GINO-Q: Learning an Asymptotically Optimal Index Policy for Restless Multi-armed Bandits

Gongpu Chen[1*], **Soung Chang Liew**[2], **Deniz Gündüz**[1]

[1]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K.
[2]Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.
gongpu.chen@imperial.ac.uk, soung@ie.cuhk.edu.hk, d.gunduz@imperial.ac.uk

Abstract

The restless multi-armed bandit (RMAB) framework is a popular model with applications across a wide variety of fields. However, its solution is hindered by the exponentially growing state space (with respect to the number of arms) and the combinatorial action space, making traditional reinforcement learning methods infeasible for large-scale instances. In this paper, we propose GINO-Q, a three-timescale stochastic approximation algorithm designed to learn an asymptotically optimal index policy for RMABs. GINO-Q mitigates the curse of dimensionality by decomposing the RMAB into a series of subproblems, each with the same dimension as a single arm, ensuring that complexity increases linearly with the number of arms. Unlike recently developed Whittle-index-based algorithms, GINO-Q does not require RMABs to be indexable, enhancing its flexibility and applicability. Our experimental results demonstrate that GINO-Q consistently learns near-optimal policies, even for non-indexable RMABs where Whittle-index-based algorithms perform poorly, and it converges significantly faster than existing baselines.

Introduction

A restless multi-armed bandit (RMAB) models a sequential decision-making problem, in which a set of resources must be allocated to N out of M ($1 \leq N < M$) arms at each discrete time step. Here, each arm represents a dynamic process that, depending on whether it is selected, generates a reward and may transition to a new state. These arms evolve independently except for simultaneously being subject to the resource constraint. The objective is to find an optimal policy that selects arms at each time step to maximize the expected reward. RMABs find applications in diverse fields, including network resource allocation (Wang et al. 2019), opportunistic scheduling (Wang and Chen 2021), public health interventions (Mate et al. 2022), and many more.

This paper investigates the design of an efficient reinforcement learning (RL) algorithm for RMABs. As we know, the curse of dimensionality is a central challenge in solving dynamic programs. Unfortunately, this problem is particularly severe in RMABs due to the exponential growth of RMAB's dimension with the number of arms. In fact, it is known that RMABs are PSPACE hard even when full system knowledge is available (Papadimitriou and Tsitsiklis 1999). As a result, directly treating RMABs as Markov decision processes (MDPs) and applying RL algorithms is inefficient, and, in many cases, computationally infeasible—particularly for large-scale RMABs. For instance, consider a moderate-scale RMAB with $M = 100$ and $N = 25$. Suppose each arm exhibits 10 states. Then the RMAB encompasses an astronomical 10^{100} possible states, along with $\binom{100}{25} \approx 2.4 \times 10^{23}$ valid actions. Such complexity presents a great challenge for conventional RL algorithms.

To address this challenge, recent studies have drawn inspiration from the well-known Whittle index policy (Whittle 1988), a planning algorithm for RMABs with full system knowledge. This approach decouples the RMAB into individual arms, computes an index for each, and schedules based on these indices. Despite its efficiency, the Whittle index policy is fundamentally limited by its reliance on the *indexability* property—a condition that does not naturally hold for all RMABs. While several sufficient conditions for indexability have been studied (Niño-Mora 2001, 2007; Glazebrook, Ruiz-Hernandez, and Kirkbride 2006), they are often stringent and apply only to a narrow class of RMABs. In general, verifying indexability requires full knowledge of the system and considerable analytical effort (Chen, Liew, and Shao 2022; Liu and Zhao 2010; Villar 2016), making it infeasible in most scenarios where full knowledge of the system is unavailable. As will be shown in our experiments, applying the Whittle index policy to a non-indexable RMAB can result in arbitrarily poor performance. This underscores a significant limitation of Whittle-index-based learning algorithms: without indexability guarantees, their application can be unreliable and potentially detrimental.

In this paper, we propose an RL algorithm for general RMABs that does not require indexability. Our method is inspired by a recently developed planning algorithm for RMABs known as *the gain index policy* (Chen and Liew 2024). This method has been shown to achieve asymptotic optimality as the number of arms grows large, given a fixed selection ratio. However, computing gain indices remains computationally demanding—even when full system parameters are known. We propose GINO-Q learning, a three-timescale stochastic approximation algorithm designed to efficiently approximate the gain index policy in model-free

*Corresponding author.

settings. Our approach integrates Q-learning, SARSA, and stochastic gradient descent (SGD), operating on distinct timescales to effectively learn gain indices without relying on system knowledge. Specifically, we begin by decomposing the RMAB across arms through a relaxation of the hard constraint, applying the Lagrange multiplier method to formulate M unconstrained single-arm subproblems. For each subproblem, Q-learning is used to estimate the Q-function, while SARSA is employed to approximate the gradient of the average reward with respect to the Lagrange multiplier. The Lagrange multiplier itself is then updated via SGD. By appropriately designing the learning rates for the three updates, the algorithm naturally operates on three distinct timescales, enabling efficient and stable learning of the gain index policy. The key advantages of GINO-Q are twofold:

1. *Scalability*: By decomposing the RMAB into a collection of subproblems, GINO-Q only needs to solve problems with the same dimension as a single arm. This decomposition circumvents the exponential growth of the joint state and action spaces, ensuring that computational complexity scales linearly with the number of arms M. As a result, GINO-Q achieves strong performance even in large-scale RMABs, where conventional RL algorithms are computationally infeasible.
2. *Applicability*: GINO-Q does not require the RMAB to be indexable, thereby significantly broadening its applicability. Our experiments show that Whittle-index-based learning algorithms can perform poorly in non-indexable settings. In contrast, GINO-Q consistently learns near-optimal policies—even in non-indexable RMABs.

We evaluate the performance of GINO-Q through extensive experiments, which show that it consistently outperforms existing algorithms across all tested settings.

Related Work

The success of deep RL has attracted considerable research interest in its application to practical problems that can be modeled as RMABs. Notable examples include wireless sensor scheduling (Leong et al. 2020), dynamic multichannel access (Wang et al. 2018; Demirel et al. 2018), intelligent building management (Wei, Wang, and Zhu 2017). However, a common limitation across these studies is the poor scalability of deep RL in RMABs. The experimental results presented in these papers are typically restricted to small-scale scenarios, with the number of arms limited to $M < 10$.

In the planning scenario, where the system knowledge is known, the Whittle index policy (Whittle 1988) is recognized as one of the most efficient heuristic algorithms for addressing RMAB problems. This has inspired a series of studies focused on applying RL algorithms to learn Whittle indices. For example, the work in (Fu et al. 2019) studied a Q-learning algorithm to learn the Whittle indices; however, their experiments revealed that the proposed algorithm struggles to accurately learn the Whittle indices. Subsequently, Avrachenkov and Borkar (2022) introduced Whittle-Index-Based Q-learning (WIBQ), a two-timescale stochastic approximation algorithm designed to learn the Whittle indices. They proved theoretically that WIBQ converges to the Whittle indices for indexable RMABs. Further advancing this work, Xiong and Li (2023) enhanced WIBQ to handle arms with large state spaces by coupling WIBQ with neural network function approximator. Additionally, Nakhleh et al. (2021) converted the computation of Whittle indices to an optimal control problem and proposed a neural network-based approach for computing Whittle indices.

In practical applications, Whittle-index-based learning algorithms have been employed in adaptive video streaming (Xiong et al. 2022), wireless edge caching (Xiong et al. 2024), and preventive healthcare (Biswas et al. 2021). Moreover, the Whittle index policy has also inspired studies on online learning for RMABs (Wang et al. 2023; Xiong, Wang, and Li 2022; Xiong, Li, and Singh 2022).

All of these studies rely on the indexability condition that makes the Whittle index policy applicable. However, as we will show through a concrete example, not all RMABs satisfy this condition. Fortunately, a gain index policy that does not require indexability was recently proposed by Chen and Liew (2024), who also introduced a gradient-based approach to compute the gain indices with full knowledge of system parameters. To the best of our knowledge, our proposed GINO-Q algorithm is the first RL method designed to learn the gain index policy in a model-free setting.

Notations. For any positive integer M, we will use $[M]$ to denote the set of positive integers between 1 and M, i.e., $[M] = \{1, 2, \cdots, M\}$. $\mathbb{E}[\cdot]$ denotes the expectation.

Problem Statement and Preliminaries

An RMAB consists of M arms $\{\mathcal{B}_i : i \in [M]\}$. Each arm $\mathcal{B}_i$ is an MDP represented by a tuple $(\mathcal{S}_i, \mathcal{A}_i, r_i, p_i)$, where $\mathcal{S}_i$ is the state space, $\mathcal{A}_i$ is the action space, $r_i : \mathcal{S}_i \times \mathcal{A}_i \to \mathbb{R}$ is the reward function, and p_i is the transition kernel. Let s_i^t and a_i^t denote the state and action of $\mathcal{B}_i$ at time t. In particular, we have $\mathcal{A}_i = \{0, 1\}$ for all $i \in [M]$. We will say that arm $\mathcal{B}_i$ is activated at time t if $a_i^t = 1$. At any time step, all arms evolve independently based on their actions; that is, $P(s_1^{t+1}, \cdots, s_M^{t+1} | s_1^t, \cdots, s_M^t, a_1^t, \cdots, a_M^t) = \prod_{i=1}^M p_i(s_i^{t+1} | s_i^t, a_i^t)$. However, there is a constraint on the actions that weakly couples all the arms. Specifically, at each time step, only N arms can be activated ($1 \le N < M$). That is, $\sum_{i=1}^M a_i^t = N$ for all t. The objective is to identify an optimal policy that maximizes the cumulative reward obtained from all the arms. Mathematically, an RMAB is a constrained optimization problem defined as follows:

$$\max_{\{a_i^t : i \in [M], t \ge 1\}} \lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{M} r_i\left(s_i^t, a_i^t\right)\right] \quad (1)$$

$$\text{subject to } \sum_{i=1}^{M} a_i^t = N, \quad \forall t. \quad (2)$$

We assume that all arms are unichain MDPs. As a result, the objective function (1) is independent of the initial state.

The Gain Index Policy. An RMAB can be viewed as an MDP with a joint state space $\mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M$ and action

space $\{0,1\}^M$. However, the size of the state spaces grows exponentially with M, and the set of feasible actions forms a combinatorial space of size $\binom{M}{N}$. As a result, the problem becomes challenging to solve when M is large. Since the arms are weakly coupled only via constraint (2), a common approach to mitigate the complexity is to decompose the RMAB into single-arm problems by relaxing the constraint. Building on this concept, Chen and Liew (2024) introduced the gain index policy and proved its asymptotic optimality.

In particular, we relax constraint (2) of the RMAB problem to the following:

$$\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}a_i^t\right]=N. \tag{3}$$

Replacing (2) by (3) leads to a relaxed RMAB. By applying the Lagrange multiplier method and invoking strong duality, the relaxed problem can be equivalently reformulated as the following inf-max problem:

$$\inf_{\lambda}\max_{\{a_i^t\}}\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}\left[r_i\left(s_i^t,a_i^t\right)-\lambda a_i^t\right]\right]+N\lambda, \tag{4}$$

where $\lambda\in\mathbb{R}$ is the Lagrange multiplier. For any fixed λ, (4) can be decoupled into M subproblems:

$$J_i(\lambda):\max_{\{a_i^t:t\geq 1\}}\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\left[r_i\left(s_i^t,a_i^t\right)-\lambda a_i^t\right]\right], \tag{5}$$

where $i\in[M]$. We refer to each $J_i(\lambda)$ as a *single-arm problem*. Note that $J_i(\lambda)$ is an MDP associated with $\mathcal{B}_i$: they share the same state space $\mathcal{S}_i$, action space $\mathcal{A}_i$, and transition kernel p_i. However, there is a distinction between them in terms of their reward functions. The reward function of $J_i(\lambda)$ is defined as $r_i(s,a)-\lambda a$, where λ can be interpreted as the cost of action $a=1$. If we denote the optimal value of problem $J_i(\lambda)$ by $g_i(\lambda)$, it can be determined by the Bellman equation as follows (Puterman 2014):

$$V_i(s,\lambda)+g_i(\lambda)=\max_{a\in\{0,1\}}\left\{Q_i(s,a,\lambda)\right\},\ s\in\mathcal{S}_i, \tag{6}$$

where $V_i(s,\lambda)$ is the state value function and $Q_i(s,a,\lambda)$ is the state-action value function:

$$Q_i(s,a,\lambda)\triangleq r_i(s,a)-\lambda a+\sum_{s'\in\mathcal{S}_i}p_i(s'|s,a)V_i(s',\lambda).$$

Here, we express V_i and Q_i as functions of λ to highlight their dependence on λ. Now, (4) reduces to

$$\inf_{\lambda}\ f(\lambda)\triangleq\sum_{i=1}^{M}g_i(\lambda)+N\lambda. \tag{7}$$

Denoting by λ^* the optimal solution to problem (7), then the gain index policy is defined as follows:

Definition 1 (Gain index policy)**.** *For each arm $i\in[M]$ and each state $s\in\mathcal{S}_i$, a gain index is defined as:*

$$W_i(s)\triangleq Q_i(s,1,\lambda^*)-Q_i(s,0,\lambda^*). \tag{8}$$

Then the gain index of the i-th arm at time t is given by $W_i(s_i^t)$. The gain index policy activates the N arms with the largest N gain indices, with ties broken arbitrarily.

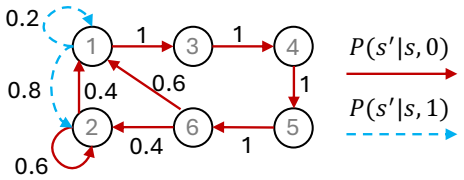


Figure 1: Transition probabilities of a non-indexable arm. Except for state 1, each of the remaining states has identical transition probabilities for both actions. For simplicity, we only plot the transitions of action 0 for those states.

Intuitively, the gain index $W_i(s)$ evaluates the "gain" of activating arm $\mathcal{B}_i$ when it is in state s. Under some mild conditions (Chen and Liew 2024), the gain index policy is asymptotically optimal in the following sense:

$$\lim_{M\to\infty}\frac{1}{M}G_M^{ind}=\lim_{M\to\infty}\frac{1}{M}G_M^{opt}, \tag{9}$$

where G_M^{ind} and G_M^{opt} denote the total time-averaged reward across all arms under the gain index policy and the optimal policy, respectively.

Whittle Index Policy and Indexability. Whittle index policy (Whittle 1988) is a classic approach for RMABs. It shares the same asymptotic optimality as described in (9) (Weber and Weiss 1990). Similar to the gain index policy, the Whittle index policy assigns an index to each state of each arm and selects at each time the top N arms with the highest indices for activation. Whittle's index for state s of arm $\mathcal{B}_i$ is defined as the infimum value of λ that ensures equal optimality between taking action 0 and action 1 in state s within the single-arm problem $J_i(\lambda)$. That is,

$$\mathrm{WI}_i(s)=\inf\{\lambda:Q_i(s,1,\lambda)=Q_i(s,0,\lambda)\}. \tag{10}$$

The Whittle index policy is only applicable to indexable RMABs. Specifically, we denote by $\mathcal{E}_i(\lambda)$ the set of states in which action 0 is optimal for problem $J_i(\lambda)$.

Definition 2 (Indexability)**.** *An arm $\mathcal{B}_i$ is considered indexable if $\mathcal{E}_i(\lambda)$ expands monotonically from the empty set to the entire state space $\mathcal{S}_i$ as λ increases from $-\infty$ to ∞. An RMAB is indexable if all its arms are indexable.*

Recall that λ can be interpreted as the cost of taking action $a=1$. If an arm $\mathcal{B}_i$ is indexable, then the optimal action for this arm in state s is $a=1$ if the cost $\lambda\leq\mathrm{WI}_i(s)$; otherwise, the optimal action is $a=0$. Therefore, $\mathrm{WI}_i(s)$ can be interpreted as the "value" of taking action $a=1$ in state s. However, in the absence of indexability, this property no longer holds—rendering $\mathrm{WI}_i(s)$ an invalid metric for prioritization. As a result, the Whittle index policy is well-defined only for indexable RMABs.

To illustrate that the indexability does not always hold, we construct a non-indexable arm as a counterexample.

Example 1 (A Non-indexable Arm)**.** *Consider an arm $\mathcal{B}_i$ with 6 states and transition probabilities illustrated in Fig. 1. The reward function is defined as $r_i(1,1)=-10, r_i(1,0)=$*

$-4, r_i(2,1) = r_i(2,0) = 4$ *and* $r_i(s,1) = 0, r_i(s,0) = -2$ *for* $s \in \{3,4,5,6\}$. *It can be verified that state* $1 \in \mathcal{E}_i(\lambda)$ *for* $-4 \le \lambda \le 2$ *and* $1 \notin \mathcal{E}_i(\lambda)$ *for* $\lambda < -4$ *and* $\lambda > 2$. *Hence the set* $\mathcal{E}_i(\lambda)$ *does not expand monotonically as* λ *increases, implying that the arm is not indexable.*

Nevertheless, the quantity defined by (10) remains well-defined, even for non-indexable arms. This raises a natural question: *what happens if the Whittle index policy is applied to a non-indexable RMAB?* Moreover, in existing Whittle-index-based algorithms such as WIBQ (Avrachenkov and Borkar 2022) and Neurwin (Nakhleh et al. 2021), the Whittle index is typically determined by learning a value of λ that satisfies the condition $Q_i(s,1,\lambda) = Q_i(s,0,\lambda)$. However, in Example 1, both -4 and 2 satisfy this condition. As a result, the algorithm may converge to one of these values arbitrarily, or even oscillate if multiple such solutions are close in value. *How do Whittle-index-based algorithms perform under such ambiguity?*

As demonstrated by our experiments, applying the Whittle index policy to an RMAB with arms as defined in Example 1 can lead to arbitrarily poor performance. This underscores the fragility of Whittle-index-based learning algorithms in the absence of indexability guarantees and highlights the importance of the gain index policy, which operates without such assumptions. These considerations motivate the development of GINO-Q, which aims to learn the gain index policy efficiently in model-free settings.

Gain-Index-Oriented Q Learning

With asymptotic optimality and no reliance on indexability, learning the gain index policy offers a promising solution for RMABs without system knowledge, particularly for large-scale problems. This section presents the GINO-Q learning algorithm for RMABs. Our approach involves decomposing the RMAB into single-arm problems and learning the gain indices for each arm.

By definition, the gain indices of an arm are determined by the Q-function of the corresponding single-arm problem under the optimal activation cost λ^*. While learning the Q-function for a fixed λ reduces to a standard Q-learning task, jointly learning λ^* introduces additional complexity due to the coupling between the dual variable λ and the value functions. Specifically, in the absence of system knowledge, the gradient required to update λ cannot be computed directly—and, moreover, it cannot be estimated directly during the Q-learning process.

To overcome this challenge, we propose a three-timescale stochastic approximation algorithm. The algorithm updates the Q-function of $J_i(\lambda)$ and the dual variable λ on medium and slow timescales, respectively. A third, faster timescale is employed to estimate the derivative of $g_i(\lambda)$ with respect to λ, which plays a critical role in guiding the update of λ.

Useful Properties. We begin by introducing some useful definitions and establishing key properties of the optimization problem (7), which form the basis of our method. We first define an auxiliary MDP for each arm as follows:

Definition 3 (Auxiliary MDP)**.** *The auxiliary MDP associated with arm* $\mathcal{B}_i$ *is defined as* $\mathcal{M}_i = (\mathcal{S}_i, \mathcal{A}_i, c, p_i)$, *where the state space* $\mathcal{S}_i$, *action space* $\mathcal{A}_i$, *and transition kernel* p_i *are identical to those of* $\mathcal{B}_i$. *The cost function is given by* $c(s,a) = a$ *for all* $s \in \mathcal{S}_i$ *and* $a \in \mathcal{A}_i$.

Since $\mathcal{B}_i$ is a unichain MDP, so is $\mathcal{M}_i$. As a result, the average cost of $\mathcal{M}_i$ under any stationary policy π is independent of the initial state s. We denote this average cost by $h_i^\pi \triangleq \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi\left[\sum_{t=1}^T a_i^t\right]$. Let $D_i^\pi(s,a)$ denote the state-action value function of $\mathcal{M}_i$ under policy π, and $\pi(a'|s')$ the probability of taking action a' in state s'. Then we have

$$D_i^\pi(s,a) + h_i^\pi = a + \sum_{s',a'} p_i(s'|s,a)\pi(a'|s')D_i^\pi(s',a').$$

Clearly, $J_i(\lambda)$ and $\mathcal{M}_i$ share the same policy space. Let π be a policy for the single-arm problem $J_i(\lambda)$, and $V_i^\pi(s)$ and g_i^π be the associated value function and long-term average reward, respectively. It is easy to see that, for any policy π,

$$\frac{dg_i^\pi}{d\lambda} = -h_i^\pi,\ \forall i \in [M]. \tag{11}$$

Let π_i^λ denote the optimal policy for problem $J_i(\lambda)$, then the derivative of function $f(\lambda)$ is given by

$$f'(\lambda) = \sum_{i=1}^M \frac{dg_i(\lambda)}{d\lambda} + N = N - \sum_{i=1}^M h_i^{\pi_i^\lambda}. \tag{12}$$

Drawing from these concepts, we can establish the existence of a bounded optimal solution λ^*, which supports the practical feasibility of learning λ^*.

Lemma 1. *For any RMAB with bounded reward functions* $\{r_i\}_{i\in[M]}$, $f(\lambda)$ *is a piecewise linear and convex function. In addition, there always exist a bounded* λ^* *that achieves the minimum value of* $f(\lambda)$.

GINO-Q learning algorithm. Our objective is to learn the optimal dual variable λ^* and the corresponding Q-functions for the single-arm problems $J_i(\lambda^*)$, for all $i \in [M]$. To this end, we propose a simple yet effective algorithm that simultaneously updates λ and learns the associated Q-functions, thereby enabling the computation of gain indices for all arms. We refer to this approach as Gain-Index-Oriented Q-learning (GINO-Q).

In particular, we fix a stepsize sequence $\{\theta^t : t \ge 1\}$, and employ the stochastic gradient-descent method to update λ:

$$\lambda^{t+1} = \lambda^t - \theta^t \hat{f}'(\lambda^t), \tag{13}$$

where $\hat{f}'(\lambda^t)$ is an estimator of $f'(\lambda^t)$, to be detailed later. Meanwhile, we use the standard relative value iteration (RVI) Q-learning algorithm (Abounadi, Bertsekas, and Borkar 2001) to learn the Q-function of every single-arm problem $J_i(\lambda^t)$ for the current λ^t. That is, $\forall i \in [M]$:

$$\begin{aligned} Q_i^{t+1}(s_i^t, a_i^t) =& Q_i^t(s_i^t, a_i^t) + \beta_i^t[r_i(s_i^t, a_i^t) - \lambda^t a_i^t + \\ & \max_a Q_i^t(s_i^{t+1}, a) - Q_i^t(s_i^t, a_i^t) - g_i^t], \end{aligned} \tag{14}$$

where $\{\beta_i^t : t \ge 1\}$ is a predefined stepsize sequence. While g_i^t can be estimated in various ways (Abounadi, Bertsekas, and Borkar 2001), one of the most widely used methods is:

$$g_i^t = \frac{1}{2|\mathcal{S}|}\sum_{s\in\mathcal{S}_i}\sum_{a\in\mathcal{A}_i} Q_i^t(s,a). \tag{15}$$

We proceed by constructing the estimator $\hat{f}'(\lambda)$ according to (12). Given any λ, the optimal policy π_i^λ for $J_i(\lambda)$ selects actions greedily according to the optimal Q-function of $J_i(\lambda)$, denoted by Q_i. At time step t, Q_i^t serves as an estimate of Q_i. Hence, the policy that selects actions greedily according to Q_i^t can be regarded as an estimate of π_i^λ. To learn the average cost of this policy for the auxiliary MDP $\mathcal{M}_i$, we employ the SARSA algorithm (Sutton and Barto 2018):

$$D_i^{t+1}(s_i^t, a_i^t) = D_i^t(s_i^t, a_i^t) + \alpha_i^t[a_i^t + D_i^t(s_i^{t+1}, a_i^{t+1}) - D_i^t(s_i^t, a_i^t) - h_i^t], \quad (16)$$

where $\{\alpha_i^t : t \geq 1\}$ is the stepsize sequence, action $a_i^{t+1} = \arg\max_a Q_i^t(s_i^{t+1}, a)$ is selected greedily based on Q_i^t. The average cost of $\mathcal{M}_i$ is estimated by the well-known method from (Abounadi, Bertsekas, and Borkar 2001):

$$h_i^t = \max_a D_i^t(s_i, a), \quad (17)$$

where $s_i \in \mathcal{S}_i$ is an arbitrary but fixed reference state. We then estimate $f'(\lambda^t)$ by $\hat{f}'(\lambda^t) = N - \sum_{i=1}^M h_i^t$.

The stepsize schedules of the three coupled iterates play a critical role in learning both λ^* and the corresponding Q-functions. As shown in equation (12), for any given λ, the derivative $f'(\lambda)$ depends on the optimal policy π_i^λ for the single-arm problem $J_i(\lambda)$. Therefore, the update of the sequence $\{\lambda^t\}$ must proceed more slowly than that of Q_i^t, allowing Q-learning to sufficiently approximate Q_i. Conversely, since SARSA aims to estimate the state-action value function of $\mathcal{M}_i$ under the greedy policy induced by Q_i^t, its updates should operate at a faster timescale than Q-learning.

We define the stepsize sequences of the three coupled iterates properly so that they form a three-timescale stochastic approximation algorithm, with updates (13), (14), and (16) operating in the slow, medium, and fast timescales, respectively. In particular, define[1]

$$\alpha_i^t = \frac{C_1}{t}, \beta_i^t = \frac{C_2}{t\sqrt{\log t}}, \theta^t = \frac{C_3}{t \log t}\mathbf{1}\{t(\text{mod } C_4) = 0\},$$

where $\{C_k\}$ are constants, $\mathbf{1}\{\cdot\}$ is the indicator function, $\{\alpha_i^t\}$ and $\{\beta_i^t\}$ are invariant across $i \in [M]$. It is easy to verify that $\sum_{t=1}^\infty x^t = \infty$ and $\sum_{t=1}^\infty (x^t)^2 < \infty$ for $x = \alpha_i, \beta_i$ or θ. Update (16) is faster than (14) because $\beta_i^t = o(\alpha_i^t)$. For the same reason, update (14) is faster than (13).

The GINO-Q algorithm is summarized in Algorithm 1. A detail to note is line 14, where λ^t is updated only if the estimated derivative is decreased in absolute value. Since $f(\lambda)$ is piecewise linear and convex, this strategy helps mitigate erroneous updates caused by noise in the estimation of $f'(\lambda)$. While this may slightly slow convergence near λ^*, it improves stability and robustness during learning.

Remark: If the learner has knowledge of the arm classification (but not the specific parameters of arms), updates (14) and (16) do not need to be performed for each individual arm. Instead, it is sufficient to maintain a pair of (Q_i^t, D_i^t) for each class. This is because arms within the same class share the same Q and D functions. As a result, the complexity of GINO-Q increases linearly with respect to the number of classes K. If the arm classification is unknown, then the complexity scales linearly with respect to the number of arms M, which still represents a significant reduction compared to the exponential growth of the state space.

[1]In practice, t in the expressions of α_i^t and β_i^t can be replaced by $n(s_i^t, a_i^t)$, the number of occurrences of (s_i^t, a_i^t) up to time t.

Algorithm 1: GINO-Q Learning

Input: Integer T, learning rates $\{\alpha_i^t\}$, $\{\beta_i^t\}$, and $\{\theta^t\}$
Initialization: Let $t = 1, \lambda^1 = 0, y^0 = M$. Reset the RMAB and receive the initial arm states $\{s_i^1\}$
Output: Gain indices of all arms

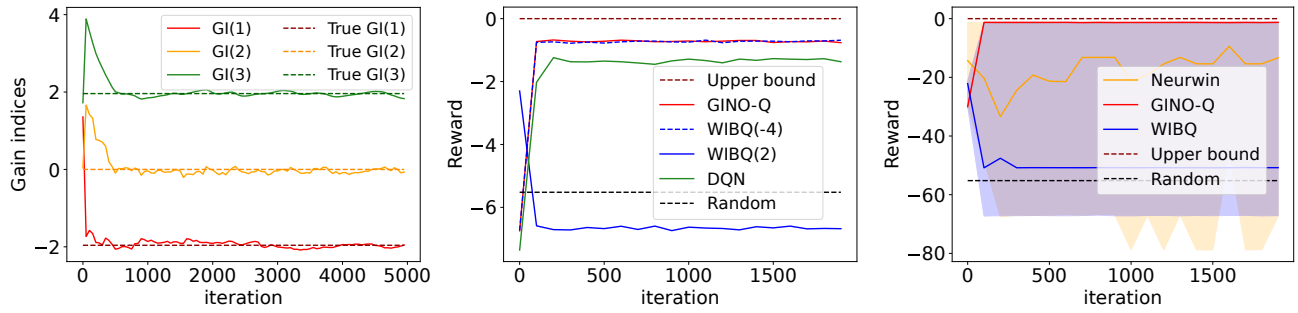
while $t \leq T$ **do**
  for $i = 1, 2, \cdots, M$ **do**
    Select action a_i^t according to Q_i^t (e.g., ϵ-greedy)
    Apply action a_i^t to the i-th arm, observe reward r_i^t and the new state s_i^{t+1}
    Compute g_i^t using equation (15)
    $\delta_i^t = r_i^t - \lambda^t a_i^t + \max_a Q_i^t(s_i^{t+1}, a) - Q_i^t(s_i^t, a_i^t)$
    $Q_i^{t+1}(s_i^t, a_i^t) = Q_i^t(s_i^t, a_i^t) + \beta_i^t(\delta_i^t - g_i^t)$
    Compute h_i^t using equation (17)
    $b_i^{t+1} = \arg\max_a Q_i^t(s_i^{t+1}, a)$
    $\sigma_i^t = a_i^t + D_i^t(s_i^{t+1}, b_i^{t+1}) - D_i^t(s_i^t, a_i^t) - h_i^t$
    $D_i^{t+1}(s_i^t, a_i^t) = D_i^t(s_i^t, a_i^t) + \alpha_i^t \sigma_i^t$
  end for
  $y^t = N - \sum_{i=1}^M h_i^t$
  $\lambda^{t+1} = \lambda^t - \theta^t \mathbf{1}\{|y^t| < |y^{t-1}|\} y^t$
  $t \leftarrow t + 1$
end while
for $i = 1, 2 \cdots, M$ **do**
  for $s \in \mathcal{S}_i$ **do**
    $W_i(s) = Q_i^T(s, 1) - Q_i^T(s, 0)$ // gain index
  end for
end for

Robustness. The definition of gain indices is dependent on λ^*, and our GINO-Q learning aims to learn λ^* and the corresponding Q function of every single-arm problem $J_i(\lambda^*)$. This part discusses the robustness of GINO-Q learning with respect to the value of λ^*. Remarkably, we show that the asymptotic optimality of the gain index policy is assured as long as the sequence $\{\lambda^t\}$ converges to a neighbourhood of λ^*—not necessarily converging precisely to λ^*. This implies that convergence to a neighbourhood of λ^* is sufficient to induce a near-optimal index policy.

Formally, for any λ and any arm $\mathcal{B}_i$, define

$$W_i^\lambda(s) \triangleq Q_i(s, 1, \lambda) - Q_i(s, 0, \lambda), \ \forall s \in \mathcal{S}_i.$$

Then the gain index is $W_i(s) = W_i^{\lambda^*}(s)$. Learning the values of $\{W_i^\lambda(s)\}$ for a given λ constitutes a well-studied RL task. However, a practical concern arises: how does the performance of the index policy get affected by inaccuracies in estimating λ^*? The following lemma partially addresses this question by showing that the policy remains asymptotically optimal to a certain degree of estimation error in λ^*.



(a) Gain index over training. (b) Reward over training ($M = 10, N = 7$). (c) $M = 100, N = 70$.

Figure 2: Performances of the GINO-Q and baseline algorithms in a non-indexable RMAB problem. In (a), GI(s) denotes the gain index of state s. States 3 to 6 share the same gain index, hence we only plot GI(3) and omit the others. In (b), WIBQ(x) corresponds to the result of a single run where the Whittle index of state 1 learned by WIBQ is $x \in \{-4, 2\}$. In (c), the results represent the average of 20 independent runs, with the shaded areas indicating confidence bounds.

Lemma 2. *There exists a non-empty interval (λ^l, λ^u) that contains λ^*, such that the gain index policy with indices $\{W_i^\lambda(s)\}$ is asymptotically optimal for any $\lambda \in (\lambda^l, \lambda^u)$.*

As demonstrated in the proof (see Appendix), (λ^l, λ^u) is the interval where $f'(\lambda) = 0$, whenever such an interval exists. If no such interval exists, (λ^l, λ^u) corresponds to the range of the two segments connected to λ^*. In this latter case, the non-smoothness of $f(\lambda)$ may pose a challenge when searching for λ^* using gradient decent methods. Specifically, the sequence $\{\lambda^t\}$ may oscillate around λ^* instead of converging to it precisely because $|f'(\lambda)|$ is bounded away from 0 for λ in the neighborhood of λ^*. Fortunately, according to Lemma 2, this situation will not affect the asymptotic optimality of the resulting index policy. This property makes GINO-Q particularly well-suited for large-scale RMABs—when M is large, GINO-Q can learn a near-optimal policy and remains robust to the inaccuracies in λ^*.

Experiments

This section showcases the performance of GINO-Q by evaluating it across three distinct RMABs. For each problem, we explore various settings characterized by different pairs of (M, N) values. The baseline algorithms include the conventional RL method DQN (Mnih et al. 2015), as well as recently developed Whittle-index-based approaches: WIBQ (Avrachenkov and Borkar 2022) and Neurwin (Nakhleh et al. 2021). Among these, DQN models the entire RMAB as a single MDP, resulting in a state space growing exponentially in M, and a combinatorial action space of size $\binom{M}{N}$. As M increases, the problem quickly becomes intractable for DQN. Therefore, we only evaluate DQN in small-scale problems. On the other hand, WIBQ and Neurwin are state-of-the-art algorithms specifically designed for RMABs, with a focus on learning the Whittle index policy. They serve as our primary baselines for comparison. Additional details of the experiments can be found in the appendix.

Non-indexable RMAB. We begin with comparing GINO-Q with the Whittle-index-based learning algorithms in a non-indexable RMAB. Both WIBQ and Neurwin assume that the given RMAB is indexable because the Whittle index policy is only defined for indexable RMABs. However, indexability is rarely guaranteed in practical learning scenarios. To explore the consequences of applying a Whittle-based algorithm to a non-indexable problem, we constructed an RMAB using the arm model from Example 1 and evaluated the performance of GINO-Q and WIBQ in this setting.

We first consider the setting with $M = 10$ and $N = 7$. Fig. 2a demonstrates that GINO-Q effectively acquires the gain indices. It is important to note that the key point of the gain index policy is the relative ordering of states by their gain indices. Consequently, the performance of GINO-Q stabilizes once this order is established (compare Fig. 2a with 2b). As discussed in Example 1, there are two values of λ (i.e., -4 and 2) that satisfy the condition $Q_i(1, 1, \lambda) = Q_i(1, 0, \lambda)$. As a result, both can be recognized by the WIBQ algorithm as valid Whittle indices for state 1. In contrast, each of the remaining states admits a unique Whittle index. In our experiments, WIBQ consistently learned the true Whittle indices for states 2 to 6. However, for state 1, the Whittle index converged to -4 in some runs and to 2 in others. State 1 has the highest priority with index 2 and the lowest priority with index -4, resulting in significantly different performances, as shown in Fig. 2b. For benchmarking purposes, we also evaluated the performance of DQN and the random policy (i.e., selecting N arms randomly at each step). Notably, WIBQ performs even worse than the random policy when the Whittle index of state 1 converges to 2, highlighting the risk of applying Whittle-index-based learning to non-indexable RMABs. For simplicity, the results of Neurwin are not plotted in this setting as WIBQ already learned the true Whittle indices.

We also compared the algorithms in the setting of $(M, N) = (100, 70)$, as depicted in Fig. 2c. The low average of WIBQ suggests that it is more likely to converge to 2 than -4. While Neurwin generally outperforms WIBQ in this RMAB, it similarly faces a high risk of learning a very poor index policy. In contrast, GINO-Q consistently learns a near-optimal index policy. Moreover, we computed upper bounds

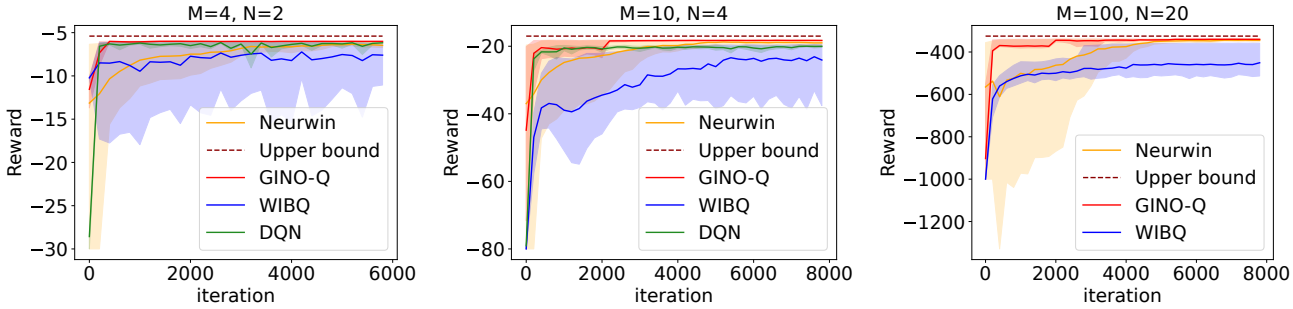


Figure 3: Performance comparisons between GINO-Q and baseline algorithms in the channel allocation problem.

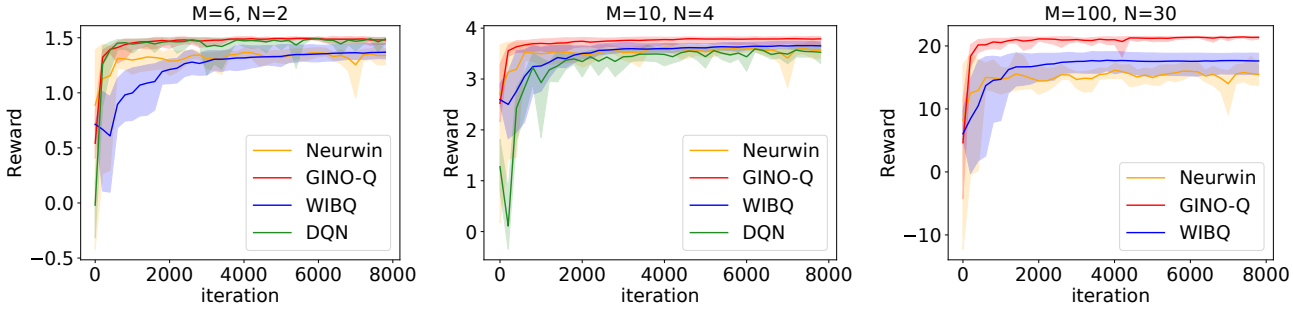


Figure 4: Performance comparisons between GINO-Q and baseline algorithms in the patrol scheduling problem.

for both settings, provided by the optimal value of the relaxed RMAB. Fig. 2c shows that when M is large, GINO-Q closely approaches the upper bound, providing strong evidence of its asymptotic optimality.

Channel Allocation. Channel allocation in communication networks involves strategically assigning communication channels to users to optimize the overall system performance. In a simple scenario, a network possesses a finite number of channels, each of which can be allocated to a single user at any time step. A recent research topic is channel allocation aimed at minimizing the average age of information (AoI), a metric that measures information freshness. Tripathi and Modiano (2019) formulated this problem as an RMAB and proved that this RMAB is indexable. See the Appendix for a detailed description of the problem.

We evaluated the GINO-Q and baseline algorithms across three different scales of M and N. The reported results represent averages from 20 independent runs, as presented in Fig. 3. It can be observed that GINO-Q consistently achieves the best performance across all settings. While Neurwin is capable of learning a Whittle index policy comparable to the gain index policy, it converges at a significantly slower rate. Additionally, compared to Neurwin and WIBQ, GINO-Q exhibits very low variance, as indicated by their confidence bounds.

Patrol Scheduling. We also assess the algorithms in a patrol scheduling problem, another application that is often formulated as an RMAB (Xu et al. 2021). The performances of GINO-Q and the baseline algorithms in this problem are reported in Fig. 4. Unfortunately, the upper bounds obtained from the relaxed RMAB appear to be loose for these settings, so they are not plotted here. Nevertheless, once again, GINO-Q outperforms all benchmark algorithms across all settings. In the small-scale setting ($M = 6$), DQN performs as well as GINO-Q. However, as M increases to 10, DQN's policy becomes less effective than both the gain and Whittle index policies due to the curse of dimensionality. When $M = 100$, the RMAB exhibits enormous state and action spaces that DQN cannot handle. In contrast, GINO-Q is able to learn a near-optimal index policy quickly, even for large-scale RMABs. Its convergence speed is not affected by the increase in M, as long as the number of classes is fixed.

Conclusion

In this paper, we introduced GINO-Q, a novel three-timescale stochastic approximation algorithm designed to address the challenges posed by RMABs. Our approach effectively tackles the curse of dimensionality by decomposing the RMAB into manageable single-arm problems, ensuring that the computational complexity grows linearly with the number of arms. Unlike existing Whittle-index-based learning algorithms, GINO-Q does not require RMABs to be indexable, significantly broadening its applicability. We showed experimentally that Whittle-index-based learning algorithms can perform poorly in non-indexable RMABs. In contrast, GINO-Q consistently learns near-optimal policies across all experimental RMABs, including non-indexable ones, and shows great efficiency by converging significantly faster than existing baselines.

Acknowledgements

The work of G. Chen and D. Gündüz was supported in part by the UKRI for the Project AI-R (ERC Consolidator) under Grant EP/X030806/1, and by INFORMED-AI under Grant EP/Y028732/1. The work of S. C. Liew was supported in part by the General Research Funds (Project No. 14200221) established under the University Grant Committee of the Hong Kong Special Administrative Region, China.

References

Abounadi, J.; Bertsekas, D.; and Borkar, V. S. 2001. Learning algorithms for Markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3): 681–698.

Avrachenkov, K. E.; and Borkar, V. S. 2022. Whittle index based Q-learning for restless bandits with average reward. *Automatica*, 139: 110186.

Biswas, A.; Aggarwal, G.; Varakantham, P.; and Tambe, M. 2021. Learn to Intervene: An Adaptive Learning Policy for Restless Bandits in Application to Preventive Healthcare. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4039–4046.

Chen, G.; and Liew, S. C. 2024. An Index Policy for Minimizing the Uncertainty-of-Information of Markov Sources. *IEEE Transactions on Information Theory*, 70(1): 698–721.

Chen, G.; Liew, S. C.; and Shao, Y. 2022. Uncertainty-of-Information Scheduling: A Restless Multiarmed Bandit Framework. *IEEE Transactions on Information Theory*, 68(9): 6151–6173.

Demirel, B.; Ramaswamy, A.; Quevedo, D. E.; and Karl, H. 2018. DeepCAS: A Deep Reinforcement Learning Algorithm for Control-Aware Scheduling. *IEEE Control Systems Letters*, 2(4): 737–742.

Fu, J.; Nazarathy, Y.; Moka, S.; and Taylor, P. G. 2019. Towards Q-learning the Whittle Index for Restless Bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*, 249–254.

Glazebrook, K. D.; Ruiz-Hernandez, D.; and Kirkbride, C. 2006. Some Indexable Families of Restless Bandit Problems. *Advances in Applied Probability*, 38(3): 643–672.

Leong, A. S.; Ramaswamy, A.; Quevedo, D. E.; Karl, H.; and Shi, L. 2020. Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems. *Automatica*, 113: 108759.

Liu, K.; and Zhao, Q. 2010. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11): 5547–5567.

Mate, A.; Madaan, L.; Taneja, A.; Madhiwalla, N.; Verma, S.; Singh, G.; Hegde, A.; Varakantham, P.; and Tambe, M. 2022. Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12017–12025.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Nakhleh, K.; Ganji, S.; Hsieh, P.-C.; Hou, I.; Shakkottai, S.; et al. 2021. NeurWIN: Neural Whittle index network for restless bandits via deep RL. *Advances in Neural Information Processing Systems*, 34: 828–839.

Niño-Mora, J. 2001. Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 33(1): 76–98.

Niño-Mora, J. 2007. Dynamic priority allocation via restless bandit marginal productivity indices. *Top*, 15(2): 161–198.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1999. The Complexity of Optimal Queuing Network Control. *Mathematics of Operations Research*, 24(2): 293–305.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Tripathi, V.; and Modiano, E. 2019. A whittle index approach to minimizing functions of age of information. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1160–1167. IEEE.

Villar, S. S. 2016. Indexability and optimal index policies for a class of reinitialising restless bandits. *Probability in the engineering and informational sciences*, 30(1): 1–23.

Wang, J.; Ren, X.; Mo, Y.; and Shi, L. 2019. Whittle index policy for dynamic multichannel allocation in remote state estimation. *IEEE Transactions on Automatic Control*, 65(2): 591–603.

Wang, K.; and Chen, L. 2021. *Restless Multi-Armed Bandit in Opportunistic Scheduling*. Springer.

Wang, K.; Xu, L.; Taneja, A.; and Tambe, M. 2023. Optimistic whittle index policy: Online learning for restless bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10131–10139.

Wang, S.; Liu, H.; Gomes, P. H.; and Krishnamachari, B. 2018. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE transactions on cognitive communications and networking*, 4(2): 257–265.

Weber, R. R.; and Weiss, G. 1990. On an index policy for restless bandits. *Journal of applied probability*, 27(3): 637–648.

Wei, T.; Wang, Y.; and Zhu, Q. 2017. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th annual design automation conference 2017*, 1–6.

Whittle, P. 1988. Restless Bandits: Activity Allocation in a Changing World. *Journal of Applied Probability*, 25: 287–298.

Xiong, G.; and Li, J. 2023. Finite-time analysis of whittle index based Q-learning for restless multi-armed bandits with neural network function approximation. In *Advances in Neural Information Processing Systems*, volume 36, 29048–29073.

Xiong, G.; Li, J.; and Singh, R. 2022. Reinforcement learning augmented asymptotically optimal index policy for finite-horizon restless bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8726–8734.

Xiong, G.; Qin, X.; Li, B.; Singh, R.; and Li, J. 2022. Index-aware reinforcement learning for adaptive video streaming at the wireless edge. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 81–90.

Xiong, G.; Wang, S.; and Li, J. 2022. Learning infinite-horizon average-reward restless multi-action bandits via index awareness. *Advances in Neural Information Processing Systems*, 35: 17911–17925.

Xiong, G.; Wang, S.; Li, J.; and Singh, R. 2024. Whittle Index-Based Q-Learning for Wireless Edge Caching With Linear Function Approximation. *IEEE/ACM Transactions on Networking*.

Xu, L.; Bondi, E.; Fang, F.; Perrault, A.; Wang, K.; and Tambe, M. 2021. Dual-mandate patrols: Multi-armed bandits for green security. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14974–14982.