A Deep Reinforcement Learning-Based Whittle Index Policy for Multibeam Allocation

Yuhang Hao
School of Automation
Northwestern Polytechnical University
Xi'an, China
yuhanghao@mail.nwpu.edu.cn

Zengfu Wang
School of Automation
Northwestern Polytechnical University
Xi'an, China
wangzengfu@nwpu.edu.cn

Jing Fu
School of Engineering
Royal Melbourne Institute of Technology University
Melbourne, Australia
jing.fu@rmit.edu.au

Quan Pan
School of Automation
Northwestern Polytechnical University
Xi'an, China
quanpan@nwpu.edu.cn

***Abstract*—In this paper, a non-myopic beam scheduling policy is proposed for multi-target tracking (MTT) in a phased-array radar network, seeking to minimize the discounted sum of tracking error of targets and improve the long-term tracking performance. The Whittle index policy based on the restless multi-armed bandit (RMAB) model can decompose the state space of the underlying optimization problem into independent spaces with reduced sizes. We consider the tracking error covariance (TEC) matrix as the state of each target (arm), which evolves based on the Kalman filter. However, for a real-world MTT, the exact calculation of the Whittle index in multiple dimensions is challenging. The neural network is established to achieve the feature extraction of TEC states and learn the corresponding Whittle index. The deep reinforcement learning (DRL) method is exploited to train the neural network by leveraging the threshold property of the Whittle index policy and engaging in interactions with a single target tracking environment. We propose the DRL-based Whittle index policy, namely DRLWI, aiming to solve the beam allocation problem for MTT with multi-dimensional TEC states. This approach effectively mitigates the exponential computational complexity of classical dynamic programming approaches and the low convergence rate caused by large joint state and action spaces in the simple application of DRL algorithms. Numerical results demonstrate the performance of the proposed DRLWI policy surpasses that of DRL algorithms and myopic policies.**

***Index Terms*—Whittle index, deep reinforcement learning, target tracking, multibeam allocation**

I. Introduction

A phased-array radar network enables multi-radar collaboration, obtaining superiority over a single-site radar [1], [2]. We consider a phased-array radar network for multi-target tracking (MTT) using limited multibeam resources, where each beam can only track one target at each time. The multibeam allocation problem is of crucial importance for the performance of MTT in the phased-array radar network [3], [4]. It becomes more challenging to allocate multibeam resources optimally for multiple targets over a long-term objective, seeking to achieve near-optimality through a non-myopic policy [5], [6].

Optimal solutions to the multibeam allocation problem are in general intractable due to the complexity of multi-step decision-making [6]. Conventional dynamic programming approaches were considered by formulating the problem as a Markov decision process (MDP) or partially observable Markov decision process (POMDP) [7]. In [6], a branch-and-bound algorithm was considered for mobile sensor scheduling with maneuvering targets, which relied on predictions of a few time slots onward. In [8], a branch-and-bound pruning algorithm was utilized to minimize the radiation cost with a tracking accuracy constraint. Here, the tracking accuracy and radiation cost were predicted over a finite time horizon. In [9], a non-myopic and fast resource scheduling algorithm with comprehensive performance indices was proposed for MTT in a space-based radar system. This algorithm employed the online POMDP model based on the Monte Carlo tree search. Unfortunately, as the time horizon increases, the action space for multi-step decisions grows exponentially, especially for large-scale MTT problems. To avoid this problem, [6]–[9] shrank the time horizon used for decision-making predictions, resulting in performance deterioration in the long-run case.

The restless multi-armed bandit (RMAB) techniques quantify the marginal rewards/costs of each arm through state-dependent, real-valued *indices* [10]. Based on such indices, the *index policies*, for RMAB problems, are priority-style policies that always prioritize those arms according to the descending/ascending orders of the indices. Howard *et al.* [11] proposed a greedy index policy based on the metric defined on the tracking error variance (TEV) state, which was updated via scalar Kalman filter dynamics. The objective function was defined as the summary cost of multiple projects over a finite time horizon. Utilizing the Whittle relaxation, a Whittle index policy can decompose the original optimization problem for multiple targets into multiple subproblems, which significantly

This work was partly supported by the National Natural Science Foundation of China (grant no. U21B2008).

reduces computational complexity [12]. In [13], [14], the real-valued TEVs were set as the states of each target, which was treated as an arm. The Whittle index policy for MTT tasks was proposed based on the scalar Kalman filter. Yang and Luo [15] transformed the trace of the channel estimation mean square error as the scalar state of bandits and proposed a method to approximate the Whittle index. In [16], for the Whittle index policy with convoluted transition kernels and two-dimensional states, a Neural Whittle Index Network (NeurWIN) algorithm was proposed to learn Whittle indices for RMAB and deep reinforcement learning (DRL) algorithms were used to train NeurWIN.

In practical scenarios with MTT tasks, it is more reasonable to take the multi-dimensional tracking error covariance (TEC) matrix as the optimization metric. Although the work in [15] considered the multi-dimensional state case, only the trace feature was utilized in the threshold policy. As pointed out by Dance and Silander [17], computing the Whittle indices for multi-dimensional states poses a mathematical challenge.

This paper aims to address the multibeam resource scheduling problem for multi-target with multi-dimensional TEC states. Our objective is to optimize the discounted long-term reward, where the immediate reward is related to the trace of the TEC state. We establish the MDP model of each target and formulate the problem through the RMAB approach, where each target corresponds to a restless project (arm). The neural network is utilized to approximate Whittle indices and the DRL technique trains the network through the interactions with the target tracking environment. We propose the DRL-based Whittle index policy, referred to as the DRLWI policy, to address the beam resource scheduling problem, where multi-feature extraction for the multi-dimensional TEC states is conducted through DRL. Through extensive simulations, we numerically demonstrate that DRLWI outperforms the Amortized Q-learning (AQL) algorithm based on the actor-network and critic-network [18] and myopic policies. The contributions of this paper are summarized as follows.

1) The multibeam allocation optimization problem is formulated through the RMAB model, where each arm represents a single target with TEC states. The MDP is established for each target. Considering the decomposition advantage for the original optimization problem, we seek to utilize DRL to learn Whittle indices through the threshold policy property of the Whittle index policy.
2) We establish a neural network with fully connected (FC) layers to perform feature extraction and approximate Whittle indices. The input vector for the neural network is normalized after the preprocessing stage. We use the mini-batch training method to compute the weighted gradient and tune the parameters of the network through interactions with the target tracking environment.
3) We set up diverse-scale MTT scenarios to demonstrate that the proposed DRLWI policy outperforms the AQL policy and the myopic policy in scheduling optimization.

The rest of this paper is organized as follows. Section II describes the target dynamic model, measurement model, and TEC update model for tracking a single target. Then the multibeam allocation problem is formulated based on the RMAB model. Section III presents the Whittle index policy, followed by an explanation of the DRL framework and training of our DRLWI policy. Section IV showcases the simulation results in three different scales of MTT. Finally, Section V draws the conclusion.

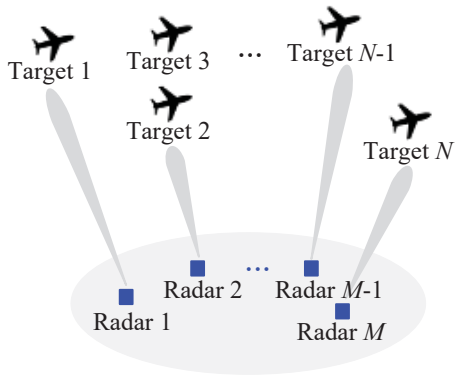

Fig. 1. The tracking scene in a phased array radar network.

II. Problem Formulation

A. Target Dynamic Model and Measurement Model

Consider a radar network, consisting of M phased array radars, which centrally steer M beams to track $N > M$ targets, as shown in Fig. 1. Given the limited beam resources relative to the number of targets, each radar is assigned to track one distinct target in each time slot. Fig. 1 shows an example of beam allocation results, where Target 3 is not tracked.

The dynamic state of target n at time t is denoted by $\boldsymbol{x}_t^n = [x_t^n, \dot{x}_t^n, y_t^n, \dot{y}_t^n]' \in \mathbb{R}^L$, where $[x_t^n, y_t^n]$ and $[\dot{x}_t^n, \dot{y}_t^n]$ are the location and velocity of the target, respectively. Without loss of generality, let $L = 4$. $[\cdot]'$ is the transpose operator. We assume that targets are independent, and follow the nearly constant velocity model in Cartesian coordinates, which is given by

$$\boldsymbol{x}_t^n = \mathbf{F}^n \boldsymbol{x}_{t-1}^n + \boldsymbol{\mu}_{t-1}^n, \tag{1}$$

where the dynamic state transition matrix is denoted as

$$\mathbf{F}^n = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \tag{2}$$

and $\boldsymbol{\mu}_{t-1}^n$ is the process noise, which follows a zero-mean white Gaussian distribution with the covariance matrix being

$$\mathbf{Q}^n = q^n \mathbf{I}_2 \otimes \begin{bmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{bmatrix}, \tag{3}$$

where $\otimes$ is the Kronecker product and q^n denotes the intensity of the process noise. $\mathbf{I}_2$ denotes the 2×2 identity matrix and T_s denotes sampling interval of target tracking.

This paper considers that the measurements of the radar are in the Cartesian coordinate, which can be carried out by the maximum likelihood estimation of the received signal of the

radar. When a radar tracks target n at time t, a measurement $\boldsymbol{z}_t^n$ is derived with the following linear measurement function.

$$\boldsymbol{z}_t^n = \mathbf{H}^n \boldsymbol{x}_t^n + \boldsymbol{v}_t^n, \tag{4}$$

where

$$\mathbf{H}^n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{5}$$

and $\boldsymbol{v}_t^n$ denotes the measurement noise, which is assumed to be a zero-mean white Gaussian noise with covariance matrix being $\mathbf{R}^n$. $\boldsymbol{\mu}_t^n$ and $\boldsymbol{v}_t^n$ are assumed to be independent.

B. TEC Update Model

TEC quantifies the spread of errors between the estimation of the target's state and the actual target state. It evolves in the manner of the Kalman filter [19]. Consider TEC as the optimization criterion, and define it as the state of the bandits. The action a_{t-1}^n for target n will affect updates of the subsequent TEC state at time slot t. If $a_{t-1}^n = 1$, target n is tracked by radars at time t; otherwise, $a_{t-1}^n = 0$, signifying that the radar network will not obtain measurements of target n. Through the Kalman filter, the TEC $\mathbf{P}_t^n$ of target n at time t can be recursively updated. Specifically, if a target is tracked at time t, the TEC will be updated under action $a_{t-1}^n = 1$,

$$\mathbf{P}_t^n = f_n^1\left(\mathbf{P}_{t-1}^n\right) = \left(\mathbf{I}_L - \mathbf{K}_t^n \mathbf{H}^n\right) \bar{\mathbf{P}}_{t|t-1}^n, \tag{6}$$

where

$$\bar{\mathbf{P}}_{t|t-1}^n = \mathbf{F}^n \mathbf{P}_{t-1}^n \left(\mathbf{F}^n\right)' + \mathbf{Q}^n, \tag{7}$$

$$\mathbf{K}_t^n = \bar{\mathbf{P}}_{t|t-1}^n (\mathbf{H}^n)' \left(\mathbf{H}^n \bar{\mathbf{P}}_{t|t-1}^n (\mathbf{H}^n)' + \mathbf{R}^n\right)^{-1}. \tag{8}$$

Otherwise,

$$\mathbf{P}_t^n = f_n^0\left(\mathbf{P}_{t-1}^n\right) = \bar{\mathbf{P}}_{t|t-1}^n. \tag{9}$$

It is important to note that the state update functions $f_n^1\left(\mathbf{P}_{t-1}^n\right)$ and $f_n^0\left(\mathbf{P}_{t-1}^n\right)$ are deterministic, that is, the measurement $\boldsymbol{z}_t^n$ is not involved in the state update. Consequently, we can establish the MDP model [20] as a 4-tuple notation $(\mathcal{P}^n, \mathcal{A}, f_n, R)$ to describe the state transitions, actions, and rewards of each target n, where $\mathcal{P}^n$ is a positive definite symmetric matrix space, i.e., $\mathbf{P}_t^n \in \mathcal{P}^n$; $\mathcal{A} = \{0, 1\}$ is the action space, i.e., $a_t^n \in \mathcal{A}$; f_n consists of the state update functions $f_n^1\left(\mathbf{P}_{t-1}^n\right)$ and $f_n^0\left(\mathbf{P}_{t-1}^n\right)$; R represents the rewards function $R(\mathbf{P}_t^n) \triangleq -\mathrm{tr}(\mathbf{P}_t^n)/L$, indicating our goal of maximizing rewards to improve tracking accuracy.

C. RMAB Model for Problem Formulation

Based on the RMAB approach, we associate each target with a restless bandit process (restless arm), which is an MDP characterized by $(\mathcal{P}^n, \mathcal{A}, f_n, R)$. Define the sum of the long-term discounted rewards of N targets as the optimization objective in the phased array radar network. The dynamic optimization problem can be formulated as

$$\begin{aligned} &\max_{\boldsymbol{\pi}} \mathrm{E}_{\mathbb{P}_0}^{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \sum_{n=1}^{N} \beta^t d^n R\left(\mathbf{P}_t^n\right)\right], \\ &\text{s.t.} \sum_{n=1}^{N} a_t^n = M, \forall t \end{aligned} \tag{10}$$

where $0 < \beta < 1$ denotes the discount factor and d^n is the target weight. Given the distribution of the initial state $\mathbb{P}_0 \triangleq (\mathbf{P}_0^n)_{n=1}^N$, we can compute the expected reward $\mathrm{E}_{\mathbb{P}_0}^{\boldsymbol{\pi}}[\cdot]$ under policy $\boldsymbol{\pi}$.

III. The DRL-based Whittle Index Policy

A. Whittle Relaxation for Multi-target Decomposition

By the Whittle relaxation, we can relax the constraint in (10) over the infinite time horizon as

$$\mathrm{E}^{\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty} \sum_{n=1}^{N} \beta^t a_t^n\right] = \frac{M}{1-\beta}. \tag{11}$$

The Whittle index policy utilizes the Whittle relaxation and the Lagrange relaxation to transform the original problem (10) to

$$\max_{\boldsymbol{\pi}} \mathrm{E}_{\mathbb{P}_0}^{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \sum_{n=1}^{N} \beta^t \{d^n R\left(\mathbf{P}_t^n\right) - \lambda a_t^n\}\right] + \frac{M\lambda}{1-\beta}, \tag{12}$$

where $\lambda \geq 0$ denotes the Lagrange multiplier.

As a result, the problem (12) can be decomposed into N independent sub-problems, given by

$$\max_{\pi^n} \mathrm{E}_{\mathbf{P}_0^n}^{\pi^n} \left[\sum_{t=0}^{\infty} \beta^t \{d^n R\left(\mathbf{P}_t^n\right) - \lambda a_t^n\}\right]. \tag{13}$$

Hence, the number of the action space decreases from $\binom{N}{M}$ to $2N$.

Then, we have the definition of the *indexability* of subproblem (13), following [13, Definition 1].

Definition 1 (Indexability): We say that subproblem (13) is indexable, if there exists a real-valued index $\lambda^{*,n}(\mathbf{P}_t^n)$ which is a scalar function of the target's state $\mathbf{P}_t^n \in \mathcal{P}^n$ such that, for any value of multiplier $\lambda \in \mathbb{R}$, the active action $a_t^n = 1$ is optimal in state $\mathbf{P}_t^n$ iff $\lambda^{*,n}(\mathbf{P}_t^n) \geq \lambda$.

In general, the sufficient and necessary condition for the indexability of a multi-dimensional-state RMAB problem remains open, as does computing the Whittle indices in this case [17]. In this paper, we resort to approximating the Whittle indices for multi-dimensional TEC states through the DRL learning technique.

B. DRL Framework for Whittle Index

We establish the neural network consisting of multiple FC layers with several neurons. The TEC state $\mathbf{P}_t^n$ is flattened into a $L^2 \times 1$ vector after the normalization operation, which serves as the input to the neural network's input layer. Hence, the input layer consists of L^2 neurons. Through multiple hidden layers, the neural network outputs the estimated scalar Whittle index $\lambda_\omega^n(\mathbf{P}_t^n)$ in the output layer with one neuron, where ω consists of weights and biases parameters of the established neural network. Recall that the established neural network focuses on estimating Whittle indices for each target n, as shown in the top part of Fig. 2.

Subsequently, based on the threshold property between the Whittle index $\lambda^{*,n}(\mathbf{P}_t^n)$ and the multiplier λ in Definition 1, we aim to learn the correct order of Whittle index values in

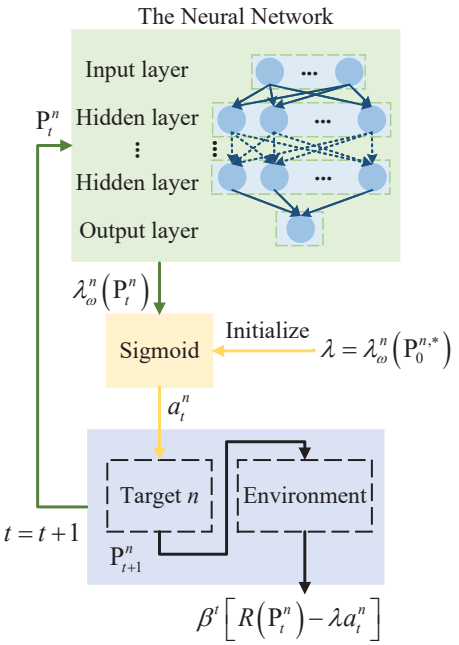

Fig. 2. The training scheme for the established neural network.

different states. We set an estimated Whittle index $\lambda_\omega^n(\mathbf{P}_0^{n,*})$ as λ, where ω is the set of parameters of the neural network in the current episode and $\mathbf{P}_0^{n,*}$ is another randomized initialization TEC state. Then, we can determine the correct order of Whittle index values between TEC states $\mathbf{P}_t^n$ and $\mathbf{P}_0^{n,*}$ through cumulative discounted reward of each episode. An episode represents the time slot length of a simulation, encompassing all states between an initial state and a terminal state. To ensure the scalability of the neural network, we set the target weight d^n to 1 in the environmental reward feedback. To this end, we develop the training scheme of the neural network, as shown in Fig. 2.

At the beginning time slot $t = 0$ of each episode, we initialize two states $\mathbf{P}_0^n$ and $\mathbf{P}_0^{n,*}$, and assign $\lambda_\omega^n(\mathbf{P}_0^{n,*})$ to λ. The target n is tracked with the probability $\sigma\left(\lambda_\omega^n(\mathbf{P}_t^n), \lambda\right)$ at each time slot t. This paper samples the action a_t^n from a uniform distribution $\mathrm{U}(0,1)$. The probability is computed through the Sigmoid function, given by

$$\sigma\left(\lambda_\omega^n(\mathbf{P}_t^n), \lambda\right) \triangleq \frac{1}{1+e^{-(\lambda_\omega^n(\mathbf{P}_t^n)-\lambda)}}. \tag{14}$$

We regard the action selection as a classification problem and define the classical cross-entropy loss function [21] as the loss function of the neural network, aiming to maximize $\lambda_\omega^n(\mathbf{P}_t^n) - \lambda$, if the action $a_t^n = 1$ (or $\lambda - \lambda_\omega^n(\mathbf{P}_t^n)$ if $a_t^n = 0$). More precisely, define the loss function as

$$\begin{aligned} C\left(\mathbf{P}_t^n, a_t^n\right) \triangleq & a_t^n \cdot \ln\left(\lambda_\omega^n(\mathbf{P}_t^n) - \lambda\right) \\ & + \left(1 - a_t^n\right) \cdot \ln\left(1 - \left(\lambda_\omega^n(\mathbf{P}_t^n) - \lambda\right)\right). \end{aligned} \tag{15}$$

We train the neural network by the mini-batch training method [22], where each batch possesses K episodes. The K episodes share the same initial states $\mathbf{P}_0^n$ and $\mathbf{P}_0^{n,*}$. Then, we can calculate the gradient of the loss function for target n at time t in episode k through backward propagation [21], $\delta_{k,t}^n \triangleq \nabla_\omega C\left(\mathbf{P}_t^n, a_t^n\right)$, with respect to ω. With the discounted reward

$$r_k \triangleq \sum_{t=0}^{T-1} \beta^t \left[R\left(\mathbf{P}_t^n\right) - \lambda a_t^n\right] \tag{16}$$

of episode $k = 1, 2, \ldots, K$, we utilize the weighted gradient ascent method to update the parameters ω of the neural network, where the weighted fusion gradient is defined as

$$\delta^n = \sum_{k=1}^{K} (r_k - \bar{r}) \cdot \left[\sum_{t=0}^{T-1} \delta_{k,t}^n\right], \tag{17}$$

where $\bar{r} \triangleq \sum_{k=1}^{K} r_k / K$ and T denotes the time horizon. In this approach, the neural network will intend to follow the action sequence with a larger reward r_k.

The neural network interacts with the tracking environment through multiple batches of episodes and tunes the parameters ω based on the gradient δ^n after every K episodes. Ultimately, extensive training can enable the neural network to compute the Whittle index $\lambda^{*,n}(\cdot)$ on diverse TEC states.

C. Scheduling of DRLWI Policy

Since all the targets are stochastically identical, we utilize the neural network to compute the estimated Whittle index $\lambda_\omega^n(\mathbf{P}_t^n)$ for each target at time slot t. Considering the target weight d^n for each target, define the weighted Whittle index $d^n\lambda_\omega^n(\mathbf{P}_t^n)$. The M targets with the M highest weighted Whittle indices will be tracked, i.e., $a_t^n = 1$, while the remaining targets are not tracked. Then, the TEC state transition in (6) and (9) follow the actions $(a_t^n)_{n=1}^N$. Accordingly, the radar network obtains immediate rewards $\sum_{n=1}^N R\left(\mathbf{P}_t^n\right)$ for time slot t.

IV. Simulations

We consider MTT tracking scenarios with four scales, that is, $(M, N) = (1,4), (2,8), (3,10), (10,40)$. The discounted rewards are evaluated with 100 independent runs, where the time horizon $T = 100$ and the discount factor $\beta = 0.9$. Each independent run stochastically initializes the TEC states of N targets. The initial state $\mathbf{P}_{n,0}$ is generated by $\boldsymbol{\rho}_0\boldsymbol{\rho}_0'$, where each components of $\boldsymbol{\rho}_0$ follows the uniform distribution $\mathrm{U}(0, 40)$. Then divide all components of $\boldsymbol{\rho}_0\boldsymbol{\rho}_0'$ by 10^2, except for the position components on the diagonal. The intensity of the process noise q^n is 1, and the measurement covariance is given as $\mathbf{R}^n = \mathrm{diag}([10^4, 10^4])$, $n = 1, 2, \ldots, N$. We define the target weight d^n in Table I.

The three baselines are as follows. 1) The Greedy policy selects the M targets with the M highest indices. The index is defined by the product of target weight d^n and the trace of the current TEC state at each time slot [13]. 2) The Balanced policy uniformly allocates the M beam resources during the tracking period and achieves cyclic tracking of N targets. 3) AQL [18] implements the Actor-Critic method. The input state of the algorithm is the flattened joint vector of multi-target TEC states after normalization. The outputs are the Q

TABLE I
TARGET WEIGHT d^n FOR TARGETS

(M, N)	d^n, $n = 1, 2, \ldots, N$
$(1, 4)$	0.7, 0.1, 0.1, 0.1
$(2, 8)$	0.7, 0.7, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1
$(3, 10)$	0.7, 0.7, 0.7, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1
$(10, 40)$	0.7 for $n = 1, 2, \ldots, 10$, and 0.1 for $n = 11, 12, \ldots, N$

TABLE II
TRAINING PARAMETERS FOR AQL

(M, N)	L_r	Hidden layers
$(1, 4)$	0.01	$\{256, 128, 128, 64, 32\}$
$(2, 8)$	0.01	$\{512, 256, 256, 128, 128, 128, 64, 64\}$
$(3, 10)$	0.01	$\{640, 512, 256, 256, 128, 128, 128, 128\}$

values from the critic-network and selected probabilities of combinatorial actions from the actor-network.

In our DRLWI policy, the established neural network consists of one input layer, five hidden layers, and one output layer, each of which has L^2, 256, 256, 128, 128, 64, and 1 neurons, respectively. In the training process, we select the $K = 5$ episodes as a batch and the K episodes share the same TEC initial states $\mathbf{P}_0^n$ and $\mathbf{P}_0^{n,*}$. The normalization of input data contributes to improved performance, including faster convergence and increased stability during training of network models [21]. This paper utilizes the Min-Max scaling method, where the TEC state $\mathbf{P}_t^n$ is normalized by scaling its value between 0 and 5000 at each time slot. The Adam optimizer [21] is used to tune the parameters ω of the neural network. The input learning rate L_r is initially set to be 0.01 in episode 1 and decays by a factor of 0.99 for every 200 batches. Recall that the training process focuses on a single target - computing the Whittle indices over various TEC states. As a result, the DRLWI policy can be applied to different scales of (M, N) without necessitating additional multiple training operations.

In the AQL algorithm, a batch consists of $K = 5$ episodes. The input layer and output layer of the AQL network have $L^2 * N$ and $\binom{N}{M}$ neurons, respectively. The training parameters for the scales of $(M, N) = (1, 4), (2, 8), (3, 10)$ are set in Table II, where $\{\cdot\}$ represents the numbers of neurons in the multiple hidden layers, respectively. It is noted that since the AQL algorithm requires a complex neural network framework and training process for the scale of $(M, N) = (10, 40)$, no comparisons will be made here.

Fig. 3, Fig. 4, Fig. 5, and Fig. 6 plot the average discounted rewards for the 100 runs across the 10000 training episodes in the scales of $(M, N) = (1, 4), (2, 8), (3, 10), (10, 40)$, respectively. Each episode shares the same initial TEC states of targets. It is seen that since the learning process focuses on joint states and combinatorial actions, the AQL algorithm results in a low convergence rate of Q values and converges to the worst performance after episode 2000. In practical applications, the AQL algorithm may require a complex neural network framework and a large amount of interaction data. Hence, as the scale of (M, N) grows, AQL struggles to achieve acceptable performances. The Greedy policy and Balanced policy obtain higher reward performances than that of AQL. Considering target weights across N targets, the Greedy policy can earn higher rewards than the Balanced policy. Note that since the Greedy policy and Balanced policy are deterministic, they will always receive the same reward for each episode given the same initial TEC state, as shown by the constant lines in Figs. 3, 4, 5, and 6. The DRLWI policy obtains the highest reward after episode 3000 and converges to the highest reward performance at episode 10000. Note that the trained neural network for a single target is directly applied to each target, so the convergence trends across the 10000 episodes are similar for all four scales of (M, N). We record the reward performances at episode 10000 and calculate the relative improvement rates, as shown in Table III. The improvement rates describe the obtained higher reward of the DRLWI policy over the baselines, as indicated below the baseline results. After the convergence of DRLWI and AQL, the DRLWI policy achieves at least 6.54% reward improvement over the baselines. Above all, the proposed DRLWI policy exhibits superior performance compared to other baselines in addressing the MDP problem with multi-dimensional TEC states.

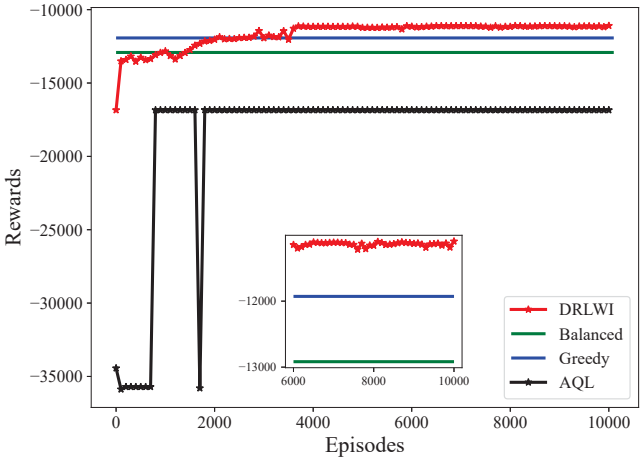

Fig. 3. Rewards over training episodes with the scale of $(M, N) = (1, 4)$.

TABLE III
REWARD COMPARISON AND IMPROVEMENT RATES AT EPISODE 10000

(M, N)	DRLWI	Greedy	Balanced	AQL
$(1, 4)$	-11093.32	-11928.04 (7.00%)	-12917.51 (14.12%)	-16842.34 (34.13%)
$(2, 8)$	-22018.08	-23661.74 (6.95%)	-28176.37 (21.86%)	-52586.86 (58.13%)
$(3, 10)$	-28941.63	-30965.91 (6.54%)	-36972.85 (21.72%)	-80021.46 (63.83%)
$(10, 40)$	-110593.57	-118941.14 (7.02%)	-128797.88 (14.13%)	– (–%)

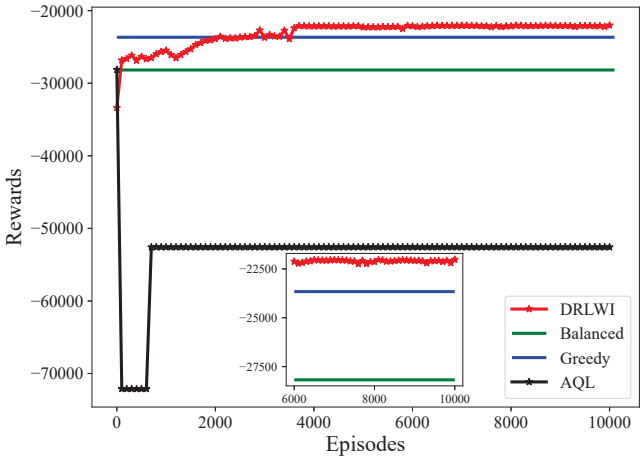

Fig. 4. Rewards over training episodes with the scale of $(M, N) = (2, 8)$.

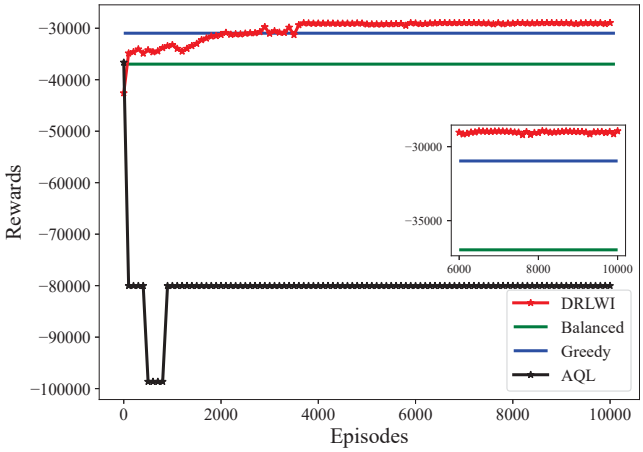

Fig. 5. Rewards over training episodes with the scale of $(M, N) = (3, 10)$.

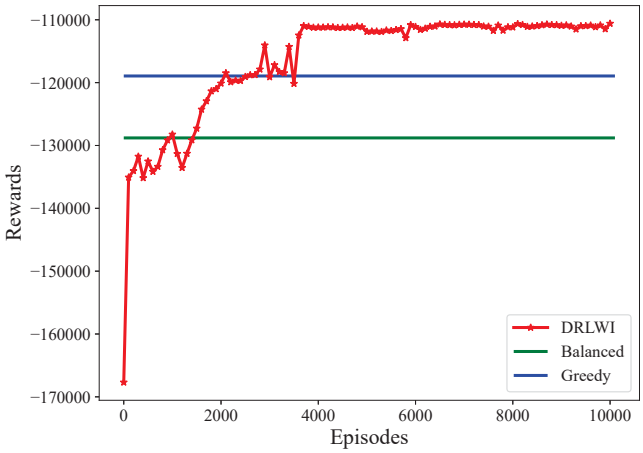

Fig. 6. Rewards over training episodes with the scale of $(M, N) = (10, 40)$.

V. Conclusion

We addressed the dynamic beam allocation problem for MTT using the RMAB technique. For the bandits with multi-dimensional TEC states, the DRL algorithm was utilized to approximate the Whittle index, where the neural network was established based on the threshold property between the estimated Whittle index and the preset action cost. After the training process, the neural network computed the estimated Whittle index for each target at each time slot, allowing the radar network to compute the activated set of targets. The simulation results validated the superiority and effectiveness of our proposed DRLWI policy over other baselines. Future work entails developing an index-based beam and power resource scheduling policy for enhancing multi-target tracking and detection.

References

[1] W. Zhang, C. Shi, J. Zhou, and R. Lv, "Joint aperture and transmit resource allocation strategy for multitarget localization in the phased array radar network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1551–1565, 2023.

[2] C. Shi, Y. Wang, S. Salous, J. Zhou, and J. Yan, "Joint transmit resource management and waveform selection strategy for target tracking in distributed phased array radar network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 2762–2778, 2021.

[3] J. Dai, J. Yan, S. Zhou, P. Wang, B. Jiu, and H. Liu, "Sensor selection for multi-target tracking in phased array radar network under hostile environment," in *2020 IEEE Radar Conference (RadarConf20)*. IEEE, 2020, pp. 1–5.

[4] W. Yi, Y. Yuan, R. Hoseinnezhad, and L. Kong, "Resource scheduling for distributed multi-target tracking in netted colocated MIMO radar systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1602–1617, 2020.

[5] C. Pang and G. Shan, "Sensor scheduling based on risk for target tracking," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8224–8232, 2019.

[6] X. Gongguo, S. Ganlin, and D. Xiusheng, "Non-myopic scheduling method of mobile sensors for manoeuvring target tracking," *IET Radar, Sonar & Navigation*, vol. 13, no. 11, pp. 1899–1908, 2019.

[7] B. Van Der Werk, M. Ian Schöpe, and H. Driessen, "Approximately optimal radar resource management for multi-sensor multi-target tracking," in *2021 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–8.

[8] G. Shan, G. Xu, and Q. Chenglin, "A non-myopic scheduling method of radar sensors for maneuvering target tracking and radiation control," *Defence Technology*, vol. 16, no. 1, pp. 242–250, 2020.

[9] Z. Wang, G. Yang, and S. Jin, "A non-myopic and fast resource scheduling algorithm for multi-target tracking of space-based radar considering optimal integrated performance," *Journal of Radars*, vol. 13, no. 1, pp. 253–269, 2024.

[10] Y. Hao, Z. Wang, J. Niño-Mora, J. Fu, M. Yang, and Q. Pan, "Non-myopic beam scheduling for multiple smart target tracking in phased array radar network," *arXiv preprint arXiv:2312.07858*, 2023.

[11] S. Howard, S. Suvorova, and B. Moran, "Optimal policy for scheduling of Gauss-Markov systems," in *Proceedings of the Seventh International Conference on Information Fusion*. Citeseer, 2004, pp. 888–892.

[12] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of Applied Probability*, vol. 25, no. A, pp. 287–298, 1988.

[13] J. Niño-Mora and S. S. Villar, "Multitarget tracking via restless bandit marginal productivity indices and Kalman filter in discrete time," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 2905–2910.

[14] J. Niño-Mora, "Whittle's index policy for multi-target tracking with jamming and nondetections," in *International Conference on Analytical and Stochastic Modeling Techniques and Applications*. Springer, 2016, pp. 210–222.

[15] F. Yang and X. Luo, "A restless mab-based index policy for UL pilot allocation in massive MIMO over Gauss–Markov fading channels," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3034–3047, 2020.

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

[17] C. Dance and T. Silander, "When are Kalman-filter restless bandits indexable?" in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1, 2015, pp. 1711–1719.

[18] T. Van de Wiele, D. Warde-Farley, A. Mnih, and V. Mnih, "Q-learning in enormous action spaces via amortized approximate maximization," *arXiv preprint arXiv:2001.08116*, 2020.

[19] Y. Kim, H. Bang *et al.*, "Introduction to Kalman filter and its applications," *Introduction and Implementations of the Kalman Filter*, vol. 1, pp. 1–16, 2018.

[20] Y. Shi, B. Jiu, J. Yan, and H. Liu, "Data-driven radar selection and power allocation method for target tracking in multiple radar system," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 296–19 306, 2021.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.

[22] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 661–670.