Deep Index Policy for Multi-Resource Restless Matching Bandit and Its Application in Multi-Channel Scheduling

Nida Zamir
nidazamir@tamu.edu
Texas A&M University
College Station, Texas, USA

I-Hong Hou
ihou@tamu.edu
Texas A&M University
College Station, Texas, USA

ABSTRACT

Scheduling in multi-channel wireless communication system presents formidable challenges in effectively allocating resources. To address these challenges, we investigate a multi-resource restless matching bandit (MR-RMB) model for heterogeneous resource systems with an objective of maximizing long-term discounted total rewards while respecting resource constraints. We have also generalized to applications beyond multi-channel wireless. We discuss the Max-Weight Index Matching algorithm, which optimizes resource allocation based on learned partial indexes. We have derived the policy gradient theorem for index learning. Our main contribution is the introduction of a new Deep Index Policy (DIP), an online learning algorithm tailored for MR-RMB. DIP learns the partial index by leveraging the policy gradient theorem for restless arms with convoluted and unknown transition kernels of heterogeneous resources. We demonstrate the utility of DIP by evaluating its performance for three different MR-RMB problems. Our simulation results show that DIP indeed learns the partial indexes efficiently.

CCS CONCEPTS

• **Theory of computation → Sequential decision making**.

KEYWORDS

Online learning, multi-armed bandit, multi-channel scheduling

ACM Reference Format:
Nida Zamir and I-Hong Hou. 2024. Deep Index Policy for Multi-Resource Restless Matching Bandit and Its Application in Multi-Channel Scheduling. In *The Twenty-fifth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MOBIHOC '24), October 14–17, 2024, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641512.3686381

1 INTRODUCTION

Scheduling in wireless communication systems is a critical aspect that plays a pivotal role in optimizing resource utilization and enhancing system performance. The importance of scheduling stems from the inherent limitations and complexities of wireless channels, including limited bandwidth, varying channel conditions, and the presence of interference. Effective scheduling strategies enable efficient allocation of scarce resources, such as bandwidth and transmission slots, to users or applications, thereby maximizing system throughput, minimizing latency, and enhancing overall network performance [26]. Scheduling problems have been extensively studied in the literature in various applications such as Age of Information (AoI) [14], Quality of Experience (QoE) [2], Mobile Edge Computing (MEC) for wireless Virtual Reality (VR) [36] and opportunistic scheduling problems for downlink data traffic [1].

Most modern wireless systems, such as those employing orthogonal frequency-division multiple access (OFDMA), have the capability to schedule different users for transmissions on different channels or subcarriers simultaneously. Scheduling problems in such multi-channel wireless networks are especially difficult because a user may experience different channel qualities on different channels. Hence, a controller of the system needs to decide not only whom to schedule, but also which channels to schedule each user on.

In this paper, we investigate a multi-resource restless matching bandit (MR-RMB) model to address challenges in multi-channel scheduling problems. This model considers each user as a restless arm whose state, such as queue status, packet delay, recent deliveries, evolves according to both application behaviors and the perceived network services. In each time step, a controller observes the states of all restless arms and then matches each restless arm to a resource (channel), subject to the capacity of the resource. In addition to multi-channel scheduling problems, we also show that the MR-RMB model can be applied to a wide range of applications such as advertisement placements on social media websites and call center scheduling.

The MR-RMB model extends the restless bandit problem, which is a special case with a single resource. The restless bandit problem is intractable due to its exponentially growing state space. To address this challenge, we adopt the technique in Zou et al. [38], which proposes a partial index policy for minimizing AoI in multi-channel wireless networks. This policy calculates a partial index between each restless arm n and each resource h. An important feature of the partial index is that it only depends on the state of the restless arm n and is independent of the states of all other restless arms. This feature makes calculating the partial index tractable. We generalize this policy for generic MR-RMB models.

An important limitation of the partial index policy is that it requires the complete knowledge of the transition kernel of each restless arm, which, in the case of multi-channel scheduling, consists of the application behaviors and channel qualities of each user, to calculate the partial index. In practice, such knowledge may not be available. We propose using deep reinforcement learning for the

MOBIHOC '24, October 14–17, 2024, Athens, Greece

ACM ISBN 979-8-4007-0521-2/24/10.
https://doi.org/10.1145/3641512.3686381

controller to learn the partial indexes on the fly without any prior knowledge about the transition kernel of each restless arm. We show that finding the partial indexes is equivalent to finding the optimal control policy for a family of auxiliary Markov decision processes (MDP). We further derive the policy gradient theorem for the entire family of auxiliary MDPs.

Based on the policy gradient theorem, we propose Deep Index Policy (DIP), a new deep reinforcement learning policy that learns the partial indexes by using actor-critic networks. The utility of DIP is comprehensively evaluated in three different MR-RMB problems, two problem are scheduling problems in multi-channel wireless networks and the third one is an advertisement placement problem in social media websites. For scenarios where partial indexes can be calculated, DIP indeed converges to partial indexes efficiently. For other scenarios, DIP significantly outperforms other policies.

Our primary contributions are the introduction of DIP, which generalizes the partial index policy for generic MR-RMB models, and the development of a learning algorithm that efficiently computes the partial index without requiring prior knowledge of the system. Zou et al. [38] define partial index but calculating the partial index requires a full knowledge about the system, which may not be available in practice. In contrast, we have a learning algorithm that can efficiently find the partial index without knowing anything about the system in advance. Additionally, we advance the policy gradient theorem by incorporating multiple resources, a more complex scenario than the single-resource setting explored by Nakhleh and Hou [24]. Unlike the restless bandit problem, where the comparison is between activation and idle, our work involves comparing activation on resource h with activation on any other resource. This complexity makes a policy gradient theorem more challenging to drive.

The rest of the paper is organized as follows: Section 2 reviews recent studies on wireless scheduling and restless bandits. Section 3 introduces the model for MR-RMB. Section 4 discusses the partial index policy for MR-RMB. Section 5 establishes the policy gradient theorem for finding partial indexes. Section 6 introduces the deep index policy that finds partial indexes through deep reinforcement learning. Section 7 presents the simulation results. Finally, Section 8 concludes the paper.

1.1 Notations

Throughout this paper, we use $\vec{x}$ to denote the vector containing $[x_1, x_2, \dots]$. We use $\vec{x}_{-m}$ to denote the vector $\vec{x}$ without x_m, i.e., $[x_1, x_2, \dots, x_{m-1}, x_{m+1}, \dots]$, and use $[\vec{x}_{-m}, y]$ to denote the vector $\vec{x}$ with x_m being replaced by y, i.e., $[x_1, x_2, \dots, x_{m-1}, y, x_{m+1}, \dots]$

2 RELATED WORK

Scheduling problems have been extensively studied in wireless networks, with many focusing on scenarios with a single shared resource [14, 36], potentially limiting the generalizability of their findings. However, significant research has also been investigated in multi-channel wireless networks [6, 8, 16, 19, 32]. Gopalan, Caramanis and Shakkottai explore an online scheduling algorithm to allocate multiple channels for a queueing system [11]. Krishnasamy et al. examine the optimization of energy costs in systems with multiple Base Stations (BSs) [17]. Bodas et al. focus on the allocation of multiple channels for the downlink of cellular wireless networks [4]. Additionally, Sombabu and Moharir [28], Xu et al. [35], Talak, Karaman and Modiano [29], Li and Duan [20] and Fountoulakis et al. [9] address the AoI minimization problem to enhance reliability under unstable conditions. All these studies assume that the application behaviors and channel conditions are known in advance.

Some researchers have explored learning approach for scheduling problems with unknown application behaviors or channel conditions. Huang et al. optimize task offloading in the downlink of mobile edge computing systems [13]. Leong et al. optimize sensor transmission scheduling for remote state estimation in cyber-physical systems.[18]. Zakeri et al. develop transmission scheduling policies to minimize AoI [37]. Naderializadeh et al. study resource management in the downlink of a wireless network with multiple access points (APs) transmitting data to user equipment devices (UE), where each UE can only be served by one AP at a time during scheduling [23]. These studies focus on learning scheduling strategies for a single shared resource. They cannot be easily extended to multi-channel wireless networks.

Scheduling problem has frequently been formulated as a Restless Multi-armed Bandit (RMAB) problem [7, 34]. RMAB is notoriously hard to solve as the size of its state space grows exponentially with the number of arms. The Whittle index is widely embraced as a scalable solution for addressing RMAB problems and has been extensively studied in various applications [5, 22, 24, 30]. We note that in the RMAB literature there is also a line of work on multi-action bandits considering a single resource [12, 15]. Some of the researcher also considered multiple resources. Wu et al. [33] calculate the index via brute force, which is not feasible for larger state spaces or unknown system behaviors. Simchi-Levi, Sun and Wang [27] do not consider states of the user. Gai, Krishnamachari and Liu [10] consider states, but during the assignment, it does not look at the states. Assignments are made based on the best performance for the long-term reward, and then these assignments are maintained. Therefore, it is only looking for a stationary assignment not based on states. Zou et al. [38] study the heterogeneous and unreliable channels to minimize the AoI problem where they have considered multiple dual costs of each resources. In contrast, our work is more general and is applicable to any problem that admits heterogeneous resources.

3 SYSTEM MODEL

In this section, we explore the generic MR-RMB model and its application to various networked systems, including multi-channel wireless networks.

We consider a system composed of N restless arms (wireless clients), numbered by $n = 1, 2, \dots, N$, H heterogeneous resources (wireless channels), numbered as $h = 1, 2, \dots, H$, and a controller (cellular base station) in charge of matching resources to restless arms. Each resource h has a capacity of serving up to C_h restless arms in each time step. To simplify notation, we also introduce a *Null* resource $h = 0$ with infinite capacity, i.e., $C_0 = \infty$. In each time step t, a controller observes the state of each arm n, denoted by $s_{n,t} \in S_n$, and then chooses a resource $a_{n,t} \in \mathbb{A}$, where $\mathbb{A} = \{0, 1, 2, \dots, H\}$, to serve each arm n such that at most c_h arms are served by resource h. If $a_{n,t} = 0$, then arm n is not served by any resource in time step t. After being served by resource $a_{n,t}$, arm n generates a reward $r_{n,t}$ and changes its state to $s_{n,t+1}$ in the next time step. We assume that the reward and state evolution of each arm follows a MDP . Specifically, when arm n is in state s_n and is being served by resource a_n, then it generates a random reward with unknown mean $R_n(s_n, a_n)$ and changes its state to state s'_n with unknown probability $P_n(s'_n|s_n, a_n)$.

There are many real-world scenarios where the controller needs to decide not only which restless arms to serve, but also which resource to use for each arm. We demonstrate three examples of such systems below

Example 1: In multi-channel wireless systems, a base station (BS) serves N data flows (restless arms) with H heterogeneous channels (resources) and $p_{n,h}$ represents the channel quality between user n and channel h. The model for state and reward of a data flow is specified by its application and can be defined based on, for example, queue status, packet delay, data freshness, etc. Each data flow experiences different $p_{n,h}$ on different channels. In each time step, the BS chooses data flows to transmit over each channel.

Example 2: In social media website advertisement, the website has three places to display advertisements: the overhead banner, the sidebar, and the newsfeed. Hence, $H = 3$ and C_h represents the number of spots in each place. Each restless arm is an advertisement whose state includes the time and place it was last displayed. In each time step, the website administrator determines whether and where to display each advertisement. The reward of each advertisement, measured in terms of the click-through rate, depends on the state of the advertisement and the place that it is displayed in.

Example 3: Within a Maternal and Child Health Program, call center services employ live voice scheduling to provide timely preventive care information to expecting and new mothers throughout pregnancy and up to one year post-birth. Call center has H callers available, each caller has a different language expertise and can call up to C_h expecting/new mothers each week. Each expecting/new mother is a restless arm with her own language preference, and her state is either engaged in preventive care or not. Each time step, the call center determines which caller to contact to increase her engagement in preventive care. The effectiveness of a call depends on the mother's state and whether the caller's language expertise matches the mother's. A recent study [31] has studied the special case when all callers have the same language expertise.

The controller employs a matching policy $\vec{\pi}$ to match arms to resources. Let $\vec{s}_t := [s_{1,t}, s_{2,t}, \dots, s_{N,t}]$ and $\vec{a}_t := [a_{1,t}, a_{2,t}, \dots, a_{N,t}]$, then the matching policy can be viewed as a function $\vec{\pi}$ that determines $\vec{a}_t = \vec{\pi}(\vec{s}_t) := [\pi_1(\vec{s}_t), \pi_2(\vec{s}_t), \dots]$. We evaluate $\vec{\pi}$ by its long-term discounted total reward. Specifically, let β be the discount factor and let $\mathbb{I}(\cdot)$ be the indicator function, then the controller's goal is to find the optimal $\vec{\pi}$ for the following optimization problem:

$$\textbf{SYSTEM:} \quad \max_{\vec{\pi}} \; E[\sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t R_n\big(s_{n,t}, \pi_n(\vec{s}_t)\big)] \tag{1}$$

$$\text{s.t.} \sum_{n=1}^{N} \mathbb{I}\big(\pi_n(\vec{s}) = h\big) \le C_h, \forall \vec{s}, h, \tag{2}$$

$$\text{and } \pi_n(\vec{s}) \in \{0, 1, 2, \dots, H\}, \forall \vec{s}, n. \tag{3}$$

4 PRELIMINARY: DECOMPOSITION AND INDEX POLICY

The problem **SYSTEM** is intractable to solve because the state space of $\vec{s}$ is the product of the state spaces of s_n and its size increases exponentially with N. A recent study [38] has employed a decomposition technique to develop an index policy for the problem of minimizing AoI in multi-channel wireless systems. In this section, we generalizes its result for generic MR-RMB problems.

4.1 Lagrange Decomposition

To simplify the **SYSTEM** problem, we first relax the per-$\vec{s}$ constraint (2) to an average constraint:

$$E\left[\sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t \mathbb{I}(\pi_n(\vec{s_t}) = h)\right] \le \sum_{t=0}^{\infty} \beta^t C_h = \frac{C_h}{1-\beta}, \forall h, \tag{4}$$

and then introduce a Lagrange multiplier $\vec{\lambda} := [\lambda_1, \lambda_2, \dots, \lambda_H]$ for this relaxed constraint. We can view $\vec{\lambda}$ as a shadow price so that the controller needs to pay λ_h for every arm that is matched to resource h. The Lagrangian of the relaxed problem is then

$$\textbf{Lagr}(\vec{\lambda})\textbf{:} \max_{\vec{\pi}} \; E\left[\sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t \Big(R_n\big(s_{n,t}, \pi_n(\vec{s}_t)\big) - \lambda_{\pi_n(\vec{s}_t)}\Big)\right],$$
$$\text{s.t. } \pi_n(\vec{s}) \in \{0, 1, 2, \dots, H\}, \forall \vec{s}, n. \tag{5}$$

An important feature of $\textbf{Lagr}(\vec{\lambda})$ is that it can be decomposed into N subproblems, one for each arm. Specifically, let $\sigma_{n,\vec{\lambda}}(\cdot)$ be the optimal solution to the $\textbf{Arm}_n(\vec{\lambda})$ problem,

$$\textbf{Arm}_n(\vec{\lambda})\textbf{:} \max_{\sigma_n} \; E\left[\sum_{t=0}^{\infty} \beta^t \Big(R_n\big(s_{n,t}, \sigma_n(s_{n,t})\big) - \lambda_{\sigma_n(s_{n,t})}\Big)\right],$$
$$\text{s.t. } \sigma_n(s_n) \in \{0, 1, 2, \dots, H\}, \forall s_n, \tag{6}$$

then choosing $\pi_n(\vec{s}) = \sigma^*_{n,\vec{\lambda}}(s_n)$ solves $\textbf{Lagr}(\vec{\lambda})$.

There are three important advantages of the above decomposition. First, this decomposition addresses the curse of dimensionality since each $\textbf{Arm}_n(\vec{\lambda})$ problem only involves the state space of arm n. Second, the decomposition preserves optimality. Specifically, when $\vec{\lambda}$ is chosen as the minimizer of $L(\vec{\lambda}) := \min_{\vec{\lambda}} \max_{\vec{\pi}} E[\sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t \Big(R_n\big(s_{n,t}, \pi_n(\vec{s}_t)\big) - \lambda_{\pi_n(\vec{s}_t)}\Big)]$, then the optimal solution to $\textbf{Arm}_n(\vec{\lambda})$ problem is also the optimal solution to **SYSTEM** with the relaxed constraint Eq. (4). Finally, the minimizer of $L(\vec{\lambda})$ can be found iteratively through a simple gradient algorithm:

$$\lambda_h^{(k+1)} = \left[\lambda_h^{(k)} + \rho_k\Big(E[\sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t \mathbb{I}(\sigma_{n,\lambda_n}(s_{n,t}) = h)] - \frac{C_h}{1-\beta}\Big)\right]^+, \tag{7}$$

where $\lambda_h^{(k)}$ is the value of λ_h in the k−th iteration, ρ_k is a properly chosen step size, and $[x]^+ := \max\{x, 0\}$.

Algorithm 1 Max-Weight Index Matching

```
Initialize $\vec{\lambda}$
for t=0, 1, 2, ... do
    Calculate $w_{n,h}(s_{n,t}, \vec{\lambda}_{-h})$ for all $n, h$
    Create a bipartite graph with $N$ arm nodes and $H + 1$ resource nodes
    Add an edge with weight $w_{n,h}(s_{n,t}, \vec{\lambda}_{-h})$ between arm node $n$ and resource node $h$
    Find the max-weight matching and match arms to resources accordingly
    $\lambda_h \leftarrow \left[\lambda_h + \rho_t\left(\sum_n \mathbb{I}(w_{n,h}(s_{n,t}\vec{\lambda}_{-h}) > \lambda_h) - C_h\right)\right]^+, \forall h$
end for
```

4.2 Partial Index and Max-Weight Index Matching

One important drawback of Lagrange decomposition is that it needs to relax the per-$\vec{s}$ constraint (2). For the special case when there is only one resource, i.e., $H = 1$, the Whittle index overcomes this drawback by producing a policy that both satisfies the per-$\vec{s}$ constraint (2) and is asymptotically optimal under some mild conditions. Following the approach in Zou et al. [38], one can define a *partial index* for each arm n, each state s_n, and each resource h. For a given arm n, it defines a *partial index* for each state s_n of n and each resource h, denoted by $w_{n,h}(s_n, \vec{\lambda}_{-h})$, as the following:

Definition 4.1. [Partial Index] Given an arm n and a Lagrange multiplier $\vec{\lambda}$, the partial index for state s_n and resource h is defined as

$$w_{n,h}(s_n, \vec{\lambda}_{-h}) := sup\{y|\sigma^*_{n,[\vec{\lambda}_{-h},y]}(s_n) = h\}. \tag{8}$$

Intuitively, $w_{n,h}(s_n, \vec{\lambda}_{-h})$ can be viewed as the highest shadow price that arm n is willing to pay to be matched to resource h, instead of any other resources, when its state is s_n. Hence, arm n should prefer resource h over others as long as $\lambda_h \leq w_{n,h}(s_n, \vec{\lambda}_{-h})$. Arm n is said to be *indexable* when it indeed exhibits such a behavior:

Definition 4.2. [Indexability] An arm n is indexable if, for any s_n, $\vec{\lambda}$, and h, we have, for any $y \leq w_{n,h}(s_n, \vec{\lambda}_{-h})$, $\sigma^*_{n,[\vec{\lambda}_{-h},y]}(s_n) = h$.

We now discuss the matching algorithm. Given $\vec{\lambda}$, the algorithm first calculates the partial index for each n, s_n, and h. In each time step t, the algorithm creates a bipartite graph with $N + H + 1$ nodes, such that each arm and each resource is a node. There is an edge between each arm n and each resource h with weight $w_{n,h}(s_{n,t}, \vec{\lambda}_{-h})$. The algorithm then finds the max-weight matching between the nodes, with the constraint that each resource h can be matched to at most C_h arms. We describe the algorithm along with a simplified update rule for $\vec{\lambda}$ in Alg. 1. Zou et al. [38] has proved that Alg. 1 is asymptotically optimal for a specific wireless scheduling problem of AoI minimization. However, to calculate the partial index, one needs to know the transition kernels of each restless arms. In the next two sections, we will introduce an online reinforcement learning algorithm that learns the partial index for restless arms with convoluted and unknown transition kernels.

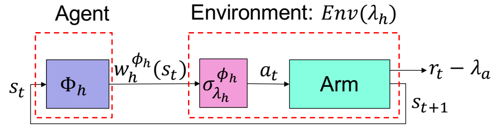


Figure 1: An illustration of Corollary 5.1

5 POLICY GRADIENT THEOREM FOR INDEX LEARNING

In this section, we study the fundamental properties of the partial index $w_{n,h}(s_n, \vec{\lambda}_{-h})$. We will show that $w_{n,h}(s_n, \vec{\lambda}_{-h})$ is the optimal solution to an auxiliary MDP. Nakhleh and Hou [24] has derived the policy gradient theorem for finding the Whittle index, which is equivalent to the special case of $H = 1$ in our paper. We expand Nakhleh and Hou [24] to address the more general case of multiple resources and derive the corresponding policy gradient theorem.

Throughout this section, we will focus on studying $w_{n,h}(s_n, \vec{\lambda}_{-h})$ for fixed n, h, and $\vec{\lambda}_{-h}$. We also assume that the partial indexes for all other resources, that is, $w_{n,h'}(s_n, \vec{\lambda}_{-h'})$ for all $h' \neq h$, are known and given. We drop the subscript n from all notations for better clarity.

Let $w_h^{\phi_h}(s)$ be a function with parameter vector ϕ_h that predicts the partial index $w_h(s, \vec{\lambda}_{-h})$. For a given resource h of an $\mathbf{Arm}_n([\vec{\lambda}_{-h}, \lambda_h])$ problem, we can construct the following policy, which we denote by $\sigma_{\lambda_h}^{\phi_h}(s)$: First, the policy compares $w_h^{\phi_h}(s)$ with λ_h and sets $\sigma_{\lambda_h}^{\phi_h}(s) = h$ if $w_h^{\phi_h}(s) \geq \lambda_h$. Second, if $w_h^{\phi_h}(s) \leq \lambda_h$, then the policy finds the largest $h' \neq h$ with $w_{h'}(s, \vec{\lambda}_{-h'}) \geq \lambda_{h'}$ and sets $\sigma_{\lambda_h}^{\phi_h}(s) = h'$. Finally, if $w_h^{\phi_h}(s) < \lambda_h$ and $w_{h'}(s, \vec{\lambda}_{-h'}) < \lambda_{h'}$ for all $h' \neq h$, then policy sets $\sigma_{\lambda_h}^{\phi_h}(s) = 0$.

We can now define the state-action function of applying $\sigma_{\lambda_h}^{\phi_h}(s)$ to $\mathbf{Arm}_n([\vec{\lambda}_{-h}, \lambda_h])$ problem, which we denote by $Q^{\phi_h}(s, a, \lambda_h)$. The Bellman equation of $Q^{\phi_h}(s, a, \lambda_h)$ is defined as:

$$\begin{aligned} Q^{\phi_h}(s, a, \lambda_h) =& R(s, a) - \lambda_a \\ &+\beta \sum_{s' \in S} P(s'|s, a) Q^{\phi_h}(s', \sigma_{\lambda_h}^{\phi_h}(s'), \lambda_h)) \end{aligned} \tag{9}$$

We then have the following Corollary:

Corollary 5.1. *If arm n is indexable, then setting $w_h^{\phi_h}(s)$ to be its partial index $w_h(s, \vec{\lambda}_{-h})$ maximizes $Q^{\phi_h}(s, a, \lambda_h)$ for any λ_h.*

Proof. If $w_h^{\phi_h}(s) \equiv w_h(s, \vec{\lambda}_{-h})$, then $\sigma_{\lambda_h}^{\phi_h}(s) = g$ if and only if $\lambda_g \leq w_g(s, \vec{\lambda}_{-g})$ for any g. By Definition 4.2, $\sigma_{\lambda_h}^{\phi_h}(s) = \sigma^*_{[\vec{\lambda}_{-h},\lambda_h]}(s)$ for all λ_h and s. Since $\sigma^*_{[\vec{\lambda}_{-h},\lambda_h]}(s)$ is the optimal solution to the $\mathbf{Arm}_n([\vec{\lambda}_{-h}, \lambda_h])$ problem, setting $w_h^{\phi_h}(s) \equiv w_h(s, \vec{\lambda}_{-h})$ maximizes $Q^{\phi_h}(s, a, \lambda_h)$. □

To understand the implication of Corollary 5.1, consider the reinforcement learning problem as illustrated in Fig. 1 that contains an agent and an environment called $Env(\lambda_h)$. In each time step, the agent observes the state s_t and chooses a real number as its control decision. When $Env(\lambda_h)$ receives the control decision, it first treats the control decision as $w_h^{\phi_h}(s_t)$ and employs $\sigma_{\lambda_h}^{\phi_h}$ to determine a_t. It then feeds a_t to the restless arm to generate the next state s_{t+1} and the net reward $r_t - \lambda_{a_t}$. The agent's goal is to find a control policy that can maximize the long-term average net reward, $\sum_{t=0}^{\infty} \beta^t E[r_t - \lambda_{a_t}]$. Corollary 5.1 states that the partial index $w_h(s, \vec{\lambda}_{-h})$ is the optimal control policy for $Env(\lambda_h)$ for all λ_h.

Based on Corollary 5.1, we define the objective function for learning ϕ_h as

$$J^{\phi_h} = \sum_{s \in S} \int_{\lambda_h=-M}^{\lambda_h=+M} Q^{\phi_h}(s, \sigma_{\lambda_h}^{\phi_h}(s), \lambda_h) d\lambda_h, \tag{10}$$

where M is a sufficiently large constant such that $\lambda_h \in [-M, +M]$. If J^{ϕ_h} has a unique maximizer, then maximizing J^{ϕ_h} is equivalent to finding ϕ_h such that $w_h^{\phi_h}(\cdot)$ is the partial index. We therefore seek to find the partial index by finding ϕ_h that maximizes J^{ϕ_h}. We will find $\nabla_{\phi_h} J^{\phi_h}$ as shown in Theorem 5.2 below.

Theorem 5.2. *Given the parameter vector ϕ_h, if all states $s \in S$ have distinct values of $w_h^{\phi_h}(s)$, then the gradient of the objective function J^{ϕ_h} with respect to the parameter vector ϕ_h is given by:*

$$\begin{aligned}\nabla_{\phi_h} J^{\phi_h} = &\sum_{s \in S} \Big[Q^{\phi_h}(s, h, w_h^{\phi_h}(s)) \; - Q^{\phi_h}(s, \sigma'_{\lambda_h}(s), w_h^{\phi_h}(s)) \Big] \\ &\nabla_{\phi_h} w_h^{\phi_h}(s),\end{aligned} \tag{11}$$

where

$$\sigma'_{\lambda_h}(s) = \begin{cases} 0, & \text{if } w_{h'}(s, \vec{\lambda}_{-h'}) < \lambda_{h'}, \forall h' \neq h, \\ \max\{h' | w_{h'}(s, \vec{\lambda}_{-h'}) \geq \lambda_{h'}\}, & \text{otherwise.} \end{cases} \tag{12}$$

Proof. Taking the gradient of Eq. (10) yields

$$\nabla_{\phi_h} J^{\phi_h} = \nabla_{\phi_h} \sum_{s \in S} \int_{\lambda_h=-M}^{\lambda_h=+M} Q^{\phi_h}(s, \sigma_{\lambda_h}^{\phi_h}(s), \lambda_h) d\lambda_h. \tag{13}$$

Renumber all states in S such that $w_h^{\phi_h}(s_{|S|}) > w_h^{\phi_h}(s_{|S|-1}) > \dots w_h^{\phi_h}(s_1)$. Let $\mathbb{M}^0 = -M$, $\mathbb{M}^k = w_h^{\phi_h}(s_k)$, for each $1 \leq k \leq |S|$, and $\mathbb{M}^{|S|+1} = +M$. Also let S_k be the subset of states $\{s_k, s_{k+1}, ..., s_{|S|}\}$, for each $1 \leq k \leq |S|$. Then, for any k, S_k is the subset of states with $w_h^{\phi_h}(s) \geq \mathbb{M}^k$. Hence, for any λ_h in the interval $(\mathbb{M}^k, \mathbb{M}^{k+1})$, we have

$$\sigma_{\lambda_h}^{\phi_h}(s) = \begin{cases} h, & \text{if } s \in S_{k+1}, \\ \sigma'_{\lambda_h}(s), & \text{otherwise.} \end{cases} \tag{14}$$

Define the policy $\hat{\sigma}_{\lambda_h}^{k+1}(s)$ as the policy that chooses action h when $\mathbb{I}(s \in S_{k+1}) = 1$ and $\sigma'_{\lambda_h}(s)$ otherwise. Let $\hat{Q}_{k+1}(s, a, \lambda_h)$ be the state-action function of applying $\hat{\sigma}_{\lambda_h}^{k+1}(s)$. Then, for any $\lambda_h \in (\mathbb{M}^k, \mathbb{M}^{k+1})$, $\sigma_{\lambda_h}^{\phi_h}(s) = \hat{\sigma}_{\lambda_h}^{k+1}(s)$ for all s, and therefore $Q^{\phi_h}(s, a, \lambda_h) = \hat{Q}_{k+1}(s, a, \lambda_h)$ for all s and a. We can now rewrite Eq. (13) as

$$\nabla_{\phi_h} J^{\phi_h} = \sum_{k=0}^{|S|} \sum_{s \in S} \nabla_{\phi_h} \int_{\lambda_h=\mathbb{M}^k}^{\lambda_h=\mathbb{M}^{k+1}} \hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \lambda_h) d\lambda_h. \tag{15}$$

Applying Leibniz integral rule and we have:

$$\begin{aligned}\nabla_{\phi_h} J^{\phi_h} = &\sum_{k=0}^{|S|} \sum_{s \in S} \Big[\hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \mathbb{M}^{k+1}) \; \nabla_{\phi_h} \mathbb{M}^{k+1} \\ &-\hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \mathbb{M}^k) \nabla_{\phi_h} \mathbb{M}^k \Big] \\ &+ \sum_{k=0}^{|S|} \sum_{s \in S} \int_{\lambda_h=\mathbb{M}^k}^{\lambda_h=\mathbb{M}^{k+1}} \nabla_{\phi_h} \hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \lambda_h) d\lambda_h.\end{aligned} \tag{16}$$

Since $\hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \lambda_h)$ is constant with respect to ϕ_h, $\nabla_{\phi_h} \hat{Q}_{k+1}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \lambda_h) = 0$. Moreover, for any k, $\hat{\sigma}_{\lambda_h}^{k+1}(s) = \hat{\sigma}_{\lambda_h}^{k}(s)$ for all s except s_k. Hence, we can further simplify Eq. (16) and obtain

$$\begin{aligned}&\nabla_{\phi_h} J^{\phi_h} \\ =&\sum_{k=0}^{|S|} \sum_{s \in S} \Big[Q^{\phi_h}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \mathbb{M}^{k+1}) \nabla_{\phi_h} \mathbb{M}^{k+1} \\ &-Q^{\phi_h}(s, \hat{\sigma}_{\lambda_h}^{k+1}(s), \mathbb{M}^k) \nabla_{\phi_h} \mathbb{M}^k \Big] \\ =&\sum_{k=1}^{|S|} \Big[Q^{\phi_h}(s_k, \hat{\sigma}_{\lambda_h}^{k}(s_k), \mathbb{M}^k) - Q^{\phi_h}(s_k, \hat{\sigma}_{\lambda_h}^{k+1}(s_k), \mathbb{M}^k) \Big] \nabla_{\phi_h} \mathbb{M}^k \\ =&\sum_{k=1}^{|S|} \Big[Q^{\phi_h}(s_k, h, w_h^{\phi_h}(s_k)) - Q^{\phi_h}(s_k, \sigma'_{\lambda_h}(s_k), w_h^{\phi_h}(s_k)) \Big] \nabla_{\phi_h} w_h^{\phi_h}(s_k) \\ =&\sum_{s \in S} \Big[Q^{\phi_h}(s, h, w_h^{\phi_h}(s)) \; -Q^{\phi_h}(s, \sigma'_{\lambda_h}(s), w_h^{\phi_h}(s)) \Big] \nabla_{\phi_h} w_h^{\phi_h}(s)\end{aligned}$$

This completes the proof. □

6 DEEP INDEX POLICY

In this section, we introduce DIP, a new Deep Index Policy, for online learning of the partial indexes by leveraging the policy gradient theorem for index learning (Thm. 5.2). For each restless arm n, DIP maintains H actor networks, parameterized by $\phi_{n,1}, \phi_{n,2}, \dots$, and a critic network, parameterized by θ_n. Each actor network $\phi_{n,h}$ takes $(s_n, \vec{\lambda}_{-h})$ as input and produces a number $w_{n,h}^{\phi_{n,h}}(s_n, \vec{\lambda}_{-h})$ as its output. DIP aims to train $\phi_{n,h}$ such that the actor network predicts the partial index, that is, $w_{n,h}^{\phi_{n,h}}(s_n, \vec{\lambda}_{-h}) \approx w_{n,h}(s_n, \vec{\lambda}_{-h})$, for all n and h. Each critic network θ_n takes $(s_n, a_n, \vec{\lambda})$ as input and produces a number $Q_n^{\theta_n}(s_n, a_n, \vec{\lambda})$. DIP aims to train θ_n such that the critic network predicts the state-action function of applying the optimal policy to the $\mathbf{Arm}_n(\vec{\lambda})$ problem, which we denote by $Q_n^*(s_n, a_n, \vec{\lambda})$. The Bellman equation of $Q_n^*(s_n, a_n, \vec{\lambda})$ is

$$\begin{aligned}Q_n^*(s_n, a_n, \vec{\lambda}) =& R_n(s_n, a_n) - \lambda_{a_n} \\ &+\beta \sum_{s'_n \in S} P(s'_n | s_n, a_n) \max_{a'} Q^*(s'_n, a', \lambda_h).\end{aligned} \tag{17}$$

Algorithm 2 Deep Index Learning

```
Initialize $\vec{\lambda}$, $\phi_{n,h}$, and $\theta_n$
Initialize a replay memory $\mathcal{M}_n$ for each $n$
$\theta'_n \leftarrow \theta_n$
for t=0, 1, 2, ... do
    $x \sim U(0,1)$
    if $x < \epsilon$ then
        Randomly match arms to resources
    else
        Create a bipartite graph with $N$ arm nodes and $H$+1 resource nodes
        Add an edge with weight $w_{n,h}^{\phi_{n,h}}(s_{n,t}, \vec{\lambda}_{-h})$ between arm node $n$ and resource node $h$
        Find the max-weight matching and match arms to resources accordingly
        $\lambda_h \leftarrow \Big[\lambda_h + \rho_t\Big(\sum_n \mathbb{I}(w_{n,h}^{\phi_{n,h}}(s_{n,t}\vec{\lambda}_{-h}) > \lambda_h) - C_h\Big)\Big]^+, \forall h$
    end if
    Save $(s_{n,t}, a_{n,t}, r_{n,t}, s_{n,t+1})$ to $\mathcal{M}_n$
    Run Alg. 3 to update all neural networks
end for
```

DIP also maintains a target critic network with parameters θ'_n for each n. The target critic networks are updated at a slower rate compared to critic parameters θ_n to ensure stability and improve the training process by providing consistent target values for the critic network [21].

We first provide an overview of the procedure of DIP. In each time step, DIP employs a exploration-exploitation policy similar to the $\epsilon-$greedy policy. With probability ϵ, DIP randomly assigns restless arms to resources for the purpose of exploration. With probability $1-\epsilon$, DIP employs Max-Weight Index Matching where the weight between restless arm n and resource h is set to be $w_{n,h}^{\phi_{n,h}}(s_{n,t}, \vec{\lambda}_{-h})$. DIP observes the actions, rewards, and state transitions of all restless arms and store them in a replay buffer. DIP then updates the value of $\vec{\lambda}$ by $\lambda_h \leftarrow \Big[\lambda_h + \rho_t\Big(\sum_n \mathbb{I}(w_{n,h}^{\phi_{n,h}}(s_{n,t}\vec{\lambda}_{-h}) > \lambda_h) - C_h\Big)\Big]^+$, for all h. Finally, DIP updates all actor networks, critic networks, and target critic networks. DIP is based on Deep Deterministic Policy Gradient (DDPG), an off-policy reinforcement learning algorithm, to learn the partial indexes. This off-policy nature is crucial because it allows the learning of the partial index that optimizes the fictitious policy described in Thm. 5.2, while the max-weight matching policy is being executed. Alg. 2 describes the detailed procedure of DIP.

We now discuss how DIP updates all neural networks. For each n, DIP randomly samples a batch of transitions $(s_{n,t}, a_{n,t}, r_{n,t}, s_{n,t+1})$ from the replay buffer and attach a randomly selected $\vec{\lambda} \in [-M, +M]^H$ to each transition. For each transition, DIP calculates

$$\begin{aligned}&[Q_n^{\theta_n}(s_{n,t}, a_{n,t}, [\vec{\lambda}_{-a_{n,t}}, w_{n,a_{n,t}}^{\phi_{n,a_{n,t}}}(s_{n,t}, \vec{\lambda}_{-a_{n,t}})])\\ &-Q_n^{\theta_n}(s_{n,t}, \sigma'_{\lambda_{a_{n,t}}}(s), [\vec{\lambda}_{-a_{n,t}}, w_{n,a_{n,t}}^{\phi_{n,a_{n,t}}}(s_{n,t}, \vec{\lambda}_{-a_{n,t}})])]\\ &\times\nabla_{\phi_{n,a_{n,t}}} w_{n,a_{n,t}}^{\phi_{n,a_{n,t}}}(s_{n,t}, \vec{\lambda}_{-a_{n,t}}) \qquad (18)\end{aligned}$$

as an approximation of Eq. (11) and uses it to update the actor networks $\phi_{n,a_{n,t}}$.

Algorithm 3 Neural Networks Updates

```
for n= 1, 2, ..., N do
    Sample a mini batch $B$ transitions $(s_{n,t_k}, a_{n,t_k}, r_{n,t_k}, s_{n,t_{k+1}})$, for $1 \le k \le B$ from memory $\mathcal{M}_n$.
    Randomly sample $B$ different shadow price $[\vec{\lambda}_1, \vec{\lambda}_2, \dots \vec{\lambda}_B]$ from $[-M, M]^H$.
    $\Delta\phi_{n,h} \leftarrow 0$, for all $h$; $\Delta\theta_n \leftarrow 0$
    for k=1, 2, ..., B do
        Set $\delta$ to be Eq. (18) for the $k^{th}$ element in the batch
        $\Delta\phi_{n,a_{n,t_k}} \leftarrow \Delta\phi_{n,a_{n,t_k}} + \delta$
        Set $\delta$ to be Eq. (20) for the $k^{th}$ element in the batch
        $\Delta\theta_n \leftarrow \Delta\theta_n + \delta$
    end for
    Update $\phi_{n,h}$ by $\Delta\phi_{n,h}$ and $\theta_n$ by $\Delta\theta_n$
    $\theta'_n \leftarrow \tau\theta_n + (1-\tau)\theta'_n$
end for
```

DIP then uses the same batch of transitions and $\vec{\lambda}$ to update the critic network. Based on the Bellman equation of $Q_n^*(s_n, a_n, \vec{\lambda})$ (Eq. (17)), DIP defines the loss function of the critic function as

$$\begin{aligned}\mathcal{L}_n^{\theta_n} =&\mathbb{E}\Bigg[\Big(Q_n^{\theta_n}(s_{n,t}, a_{n,t}, \vec{\lambda}) - r_{n,t} + \lambda_{a_{n,t}}\\ &-\beta\max_{a'} Q_n^{\theta'_n}(s_{n,t+1}, a', \vec{\lambda})\Big)^2\Bigg]. \qquad (19)\end{aligned}$$

DIP then uses the each transition and $\vec{\lambda}$ from the batch to estimate the gradient of the loss function by

$$\begin{aligned}&2\Big(Q_n^{\theta_n}(s_{n,t}, a_{n,t}, \vec{\lambda}) - r_{n,t} + \lambda_{a_{n,t}}\\ &-\beta\max_{a'} Q_n^{\theta'_n}(s_{n,t+1}, a', \vec{\lambda})\Big)\nabla_{\theta_n} Q_n^{\theta_n}(s_{n,t}, a_{n,t}, \vec{\lambda}). \qquad (20)\end{aligned}$$

and to update θ_n. Finally, DIP soft updates the target critic network by $\theta'_n \leftarrow \tau\theta_n + (1-\tau)\theta'_n$. The training complexity scales with the product of the number of arms and the number of resources i.e., $O(N \times H)$, as each (arm, resource) pair is trained independently.

7 SIMULATIONS

In this section, we present our simulation results that evaluate DIP in three different MR-RMB problems. The first two problems are scheduling problems in multi-channel wireless networks with one on minimizing AoI and the other on minimizing holding cost. The third problem considers advertisements placements in social media websites.

We compare the performance of DIP against domain-specific policies in each problem. We also evaluate DeepTOP [24] in all problems. DeepTOP is a deep online learning algorithm that finds the Whittle index when there is only one kind of resource. In order to incorporate DeepTOP in multi-resource problems, we consider a policy that selects the restless arms with the highest indexes to activate, and then randomly matches selected restless arms to resources. We then train DeepTOP with respect to this policy.

All simulation results are the average of 20 independent runs, with error bars indicating standard deviations. Each run consists of 12,000 time steps. For each time step, we obtain the running average performance of the past 100 time steps. The value of $\vec{\lambda}$ is updated every 100 time steps and we have set the discounted factor to 0.99 (i,e., $\beta = 0.99$) and the learning rate of $\vec{\lambda}$ to 0.01 (i.e., $\rho = 0.01$) for both problems. The learning rates of neural networks are determined by the ADAM optimizer. All neural networks of DIP have two fully connected hidden layers with 128 neurons in each hidden layer. We have use the same setting for DeepTOP.

7.1 Network Setting

We introduce the settings of multi-channel wireless networks that will be used in both the AoI minimization problem and the holding cost minimization problem. We consider two types of systems: heterogeneous channels and homogeneous channels. Each type of system has two different settings. In all settings, we consider the challenge that wireless transmissions are not reliable. When a mobile user n is scheduled to transmit on channel h, the transmission will be successful with probability $p_{n,h}$.

We have two settings for heterogeneous channels, one with two channels and the other with three channels. For the two-channel system, we assume that there are 20 mobile users. The first 14 users have $p_{n,1} = 0.7$ and $p_{n,2} = 0.3$. The other 6 users have $p_{n,1} = 0.3$ and $p_{n,2} = 0.7$. Each channel has a capacity of two, that is, $C_h = 2$ for all h. For the three-channel system, we assume that there are 34 mobile users. The first 20 users have $p_{n,1} = 0.9, p_{n,2} = 0.5, p_{n,3} = 0.1$, the next 4 users have $p_{n,1} = 0.1, p_{n,2} = 0.9, p_{n,3} = 0.5$, and the last 10 users have $p_{n,1} = 0.5, p_{n,2} = 0.1, p_{n,2} = 0.9$. Each channel has a capacity of two.

We also have two settings for homogeneous channels, one with two channels and the other with three channels. For the two-channel system, we assume that there are 20 mobile users. The first 14 users have $p_{n,1} = p_{n,2} = 0.7$. The other 6 users have $p_{n,1} = p_{n,2} = 0.3$. Each channel has a capacity of two. For the three-channel system, we assume that there are 34 mobile users. The first 20 users have $p_{n,1} = p_{n,2} = p_{n,3} = 0.9$, the next 4 users have $p_{n,1} = p_{n,2} = p_{n,3} = 0.7$, and the last 10 users have $p_{n,1} = p_{n,2} = p_{n,3} = 0.5$. Each channel has a capacity of two.

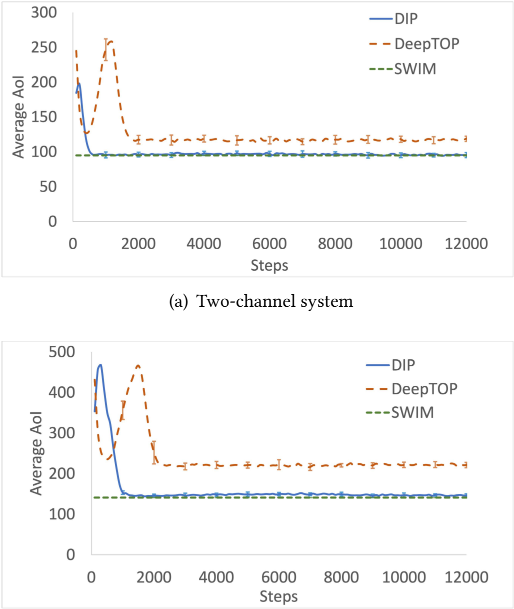
We note that the homogeneous channel systems are equivalent to single-channel systems where the capacity of the channel is HC_h. Since DeepTOP learns the Whittle index in single-channel systems, we can use the performance of DeepTOP as that by the Whittle index policy in homogeneous channel systems.

7.2 AoI Minimization

AoI has gained significant research interests due to its elegance in capturing information freshness. We define the AoI of a mobile user recursively as follows: At time $t = 0$ the AoI of the user is 1. In each subsequent time step, the AoI increases by 1 if there is no packet delivery for the user, either because the user is not scheduled or because the transmission fails, and the AoI becomes 1 if there is a packet delivery.

We can model the AoI of user n as a MDP where the state of the user $s_{n,t}$ is its AoI. To ensure a finite state space, we cap the AoI at 20. If user n is not scheduled to any channel, then its AoI will increase by one, and hence

$$P_n(s_{n,t+1} = \min\{s+1, 20\} | s_{n,t} = s, a_{n,t} = 0) = 1. \qquad (21)$$



(a) Two-channel system

(b) Three-channel system

Figure 2: AoI comparison for multi-channel wireless networks with heterogeneous channels

On the other hand, if user n is scheduled to transmit on channel h, then user n will successfully deliver a packet with probability $p_{n,h}$. Hence, we have

$$P_n(s_{n,t+1} = \min\{s+1, 20\} | s_{n,t} = s, a_{n,t} = h) = 1 - p_{n,h}, \qquad (22)$$

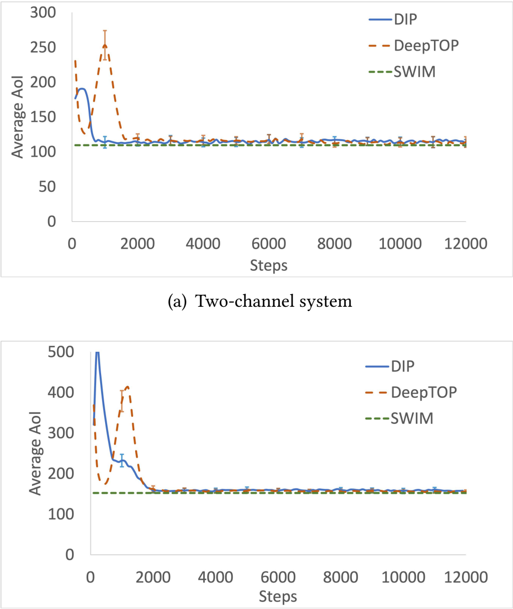
$$P_n(s_{n,t+1} = 1 | s_{n,t} = s, a_{n,t} = h) = p_{n,h}. \qquad (23)$$

The reward of user n is $R_n(s_{n,t}, a_{n,t}) = -s_{n,t+1}$. The objective is to minimize the total long-term discounted AoI of all users in the system.

A recent paper [38] has studied the AoI minimization problem in multi-channel wireless networks. It has proposed a Sum Weighted Index Matching (SWIM) policy. SWIM calculates the partial indexes of all users on all channels and then use max-weight matching to schedule users. However, SWIM requires the precise knowledge of all $p_{n,h}$ to calculate the partial indexes. Since our DIP aims to learn the partial indexes without any prior knowledge of the system, we can use SWIM as a baseline policy.

Fig. 2 shows the simulation results for heterogeneous multi-channel wireless networks. It can be observed that the AoI of DIP converges to the AoI of SWIM in less than 1,000 time steps in both the two-channel system and the three-channel system. This shows that DIP has indeed efficiently learned the partial indexes and the Lagrange multipliers. It can also be observed that DeepTOP is considerably worse than DIP. This shows the standard Whittle index, which is designed for systems with only one kind of resource, does not work well in multi-resource systems.

Simulation results for homogeneous multi-channel wireless networks are shown in Fig. 3. As discussed in the previous section, these systems are equivalent to single-channel systems. In such



(a) Two-channel system

(b) Three-channel system

Figure 3: AoI comparison for multi-channel wireless networks with homogeneous channels.

systems, the Whittle index policy is near-optimal. Indeed, the performance of DeepTOP converges to SWIM. DIP also converges to SWIM in less than 2,000 time steps.

7.3 Holding Cost Minimization

We consider the problem of minimizing holding costs. In this problem, the base station maintains a queue of undelivered packets fro each mobile user. In each time slot, a packet arrives for user n with probability ζ_n. A packet is delivered for user n whenever user n has a successful transmission. In each time slot, each user n incurs a holding cost of its queue size squared.

We can model this problem as a MDP where the state of user n, $s_{n,t}$, is its queue size. To ensure a finite state space, we cap the queue size at 20. If user n is not scheduled to any channel, then its queue size will increase if there is a packet arrival, and will remain the same, otherwise. Hence,

$$P_n(s_{n,t+1} = \min\{s+1, 20\} | s_{n,t} = s, a_{n,t} = 0) = \zeta_n, \tag{24}$$

$$P_n(s_{n,t+1} = s | s_{n,t} = s, a_{n,t} = 0) = 1 - \zeta_n. \tag{25}$$

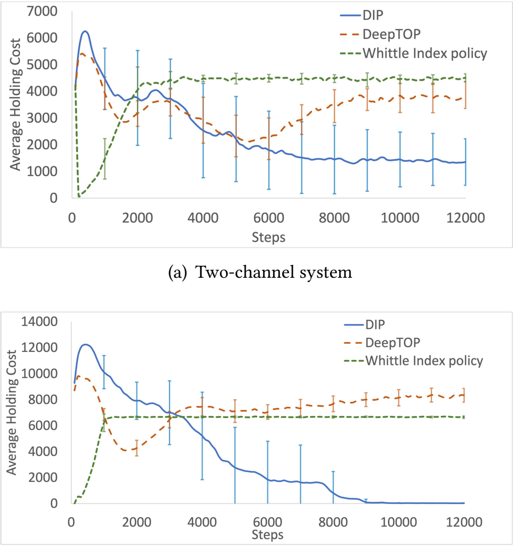
If user n is scheduled to channel h, then it will have a packet departure with probability $p_{n,h}$. It will have a packet arrival with probability ζ_n. Hence, we have

$$P_n(s_{n,t+1} = \min\{s+1, 20\} | s_{n,t} = s, a_{n,t} = h) = (1 - p_{n,h})\zeta_n, \tag{26}$$

$$P_n(s_{n,t+1} = s | s_{n,t} = s, a_{n,t} = h) = (1 - p_{n,h})(1 - \zeta_n) + p_{n,h}\zeta_n, \tag{27}$$

$$P_n(s_{n,t+1} = \max\{s-1, 0\} | s_{n,t} = s, a_{n,t} = h) = p_{n,h}(1 - \zeta_n). \tag{28}$$

The reward of user n is $R_n(s_{n,t}, a_{n,t}) = -s_{n,t}^2$.



(a) Two-channel system

(b) Three-channel system

Figure 4: Holding cost comparison for multi-channel wireless networks with heterogeneous channels

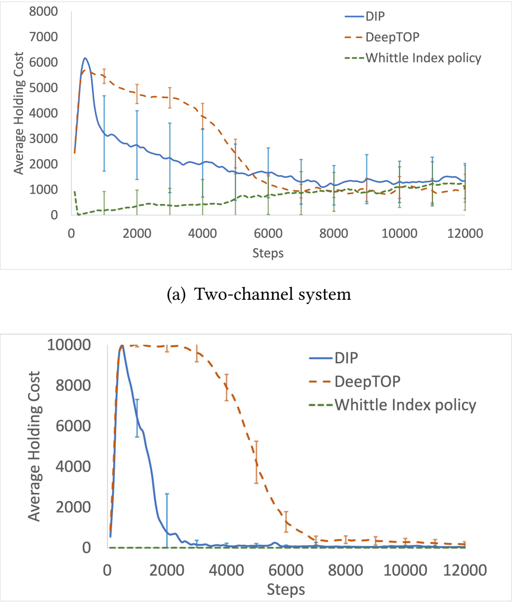
Ansell et al. [3] has studied the scheduling problem for holding cost minimization for the special case of single-channel wireless networks. It has derived the Whittle index for this special case. To employ Ansell et al. [3] for our multi-channel wireless networks, we first assign each user n to a channel h_n^* that has the highest reliability, that is, $h_n^* := \arg\max_h \{p_{n,h}\}$. We then calculate the Whittle index under this channel assignment, which Ansell et al. has shown to be

$$\frac{3\zeta_n - p_{n,h_n^*}}{p_{n,h_n^*} - \zeta_n} + 2p_{n,h_n^*} s_{n,t}, \tag{29}$$

and schedules users according to their Whittle indexes. We call this policy the Whittle index policy.

Fig. 4 shows the simulation results for heterogeneous multi-channel wireless networks, where we set $\zeta_n = 0.11$ for both the two-channel system and the three-channel system. It can be observed that DIP significantly outperforms the Whittle index policy and DeepTOP. While the Whittle index policy computes the Whittle index, it can only assign a user to its best channel. On the other hand, DIP allows a user to be matched to the worse channel when its best channel is too congested. The big performance gap between the Whittle index policy and DIP highlights the additional challenges faced in MR-RMB problems.

Simulation results for homogeneous multi-channel wireless networks are shown in Fig. 5. We set $\zeta_n = 0.1$ for the two-channel system and $\zeta_n = 0.08$ for the three-channel system so that the arrival rates are close to the boundaries of the capacity regions. We observe that all three policies converge to the same holding cost. This is to be expected since these networks are equivalent to single-channel wireless networks. The fact that DIP converges to the Whittle index policy suggests that DIP indeed learns the Whittle index.



(a) Two-channel system

(b) Three-channel system

Figure 5: Holding cost comparison for multi-channel wireless networks with homogeneous channels

7.4 Online Advertisement Placement

We consider an online advertisement placement problem on social media platforms. There are three places where advertisers can display their ads: newsfeed, overhead banner, and sidebar. At each time step, the website administrator determines whether and where to display each advertisement. The consumer interest in a particular advertisement finishes if they click its link and interest may gradually recover over time. For example, someone who recently a purchased a product may not be interested in advertisement for the same product in the immediate future but their interest might gradually revive with time. The objective is to strategically display advertisements in a manner that effectively captures and sustains consumer interest.

We can model the online advertisement placement problem as a recovering bandit, first introduced by [25]. The objective of recovering bandits is to capture the evolving behaviors of consumers over time. The effectiveness of displaying an advertisement depends on the elapsed time since the advertisement was last displayed and the placement. At time $t = 0$, the elapsed time of all advertisements are set to 1. If the advertisement is not displayed, then elapsed time increased by 1 and if the advertisement is displayed then it is set to the 1.

We formulate recovering bandit as a MDP where each arm of the recovering bandit is the advertisement. The state of the advertisement $s_{n,t}$ is the elapsed time. To ensure the finite state space, we cap the elapsed time at 20. If advertisement n is not displayed to any placement, then its wait time will increase by one, and hence

$$P_n(s_{n,t+1} = \min\{s+1, 20\} | s_{n,t} = s, a_{n,t} = 0) = 1. \tag{30}$$

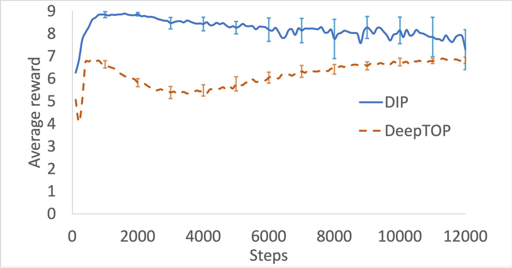


Figure 6: Average reward comparison for online advertisement placement on social media

On the other hand, if advertisement n is scheduled to display on placement h, then its wait time will set to one, and hence

$$P_n(s_{n,t+1} = 1 | s_{n,t} = s, a_{n,t} = h) = 1. \tag{31}$$

The reward of advertisement n is $R_n(s_{n,t}, a_{n,t}) = \theta^0_{n,h}(1 - e^{\theta^1_{n,h} \cdot s_{n,t}})$, where $\theta^0_{n,h}\cdot$ and $\theta^1_{n,h}$are the hyperparameters for the placement h. The objective is to maximize the total long-term discounted reward of all advertisements in the system.

We consider a setup of 30 advertisements and three advertisement placement (i.e., $N = 30$, $H = 3$). Each placement can accommodate up to two advertisements at a time (i.e., $C_h = 2$). The hyperparameters of the first 10 advertisements are $\theta^0_{n,1} = 1, \theta^0_{n,2} = 3, \theta^0_{n,3} = 5$,the next 10 advertisements are $\theta^0_{n,1} = 5, \theta^0_{n,2} = 1, \theta^0_{n,3} = 3$ and the last 10 advertisements are $\theta^0_{n,1} = 3, \theta^0_{n,2} = 5, \theta^0_{n,3} = 1$. Additionally, the hypeparameter $\theta^1_{n,h} = 0.1$ for all advertisements n and all placements h.

Fig. 6 illustrates the simulation results for online advertisement. It is evident that DIP significantly outperforms DeepTOP. Specifically, DIP demonstrates efficient scheduling of advertisements, particularly in situations of high competition for display space. Conversely, DeepTOP tends to prioritize displaying advertisements in less favorable placements when the preferred placements are congested. This indicates that DeepTOP is limited to handling a single type of resource and does not perform effectively in multi-resource systems.

8 CONCLUSION

We address the critical challenges of allocating multiple heterogeneous resources in wireless communication systems. We have derived the policy gradient theorem to learn the index. By leveraging policy gradient theorem, we have proposed DIP, an online reinforcement learning algorithm for MR-RMB. Our results show that DIP outperforms existing methods such as DeepTOP and Whittle Index policy, highlighting the limitations of Whittle index based approaches in heterogeneous multi-resource systems. We have also compared DIP algorithm with SWIM and shows that DIP learned partial index and dual cost correctly in heterogeneous channels setting. We have also shown that all the results of different policies converge in homogeneous channels setting, signifying that DIP has indeed learned the Whittle index correctly, as homogeneous channels mimic the behavior of a single-channel wireless network. Our findings underscore the versatility of DIP across a wide range of scheduling problems, from homogeneous to heterogeneous resource settings, when transition kernel is unknown and convoluted. We have also generalized DIP to applications beyond multi-channels. For future study, extending the MR-RMB model to accommodate multiple conflicting objectives could enhance optimization in wireless communication systems.

9 ACKNOWLEDGEMENT

This material is based upon work supported in part by NSF under Award Numbers ECCS-2127721 and CCF-2332800 and in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant Number W911NF-22-1-0151.

REFERENCES

[1] Samuli Aalto, Pasi Lassila, and Prajwal Osti. 2015. 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*. 57–69.

[2] Arjun Anand and Gustavo de Veciana. 2018. A Whittle's index based approach for qoe optimization in wireless networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 1–39.

[3] PS Ansell, Kevin D Glazebrook, José Nino-Mora, and M O'Keeffe. 2003. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research* 57 (2003), 21–39.

[4] Shreeshankar Bodas, Sanjay Shakkottai, Lei Ying, and R Srikant. 2013. Scheduling in multi-channel wireless networks: Rate function optimality in the small-buffer regime. *IEEE Transactions on Information Theory* 60, 2 (2013), 1101–1125.

[5] Vivek S Borkar, Gaurav S Kasbekar, Sarath Pattathil, and Priyesh Y Shetty. 2017. Opportunistic scheduling as restless bandits. *IEEE Transactions on Control of Network Systems* 5, 4 (2017), 1952–1961.

[6] Marcel C Castro, Peter Dely, Andreas J Kassler, and Nitin H Vaidya. 2009. Qos-aware channel scheduling for multi-radio/multi-channel wireless mesh networks. In *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*. 11–18.

[7] Gongpu Chen, Soung Chang Liew, and Yulin Shao. 2022. Uncertainty-of-information scheduling: A restless multiarmed bandit framework. *IEEE Transactions on Information Theory* 68, 9 (2022), 6151–6173.

[8] Wei Cheng, Xiuzhen Cheng, Taieb Znati, Xicheng Lu, and Zexin Lu. 2009. The complexity of channel scheduling in multi-radio multi-channel wireless networks. In *IEEE INFOCOM 2009*. IEEE, 1512–1520.

[9] Emmanouil Fountoulakis, Themistoklis Charalambous, Anthony Ephremides, and Nikolaos Pappas. 2023. Scheduling policies for AoI minimization with timely throughput constraints. *IEEE Transactions on Communications* (2023).

[10] Yi Gai, Bhaskar Krishnamachari, and Mingyan Liu. 2011. On the combinatorial multi-armed bandit problem with Markovian rewards. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 1–6.

[11] Aditya Gopalan, Constantine Caramanis, and Sanjay Shakkottai. 2012. On wireless scheduling with partial channel-state information. *IEEE Transactions on Information Theory* 58, 1 (2012), 403–420.

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

[13] Shanfeng Huang, Bojie Lv, Rui Wang, and Kaibin Huang. 2020. Scheduling for mobile edge computing with random user arrivals—An approximate MDP and reinforcement learning approach. *IEEE Transactions on Vehicular Technology* 69, 7 (2020), 7735–7750.

[14] Igor Kadota, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. 2018. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2637–2650.

[15] Jackson A Killian, Andrew Perrault, and Milind Tambe. 2021. Beyond" to act or not to act": Fast lagrangian approaches to general multi-action restless bandits. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 710–718.

[16] Yang G Kim and Myung J Lee. 2014. Scheduling multi-channel and multi-timeslot in time constrained wireless sensor networks via simulated annealing and particle swarm optimization. *IEEE communications Magazine* 52, 1 (2014), 122–129.

[17] Subhashini Krishnasamy, PT Akhil, Ari Arapostathis, Rajesh Sundaresan, and Sanjay Shakkottai. 2018. Augmenting max-weight with explicit learning for wireless scheduling with switching costs. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2501–2514.

[18] Alex S Leong, Arunselvan Ramaswamy, Daniel E Quevedo, Holger Karl, and Ling Shi. 2020. Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems. *Automatica* 113 (2020), 108759.

[19] Gang Li, Chunjing Hu, Tao Peng, Xiaohui Zhou, and Yueqing Xu. 2018. High-Priority Minimum-Interference Channel Assignment in Multi-Radio Multi-Channel Wireless Networks. In *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering*. 314–318.

[20] Songhua Li and Lingjie Duan. 2023. Age of Information Diffusion on Social Networks: Optimizing Multi-Stage Seeding Strategies. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 81–90.

[21] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

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

[23] Navid Naderializadeh, Jaroslaw J Sydir, Meryem Simsek, and Hosein Nikopour. 2021. Resource management in wireless networks via multi-agent deep reinforcement learning. *IEEE Transactions on Wireless Communications* 20, 6 (2021), 3507–3523.

[24] Khaled Nakhleh and I-Hong Hou. 2022. DeepTOP: Deep threshold-optimal policy for MDPs and RMABs. *Advances in Neural Information Processing Systems* 35 (2022), 28734–28746.

[25] Ciara Pike-Burke and Steffen Grunewalder. 2019. Recovering bandits. *Advances in Neural Information Processing Systems* 32 (2019).

[26] Shang-Pin Sheng, Mingyan Liu, and Romesh Saigal. 2014. Data-driven channel modeling using spectrum measurement. *IEEE Transactions on Mobile Computing* 14, 9 (2014), 1794–1805.

[27] David Simchi-Levi, Rui Sun, and Xinshang Wang. 2023. Online Matching with Bayesian Rewards. *Operations Research* (2023).

[28] Bejjipuram Sombabu and Sharayu Moharir. 2020. Age-of-information based scheduling for multi-channel systems. *IEEE Transactions on Wireless Communications* 19, 7 (2020), 4439–4448.

[29] Rajat Talak, Sertac Karaman, and Eytan Modiano. 2020. Improving age of information in wireless networks with perfect channel state information. *IEEE/ACM Transactions on Networking* 28, 4 (2020), 1765–1778.

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

[31] Shresth Verma, Aditya Mate, Kai Wang, Neha Madhiwalla, Aparna Hegde, Aparna Taneja, and Milind Tambe. 2023. Restless Multi-Armed Bandits for Maternal and Child Health: Results from Decision-Focused Learning.. In *AAMAS*. 1312–1320.

[32] Peng-Jun Wan. 2016. Joint selection and transmission scheduling of point-to-point communication requests in multi-channel wireless networks. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 231–240.

[33] Shuang Wu, Xiaoqiang Ren, Qing-Shan Jia, Karl Henrik Johansson, and Ling Shi. 2022. Towards efficient dynamic uplink scheduling over multiple unknown channels. *arXiv preprint arXiv:2212.06633* (2022).

[34] Jun Xu and Chengcheng Guo. 2019. Scheduling stochastic real-time D2D communications. *IEEE Transactions on Vehicular Technology* 68, 6 (2019), 6022–6036.

[35] Jun Xu, Jianfeng Yang, Yinbo Xie, Chengcheng Guo, and Yinbo Yu. 2016. MDP based link scheduling in wireless networks to maximize the reliability. *Wireless Networks* 22 (2016), 1659–1671.

[36] Xiao Yang, Zhiyong Chen, Kuikui Li, Yaping Sun, Ning Liu, Weiliang Xie, and Yong Zhao. 2018. Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff. *IEEE Access* 6 (2018), 16665–16677.

[37] Abolfazl Zakeri, Mohammad Moltafet, Markus Leinonen, and Marian Codreanu. 2023. Minimizing the AoI in resource-constrained multi-source relaying systems: Dynamic and learning-based scheduling. *IEEE Transactions on Wireless Communications* (2023).

[38] Yihan Zou, Kwang Taik Kim, Xiaojun Lin, and Mung Chiang. 2021. Minimizing age-of-information in heterogeneous multi-channel systems: A new partial-index approach. In *Proceedings of the twenty-second international symposium on theory, algorithmic foundations, and protocol design for mobile networks and mobile computing*. 11–20.