

## Research Article

# Intelligent Sensing and Computing in Wireless Sensor Networks for Multiple Target Tracking

Xinyu Zou <sup>1</sup>, Linling Li,<sup>2</sup> Hao Du,<sup>2</sup> and Longyu Zhou <sup>2</sup>

<sup>1</sup>IT Center, West China Hospital of Sichuan University, Chengdu 610041, China

<sup>2</sup>School of Information & Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Longyu Zhou; [zhoulyfuture@outlook.com](mailto:zhoulyfuture@outlook.com)

Received 23 October 2021; Revised 24 February 2022; Accepted 24 March 2022; Published 15 April 2022

Academic Editor: Chun-xi Yang

Copyright © 2022 Xinyu Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With sixth generation (6G) communication technologies, target sensing can be finished in milliseconds. The mobile tracking-oriented Internet of Things (MTT-IoT) as a kind of emerging application network can detect sensor nodes and track targets within their sensing ranges cooperatively. Nevertheless, huge data processing and low latency demands put tremendous pressure on the conventional architecture where sensing data is executed in the remote cloud and the short transmission distance of 6G channels presents new challenges into the design of network topology. To cope with the above difficulties, this paper proposes a new resource allocation scheme to perform delicate node scheduling and accurate tracking in multitarget tracking mobile networks. The dynamic tracking problem is formulated as an infinite horizon Markov Decision Process (MDP), where the state space that considers energy consumption, system responding delay, and target important degree is extended. A model-free reinforcement learning is applied to obtain satisfied tracking actions by frequent iterations, in which smart agents interact with the complicated environment directly. The performance of each episode is evaluated by the action-value function in search of the optimal reward. Simulation results demonstrate that the proposed scheme shows excellent tracking performance in terms of energy cost and tracking delay.

## 1. Introduction

With the new technologies in 6G, ultralow latency for huge data transfer dramatically enlarges the target sensing application field. Mobile target sensing in Wireless Sensor Networks (WSNs), which is dedicated to monitoring the supervisory field in the military and civilian fields for security, is a kind of special Internet of Things (IoT) application [1, 2]. In the military field, this application can be applied in countering terrorism, unmanned aerial vehicle (UAV) navigation, and space exploration, etc. Relatively speaking, no-flying zone monitoring of airports and information collecting also used this application for security and data acquirement [3]. Specially, the sensor nodes detect whether the suspicious target intrudes into the ground for target sensing and establish a communication connection with the sink node. Controlling data is issued for scheduling potential sensors to execute sensing task on the sink node. Consequently,

the sink node is regarded as a controller with sufficient computing capacity for the sensor nodes.

However, optimal sensing time delinquency becomes common due to high latency response in the traditional centralized architecture. Specifically, computing time is obviously increased when the massive data is required from sensor nodes in WSNs. New technologies of 6G, such as terahertz channels, may provide enough bandwidth for the huge data transmission [4], but the unstable and short transmission link put new constraints on the topology control of the sensor network [5]. Furthermore, redundant data has also emerged because more than one sensor node can detect the same target with target nodes moving and the coverage areas of sensor nodes could have overlapping in the practical scenario. Besides, transmission congestion and data queue still exist in the MTT-WSNs. As a consequence, sensing time moves backwards in turn, missing the best sensing opportunity.

In addition, the low energy consumption of WSNs has always been the focus of attention, with the characteristics of battery power. Generally speaking, energy cost is mainly divided into transmission cost and sensing cost, in which sensing cost is more than transmission cost when the detected target nodes break away from control. How to schedule the proper sensor nodes for saving energy cost is the other exposed challenge. In order to improve the target sensing performance of WSNs, many scholars and engineers have done a lot of researches.

In recent years, satisfactory results have been obtained by vast studies on disparate sensing performances. In terms of a compromise solution between sensing quality and energy efficiency, the authors of [6] design a sensing scheme named “t-tracking,” which enhances quality of sensing by increasing network throughput. Sensing performance is optimized by Luo et al. [7] in improvement of localization by a cooperative localization scheme. Tracking accuracy is promoted by the combination of the off-line phase and the on-line phase, which largely offsets the slower response time. Edge computing provides a new paradigm for good tracking performance. The authors of [8, 9] introduced a similar cooperative tracking algorithm based on edge computing technology with minimum energy consumption, in which the computing device is located near the sensor node. Wan et al. [10] propose a joint Doppler-angle estimation solution for improving multitarget tracking efficiency.

In general, the multitarget tracking of WSNs faces main challenges as follow:

- (i) When targets with different important degrees invade the monitoring area, it is difficult to accurately identify the targets’ threat level. In some important monitoring areas, the highest important degree targets may be missed
- (ii) Task-oriented WSNs have the problem of unreasonable bandwidth utilization. Low important degree intrusion targets may consume limited bandwidth, so communication for monitoring high important degree targets cannot be guaranteed
- (iii) The limited mobility and coverage ability of sensor nodes may lead to the targets with high important degree be missed or misjudged. As a result, the tracking accuracy may be not satisfactory
- (iv) The high-intensity computing consumption of the central server leads to high tracing delays and cannot meet the requirements of real-time control. Effective feedback cannot be guaranteed

Motivated by the observations, we propose a distributed multitarget tracking scheme with intelligent edge device in this paper, which is different from those of target tracking system with edge computing or localization algorithms [11]–[12]. On the one hand, compared with the edge computing system, the offloading strategy needs a joint decision, including whether to schedule, which sensor to schedule, and which target to track, which makes the offloading strat-

egy more challenging. On the other hand, compared with the traditional location algorithm, the response time can be reduced rapidly by locating the distributed edge devices. The optimal target is transferred from minimum energy cost or tracking delay to joint consideration of energy cost and tracking performance. The main contributions are summarized as follows:

- (i) We propose a new distributed multitarget tracking architecture, where real-time scheduling is enhanced by requiring computing results from resource caches. Besides, we improve the tracking efficiency by introducing mobile cloud server providing resources for sink nodes. Missing probability is reduced with the assistance of collaborative perception among sensor nodes
- (ii) Based on our proposed architecture, the performance multitarget tracking is improved with joint consideration of system cost, response time, and important degree of targets. Minimum energy cost with low tracking delay is transferred as MDP with free model representation which depends on the current system state
- (iii) Self-adaption is proposed by an elastic mechanism, which optimizes energy consumption and response time. Based on the MDP model, we establish the action-value function to evaluate the scheduling strategies with the energy cost and tracking delay of sensor nodes. A deep Q-learning Network (DQN) algorithm is utilized for optimizing real-time scheduling strategies to adapt to different scenarios for different purposes, such as targets with key tasks or high mobility. Simulation results demonstrate that our intelligent cooperative scheme shows good tracking performance according to the self-adaption scheme

The rest of this paper is organized as follows. System model is given by Section II. The scheduling strategy is analysed by Section III. Section IV presents the simulation evaluation. Finally, Section V concludes this paper.

## 2. Related Work

Energy consumption, delay, coverage, sensor deployment, and accuracy are crucial factors in sensor networks about target tracking [13]. In recent years, many researchers have done a lot of work to address these challenging issues [14]–[15].

To begin with, some researches have studied the problem of tracking coverage. The purpose of [16] is to provide an effective coverage method for applications such as target location, environmental monitoring, and vehicle target tracking. Some intelligent algorithms, such as neural networks, adaptive distributed filtering, and fuzzy framework,

are used to obtain the optimal filter. A new adaptive Kalman estimation method based on distribution consistency is proposed [17]. The average inconsistency of the optimal filter gain and the estimated value are considered in the filter design. Optimal Kalman gain is obtained by minimizing mean square estimation error to estimate the states of the target more accurately. In order to get better filter performance, an adaptive consistency factor is used to adjust the optimal gain. In the information exchange of filters, dynamic clustering selection and two-level hierarchical fusion structure are used to obtain more precise estimation [18].

Meanwhile, the architecture of WSNs is extensively studied. One of the most popular applications in WSNs is vehicle tracking. Generally speaking, vehicle tracking involves edge computing, edge intelligence, etc [19]. Many researches focus on improving the tracking accuracy and energy saving of target vehicles on WSNs. A decentralized vehicle tracking strategy is proposed to dynamically adjust the active area, improve the energy-saving effect, and tracking accuracy. [20] introduces novel mobile sensor networking architecture for a swarm of microunmaned vehicles (MAVs) using software defined network (SDN) technology. The proposed architecture was aimed at enhancing the performance of user/control plane data transmission between MAVs. In [21], the authors propose a novel system architecture and MAC protocol, which includes how to select the target cooperative sensor node and which sensing data should be retransmitted by using the decided cooperative sensor node.

Moreover, due to the extensive research on machine learning, lots of researches have upgraded and extended the algorithm, which can be widely used in WSNs [22]. In [23], they proposed a double time scale Q-learning algorithm with function approximation to alleviate the curse of dimension problems. Although all of the above algorithms alleviate the state explosion problem, it is necessary to solve the action explosion problem to obtain the scalable solution. The edge intelligent computing of WSNs has also been widely studied, such as the basic wireless cache networks, in which the source server is connected to the base station (BS) supporting cache, which serves multiple requesting users [24]. In order to solve the problem of how to improve the cache hit rate under the condition of dynamic content popularization, a dynamic content update strategy based on deep reinforcement learning is proposed. In [25], the authors considered a WSN with a great number of sensors and multiple clusters. Each cluster has a cluster head and several cluster members, and the sensors are distributed randomly. Cluster members only measure environmental parameters like humidity, atmospheric pressure, and temperature. The cluster head is responsible for fusing the measurement data from the cluster members and forwarding the data to the base station through single hop or multihops.

Furthermore, with the development of WSNs and UAV technologies, which is popularly used in target tracking, the enhancement of coverage, the comprehensive utilization of data resources, and the stability of network systems, especially the cooperative network composed of WSNs and UAV, are expected to provide immediate long-term benefits in military and civil fields. Previous work mainly focused on

how to use UAV assisted WSNs for sensor and data collection, or use a single data source for target location in the monitoring system, but the potential of multiple UAV sensor networks has not been fully explored. In [26], the problem of target tracking for a class of wireless positioning system with degradation measurement and quantization effect is investigated. In order to track the state of the object as accurately as possible, a recursive filtering algorithm is proposed. First, an upper bound of filter error covariance is derived, and then, the upper bound is minimized by properly designing the filter gain at each sampling time. [27] proposed a new network platform and a system structure for multiple UAVs' cooperative monitoring, including designing idea of cooperative resource scheduling and tasking allocation scheme for multiple UAVs. Due to the complexity of the monitoring area, [28] discussed the establishment of a suitable algorithm based on machine learning.

### 3. System Model

In this section, a state-action model is established for the distributed multitarget tracking system. State space is added which incorporates requiring latency, queue length, and different important degree of targets. Each indicator is specifically designed in the following communication model and request model.

As shown in Figure 1, the efficient computing resources enabled MTT network is made up of one UAV-assisted mobile station,  $N$  target nodes,  $M$  sensor nodes, and  $K$  sink nodes. Sink nodes are selected from the set  $M$ , i.e.,  $K \subset M$ . Each sensor node is an energy-limited device with powered battery. Let  $\mathcal{N} = \{1, 2, \dots, N\}$  denotes the set of target nodes and  $\mathcal{M} = \{1, 2, \dots, M\}$  denotes the set of sensor nodes, respectively. Let  $t \in \{1, 2, \dots, T\}$  denotes the discrete time for tracking. Sensor nodes perceive targets each time and report to sink nodes which are equipped with a certain amount of computing resources. The computing resources are set as  $C \subset N$ . Sink nodes schedule proper sensor nodes to execute tracking tasks under the assistance of UAV. We assume that each sink node owns corresponding network information, in which UAV obtains the information via each fixed period by information sharing.

*3.1. Analysis of Communication Model and Action.* In this subsection, the communication model and action space are analysed. We assume that arrival of target nodes is on the basis of an independently identically distribution (IID). Once the probability happens, energy consumption is produced at time  $t$ . Hence, we depict the energy consumption as  $e_{r,t}$ . Requiring to be scheduled is real time for sensor node  $m$ , and the overhead of fetching transmission should be considered as follows:

$$e_{r,t} = \lambda_{elec} \times 2 + \lambda_{amp} \times d^2, \quad (1)$$

where  $\lambda_{elec}$  is the circuit consumption and  $\lambda_{amp}$  is the gain consumption of amplifier consumption of one bit. Downlink of 1 bit can realize scheduling strategy, i.e., 0 is prohibition

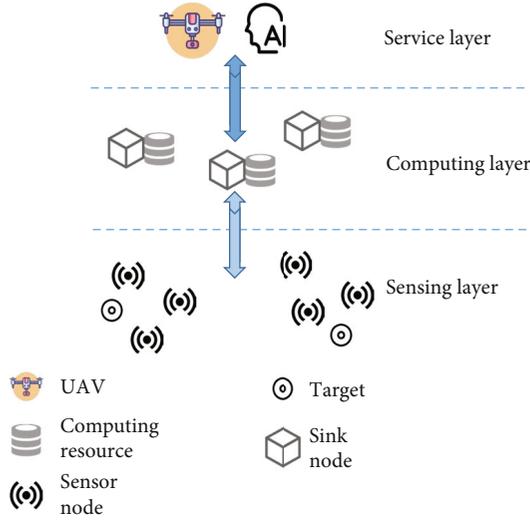


FIGURE 1: System components and transmitting modes based collaboration and scheduling solution in MTT-WSNs.

otherwise is permission.  $d$  is the distance between the sensor and the sink node.

Sink node provides a proper scheduling scheme to track wisely, and the scheduled probability of tracking target  $n$  at time  $t$  is denoted as  $p_{m,n,t}$ . The energy consumption of the sensor node  $m$  as a candidate at time  $t$  is denoted as  $e_{s,t}$ . In the dynamic self-recognized network, it is noteworthy that the information of the sensor node  $m$  is absent for the region  $r$  when the sensor node moves from the region  $h$  to the region  $r$ . However, the corresponding data can be obtained from UAV and the fetching cost is associated. Let  $b_m$  denote the cost between UAV and the sink node, the fetching cost is written as follows:

$$F_m = 1(m \notin C^m)b_m, C^m \subset \mathcal{C}, \quad (2)$$

where  $1(*)$  is the indicator function. If  $*$  is generated, cost will be incurred, otherwise, cost is zero.

In the multitarget tracking network, the energy of the sensor nodes and the relative distance between each target node are considered for evaluating the potential tracking capacity. Let  $a_t \in \mathcal{A} \triangleq \{M \cup 0\}$  denotes the decision-making variable for real time tracking at time  $t$ , where  $a_t = 0$  denotes no decision for sink node,  $a_t = m$  denotes sensor node  $m$  is scheduled at time  $t$ , and  $\mathcal{A}$  is the action space.

**3.2. Analysis of Request and Delay Model.** In this subsection, the request model and delay model are analysed, incorporating the design of the request queue and delay. At each time  $t$ , a sensor node can only receive information of one target node. Each sensor node reports to the corresponding sink node. When tracking target  $n$ , other sensor nodes may receive broadcasting data of target  $n$ . Starting at time  $t$ , sink nodes requested by the set of sensor nodes is set as  $\mathcal{M}_{n,t}^b$ , and  $\mathcal{M}_{n,t}^b \subset \{\mathcal{M} \cup 0\}$ . When the number of targets invading the monitoring area is none,  $\mathcal{M}_{n,t}^b = \emptyset$ .  $\mathcal{M}_{n,t}^d$  is the set of sensor nodes when requesting sink nodes to track target  $n$  at time  $t$ ,

and the dynamic renewal process is described as follows:

$$\mathcal{M}_{n,t+1}^b = \mathcal{M}_{n,t}^b \cup \mathcal{M}_{n,t}^d, \quad (3)$$

$$|\mathcal{M}_{n,t+1}^b| = \begin{cases} |\mathcal{M} \setminus \mathcal{M}_{n,t}^d|, & \text{if } r_t = n \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where  $r_t = n$  indicates the sink nodes receive request from sensor nodes to track target node  $n$ . Otherwise, there is no request.

At each time  $t$ , the queue for requesting sink nodes at the beginning of the next time  $t + 1$  is based on the last time. If sensor nodes have no idea for target node  $m$ , the number of requests should be computed with a view to number of requests at time  $t$  during the next time  $t + 1$ . The size of request unit length is  $\mu_{m,n,t}$ , and the request queue length exception  $R_{n,t+1}$  at time  $t + 1$  is represented as follows:

$$R_{n,t+1} = \sum_{m=1}^M \mu_{m,n,t}. \quad (5)$$

The scheduling set is explored as a subset of the request queue. In case of  $R_{n,m,t} = 0$ , the scheduling set is empty. Otherwise, the scheduling set belongs to the request queue. The  $N \times M$  state matrix  $\mathcal{R}_t$  is constructed, and  $\mathcal{R}_t \in \mathbf{R}$ , where  $\mathbf{R}$  is the state space of the request queue. The independence between sensor nodes leads the scheduling policy not to be related to the request probability but only related to the non-empty request queue.

At the beginning of time  $t$ , sensor node  $m$  transmits its own information to the sink node and is traversed according to the corresponding node ID which is stored in the computing resources. After that, the policy notifies the sink node to make processing. The fetching transmitting delay is analysed and requesting nodes with probability  $p_{n,m,t}$  is scheduled. The penalty of delayed sensor nodes is generated in a subset of request nodes. Heterogeneous target nodes with different delay constraints are considered to be associated with different delay penalties. At time  $t$ , the delay constraint range for target node  $n$  is written as  $I_{n,t} \in I_n \in [0, \bar{I}_n]$ , where the maximum delay is  $\bar{I}_n$ .  $\mathbf{I}$  denotes the delay state space and  $\bar{I}_n \in \mathbf{I}$ . Therefore, the dynamic delay is described as

$$I_{n,t+1} = \begin{cases} 0, & \text{if } \mu_{n,t} = 0 \\ \min [I(p_{n,t}), \bar{I}_{n,t+1}], & \text{otherwise} \end{cases}, \quad (6)$$

where  $I(*)$  is a utility function of the delay, which mainly includes transmitting delay, and fetching delay between the sink nodes and the UAV.

In the MTT-WSNs, penalties will increase by approaching the upper bound of delay constraint. In order to avoid high transmission delays and interminable request queues, let  $\sigma_{i,t}, \forall i \in N$  denotes the important degree of target node  $m$  which also expresses the degree of not missing target  $n$ . Important degrees for target node are denoted as  $\sigma_t = [$

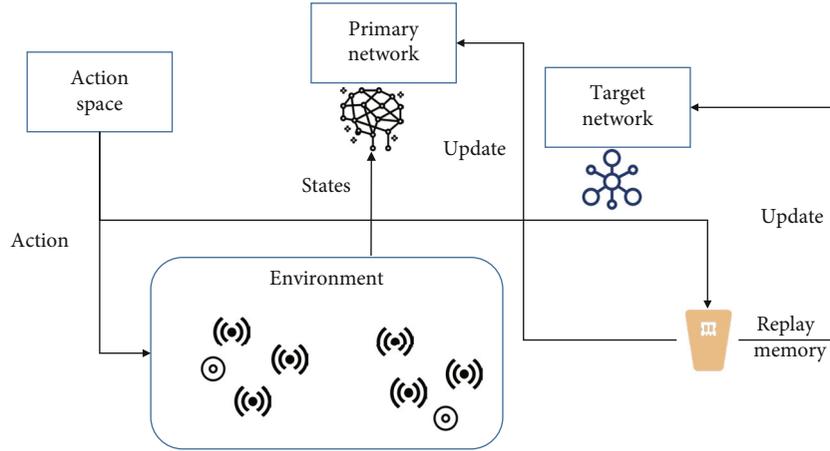


FIGURE 2: DQN-based scheduling strategy and interacting process between the agent and MTT-WSNs.

```

Initialize the replay memory  $D$  to capacity  $N$  and action- value  $Q$  with random weights
Input: set vector  $s_t$ , step number  $N$ , discount factor  $\gamma$ , learning rate  $\alpha$ , the number of step  $M$ 
1: for each interaction with environment episode = 1,  $M$ 
  do
2:   for  $t = 1, T$  do
3:     With probability  $\epsilon$  select a random action  $\alpha_t$ 
4:     The corresponding penalty  $D$  is computed and stored in memory;
5:     Get a batch of samples from replay memory, and compute difference function based on (17);
6:     Calculate the gradient of weight  $\alpha$ ;
7:     Updating: according to (18);
8:     if step =  $N$  then
9:        $Q \leftarrow Q'$ 
10:    end if
11:  end for
12: end for
Output:  $Q$ 

```

ALGORITHM 1: DQN-based scheduling strategy.

TABLE 1: Simulation parameters of MTT-WSNs' environment.

Parameter description	Value
Area of each region	100 m × 100 m
Number of target nodes	2
Velocity of target nodes	1 m/s
Number of sensor nodes	40
Velocity of sensor nodes	2 m/s
Maximum tolerate delay of target nodes	5 s, 6 s
Primary energy of each sensor nodes	40 J
Primary coordinate of target nodes	(0 m, 30 m), (0 m, 60 m)
Energy consumption for static nodes	0.2 J/unit time
Energy consumption for moving nodes	0.8 J/unit time

$\sigma_{1,t}, \sigma_{2,t}, \sigma_{n,t}$ ], and  $\sigma_t \in \sigma$ , where  $\sigma$  belongs to the following state space.

As consequence, we describe the system state as  $\mathbf{S} = \mathbf{R} \times \mathbf{I} \times \sigma$  by integrating the state space of delay, request queue, and important degree. Meanwhile, we take a single self-

organized network in consideration to simplify the complexity for the reason that the same policy can be realized in different self-organized regions. The state space  $\mathbf{S}$  is reduced to  $\mathbf{S}_1 = \mathbf{R}_1 \times \mathbf{I}_1 \times \sigma_1$ . At time  $t$ , the state space is given by

$$\mathbf{S}_1 = \{s_t = \mathbf{R}_{1,t} \times \mathbf{I}_{1,t} \times \sigma_{1,t} | p(I_{1,m,t} \times \sigma_{1,m,t} | R_{1,m,t})\}, \quad (7)$$

where delay constraint only appeared when request queue is nonempty.

#### 4. Scheduling Strategy for MTT-WSNs

The exploring strategy is analysed in this section by means of self-adaptation deep learning, in which the interacting learning is designed between a smart agent and a complex monitoring environment.

**4.1. Problem Formulation.** Given a state space  $s \in \mathbf{S}_1$ , the action, which is evaluated with a corresponding penalty or reward, resulted from the state space. Policy  $\pi$  is a function

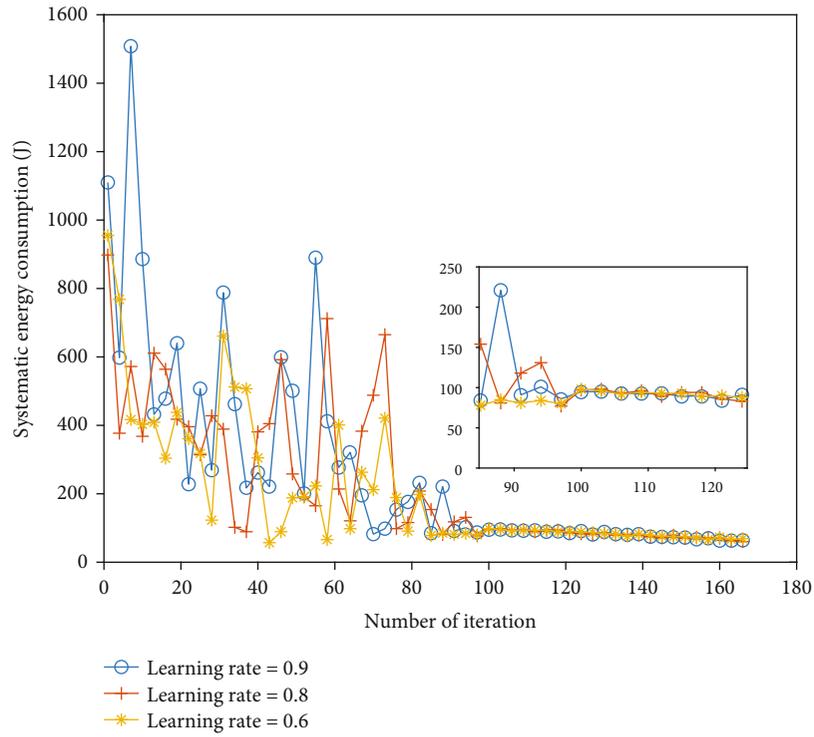


FIGURE 3: The systematic energy consumption under the condition of  $\xi_1 = \xi_2 = 0.5$ .

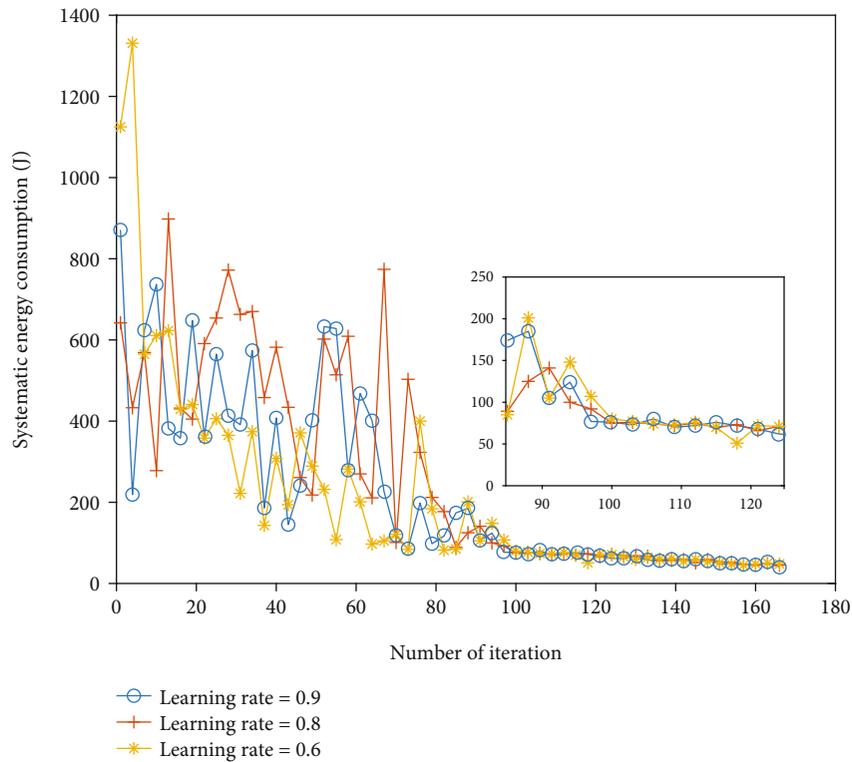


FIGURE 4: The systematic energy consumption under the condition of  $\xi_1 = 0.9, \xi_2 = 0.1$ .

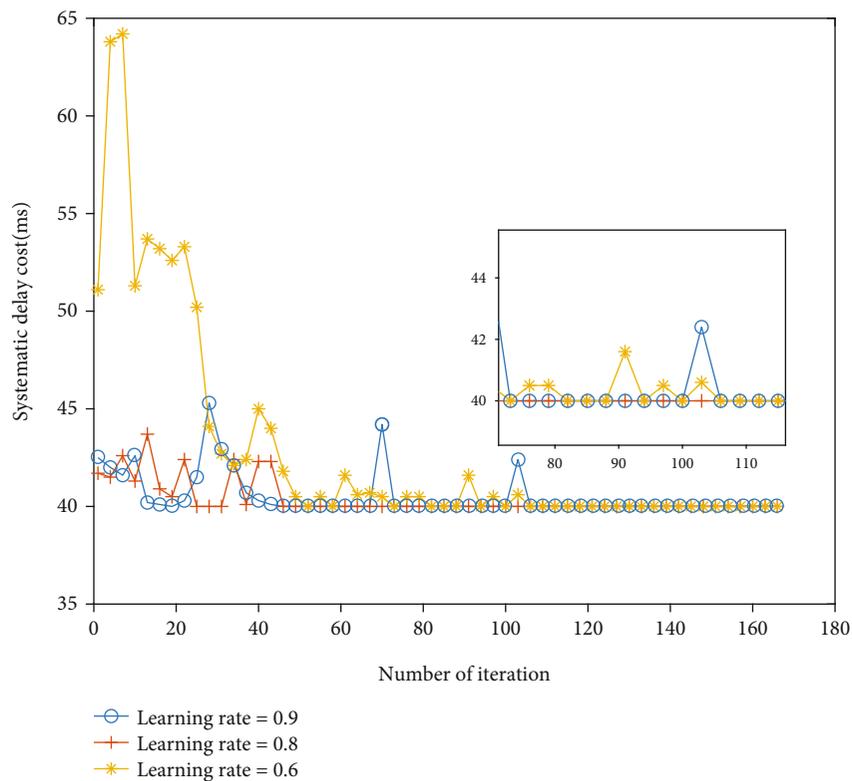


FIGURE 5: The systematic delay cost under the condition of  $\xi_1 = 0.9, \xi_2 = 0.1$ .

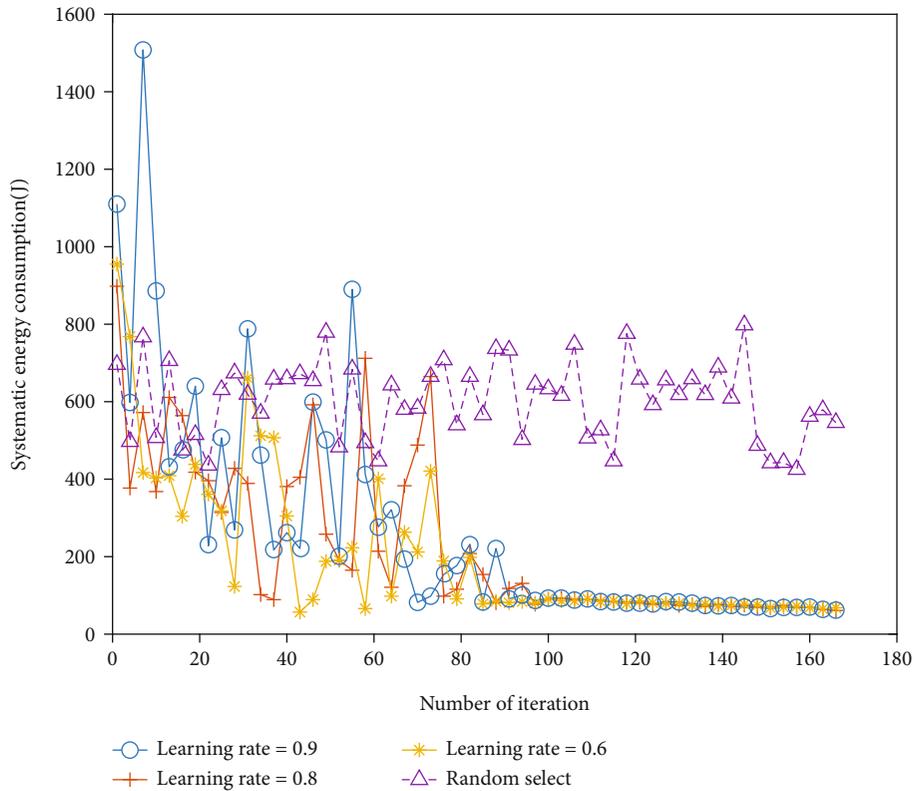


FIGURE 6: The comparison with RSN under the condition of  $\xi_1 = \xi_2 = 0.5$ .

mapping from state space to action space, which is given by

$$s \xrightarrow{\pi} \mathcal{A}. \quad (8)$$

In the MTT-WSNs, the optimal solution is unavailable by using existing deep learning algorithms such as deep Q-learning for the state space  $\mathbf{S}_1$  with multiple dimensions. Thus, the problem is solved by training in the neural network instead of finding in the Q-table. The solution can cope with the curse of dimension because of multiple tensors of the neural network. Specifically, multidimensional state space as input is learned to obtain a policy through deep neural network with weighted values. Action is obtained by the branch of policy and is executed in a dynamic and complex environment, which generates a new state and the corresponding reward or penalty. If this evaluation is a positive reward, a new state value is set as the next input. Otherwise, this action will be punished. Corresponding reward or penalty is responsible for adjusting weight parameters for the deep neural network [23].

In a practical mobile target tracking system, energy consumption can be represented for sensor node  $m$  at time  $t$  by the (state and action) pair. For a given target node  $n$ , the systematic energy consumption is written as

$$E(\pi) = \sum_{t=1}^T E_t(s_t, a_t), \quad (9)$$

$$E_t(s_t, a_t) = \sum_{i=1}^T \mu_{m,t} e_{r,t} + p_{m,t} e_{s,t} + F_{m,t}, \quad (10)$$

where  $E_t(s_t, a_t)$  denotes the system energy consumption at time  $t$ , and we give expectations for whole MTT-WSNs.  $F_{m,t}$  is the fetching cost corresponding to the Equation (2).

When execution latency exceeds the given deadline, the delay penalty is generated and is denoted by

$$D(\pi) = \sum_{t=1}^T D_t(s_t, a_t), \quad (11)$$

$$D_t(s_t, a_t) = \sum_{n=1}^N x_n(I_{n,t}), \quad (12)$$

where  $D_t(s_t, a_t)$  denotes the penalty of the system delay, and  $x_n(*)$  denotes the delay of total target nodes with diverse delay constraints. In order to ensure real-time tracking, target nodes with different important degrees are detected and tracked, and the trajectories in dynamic scenarios are random. Therefore, our purpose is to minimum the delay and the penalty of the system delay that can be written as

$$D_t(s_t, a_t) = \begin{cases} x_n(\theta_{n,t}), & \text{if } \theta_{n,t} \leq \bar{I}_n \\ y_n(\theta_{n,t}), & \text{if } \theta_{n,t} > \bar{I}_n \end{cases}, \quad (13)$$

where  $\theta_{n,t}$  denotes the practice delay for tracking target  $n$  at time  $t$ . Any delay will be punished, and the punishment

intensity  $y_n(*)$  is much weighted than  $x_n(*)$ . Once the delay  $\theta_{n,t}$  exceeds the threshold  $I_{n,t}$ , the punishment will gradually increase.

For a dynamic MTT-WSNs, it is difficult to use polynomial time to obtain the optimal policy, so the deep Q-network algorithm is employed in the MTT-WSNs. Our object is to minimize the systematic energy consumption and the system delay jointly. Therefore, the utility function of the joint optimization can be described as

$$U(\pi) = \xi_1 E(\pi) + \xi_2 D(\pi), \quad (14)$$

where  $\xi_1$  and  $\xi_2$  are the weighted parameters which satisfy  $\xi_1 + \xi_2 = 1$ , respectively. At time  $t$ ,

$$U(s_t, a_t) = \xi_1 E(s_t, a_t) + \xi_2 D(s_t, a_t), \quad (15)$$

denotes the utility function to minimize the system cost.

As mentioned above, an optimal strategy  $\pi$  that minimizes the system cost should be obtained. Consequently, the scheduling problem is described as

$$P1 : u^* = \min U(s_t, a_t), \quad (16)$$

where  $u^*$  is the optimal joint solution produced by optimal policy  $\pi^*$ . Then, the optimal strategy is learned through interaction between the agent and the MTT-WSNs.

**4.2. DQN-Based Scheduling Strategy.** The scheduling process is modelled as MDP with quaternary-state which includes state space, action space, penalty, and the state transformed equation. We use a value function based on the Bellman equation to define the potential value for state  $s$  at time  $t$  in the future which is given by

$$P2 : v_t(s) = \frac{1}{T} E \left\{ \sum_{t=1}^T \gamma^t U(s_t, a_t) | s_1 = 0 \right\}, \quad (17)$$

where  $\gamma$  is the discount factor, and  $P2$  denotes the exception function for a long-term accumulation process. The iteration process is represented as follows

$$v_{t+1}(s) = \min_{s', D_t} \sum p(s', D_t | s, a) (D_t + \gamma v_{t+1}(s')). \quad (18)$$

In DQN, the agent interacts with the dynamic environment using a dynamic programming method to alleviate the above issue. The learning process with Q value is represented as follows, which is embodied as an approximating value function,

$$Q(s_t, a_t) = E[D_t + E[Q(s_{t+1}, a_{t+1})]], \quad (19)$$

and the updating process is given by

$$Q(s, a) = Q(s, a) + \alpha \left[ D + \max_{a'} \left( Q(s', a') - Q(s, a) \right) \right], \quad (20)$$

where  $\alpha$  is the learning rate.

In Figure 2, we describe the specific interaction process. In order to minimize the loss, state variables as input of primary network are learned and updated through replay memory. Each time the network is trained, a sample is selected randomly to reduce the relevance of data. To improve the stability of the algorithm, target network is established, the same as the primary network which is used to update the network parameters. Target network is applied to update Q value. It is noteworthy that the two networks are asynchronous.

Meanwhile, the DQN-based scheduling strategy algorithm is shown below,

The time complexity is given for algorithm analysis. Based on the mention above discussions, the feedback process of neural networks is mainly considered with high resource consumption. It implements the matrix computing with inverse operation. For our designed intelligent computing algorithm, we focus on the constructed hidden layers of neural networks and the number of iterations. Explicitly, the action  $a$  is sampled from a set of action  $\mathcal{A}$  with a time complexity  $\mathcal{O}(1)$  for each iteration [22]. In the hidden layers, the main consideration focuses on back propagation. According to [12], its time complexity is  $\mathcal{O}(k(v)TM_1)$ . Thereafter, the algorithmic complexity is given as  $\mathcal{O}(k(\phi)T\mathcal{M}_1|\mathcal{C}||\mathcal{A}_1|)$ , where  $\mathcal{A}_1$  and  $\mathcal{M}_1$  are the reduced action space and set of target nodes,  $\mathcal{C}$  is the number of sink nodes, and  $k(v)$  is the a function of the depth and number of the hidden layers  $v$ .

## 5. Performance Evaluation

In this section, we present numerical results to illustrate the performance of the proposed optimal policy.

*5.1. Simulation Set-up.* To begin with, the DQN agent obtains policy from MTT-WSNs' environment. Considering the mobile target tracking scenario in a stable region, we set the number of target nodes as  $\mathcal{N} = 2$ , the number of sensor nodes as  $\mathcal{M} = 40$ , and the distribution of sensor nodes is uniform. Meanwhile, the velocity of sensor nodes and targets is 2 m/s and 1 m/s, respectively. The maximum tolerate delays are 5 seconds and 6 seconds, respectively. We set the primary energy of each sensor node as 40 J. The coordinates of running into each region for two targets are given, and the moving trajectory is the stochastic distribution. The energy consumption of static nodes is set as 0.2 J and 0.8 J for each moving sensor. Parameters of the MTT-WSNs' environment in our system are listed in Table 1. The simulation is based on Python 3.0 and Tensor Flow framework.

## 6. Simulation Results and Discussion

Figure 3 depicts the systematic energy consumption under the consideration of fairness between energy and delay, i.e.,  $\xi_1 = \xi_2 = 0.5$ . With the different learning rates, system consumption is converging quickly when the number of iterations arrives at 120 steps approximately. It is obvious that different learning rates give diverse capability for convergence, and convergent velocity is faster correspondingly when the learning rate is equal to 0.6 which has the least fluctuations and fastest convergence speed after 90 iterations. When the value of the horizontal ordinate is 90, the agent can use the learned experience to schedule diverse sensor nodes considering self-energy and relative distance from each target.

Under the consideration of collaborative perception, we also obtain the performance of trade-off between energy and delay penalty. When we mainly consider energy, i.e.,  $\xi_1 = 0.9, \xi_2 = 0.1$  in Figure 4, the performance of convergent velocity is also compared with Figure 3. In this case, the lower energy consumption is realized while guaranteeing the convergence. When systematic energy consumption is convergent, energy consumption is reduced to 42.85% approximately compared with Figure 3, wherein energy consumption and delay are analysed simultaneously.

Figure 5 depicts the systematic delay cost, which mainly considers the delay, i.e.,  $\xi_1 = 0.1, \xi_2 = 0.9$ . Tendency is convergent when the value of the horizontal ordinate is 75 and the learning rate is 0.9. Due to the randomly sampled replay and updated memory, the convergence exists a little tremble, and the performance is better than convergence of systematic energy consumption.

Figure 6 shows the comparison with the Randomly Selecting Node (RSN) algorithm. Considering the energy and delay fairly, we obviously extract that the capability of DQN-based collaborative perception algorithm is better than RSN. We analyse the whole interacting process of the agent and MTT-WSNs when the value of horizontal ordinate is fewer than 120, and the energy consumption is reduced to 76.2%, 67.3%, and 70.9% when learning rate are 0.6, 0.8, and 0.9, respectively. The effect is obviously remarkable compared with RSN not only in energy consumption but also when the convergence outperforms RSN algorithm.

## 7. Conclusions

In this paper, we have studied the mobile multitarget tracking scheduling management in the complex MTT-WSNs. The problem of mobile scheduling is formulated as a MDP, taking account of diverse important degree for different target nodes and diverse tracking capabilities for different sensor nodes. Moreover, the multiregional tracking and scheduling strategy is treated as an important component to reduce the size of state space with a low computational complexity. To cope with the mobile multitarget tracking in a complex scenario, collaborative perception among sensor nodes and the deep Q-learning algorithm is adopted to obtain the optimal scheduling strategy. Since the agent learns important information from MTT-WSNs through

the interacting process, the proposed scheme can enhance the network performance obviously and reduce the tracking delay as illustrated in simulation experiments. Furthermore, the systematic energy consumption can be reduced by the scheduling policy through sink nodes. Finally, the time complexity is analysed to promote further work smoothly.

## Data Availability

All data included in this study are available upon request by contact with the corresponding author.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the Research Funds of West China Hospital of Sichuan University.

## References

- [1] K. Xiong, S. Leng, J. Hu, X. Chen, and K. Yang, "Smart network slicing for vehicular fog-RANs," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3075–3085, 2019.
- [2] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.
- [3] C. H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 685–692, CA, USA, June 2010.
- [4] C. Liang and Y. U. Shao-Hua, "Preliminary study on the trend of 6G mobile communication," *Study on Optical Communications*, vol. 45, no. 4, pp. 1–8, 2019.
- [5] E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Kténas, N. Cassiau, and C. Dehos, "6G: the next frontier," 2019, <http://arxiv.org/1901>.
- [6] M. Z. A. Bhuiyan, G. Wang, and A. V. Vasilakos, "Local area prediction-based mobile target tracking in wireless sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1968–1982, 2015.
- [7] J. Luo, Z. Zhang, C. Liu, and H. Luo, "Reliable and cooperative target tracking based on WSN and WiFi in indoor wireless networks," *Access*, vol. 6, pp. 24846–24855, 2018.
- [8] C. Fan, B. Li, J. Hou, Y. Wu, W. Guo, and C. Zhao, "Robust fuzzy-learning for partially overlapping channels allocation on UAV Communication Networks," *IEEE Transactions on Mobile Computing*, 2018, <https://arxiv.org/pdf/1806.10756>.
- [9] T. Zhao, S. Zhou, X. Guo, and Z. Niu, "Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing," *2017 IEEE International Conference on Communications*, 2017, pp. 1–7, Paris, France, May 2017.
- [10] L. Wan, X. Wang, and G. A. Bi, "Range-Doppler-Angle Estimation Method for Passive Bistatic Radar," 2017.
- [11] G. Qiao, S. Leng, Y. Zhang, and L. Xu, "Joint multi-RATs and cloud-service matching scheme in wireless heterogeneous networks," *IEEE International Conference on Communications*, 2016, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.
- [12] G. Qiao, S. Leng, M. Zeng, and Y. Zhang, "Matching game approach for charging scheduling in vehicle-to-grid networks," *ICC 2017 IEEE International Conference on Communications*, 2017, pp. 1–6, Paris, France, May 2017.
- [13] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6G communication networks: state-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019.
- [14] C. Sergiou, M. Lestas, P. Antoniou, C. Liaskos, and A. Pitsillides, "Complex systems: a communication networks perspective towards 6G," *Access*, vol. 8, pp. 89007–89030, 2020.
- [15] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.
- [16] K. Chia-Hsu, C. Tzung-Shi, and S. Siou-Ci, "Robust mechanism of trap coverage and target tracking in mobile sensor networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3019–3030, 2018.
- [17] X. Wang, J. Guo, and N. Cui, "Adaptive extended Kalman filtering applied to low-cost MEMS IMU/GPS integration for UAV," *International Conference on Mechatronics and Automation*, vol. 2009, pp. 2214–2218, 2009.
- [18] H. Zhang, X. Zhou, Z. Wang, H. Yan, and J. Sun, "Adaptive consensus-based distributed target tracking with dynamic cluster in sensor networks," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1580–1591, 2018.
- [19] J. Li, Z. Xing, W. Zhang, Y. Lin, and F. Shu, "Vehicle tracking in wireless sensor networks via deep reinforcement learning," *IEEE Sensors Letters*, vol. 4, no. 3, pp. 1–4, 2020.
- [20] Z. Yuan, X. Huang, L. Sun, and J. Jin, "Software defined mobile sensor network for micro UAV swarm," *IEEE International Conference on Control and Robotics Engineering (ICCRE)*, vol. 2016, pp. 1–4, 2016.
- [21] S. Mori, "Cooperative sensing data collecting framework by using unmanned aircraft vehicle in wireless sensor network," *IEEE International Conference on Communications (ICC)*, vol. 2016, pp. 1–6, 2016.
- [22] L. Zhou, S. Leng, Q. Liu, S. Mao, and Y. Liao, "Intelligent resource collaboration in mobile target tracking oriented mission-critical sensor networks," *IEEE Access*, vol. 8, pp. 10971–10980, 2020.
- [23] L. A. Prashanth, A. Chatterjee, and S. Bhatnagar, "Two time-scale convergent Q-learning for sleep-scheduling in wireless sensor networks," *Wireless Networks*, vol. 20, no. 8, pp. 2589–2604, 2014.
- [24] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, "Dynamic content update for wireless edge caching via deep reinforcement learning," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1773–1777, 2019.
- [25] J. Wu and G. Li, "Drift calibration using constrained extreme learning machine and Kalman filter in clustered wireless sensor networks," *IEEE Access*, vol. 8, pp. 13078–13085, 2020.
- [26] X. Bai, Z. Wang, L. Zou, and C. Cheng, "Target tracking for wireless localization systems with degraded measurements and quantization effects," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9687–9697, 2018.

- [27] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-UAV," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 82–89, 2018.
- [28] L. Zhou, S. Leng, Q. Liu, and Q. Wang, "Intelligent UAV swarm cooperation for multiple targets tracking," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 743–754, 2022.