linear $\tilde{\mathcal{O}}(\sqrt{T})$ regret with a polynomial pre-factor *all at once*, while none of the aforementioned state of the arts achieve so.

2 Model Description and Problem Formulation

We consider the restless multi-action bandits (R2MAB) problem in continuous time. There are a total of N arms. Each arm $n \in \mathcal{N}$ is described by a MDP $(\mathcal{S}, \mathcal{A}, P_n, r_n)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P_n(s'|s,a)$ indicates the probability of reaching state s' by taking action a in state s, and $r_n(s,a)$ is the reward of each state-action pair (s,a). We assume that $\mathcal{S}$ and $\mathcal{A}$ are finite sets with cardinalities S and A, respectively. Our results and analysis will still apply when each arm has its own sets of states and actions; so, without loss of generality, we will simply assume that all arms share the same state and action sets. We consider the unichain MDPs in this paper, which in fact is known to be necessary to guarantee the existence of stationary policies for the infinite-horizon average-reward R2MAB (and MDPs) regardless of initial states [51, 3]. We refer to action $a = 0$ as *passive action* and any action $a > 0$ as an *active action*. Moreover, using the standard terminology from the restless bandits literature [59], we call an arm *active* when an active action is applied to it and *passive* otherwise. A cost is incurred when action a is applied to arm n in state s. For abuse of notation, we denote the cost as a itself. We assume that the maximum cost to activating arms at any moment in time is B, which we call the *activation budget*.

The decision-making scenario is as follows. At any moment in time t, each arm can be either active or passive. When action $a_n(t) = a$ is applied to arm n in state $s_n(t) = s$, it takes an exponentially distributed amount of time to transition to state s' with rate $P_n(s,a,s')$, $\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$. Decision epochs/time are defined as the moments when the state of an arm changes. A *policy* determines what actions to be applied to each arm at each decision epoch, with the restriction that at most B activation budgets can be used for activating arms. After receiving an action, each arm incurs an immediate reward $r_n(t) = r_n(s,a)$, which is a random variable with support $[0,1]$ and mean $\bar{r}_n(s,a)$. Without loss of generality, we further assume that only active arms yield rewards, i.e., $r_n(s,0) = 0, \forall s \in \mathcal{S}$.

Let Π be the set of all possible policies for the considered R2MAB problem and π is a feasible policy in Π, satisfying $\pi \in \Pi : \mathcal{F}_t \mapsto \mathcal{A}^N$, where $\mathcal{F}_t$ is the sigma-algebra [53] generated by random variables $\{s_n(h), a_n(h) : \forall n \in \mathcal{N}, h \leq t\}$. The objective of the decision maker is to maximize the expected long-term average reward of activating arms subject to the activation budget constraint, i.e.,

$$\texttt{R2MAB:} \quad \max_{\pi \in \Pi} \ \liminf_{T \to \infty} \frac{1}{T} \mathbb{E}_\pi \left(\int_{t=0}^{T} \sum_{n=1}^{N} r_n(t) dt \right), \quad \text{subject to} \sum_{n=1}^{N} a_n(t) \leq B, \quad \forall t. \tag{1}$$

When the underlying MDP (i.e., transition kernel and reward function) of each arm is known, the R2MAB problem (1) in theory can be solved using relative value iteration [51, 11]. Unfortunately, this approach suffers from the curse of dimensionality [10, 11], and lack of insights for the solution structure. When the underlying MDPs are not known, off-the-shelf RL algorithms are either computationally expensive or without an $\tilde{\mathcal{O}}(\sqrt{T})$ regret guarantee. In the remaining of the paper, we first build a low-complexity index based policy for (1) with optimality guarantee when the MDPs are known in Section 3, and then design index-aware RL algorithms for (1) that not only are computationally efficient but also achieve an $\tilde{\mathcal{O}}(\sqrt{T})$ regret when the MDPs are unknown in Section 4.

3 An Index-based Policy and Asymptotic Optimality

Rather than solving (1) exactly, we instead construct an index based policy for the original R2MAB (1) that we prove to be asymptotically optimal. To describe our index policy design in Section 3.1, we first need to introduce the following linear programming (LP), in which the decision variables are the occupancy measures [3] of the controlled MDP processes:

$$\texttt{LP}(P_n, r_n, \forall n): \quad \max_{\Omega_\pi} \sum_{n=1}^{N} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \omega_n(s,a) \bar{r}_n(s,a) \tag{2}$$

$$\text{subject to} \sum_{n=1}^{N} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} a \omega_n(s,a) \leq B, \tag{3}$$

$$\sum_{a} \omega_n(s,a) = \sum_{s'} \sum_{a'} \omega_n(s',a') P_n(s',a',s), \quad \forall n \in \mathcal{N}, \tag{4}$$

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \omega_n(s,a) = 1, \;\; \omega_n(s,a) \geq 0, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, a \in \mathcal{A}. \quad (5)$$

One can arrive at the LP (2)-(5) by replacing all random variables in the relaxed version of (1) with their *expected values* via introducing a new set of variables $\omega_n(s,a)$, which are called *occupancy measures* [3] of the controlled MDP corresponding to arm n. The description of reducing (1) to the LP (2)-(5) is provided in supplementary materials. Specifically, the occupancy measure Ω_π of a stationary policy π for the N controlled infinite-horizon MDPs is defined as the expected average number of visits to a state-action pair (s,a), i.e.,

$$\Omega_\pi = \left\{ \omega_n(s,a) \triangleq \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi \left(\int_{t=1}^{T} \mathbb{1}(s_n(t)=s, a_n(t)=a)dt \right) : n \in \mathcal{N}, s \in \mathcal{S}, a \in \mathcal{A} \right\}.$$

It can be easily checked that the occupancy measure satisfies $\sum_{(s,a)} \omega_n(s,a) = 1$, and hence $\omega_n, \forall n \in \mathcal{N}$ is a probability measure. To this end, (4) represents the fluid transition of the occupancy measure, which holds due to the ergodic theorem [17] for finite MDPs [56]; and (5) follows the fact that the occupancy measure is a probability measure. Thus the feasible set of LP (2)-(5) is non-empty. We denote the optimal solution to the LP (2)-(5) as $\Omega_{\pi^*} = \{\omega_n^*(s,a) : n \in \mathcal{N}, s \in \mathcal{S}, a \in \mathcal{A}\}$, and the corresponding optimal value as $V^* := \sum_{n=1}^{N} \sum_{(s,a)} \omega_n^*(s,a) r_n(s,a)$, which serves as an upper bound on the reward of the original `R2MAB` (1).

Lemma 1. *The optimal value achieved by the LP (2)-(5) is an upper bound of that of the* `R2MAB` (1).

3.1 An Index Based Policy

Unfortunately, the solution to the LP (2)-(5) does not always provide a feasible decision to the original `R2MAB` (1). This is because the activation budget constraint in (1) must be met at all time, instead of just in the average sense as in (3). Exacerbating this problem is the fact that the average constraint may be violated severely during the decision epochs, resulting in poor policy performance. We overcome these challenges by introducing a computationally appealing index policy that we prove to be asymptotically optimal. Specifically, our index policy is derived from the optimal solutions $\Omega_{\pi^*} = \{\omega_n^*(s,a)\}$ and the index assigned to arm n in state $s_n(t) = s$ at time t is defined as

$$\mathcal{I}_n(s) := \sum_{a\in\mathcal{A}} \frac{\omega_n^*(s,a)\bar{r}_n(s,a)}{\sum_{a'\in\mathcal{A}} \omega_n^*(s,a')}, \quad (6)$$

where $\xi_n(s,a) \triangleq \frac{\omega_n^*(s,a)}{\sum_{a'\in\mathcal{A}} \omega_n^*(s,a')}$ represents the probability of applying action $a_n(t) = a$ to arm n in state $s_n(t) = s$ at time t [3] when $\sum_{a'\in\mathcal{A}} \omega_n^*(s,a') > 0$, and otherwise arm n can be simply made passive. Hence, the index of arm n in (6) represents the expected obtained reward of activating arm n in state s. To this end, we rank all arms based on their indices (6) in a non-increasing order, and activate the set of highest indexed arms, denoted as $\mathcal{N}(t) \subset \mathcal{N}$, such that the corresponding activation cost of arms in $\mathcal{N}(t)$ is within the activation budget B, i.e., $\sum_{n\in\mathcal{N}(t)} a_n^\star(s_n(t)) \leq B$. Here $a_n^\star(s_n(t))$ is the action for arm n, which is determined according to the probability $\xi_n(s,a)$ based on its current state $s_n(t) = s$ at time t. When multiple arms share the same indices, we randomly activate one arm and allocate the remaining activation costs across all possible actions according to the probability $\xi_n(s,a)$. All remaining arms are kept passive in case they have zero indices. We call our index based policy, `ERC`, since it *ranks arms by **E**xpected **R**eward and pull arms constrained by activation **C**ost.* We denote the resultant index based policy as $\pi^\star = \{\pi_n^\star, n \in \mathcal{N}\}$.

Remark 1. *Unlike Whittle-based index policies [59, 28, 25, 65, 35], our* `ERC` *does not require the indexability condition, which is often hard to establish especially when the transition kernel of the underlying MDP is convoluted [47]. Like Whittle policies, our* `ERC` *is computationally efficient since it is merely based on solving a LP. We remark that [56] also considered average-reward* `R2MAB`*, however, only one action can be chosen deterministically for each state with no difference in activation cost. Hence it cannot be generalized to ours since we consider a randomized policy for each state with heterogeneous activation costs across different actions. Finally, another line of works [30, 62, 63, 60] designed index policies without indexability requirement for finite-horizon restless bandits, and hence cannot be directly applied to our infinite-horizon average-reward* `R2MAB`*.*

3.2 Asymptotic Optimality

We now show that ERC is asymptotically optimal in the same asymptotic regime as that in Whittle [59] and many others [58, 56, 65], where all N arms are generalized to N classes, and both the number of class-n arms and the activation budget B are scaled by ρ, with their ratio holding constant. Denote $X_n^\rho(\pi^\star, s, a; t)$ as the number of class-n arms at state s taking action a at time t under ERC $\pi^\star$. We will be interested in the this fluid-scaling process with parameter ρ, and define the expected long-term average reward as $V_{\pi^\star}^\rho := \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}_{\pi^\star}(\int_{t=1}^T \sum_{n=1}^N \sum_{(s,a)} r_n(s,a)\frac{X_n^\rho(\pi^\star,s,a;t)}{\rho}dt)$. Our ERC $\pi^\star$ is asymptotically optimal only when $V_{\pi^\star}^\rho \geq V_\pi^\rho, \forall \pi \in \Pi$. Before presenting our main result in this section, we first state the following technical condition called "global attractor" [58].

Definition 1. *An equilibrium point $X^{\rho,\star}/\rho$ under ERC $\pi^\star$ is a global attractor for the process $X^\rho(\pi^\star;t)/\rho$, if, for any initial point $X^\rho(\pi^\star;0)/\rho$, the process $X^\rho(\pi^\star;t)/\rho$ converges to $X^{\rho,\star}/\rho$.*

Remark 2. *The global attractor indicates that all trajectories converge to $X^{\rho,\star}$. Though it may be difficult to establish analytically that a fixed point is a global attractor for the process [56], such assumption has been made in [58, 28, 56, 65, 22] and is only verified numerically. Our experimental results in supplementary materials show that such convergence indeed occurs for our ERC $\pi^\star$.*

Theorem 1. *Our ERC $\pi^\star$ is asymptotically optimal under Definition 1, i.e., $\lim_{\rho\to\infty} V_{\pi^\star}^\rho - V_{\pi^{opt}}^\rho = 0$, where π^{opt} represents the optimal policy for the original R2MAB (1).*

4 Index-aware Reinforcement Learning Solutions

We now consider to learn the R2MAB (1) when the transition kernel P_n and reward function r_n, $\forall n \in \mathcal{N}$ are unknown. Our goal is to provide low-complexity index-aware RL algorithms and determine their finite-time performance measured by the regret [39], which is defined as follows:

Definition 2. *The regret of policy π is defined as the expected gap between the offline optimum, i.e., the best policy under which both the transition kernels and reward functions are known, and the cumulative reward of the arm selecting algorithm, i.e., Reg$(\pi, T) := T\mu^{opt} - \mathbb{E}_\pi[R(\pi,T)]$, where $\mu(\pi) := \lim_{T\to\infty}\frac{1}{T}\mathbb{E}[R(\pi,T)] = \lim_{T\to\infty}\frac{1}{T}\mathbb{E}[\sum_{t=1}^T\sum_{n=1}^N r_n(t)]$ is the expected average reward under policy π, and $\mu^{opt} := \sup_\pi \mu(\pi)$ is the optimal average reward, which is independent of the initial state for MDPs with finite diameter [51].*

4.1 GM-R2MAB: Generative Model Based Index-Aware Reinforcement Learning

We first introduce a generative model based R2MAB learning algorithm called GM-R2MAB, which contains two phases: (i) the exploration phase (lines 1-3 in Algorithm 1) and the exploitation phase (lines 4-6 in Algorithm 1). During the exploration phase, GM-R2MAB samples each state-action pair (s,a) for $J(T)$ times ($J(T)$ to be specified later) for each arm n, counts the number of times $T_n(s,a,s')$ for each transition to the next state s', and constructs an empirical model of transition kernel and reward function, denoted by $\hat{P}_n(s'|s,a) = \frac{T_n(s,a,s')}{J(T)}$ and $\hat{r}_n(s,a) = \frac{1}{J(T)}\sum_{h=1}^{J(T)} r_n(s,a;h)\mathbb{1}(s_n(h) = s, a_n(h) = a)$, $\forall (s,a,s'), n \in \mathcal{N}$, respectively. Using these empirical estimates, GM-R2MAB creates a set of plausible MDPs such that the transition kernels and reward functions are close to the true ones, which are defined as

$$\mathcal{M} = \{M_n = (\mathcal{S}, \mathcal{A}, \tilde{P}_n, \tilde{r}_n) : |\tilde{P}_n(s'|s,a) - \hat{P}_n(s'|s,a)| \leq \delta, \tilde{r}_n(s,a) = \hat{r}_n(s,a) + \delta, \forall n, s, a\}, \quad (7)$$

where $\delta = \sqrt{\frac{1}{2J(T)}\log\frac{2SANJ(T)}{\eta}}$ for $\eta \in (0,1)$ is built using the Hoeffding inequality [45].

In the exploitation phase, GM-R2MAB computes the ERC index policy $\pi^\star$ by performing optimistic planning. In other words, GM-R2MAB selects an optimistic transition kernel, an optimistic reward function, and an optimistic policy to maximize the objective function while satisfying the constraints. More specifically, it can be expressed as the following optimization problem,

$$(\tilde{P} = \{\tilde{P}_n, \forall n\}, \pi^\star) = \arg\max_{M_n \in \mathcal{M}} \ \texttt{LP}(\tilde{P}_n, \tilde{r}_n, \forall n). \quad (8)$$

To solve (8), GM-R2MAB uses **Extended LP (ELP)** by leveraging the state-action-state occupancy measure $z_n(s,a,s')$ defined as $z_n(s,a,s') = P_n(s'|s,a)\omega_n(s,a)$ to express the confidence intervals of transition probabilities. The description of **ELP** is provided in supplementary materials. Once we

Algorithm 1 GM-R2MAB

Require: Time horizon T, learning function $J(T) < T$.
1: **for** $n = 1, 2, ..., N$ **do**
2: Observe arm n until there are $J(T)$ visits of pairs $(s_n(t) = s, a_n(t) = a), \forall s \in \mathcal{S}, a \in \mathcal{A}$.
3: **end for**
4: Construct the set of plausible MDPs $\mathcal{M}$ as in (7);
5: Compute the corresponding ERC index policy $\pi^\star$ by solving **ELP**($\mathcal{M}$);
6: Execute $\pi^\star$ to the end.

compute $\{z_n^\star, \forall n\}$, the probabilities $\xi_n(s, a)$ and hence the indices $\mathcal{I}_n(s)$ in (6) are recovered from the computed occupancy measures as $\mathcal{I}_n(s) := \frac{\sum_{s' \in \mathcal{S}} z_n^\star(s,a,s')\bar{r}_n(s,a)}{\sum_{b \in \mathcal{A}, s' \in \mathcal{S}} z_n^\star(s,b,s')}$, from which we can construct our ERC index policy $\pi^\star$, and execute $\pi^\star$ to the end.

4.1.1 Regret Analysis of GM-R2MAB

To characterize the regret, we first introduce the definition of ergodicity coefficient [4, 2]. For ease of readability, we denote the state for all arms as a stacked vector $\mathbf{s} \in \mathcal{S}^N := [s_1, s_2, \ldots, s_N]$, the corresponding actions under policy $\pi^\star$ as $\pi^\star(\mathbf{s})$, and the unknown MDPs as $\Theta := [\theta_1, \theta_2, \ldots, \theta_N]$ with $\theta_n := (P_n, r_n)$. The transition kernel of the stacked system is then $P_\Theta(\cdot|\mathbf{s}, \pi^\star(\mathbf{s})), \forall \mathbf{s} \in \mathcal{S}^N$.

Definition 3. $D_{P_\Theta} := 1 - \min_{\mathbf{s},\mathbf{s}'} \sum_{\mathbf{z} \in \mathcal{S}^N} \min\{P_\Theta(\mathbf{z}|\mathbf{s}, \pi^\star(\mathbf{s})), P_\Theta(\mathbf{z}|\mathbf{s}', \pi^\star(\mathbf{s}'))\}$ *is defined as the ergodicity coefficient of* P_Θ*, and* $D := \sup_\Theta D_{P_\Theta}$ *as the maximum value.*

Theorem 2. *The regret of GM-R2MAB with* $J(T) = \mathcal{O}(T^{1/2})$ *satisfies:*

$$\textit{Reg}(\pi^\star, T) = \mathcal{O}\left(\sqrt{T}\left(SAB + \frac{BN}{1-D}\sqrt{\log\frac{4SANT}{\eta}}\right)\right). \tag{9}$$

The regret comes from the explortation and exploitation phases in GM-R2MAB, respectively. Specifically, the first term $\mathcal{O}(SAB\sqrt{T})$ in (9) is the worst regret from explorations of each state-action pair under the generative model with $J(T)$ time steps for sampling, and the second term comes from the policy execution phase of GM-R2MAB due to MDP model mismatch. The proof of Theorem 2 differs from the traditional analysis framework of generative based RL for restless bandits [57, 60], particularly in the way to track regrets due to model mismatch. We leverage the form of Bellman equations of long-term average MDPs and transfer the regret to the difference of relative value functions. This enables us to track the regret relying only on the first moment behavior of transition kernels. However, [57] depended on higher order moment behavior of transition kernels to track regrets. The higher moment behavior is typically hard to analyze for a general MDP other than the birth-death process considered in [57]. [60] tracked regrets by leveraging the optimistic properties of a linear system, which is only applicable to the finite-horizon setting considered in [60].

Remark 3. *We emphasize that although GM-R2MAB has a similar form as the "explore-then-commit" policy [24], a key differentiator between our GM-R2MAB and state-of-the-art methods stems from two perspectives. First, GM-R2MAB has access to a generative model and samples are taken initially to estimate a model. As a result, GM-R2MAB only needs to solve an ELP once to construct a policy, which is computationally efficient. This is in contrast to the state-of-the-art colored-UCRL2 [49], which needs to solve a recursive Bellman equation to derive the policy. One contribution here is to determine the right choice of* $J(T)$*, which plays a key role in balancing the tradeoff between model accuracy and complexity. Second, GM-R2MAB executes our proposed provably optimal ERC index policy in the exploitation phase rather than using a heuristic one as in state of the arts [41, 55, 42, 54]. This contributes to the polynomial prefactor in the regret, compared to an exponential one in colored-UCRL2. Finally, we note that such a generative model based approach has also been used in the context of CMDPs [26, 27] and restless bandits [57, 60]. However, they considered either a finite-time [26, 60] or a discounted setting [27], and hence cannot be directly applied to the infinite-horizon average-reward setting studied in this paper. Furthermore, [26, 27] focused on the sample complexity analysis, which is not directly translatable to the regret [20]. Though Restless-UCB [57] operated in a similar manner as our GM-R2MAB, it depends on the performance of an offline "black-boxed" oracle approximator in the exploitation phase, while our GM-R2MAB leverages an explicit and provably optimal ERC index policy.*

Algorithm 2 `UC-R2MAB`

Require: Initialize $C_n^0(s,a)=0$, and $\hat{P}_n^0(s'|s,a)=1/S, \forall n\in\mathcal{N}, s,s'\in\mathcal{S}, a\in\mathcal{A}$.

1: **for** $k=1,2,\cdots,K$ **do**
2: Construct the set of plausible MDPs $\mathcal{M}^k$ as in (10);
3: Compute the corresponding `ERC` index policy $\pi^{\star,k}$ by solving **ELP**($\mathcal{M}^k$);
4: Execute $\pi^{\star,k}$ in the current episode.
5: **end for**

4.2 `UC-R2MAB`: Online Index-Aware Reinforcement Learning

The `GM-R2MAB` approach operates in an "offline" manner in the sense that it first estimates a model by sampling every state-action pair in the system for a certain number of times, and then computes a policy to be executed throughout the exploitation phase. Unfortunately, such an "offline" approach may not be feasible in many real-world applications since some states may not be reachable without executing the policy. To this end, we further develop an "online" approach, dubbed as `UC-R2MAB`, via interleaving the process of collecting samples from the environments and model updates, which is summarized in Algorithm 2. Specifically, `UC-R2MAB` operates in an episodic manner where each episode consists of H consecutive frames. Let K be the total number of episodes until time T, hence we have $T=KH$. Denote the k-th episode as $\mathcal{H}_k$ and let τ_k be the time when it starts.

At the beginning of each episode, i.e., $\tau_k, \forall k$, `UC-R2MAB` estimates the true transition kernel and the true reward by the corresponding empirical averages as $\hat{P}_n^k(s'|s,a)=\frac{C_n^{k-1}(s,a,s')}{\max\{C_n^{k-1}(s,a),1\}}$, $\hat{r}_n^k(s,a)=\frac{1}{\max\{C_n^{k-1}(s,a),1\}}\sum_{\tau=1}^{k-1}\sum_{h=1}^{H} r_n(s,a)\mathbb{1}(s_n^\tau(h)=s, a_n^\tau(h)=a)$, where $C_n^{k-1}(s,a)$ is the number of visits to state-action pairs (s,a) until τ_k, and $C_n^{k-1}(s,a,s')$ is the number of transitions from s to s' under action a, satisfying $C_n^k(s,a)=C_n^{k-1}(s,a)+\sum_{h=1}^{H}\mathbb{1}(s_n^k(h)=s, a_n^k(h)=a)$, and $C_n^k(s,a,s')=C_n^{k-1}(s,a,s')+\sum_{h=1}^{H}\mathbb{1}(s_n^k(h+1)=s'|s_n^k(h)=s, a_n^k(h)=a), \forall(s,a)\in\mathcal{S}\times\mathcal{A}$ and $\forall(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}, \forall n$, where $s_n^k(h)$ is the state of arm n at the h-th time frame in episode k.

Similar to `GM-R2MAB`, `UC-R2MAB` creates a set of plausible MDPs using these empirical estimates for each episode as in (7) by replacing $J(T)$ with $C_n^{k-1}(s,a)$ for each arm n. Thus for $\eta\in(0,1)$, the set of plausible MDPs in the k-th episode is defined as

$$\mathcal{M}^k=\{M_n^k=(\mathcal{S},\mathcal{A},\tilde{P}_n^k,\tilde{r}_n^k):|\tilde{P}_n^k(s'|s,a)-\hat{P}_n^k(s'|s,a)|\leq\delta_n^k(s,a), \tilde{r}_n^k(s,a)=\hat{r}_n^k(s,a)+\delta_n^k(s,a)\}. \quad (10)$$

where $\delta_n^k(s,a)=\sqrt{\frac{1}{2C_n^{k-1}(s,a)}\log\frac{4SAN(k-1)H}{\eta}}$ is built using the Hoeffding inequality [45].

Once constructing the model, `UC-R2MAB` computes the `ERC` index policy $\pi^{\star,k}$ in episode k by solving an ELP, which is described in supplementary materials. Specifically, it solves the following optimization problem,

$$(\tilde{P}^k=\{\tilde{P}_n^k,\forall n\}, \pi^{\star,k})=\arg\max_{M_n^k\in\mathcal{M}^k} \texttt{LP}(\tilde{P}_n^k, \tilde{r}_n^k, \forall n), \quad (11)$$

from which it recovers the indices $\mathcal{I}_n^k(s)$ to construct `ERC` and then execute the index policy to the end of this episode. This is the same problem as in (8) except for substituting $\mathcal{M}$ with $\mathcal{M}^k$.

This algorithm draws inspiration from the infinite-horizon algorithm UCRL [32], which uses the sampled trajectory of each episode to update the plausible MDPs of next episode. However, there exists two major differences. First, `UC-R2MAB` updates the models at the beginning of each episode, which allows for faster model construction as [26]. Second, `UC-R2MAB` deploys the proposed `ERC` at each episode, and thus results in solving a low-complexity ELP, which is exponentially better than that of UCRL-based algorithm [32] that needs to solve extended value iterations.

4.2.1 Regret Analysis of `UC-R2MAB`

Theorem 3. *The regret of `UC-R2MAB` satisfies:*

$$Reg(\{\pi^{\star,k},\forall k\},T)=\mathcal{O}\left(\sqrt{T}\left(\frac{1}{2}+\frac{2\sqrt{2}B\sqrt{N}}{1-D}\sqrt{\log\frac{4SANT}{\eta}}\right)\right), \quad (12)$$

Though the design of UC-R2MAB is inspired by UCRL type algorithms, the regret analysis of UC-R2MAB differs from colored-UCRL2 [49], a state-of-the-art method for online restless bandits. The major difference comes from the fact that UC-R2MAB leverages the relative value function of Bellman equation for long-term average MDPs to track regrets caused by model mismatch as GM-R2MAB. There are also recent Thompson sampling based results [2] on characterizing Bayesian regret for RMAB while using techniques reminiscent of UC-R2MAB. The sub-linear regret in [2] was achieved by upper bounding the number of episodes to be $\sqrt{T}$ under a specific requirement for the length of each episode. In contrast, UC-R2MAB directly bounds the sum of regret from each episode without such a episode length constraint.

Remark 4. *Similar to the state-of-the-art colored-UCRL2 [49] and Thompson sampling based algorithms [33, 34, 2], our UC-R2MAB also operates in an episodic manner and achieves an $\tilde{\mathcal{O}}(\sqrt{T})$ regret. However, the color-UCRL2 is known to be computationally expensive and the regret is exponentially in the number of arms and states. Likewise, Thompson sampling based methods provide theoretical guarantee in the Bayesian settings and need to implement a computationally expensive method to update the posterior beliefs due to the complex likelihood functions. In contrast, our UC-R2MAB is computationally appealing since it only needs to solve an ELP in each episode, from which UC-R2MAB constructs and implements our proposed ERC index policy. The regret of UC-R2MAB is polynomial in the number of arms and states, which is exponentially better than that of colored-UCRL2. Finally, we remark that the multiplicative "pre-factor" that goes with the time dependent function in the regret of UC-R2MAB is smaller than that of GM-R2MAB. This is due to the fact that UC-R2MAB operates in an episodic manner while GM-R2MAB only samples once initially. This leads to a higher computation complexity of UC-R2MAB, which is $\mathcal{O}(NSAK)$ compared to that of GM-R2MAB, which is $\mathcal{O}(NSA)$.*

5 Experiments

In this section, we present some of our experimental results. We also demonstrate the utility of our GM-R2MAB and UC-R2MAB by evaluating them under two real-world applications of restless bandits.

5.1 Experiments on Constructed Instance

Instance construction. We consider a setup with 10 classes of arms. The state space is $\mathcal{S} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Class-$n$ arm arrives with rate $3n$ for $n = 1, \cdots, 10$, and departs with a fixed rate of $\mu = 20$. Since Whittle index policy is only designed for RMAB with two actions, we evaluate our algorithms under two settings: (1) two-action setting; and (2) multi-action setting. For two-action setting, we consider a controlled Markov process in which states evolve as a specific birth-and-death process, i.e., state s only transit to $s+1$ or $s-1$ with probability $P(s, s+1) = \lambda/(\lambda+\mu)$ and $P(s, s-1) = \mu/(\lambda+\mu)$. Class-$n$ arm generates a random reward $r_n(s) \sim Ber(sp_n)$, with p_n uniformly sampled from $[0.01, 0.1]$. The activation budget is set to $0.3N$, where we vary the number of arms N from 50 to 350. For multi-action setting, we consider a total of $3, 5, 10$ actions, and the state of arm n transition to any state in $\mathcal{S}$ with a randomized non-zero transition probability. We only present results with 10 actions and provide other results in supplementary materials.

Baselines. We compare our ERC index policy with Whittle index policy (Whittle) [59], a priority policy (Priority) [56] and a lagrangian based policy (Lagrangian) [36]. We compare our RL solutions GM-R2MAB and UC-R2MAB with a Q-learning based policy (MAIQL) [35], a Thompson sampling policy (TS) [34], Restless-UCB [57], and a set of learning Whittle index based policy including Fu [23], AB [8] and NeurWin [46]. Note that we do not include results of colored-UCRL2 in our comparions since it is known to suffer from a high computation complexity and is outperformed by existing policies, e.g., Restless-UCB [57]. For ease of readability, we only present results of two "online" baselines for comparisons. More experimental results and the parameter settings for baselines are provided in supplementary materials.

Asymptotic optimality. We compare the rewards obtained by an index policy with that from the theoretical upper bound obtained by solving the LP (2)-(5). We call the ratio between this award difference and the number of arms as *optimality gap*. From Figures 1 and 2, we observe that all policies are asymptotically optimal, which is consistent with their theoretical performance guarantees. Our ERC slightly outperforms these baselines in terms of the vanishing speed of optimality gap.

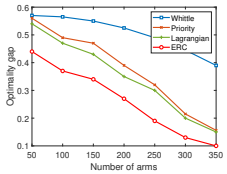

Figure 1: Optimality gap for two actions.

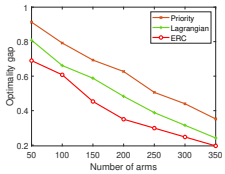

Figure 2: Optimality gap for 10 actions.

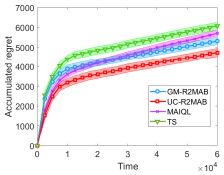

Figure 3: Regret for two actions.

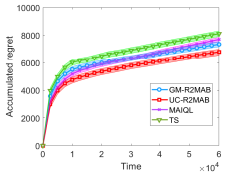

Figure 4: Regret for 10 actions.

Regret and running time. The accumulated regrets are presented in Figures 3 and 4, where we use the Monte Carlo simulation with $1,000$ independent trials of a single-threaded program on AMD Ryzen 5800x desktop with 64GB RAM. For simplicity, we choose 200 arms and a time horizon of $T = 60,000$ slots. Each episode consists of $2,500$ slots. We observe that `UC-R2MAB` achieves the lowest cumulative regret and is better than our "offline" method `GM-R2MAB`. This is consistent with our theoretical analysis (see Remark 4). A key observation is that for a large horizon T, our "offline" `GM-R2MAB` even outperforms the "online" MAIQL. This is because `GM-R2MAB` leverages our proposed provably optimal `ERC` index policy in the exploitation phase. We also compare the average running time of these algorithms. For two-action (10-action) setting, the average running time of `GM-R2MAB`, `UC-R2MAB`, MAIQL and TS is 86s (144s), 308s (607s), 348s (702s) and 359s (681s), respectively. It is clear that our `GM-R2MAB` and `UC-R2MAB` is more efficient in running time. These improvements come from the intrinsic design of our algorithms that merely need to solve an ELP, while TS needs to solve Bellman equations. More results are provided in supplementary materials.

5.2 Experiments on Real-World Datasets

We demonstrate the utility of `GM-R2MAB` and `UC-R2MAB` by evaluating them under two recently studied applications of restless bandits: wireless scheduling with two actions, and tuberculosis care with multiple actions. Due to space constraints, we relegate the detailed descriptions of these two problems to supplementary materials.

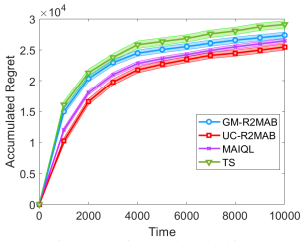

Figure 5: Scheduling.

Wireless scheduling over fading channels [1]. A wireless client is modeled as an arm, which has some data to transmit. Each arm suffers from 1 unit holding cost in each time slot until the data is transmitted. The quality of wireless channel, either good or bad, via which data is transmitted determines the amount of transmitted data and varies over time. The goal is to maximize the negative of total holding cost. We adopt the settings in [1], where 1 out of 10 arms is activated at any moment in time. The regret is shown in Figure 5. It is observed that `UC-R2MAB` outperforms all baselines. Importantly, `UC-R2MAB` has a sub-linear regret guarantee while the best-performing baseline MAIQL lacks of finite-time analysis. Since the environment is dynamically changing over time, it may be hard to build a perfect simulator and hence `UC-R2MAB` is preferable than the offline `GM-R2MAB`.

Tuberculosis care in India [36]. A health worker takes three actions on 200 patients to improve their adherence over a course of 6 months. Each action has varying cost and effective: cheap (call the patient), semi-expensive (visit the patient) and very expensive (escalate the patient). The goal is to maximize patients' adherence subject to a daily time budget due to the limited worker time and resources. We adopt the settings in [36] with the budget being 20 and the reward defined as adherence level$/3$. From Figure 6, we again observe that `UC-R2MAB` achieves a sub-linear regret and outperforms all baselines.

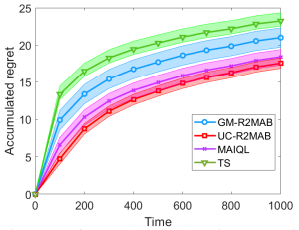

Figure 6: Tuberculosis care in India.

6 Conclusion

In this paper, we developed two low-complexity index-aware RL algorithms, `GM-R2MAB` and `UC-R2MAB` for online infinite-horizon average-reward restless multi-action bandits. We proved that both algorithms achieved a sub-linear regret that is only polynomial in the number of arms and states. Our key design to reduce both the computation complexity and exponential factor in regret analysis is via exploiting the inherent structure encoded in restless bandits and leveraging our proposed provably optimal index policies for decision-makings.

References

[1] Samuli Aalto, Pasi Lassila, and Prajwal Osti. Whittle index approach to size-aware scheduling with time-varying channels. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 57–69, 2015.

[2] Nima Akbarzadeh and Aditya Mahajan. On learning whittle index policy for restless bandits with scalable regret. *arXiv preprint arXiv:2202.03463*, 2022.

[3] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[4] Aristotle Arapostathis, Vivek S Borkar, Emmanuel Fernández-Gaucherand, Mrinal K Ghosh, and Steven I Marcus. Discrete-time controlled markov processes with average cost criterion: A survey. *SIAM Journal on Control and Optimization*, 31(2):282–344, 1993.

[5] Thomas W Archibald, DP Black, and Kevin D Glazebrook. Indexability and index heuristics for a simple class of inventory routing problems. *Operations research*, 57(2):314–326, 2009.

[6] K Avrachenkov, VS Borkar, and S Pattathil. Controlling g-aimd using index policy. In *The 56th IEEE Conference on Decision and Control, Melbourne, December*, pages 12–15, 2017.

[7] Konstantin Avrachenkov, Urtzi Ayesta, Josu Doncel, and Peter Jacko. Congestion control of tcp flows in internet routers by means of index policy. *Computer Networks*, 57(17):3463–3478, 2013.

[8] Konstantin E Avrachenkov and Vivek S Borkar. Whittle index based q-learning for restless bandits with average reward. *Automatica*, 139:110186, 2022.

[9] Saeed Bagheri and Anna Scaglione. The restless multi-armed bandit formulation of the cognitive compressive sensing problem. *IEEE Transactions on Signal Processing*, 63(5):1183–1198, 2015.

[10] Richard Bellman. *Dynamic Programming*. Princeton University Press, USA, 2010.

[11] Dimitri P Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific Belmont, MA, 1995.

[12] Dimitris Bertsimas and José Niño-Mora. Restless Bandits, Linear Programming Relaxations, and A Primal-Dual Index Heuristic. *Operations Research*, 48(1):80–90, 2000.

[13] Biswarup Bhattacharya. Restless bandits visiting villages: A preliminary study on distributing public health services. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–8, 2018.

[14] Arpita Biswas, Gaurav Aggarwal, Pradeep Varakantham, and Milind Tambe. Learn to intervene: An adaptive learning policy for restless bandits in application to preventive healthcare. In *Proc. of IJCAI*, 2021.

[15] Vivek S Borkar and Sarath Pattathil. Whittle indexability in egalitarian processor sharing systems. *Annals of Operations Research*, pages 1–21, 2017.

[16] Vivek S Borkar, K Ravikumar, and Krishnakant Saboo. An index policy for dynamic pricing in cloud computing under price commitments. *Applicationes Mathematicae*, 44:215–245, 2017.

[17] Erhan Cinlar. Introduction to stochastic processes prentice-hall. *Englewood Cliffs, New Jersey (420p)*, 1975.

[18] Kobi Cohen, Qing Zhao, and Anna Scaglione. Restless multi-armed bandits under time-varying activation constraints for dynamic spectrum access. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 1575–1578. IEEE, 2014.

[19] Wenhan Dai, Yi Gai, Bhaskar Krishnamachari, and Qing Zhao. The Non-Bayesian Restless Multi-Armed Bandit: A Case of Near-Logarithmic Regret. In *Proc. of IEEE ICASSP*, 2011.

[20] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[21] Sarang Deo, Seyed Iravani, Tingting Jiang, Karen Smilowitz, and Stephen Samuelson. Improving Health Outcomes Through Better Capacity Allocation in A Community-based Chronic Care Model. *Operations Research*, 61(6):1277–1294, 2013.

[22] Santiago Duran and Ina Maria Verloop. Asymptotic optimal control of markov-modulated restless bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–25, 2018.

[23] Jing Fu, Yoni Nazarathy, Sarat Moka, and Peter G Taylor. Towards q-learning the whittle index for restless bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*, pages 249–254. IEEE, 2019.

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

[25] Kevin D Glazebrook, David J Hodge, and Chris Kirkbride. General notions of indexability for queueing control and asset management. *The Annals of Applied Probability*, 21(3):876–907, 2011.

[26] Aria HasanzadeZonuzy, Dileep Kalathil, and Srinivas Shakkottai. Learning with Safety Constraints: Sample Complexity of Reinforcement Learning for Constrained MDPs. In *Proc. of AAAI*, 2021.

[27] Aria HasanzadeZonuzy, Dileep Kalathil, and Srinivas Shakkottai. Model-based reinforcement learning for infinite-horizon discounted constrained markov decision processes. In *Proc. of IJCAI*, 2021.

[28] David J Hodge and Kevin D Glazebrook. On the asymptotic optimality of greedy index heuristics for multi-action restless bandits. *Advances in Applied Probability*, 47(3):652–667, 2015.

[29] DJ Hodge and Kevin D Glazebrook. Dynamic Resource Allocation In A Multi-Product Make-to-Stock Production System. *Queueing Systems*, 67(4):333–364, 2011.

[30] Weici Hu and Peter Frazier. An Asymptotically Optimal Index Policy for Finite-Horizon Restless Bandits. *arXiv preprint arXiv:1707.00205*, 2017.

[31] Peter Jacko. Restless Bandits Approach to the Job Scheduling Problem and Its Extensions. *Modern trends in controlled stochastic processes: theory and applications*, pages 248–267, 2010.

[32] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-Optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 11(4), 2010.

[33] Young Hun Jung, Marc Abeille, and Ambuj Tewari. Thompson Sampling in Non-Episodic Restless Bandits. *arXiv preprint arXiv:1910.05654*, 2019.

[34] Young Hun Jung and Ambuj Tewari. Regret Bounds for Thompson Sampling in Episodic Restless Bandit Problems. *Proc. of NeurIPS*, 2019.

[35] Jackson A Killian, Arpita Biswas, Sanket Shah, and Milind Tambe. Q-learning lagrange policies for multi-action restless bandits. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 871–881, 2021.

[36] Jackson A Killian, Andrew Perrault, and Milind Tambe. Beyond" To Act or Not to Act": Fast Lagrangian Approaches to General Multi-Action Restless Bandits. In *Proc.of AAMAS*, 2021.

[37] Maialen Larrañaga, Urtzi Ayesta, and Ina Maria Verloop. Index Policies for A Multi-Class Queue with Convex Holding Cost and Abandonments. In *Proc. of ACM Sigmetrics*, 2014.

[38] Maialen Larrnaaga, Urtzi Ayesta, and Ina Maria Verloop. Dynamic Control of Birth-and-Death Restless Bandits: Application to Resource-Allocation Problems. *IEEE/ACM Transactions on Networking*, 24(6):3812–3825, 2016.

[39] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[40] Elliot Lee, Mariel S Lavieri, and Michael Volk. Optimal Screening for Hepatocellular Carcinoma: A Restless Bandit Model. *Manufacturing & Service Operations Management*, 21(1):198–212, 2019.

[41] Haoyang Liu, Keqin Liu, and Qing Zhao. Logarithmic Weak Regret of Non-Bayesian Restless Multi-Armed Bandit. In *Proc of IEEE ICASSP*, 2011.

[42] Haoyang Liu, Keqin Liu, and Qing Zhao. Learning in A Changing World: Restless Multi-Armed Bandit with Unknown Dynamics. *IEEE Transactions on Information Theory*, 59(3):1902–1916, 2012.

[43] Aditya Mate, Jackson Killian, Haifeng Xu, Andrew Perrault, and Milind Tambe. Collapsing bandits and their application to public health intervention. *Advances in Neural Information Processing Systems*, 33:15639–15650, 2020.

[44] Aditya Mate, Andrew Perrault, and Milind Tambe. Risk-Aware Interventions in Public Health: Planning with Restless Multi-Armed Bandits. In *Proc.of AAMAS*, 2021.

[45] Andreas Maurer and Massimiliano Pontil. Empirical Bernstein Bounds and Sample Variance Penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[46] Khaled Nakhleh, Santosh Ganji, Ping-Chun Hsieh, I Hou, Srinivas Shakkottai, et al. Neurwin: Neural whittle index network for restless bandits via deep rl. *Advances in Neural Information Processing Systems*, 34, 2021.

[47] José Niño-Mora. Dynamic Priority Allocation via Restless Bandit Marginal Productivity Indices. *Top*, 15(2):161–198, 2007.

[48] José Niño-Mora. Admission and routing of soft real-time jobs to multiclusters: Design and comparison of index policies. *Computers & operations research*, 39(12):3431–3444, 2012.

[49] Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret Bounds for Restless Markov Bandits. In *Proc. of Algorithmic Learning Theory*, 2012.

[50] Christos H Papadimitriou and John N Tsitsiklis. The Complexity of Optimal Queueing Network Control. In *Proc. of IEEE Conference on Structure in Complexity Theory*, 1994.

[51] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[52] Shang-Pin Sheng, Mingyan Liu, and Romesh Saigal. Data-Driven Channel Modeling Using Spectrum Measurement. *IEEE Transactions on Mobile Computing*, 14(9):1794–1805, 2014.

[53] Albert N Shiryaev. *Optimal Stopping Rules*, volume 8. Springer Science & Business Media, 2007.

[54] Cem Tekin and Mingyan Liu. Adaptive Learning of Uncontrolled Restless Bandits with Logarithmic Regret. In *Proc. of Allerton*, 2011.

[55] Cem Tekin and Mingyan Liu. Online Learning of Rested and Restless Bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

[56] Ina Maria Verloop. Asymptotically Optimal Priority Policies for Indexable and Nonindexable Restless Bandits. *The Annals of Applied Probability*, 26(4):1947–1995, 2016.

[57] Siwei Wang, Longbo Huang, and John Lui. Restless-UCB, an Efficient and Low-complexity Algorithm for Online Restless Bandits. In *Proc. of NeurIPS*, 2020.

[58] Richard R Weber and Gideon Weiss. On An Index Policy for Restless Bandits. *Journal of applied probability*, pages 637–648, 1990.

[59] Peter Whittle. Restless Bandits: Activity Allocation in A Changing World. *Journal of applied probability*, pages 287–298, 1988.

[60] Guojun Xiong, Jian Li, and Rahul Singh. Reinforcement learning for finite-horizon restless multi-armed multi-action bandits. *arXiv preprint arXiv:2109.09855*, 2021.

[61] Zhe Yu, Yunjian Xu, and Lang Tong. Deadline Scheduling as Restless Bandits. *IEEE Transactions on Automatic Control*, 63(8):2343–2358, 2018.

[62] Gabriel Zayas-Cabán, Stefanus Jasin, and Guihua Wang. An Asymptotically Optimal Heuristic for General Nonstationary Finite-Horizon Restless Multi-Armed, Multi-Action Bandits. *Advances in Applied Probability*, 51(3):745–772, 2019.

[63] Xiangyu Zhang and Peter I Frazier. Restless bandits with many arms: Beating the central limit theorem. *arXiv preprint arXiv:2107.11911*, 2021.

[64] Qing Zhao, Bhaskar Krishnamachari, and Keqin Liu. On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance. *IEEE Transactions on Wireless Communications*, 7(12):5431–5440, 2008.

[65] Yihan Zou, Kwang Taik Kim, Xiaojun Lin, and Mung Chiang. Minimizing Age-of-Information in Heterogeneous Multi-Channel Systems: A New Partial-Index Approach. In *Proc. of ACM MobiHoc*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [N/A]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]