Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1177544, 18 pages
https://doi.org/10.1155/2022/1177544

Research Article

Deep Reinforcement Learning-Based Joint Satellite Scheduling and Resource Allocation in Satellite-Terrestrial Integrated Networks

Yabo Yin,[1,2] **Chuanhe Huang**,[1,2] **Dong-Fang Wu**,[1,2] **Shidong Huang**,[1,2] **M. Wasim Abbas Ashraf**,[1,2] **Qianqian Guo**,[3] **and Lin Zhang**[4]

[1]*School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*Hubei LuoJia Laboratory, Wuhan 430072, China*
[3]*School of Information Engineering, Zhengzhou Institute of Finance and Economics, Zhengzhou 450053, China*
[4]*Wuhan Maritime Communication Research Institute, Wuhan 430072, China*

Correspondence should be addressed to Chuanhe Huang; huangch@whu.edu.cn

Received 21 December 2021; Revised 6 January 2022; Accepted 31 January 2022; Published 24 February 2022

Academic Editor: Junjuan Xia

Satellite-terrestrial integrated networks (STINs) are considered to be a new paradigm for the next generation of global communication because of its distinctive merits, such as wide coverage, high reliability, and flexibility. When the satellite associates with different base stations (BSs) and adopts different channels for communication, the utility of offloading data to BSs is different. In our work, we study how to jointly associate satellites with appropriate BSs and allocate channels to satellites. Our purpose is to maximize the utility of the data offloaded from satellites to BSs while considering the load balance of BSs. However, some satellites are often unable to connect to BSs because of their periodic flight characteristic, which makes the joint satellite-BS association and channel allocation more challenging. To solve the problem that satellites sometimes cannot connect to BSs, we abstract the communication model between satellites and BSs into a bipartite graph and add a virtual BS to ensure that all satellites can connect to at least one BS. Then, in the constructed joint optimization problem, we solve the assignment of satellites and channels simultaneously. Considering that the joint optimization problem is nonconvex, we use double deep Q-Network (DDQN) for achieving the optimal strategy of satellite association and channel allocation. Furthermore, the reward value in most state transition information generated by satellites is 0, which leads to the low learning efficiency of DDQN. Aiming at enhancing the learning efficiency of DDQN, the priority sampling-based DDQN (PSDDQN) algorithm is proposed. Experimental results demonstrate that PSDDQN gets better utility and achieves the load balance of BSs compared with other algorithms.

1. Introduction

Recently, 5G technology has developed rapidly, and many 5G base stations have been deployed at the same time. Therefore, 5G network communication services can be provided in many places, such as urban and metropolitan areas. However, for those remote areas without communication facilities, such as deserts, oceans, and other places, it is difficult to provide people with communication services. LEO satellite network has attracted researchers' attention due to its characteristics of wide coverage, low delay, and high bandwidth, and it can provide communication services for those places which are not covered by base stations [1]. Therefore, the STINs become a new paradigm for providing seamless communication. Meanwhile, people's demands for high quality video, voice, and other multimedia services grow explosively. Therefore, how to efficiently manage resources in STINs for providing people with better Internet services has become a new challenge [2, 3].

Resource allocation is a key technology that affects the performance of STINs. But most of researches focus on radio spectrum allocation [4, 5], power allocation [6, 7],

and other issues. And there is little research that studies the joint satellite association and channel allocation. Hu et al. [8] proposed a competitive market scheme to solve user association issue of satellite-drone networks. To address the user association issue of heterogeneous networks, a distributed belief propagation method was proposed [9]. Khalili et al. [10] investigated user association of the uplinks in heterogeneous networks and used maximization-minimization theory and augmented Lagrange method to guarantee users' data transmission rate. Zhao et al. [11] used multiagent RL approach for achieving the optimal user association strategy while considering the service requirements of users in heterogeneous cellular networks. However, these approaches are not suitable in STINs. The main reasons are as follows: (1) In the network scenarios mentioned above, when optimizing user association problem, users are within the communication range of BSs. That is, users are covered by at least one BS at any time. However, in STINs, many satellites cannot connect to BSs because satellites communicate with BSs by the Line-of-Sight (LoS) way. (2) Because of the periodic movement of satellites, the communication process between satellites and BSs is discontinuous rather than continuous. (3) The load balance of BSs seriously affect the performance of STINs. If the load balance of BSs is not considered, the capacity of some BSs will be exhausted in advance. And this degrades the performance of STINs. However, the above literature ignores the load balance of BSs.

In our research, the joint satellite scheduling and resource allocation in STINs are investigated. When satellites communicate with BSs, allocating appropriate BSs and channels to satellites for transmitting data has a great impact on the utility of STINs. Because the problem of joint satellite association and channel allocation is coupled and nonconvex, conventional methods cannot effectively obtain the optimal satellite association and channel allocation strategy.

Fortunately, in recent years, artificial intelligence algorithm such as reinforcement learning (RL) has developed rapidly and applied to many fields. RL has been used to solve system control problems [12]. Moreover, it has been applied in combinatorial optimization [13, 14], capacity management [15], resource management [16, 17], etc., especially in games and chess. Furthermore, RL also has good performance in dynamic network environment. Inspired by this, we try to use RL approach to achieve the optimal strategy in STINs.

This paper mainly studies the joint satellite scheduling and resource allocation in STINs. Our purpose is to maximize the utility of data offloaded from satellites to BSs as much as possible while considering the load balance of BSs. The joint optimization problem of the satellites and channel assignments is constructed. Then, a priority sampling-based double DQN (PSDDQN) method is used to address the aforementioned problem. We give the main contributions in the following:

(1) We focus on the joint satellite association and channel allocation while considering the load balance of BSs in STINs. We formulate the problem of joint satellite association and channel allocation as a joint optimization problem. And, we use RL approach to solve it

(2) We propose a bipartite graph with virtual BS to describe the communication model between satellites and BSs. By adding a virtual BS, we ensure that the action space of all satellites is the same. So we can easily determine the action space of our proposed RL model

(3) Considering that the reward value in most state transition information is 0, we propose a priority sampling-based DDQN (PSDDQN) to enhance the performance of DDQN. Moreover, to reduce the consumption of computing resources caused by sorting state transition information, we use SumTree structure to store state transition information. Experimental results demonstrate that PSDDQN can converge quickly and get better performance than the selected baseline approaches

The rest of our paper unfolds below. The recent research developments are summarized in Section 2. In Section 3, the system framework and problem formulation are illustrated. Section 4 describes our PSDDQN method. Section 5 evaluates our PSDDQN method and discusses the experimental results. We conclude this article and give the future research in Section 6.

2. Related Work

Many works investigated user association in wireless networks. Feng et al. [18] proposed a repeated game-based user association scheme while considering spectrum allocation to maximize users' data transmission rate of the MIMO system. Liu et al. [19] used Lagrange dual decomposition approach to address the user association issue in two different access schemes. Liu et al. [20] jointly optimized user association and power control and designed a semidistributed solution to obtain the optimal results. Considering that users' mobility affects the performance of heterogeneous networks, Cheng et al. [21] introduced the users' mobility and proposed a multiagent RL approach to maximize the system capacity. Based on users' preferences information, Zalghout et al. [22] proposed a priority-based user association approach to maximize user's QoS in heterogeneous wireless networks.

There are many works on the resource allocation of satellite networks. Zhu et al. [23] used multidimensional knapsack theory to minimize energy consumption while considering user's delay constraint. To improve the spectrum utilization of satellites, Zuo et al. [24] jointly optimized allocation of time, spectrum, power, and beam and solved it by heuristic algorithm. Mai et al. [25] used Stackelberg game method to reduce the transmission delay of remote sensing data. Deng et al. [26] jointly optimized virtual machine assignment and power allocation to minimize energy consumption in cloud-based satellite communication networks. Shahid et al. [27] used radio utilization as load metric and

proposed a load balancing-based resource management method for improving STINs' performance. Ji et al. [28] presented a data offloading method for solving the energy overhead issue in multicell STINs. Deng et al. [29] designed a satellite constellation deployment solution with minimum number of satellites while meeting backhaul requirements of users in STINs. However, none of them paid attention to the joint satellite association and channel allocation.

In recent years, researchers have used artificial intelligence algorithms to solve network problems. He et al. applied deep learning algorithm to memory optimal detection [30] and optimal strategy search [31] in MIMO system. Lai [32] used federated learning to select outdated access point in MEC networks. Tang and Chen [33] designed a federated edge learning framework to reduce the computational latency of tasks in IoT networks. RL has been applied to ubiquitous computing [34], network security [35], and resource allocation. For reducing the task execution time, the Q-learning approach was adopted to manage computing resources in IoT [36]. The Q-learning-based task offloading scheme was used to maximize the utility of system [37]. The designed deep RL-based radio resource management method was used to improve the radio utilization [38]. Zhang et al. [39] adopted deep RL approach to allocate resource while meeting users' reliability requirements. The multiagent RL-based hierarchical task management strategy can satisfy users' communication requirements [40]. Luong et al. [41] jointly optimized UAV position and transmitted beamforming and UAV-UE association in UAV-assisted wireless networks. Considering that this optimization problem was nonconvex, a method based on deep Q-learning was proposed for allocating resources.

3. System Model and Problem Formulation

3.1. Network Model. Figure 1 shows the structure of STINs. The STINs consist of N satellites and L BSs. For the terrestrial networks, it can connect to LEO satellite networks through BSs with satellite gateways. Because the LEO satellite network topology changes dynamically over time, we use snapshot method to discretize the satellite network topology according to reference [42]. During a snapshot, the network topology is fixed. The whole operation time is T, and the number of snapshots is N_T; then, we can easily get

$$\sum_{t=1}^{N_T} T_t = T. \tag{1}$$

The more snapshots, the higher the precision of the representation of satellite network topology. But extensive storage resources and computing resources are needed to deal with these snapshots. To ensure the precision of satellite network topology and reduce the consumption of storage resources, the duration of the snapshot should be less than the communication time between satellites and BSs. We define it as

$$T_t \le \min\left\{t_{i,j}\right\}, \forall i \in \{1, 2, 3, \cdots, N\}, j \in \{1, 2, 3, \cdots, L\}, \tag{2}$$

where variable $t_{i,j}$ expresses the communication time between ith satellite and jth BS. The structure of STINs is expressed by $G = (V, E)$. And graph G is undirected graph. The node set in STINs is denoted by V. And E represents the communication links between satellites and BSs.

3.2. Communication Model. When BSs are not within the coverage of satellites, satellites cannot communicate with BSs. Moreover, each satellite may cover many BSs sometimes. Therefore, the communication model between satellites and BSs can be abstracted into a bipartite graph. We show the structure of bipartite graph.

In Figure 2, grey nodes represent satellites, and orange nodes represent BSs. The grey lines indicate the communication links between satellites and BSs. From Figure 2, we observe that only four satellites can connect to BSs, and other satellites of the satellite networks cannot connect to any BSs. Considering the BSs covered by different satellites are different, the action space of different satellites is different. Therefore, the RL method cannot be directly applied to STINs. To solve this problem, we introduce the virtual BS. By adding a virtual BS, each satellite can be connected to a BS, and the action space of each satellite is same. We show the bipartite graph with virtual BS.

In Figure 3, the blue node is the added virtual BS. When a satellite cannot connect to any BS, we connect this satellite to the added virtual BS. Then, each satellite in STINs can be connected to at least one BS at any snapshot. Hence, the satellite can select one BS from $L + 1$ BSs to establish a communication link at any time. This operation ensures that the action space of each satellite is same, which is convenient for the definition of action space of RL in subsequent section.

3.3. Channel Model. Considering that we have added a virtual BS to STINs, we use set $\mathscr{L} = \{0, 1, 2, 3, \cdots, L\}$ to represent the BSs of STINs. And the index of the added virtual BS is 0. Only when the satellite is associated with one BS and occupies one channel of this BS can the satellite communicate with this BS. We use binary decision variable $s_{i,l}(t)$ that indicates whether satellite i associate with BS l at time t. If satellite i is assigned to BS l, then $s_{i,l}(t) = 1$; otherwise, $s_{i,l}(t) = 0$. For convenience, the satellite can only connect to one BS during one snapshot. So we can get

$$\sum_{l=0}^{L} s_{i,l}(t) \le 1, \forall i \in \{1, 2, 3, \cdots, N\}. \tag{3}$$

Moreover, we assume that each BS has M channels, and each channel can only be assigned to one satellite at any time. Therefore, each BS can connect at most M satellites simultaneously; we can get

$$\sum_{i=1}^{N} s_{i,l}(t) \le M, \forall l \in \{1, 2, 3, \cdots, L\}. \tag{4}$$

Here, we assume that the number of satellites connected to the virtual BS is not limited. So the added virtual BS can

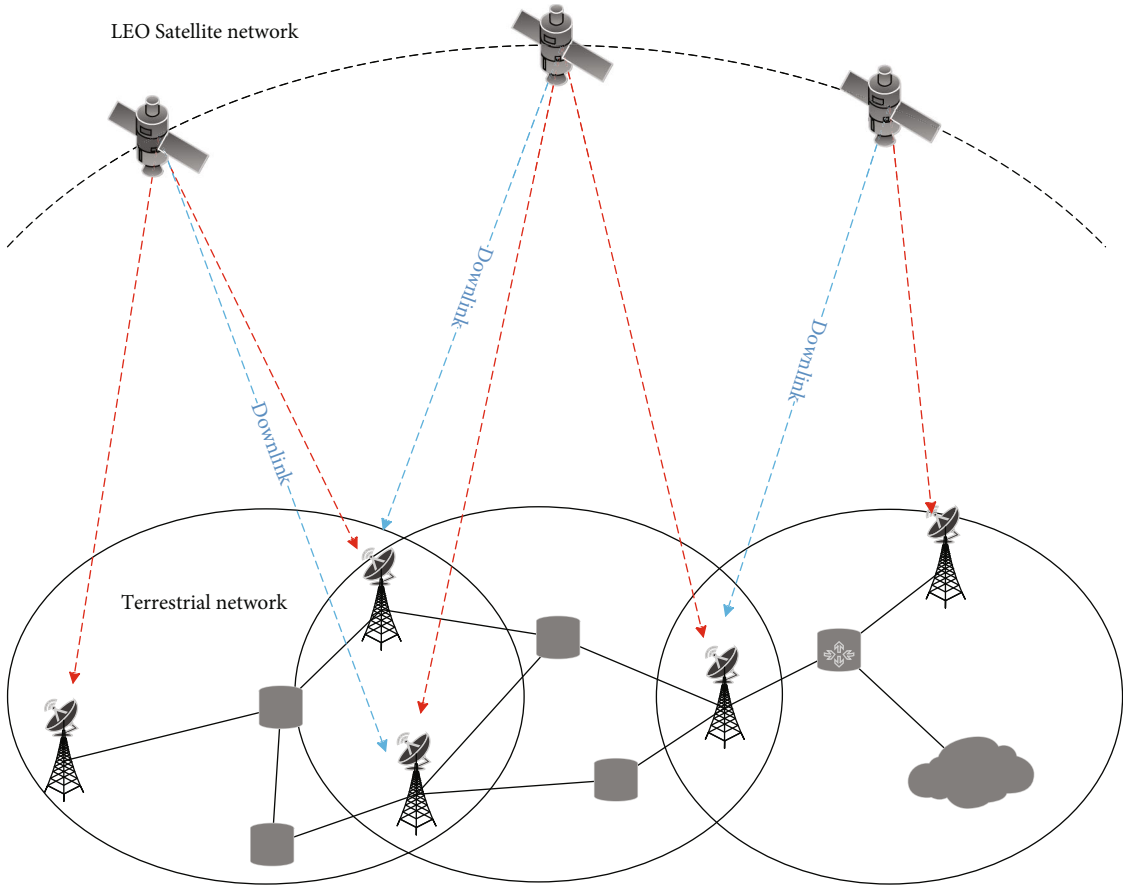

Figure 1: The structure of STINs. The dotted lines (including red and blue) indicate that the satellite is capable of connecting to the BSs under its coverage. The blue dotted line means the communication link established between the satellite and the BS. And each satellite is able to associate with one BS for communication in a time slice.

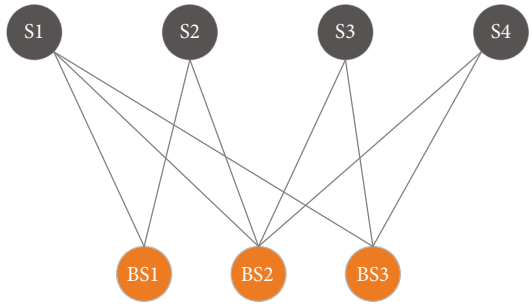

Figure 2: Structure of bipartite graph.

connect to all satellites. Considering that the index of the virtual BS is 0, we can get

$$\sum_{i=1}^{N} s_{i,0}(t) \le N. \tag{5}$$

Assigning different channels to satellites has an important impact on the data transmission rate between satellites and BSs. Assumed that different channels of the BS are orthogonal. Here, we use binary channel-allocation variable $c_{i,m}(t)$ that indicates whether satellite i communicate with BS on channel m. If satellite i uses channel m for communication at time t, $c_{i,m}(t) = 1$; otherwise, $c_{i,m}(t) = 0$. Here, one channel can only be allocated to one satellite. We can easily get

$$\sum_{m=1}^{M} c_{i,m}(t) \le 1, \forall i \in \{1, 2, 3, \cdots, N\}. \tag{6}$$

Considering that the satellite may cover many BSs, when satellite i communicates with BS l, other BSs will also interfere with the communication process. Therefore, the cochannel influence of other BSs should be considered. We obtain the SINR by

$$\mathrm{SINR}_{i,l,m}(t) = \frac{p_{i,l,m}(t) g_{i,l,m}(t) s_{i,l}(t) c_{i,m}(t)}{\sum_{b \in L \setminus \{0,l\}}^{L} p_{i,b,m}(t) g_{i,b,m}(t) s_{i,b}(t) c_{i,m}(t) + BN_0}. \tag{7}$$

The virtual BS does not interfere with the communication process of other BSs. Variable $p_{i,l,m}$ represents the transmit power operating on channel m between satellite i and BS

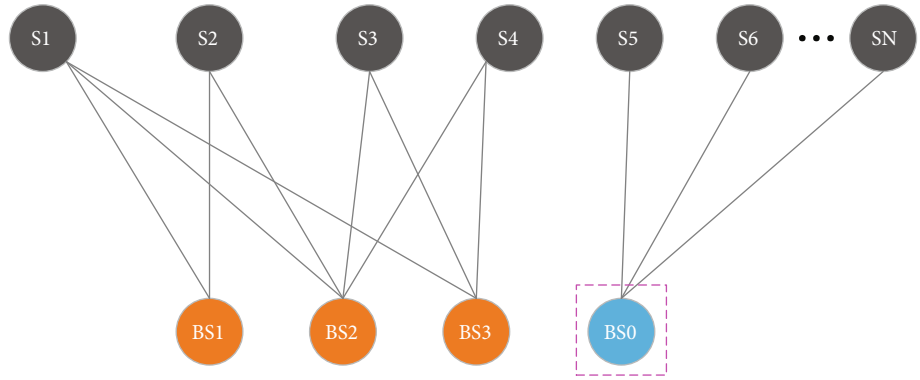

Figure 3: Bipartite graph with virtual BS.

l, and variable $g_{i,l,m}$ indicates the channel gain of channel m. The channel m is between ith satellite and lth BS. The variable N_0 indicates Gaussian white noise. B expresses the channel bandwidth.

Let the longitude and latitude of BS l be λ_l and φ_l, respectively. And λ_i and φ_i represent the longitude and latitude of satellite i. The height of satellite orbit is h, so the distance between satellite i and BS l is given by

$$d_{i,l} = (R_E + h)\sqrt{1 + \left(\frac{R_E}{R_E + h}\right)^2 - 2\frac{R_E}{R_E + h}(\cos(\lambda_l - \lambda_i)\cos\varphi_i \cos\varphi_l + \sin\varphi_i \sin\varphi_l)}, \tag{8}$$

where variable R_E denotes the earth's radius. Thus, we can get the path loss of the signal in the atmosphere

$$L_{i,l}(t) = 32.44 + 20\log d_{i,l} + 20\log f, \tag{9}$$

where variable f represents the carrier frequency. Thus, we can get the channel gain of satellite i

$$p_{i,l,m}(t) = A_{i,m} - L_{i,l}(t), \tag{10}$$

where variable $A_{i,m}$ represents the antenna gain operating on channel m of satellite i. Based on the computed SINR, we can get the data transmission rate by

$$r_{i,l,m}(t) = B\log_2(1 + \mathrm{SINR}_{i,l,m}(t)). \tag{11}$$

Variable B represents the channel bandwidth. Considering that the communication time between satellite i and BS l is different in a time slice, we can obtain the valid communication time between satellite i and BS l

$$\tau_{i,l} = \min(T_t, t_{i,l}). \tag{12}$$

3.4. Problem Formulation. Each satellite wants to get its maximum utility. Our object is to obtain the maximum utility of the data offloaded from satellites to BSs while considering the load balance of BSs. The utility of the data offloaded by satellite i is u_i. And it consists of the utility generated by the offloaded data and the cost of the BS storing these offloaded data. We define it as

$$u_i(t) = (\omega p_i - \xi f_l) r_{i,l,m}(t)\tau_{i,l}, \tag{13}$$

where variables ω and ξ are weight coefficients. Variable f_l represents the cost of storing unit data in BS l. As the storage capacity of BSs is consumed, the storage cost will gradually increase. Here, we define it as

$$f_l = \frac{1}{C_l - \sum_{i=1}^{N} r_{i,l,m}\tau_{i,l} + Q}, \tag{14}$$

where variable C_l represents the remaining capacity of BS l and variable Q is a constant. Based on the abovementioned, we can get the utility generated by the data offloaded from all satellites to BSs in a time slice,

$$\Gamma(t) = \sum_{i=1}^{N}\sum_{l=0}^{L}\sum_{m=1}^{M} u_i(t). \tag{15}$$

The running time T of satellites is composed of a series of time slices. Therefore, we can get the utility generated by the data offloaded from all satellites to BSs during the whole operation time T. We define it as

$$\begin{aligned} \Upsilon &= \max\left(\sum_{t=1}^{N_T}\Gamma(t)\right), \\ s.t. &\sum_{l=0}^{L} s_{i,l}(t) \le 1, \forall i \in \{1, 2, 3, \cdots, N\}, \\ &\sum_{i=1}^{N} s_{i,l}(t) \le M, \forall l \in \{1, 2, 3, \cdots, L\}, \\ &\sum_{i=1}^{N} s_{i,l}(t) \le N, l = 0, \\ &\sum_{m=1}^{M} c_{i,m}(t) \le 1, \forall i \in \{1, 2, 3, \cdots, N\}, \end{aligned} \tag{16}$$

where variable N_T denotes the number of snapshots. Considering that this problem is nonconvex, traditional approaches cannot be directly used to obtain the optimal results. Therefore, we adopt deep RL for achieving the optimal satellite association and channel allocation strategy.

4. Priority Sampling-Based DDQN Algorithm for Joint Satellite Scheduling and Resource Allocation

4.1. Reinforcement Learning Model

4.1.1. State Space. The state space mainly includes the real-time status of all satellites and the status of BSs. For the satellite, the status information includes its position at time t and valid communication time with different BSs. For the BS, the status information includes its remaining storage capacity and channel allocation state. We get the state space of satellite i by

$$S_i = \left\{ \begin{matrix} \lambda_i, \varphi_i, \tau_{i,0}, \tau_{i,1}, \tau_{i,2}, \cdots, \tau_{i,L}, s_{i,0}, s_{i,1}, s_{i,2}, \cdots, s_{i,L}, c_{i,1}, c_{i,2}, \cdots, c_{i,M}, \\ C_0, C_1, C_2, \cdots, C_L \end{matrix} \right\}, \forall i \in \{1, 2, 3, \cdots, N\}, \tag{17}$$

where variables λ_i and φ_i are introduced in Equation (8). Variable C_l represents the remaining capacity of BS l. Variable $s_{i,l}$ indicates whether satellite i is associated with BS l. Variable $c_{i,m}$ indicates whether satellite i adopts channel m for communication.

4.1.2. Action space. When the satellite communicates with the BS, the satellite must be associated with one BS and occupy one channel of this BS. Therefore, the action of satellite mainly includes two parts: satellite association and channel allocation. We define the action space of satellite i as

$$A_i = \left\{ s_{i,0}, s_{i,1}, s_{i,2}, \cdots, s_{i,L}, c_{i,0}, c_{i,1}, c_{i,2}, \cdots, c_{i,M} \right\}, \forall i \in \{1, 2, 3, \cdots, N\}. \tag{18}$$

From Equation (18), we can see that satellite association and channel allocation are coupled. For convenience, we use matrix **E** that represents the action space of the satellite

$$E = \begin{bmatrix} e_{0,1}, e_{0,2}, & e_{0,3}, \cdots, & e_{0,M} \\ e_{1,1}, e_{1,2}, & e_{1,3}, \cdots, & e_{1,M} \\ \vdots & \vdots & \vdots \\ e_{L,1}, e_{L,2}, & e_{L,3}, \cdots, & e_{L,M} \end{bmatrix}, \forall l \in \{0, 1, 2, 3, \cdots, L\}, m \in \{1, 2, \cdots, M\}. \tag{19}$$

We use binary variable $e_{l,m}$ that represents the joint satellite association and channel allocation of the satellite. For a satellite, when BS l and its channel m are assigned to this satellite simultaneously, $e_{l,m} = 1$; otherwise, $e_{l,m} = 0$.

4.1.3. Reward value. Reward value seriously affects the performance of RL, and the agent use it to obtain the optimal strategy. When satellites are associated with different BSs and adopt different channels for communication, the utility of data offloaded from satellites to BSs is different. Moreover, the storage cost of the BS also changes with the consumption of its capacity. In our research, our purpose is to obtain the maximum utility of offloaded data while considering the load balance of BSs. The reward value consists of the utility generated by the offloaded data and the cost of BS for storing these data. We define it as

$$r_i = (\omega p - \xi f_l) r_{i,l,m}(t) \tau_{i,l}, \tag{20}$$

where variables ω and ξ are the weight coefficients, which are initialized by analytic hierarchy process method. And variable $r_{i,l,m}(t)$ denotes the downlink data transmission rate between satellite i and BS l operating on channel m. Variable $\tau_{i,l}$ denotes the communication time between satellite i and BS l. Furthermore, we also normalize the utility of offloaded data and the cost of BSs.

4.2. Double Deep Q-Network (DDQN) Method. DQN is used widely in discrete scenarios. Meanwhile, the Q value is approximated by deep neural network. After the iterative training process, the obtained Q value is close to the true Q value

$$Q^*(s, a) \approx Q(s, a; \theta). \tag{21}$$

Moreover, the experience relay buffer scheme and gradient descent method are used for updating the parameter θ of neural network

$$L(\theta) = (y_i - Q(s, a; \theta))^2, \tag{22}$$

$$\nabla_\theta L(\theta) = (y_i - Q(s, a; \theta)) \nabla_\theta Q(s, a; \theta), \tag{23}$$

where variable y_i represents the Q value obtained by DQN; we can get it by

$$y_i = \begin{cases} r_i & , \text{is\_end}_i \text{ is true} \\ r_i + \gamma \max_{a'} Q\left(s', a'; \theta\right), & \text{is\_end}_i \text{ is false} \end{cases}. \tag{24}$$

Because greedy strategy is adopted for selecting action in DQN, the Q value obtained by DQN is over estimated. To solve this problem, double DQN (DDQN) selects actions and calculates the target Q value by two independent network structures. These two independent network structures are online Q-Network and target Q-Network, respectively, and they have the same network structure. The target Q-Network is used to decouple the action selection and the Q value estimation, which can solve the over estimation problem of Q value in DQN. When updating the parameters of DDQN, we select the action with the largest Q value by

$$a^{\max}\left(s'; \theta\right) = \arg \max_{a'} Q\left(s', a; \theta\right). \tag{25}$$

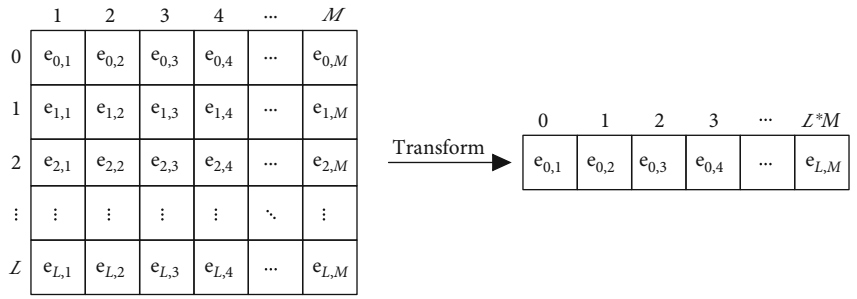

Figure 4: Action space transformation process.

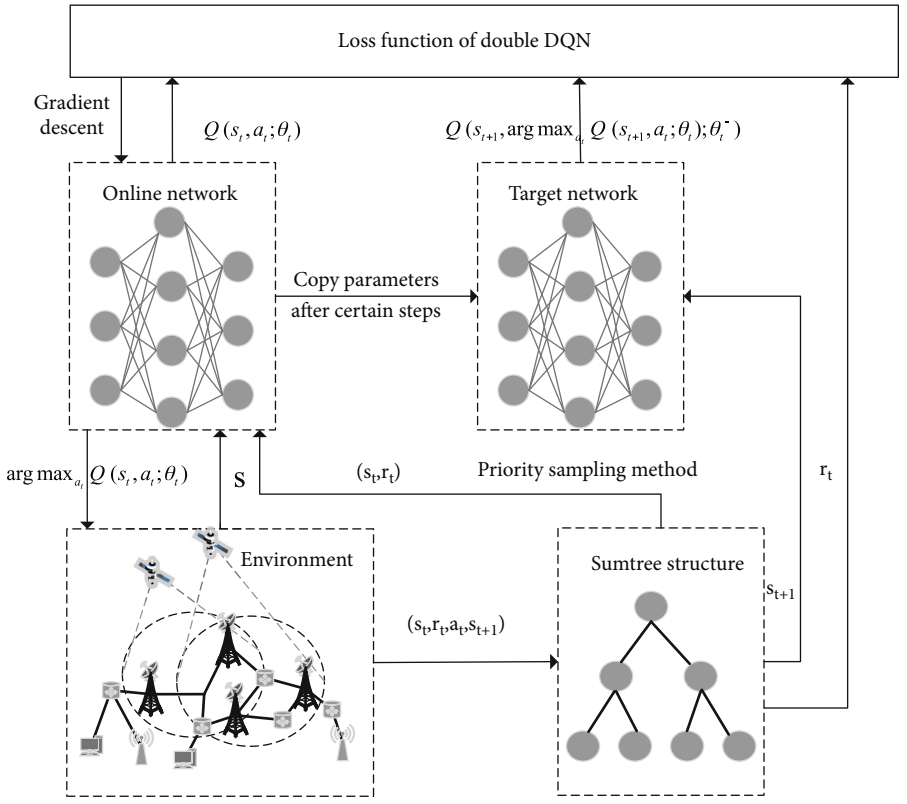

Figure 5: Structure of PSDDQN.

Then, the target Q value in DDQN is calculated by target Q-Network

$$y_i^{\mathrm{DDQN}} = \begin{cases} r_i & , \text{is\_end}_i \text{ is true} \\ r_i + \gamma Q'\left(s', a^{\max}\left(s'; \theta\right); \theta'\right), & \text{is\_end}_i \text{ is false} \end{cases}, \tag{26}$$

where r_i is the instant reward value. Variable γ denotes the discount rate. And the discount rate indicates the importance of instant reward and future reward. $\gamma = 0$ indicates that the agent is myopic, and the agent only considers the instant reward. When the discount rate is 1, the agent is farsighted and only considers the future reward. In practice, the value of γ is generally between 0 and 1.

Then, we use the target Q value to calculate the TD-error. Finally, we calculate the gradients according to the obtained TD-error and update the weights of online Q-Network. In the process of training, online Q-Network assigns its weights to target Q-Network every certain number of training steps to update the parameters of target Q-Network.

The last layer of DDQN is a fully connected layer. However, the action space of the satellite is a two-dimensional matrix. In order to train DDQN, we transform the action space of the satellite into a one-dimensional vector with size $(L + 1) * M$. The specific transformation process is shown in Figure 4.

Input: number of episodes, *Num_Episodes*; number of time slices, *Num_Timeslices*; number of leaves of Sumtree structure, B; exploration rate, ε; update frequency, F; learning rate, α ; number of satellites, N;
Output: the weight of online Q-Network, θ;
1. Initialize the state of STINs, including the capacity C of BSs, antenna model, channel model, satellite orbit parameters, and the initial positions of satellites;
2. Randomly initialize the weight θ of online Q-Network; for the weight in target Q-Network, $\theta'=\theta$;
3.For *episode* =1 to *Num_Episodes* do
4. For *time* =1 to *Num_Timeslices* do
5. For i =1 to N do
6. Get the state information of satellite i from the ground control centre at time t, s_i;
7. End for
8. Get the sate information of BSs from the ground control centre at time t, H;
9. Obtain the state information of STINs at time t, $S=(s_1,s_2,s_3,\ldots,s_N, H)$;
10. Get the next state of STINs S' from the ground control centre at t+1 and its termination *flag*;
11. For i =1 to N do
12. Use ε-greedy strategy to select an action, a_i;
13. The agent execute action a_i and obtain the instant reward r_i by Equation (20);
14. Store state transition information (s_i, a_i, r_i, s_i') in the Sumtree structure;
15. End for
16. $S= S'$;
17. The BSs and satellites send their state information to the ground control centre for updating the state information of STINs;
18. Sample samples from the Sumtree structure, and compute the loss of Q-value of each sample according to Equation (29);
19. Compute the gradient of each sample according to Equation (30);
20. Update the weight of online Q-Network according to the back propagation algorithm;
21. Compute the *TD-error* value of each sample according Equation (32) and update its priority by Equation (33);
22. Update the parameters of target Q-Network every frequency F, let $\theta'=\theta$;
23. End for
24.End for

Algorithm 1: PSDDQN for joint satellite association and channel allocation.

Table 1: Parameters of simulation.

Parameters	Value
Channel bandwidth B	41.67 kHz
Transmitter EIRP p	51.6 dBW
Antenna gain g	53.7 dBi
Antenna diameter of BS D	3.5 m
Carrier frequency f	20 GHz
Path loss model Loss	$32.44 + 20 \log d + 20 \log f$ (f:MHz)
Noise power N_0	-174 dBm/Hz
Number of satellites N	120
Number of BSs L	6
Number of channels M	4
Number of time slices *Num_timeslices*	30
Weight of utility ω	0.7
Weight of storage cost ξ	0.3

We assume the index of action $e_{l,m}$ is index$(e_{l,m})$, and we can get the location of $e_{l,m}$ in the last layer of DDQN

$$\text{index}(e_{l,m}) = l * M + m, \forall i \in \{1, 2, 3, \cdots, L\}, m \in \{1, 2, 3, \cdots, M\}. \tag{27}$$

By transforming the action space into a one-dimensional vector, we can use the state transition information to train DDQN.

4.3. Priority Sampling-Based DDQN Algorithm for Joint Satellite Scheduling and Resource Allocation. In DDQN, the

Table 2: Parameters of PSDDQN algorithm.

Parameters	Value
Batch size	32
Learning rate α	0.005
Number of leaves of SumTree structure	2000
Number of neurons in input layer	156
Number of neurons in hidden layers	64, 32
Number of neurons in output layer	24
Discount rate γ	0.9
Activation function	ReLU
Optimizer	Gradient descent optimizer

uniform sampling method is used to select samples when calculating the gradient and updating the model's parameters. If samples in experience replay buffer are evenly distributed, uniform sampling method is effective. However, because satellites communicate with BSs by LoS, few satellites can communicate with BSs at any time. Moreover, most satellites are connected to virtual BS. Therefore, most of state transition information generated by satellites is from virtual BS to virtual BS because the communication time between satellites and BSs is short. At this time, the reward received by the satellite is 0. Thus, the reward of most state transition information in experience replay buffer is 0. If these samples are uniformly sampled, the learning performance of DDQN model will be degraded. Therefore, we use priority sampling method to enhance the performance of DDQN.

For a sample, the greater its priority, the greater the probability of being sampled. We define the sampling probability of the sample as

$$P(j) = \frac{p(j)}{\sum_i p(i)}, \tag{28}$$

where $p(j)$ and $P(j)$ are the priority and sampling probability of sample j, respectively. Considering that sorting these samples wastes extensive computing resources, we use SumTree structure to store these samples. The specific structure of PSDDQN is shown in Figure 5.

When training PSDDQN model, we firstly select M samples from SumTree structure and compute their loss functions and gradients,

$$\text{Loss}(\theta) = \sum_{j=1}^{M} w_j \left(y_j^{\text{DDQN}} - Q(s_j, a_j; \theta) \right)^2, \tag{29}$$

$$\nabla_\theta \text{Loss}(\theta) = w_j \left(y_j^{\text{DDQN}} - Q(s_j, a_j; \theta) \right) \nabla Q(s_j, a_j; \theta), \tag{30}$$

$$w_j = \frac{(N * P(j))^{-\beta}}{\max_i(w_i)} = \frac{(N * P(j))^{-\beta}}{\max_i (N * P(i))^{-\beta}} = \frac{(P(j))^{-\beta}}{\max_i (P(i))^{-\beta}}, \tag{31}$$

where w_j represents the weight of priority of sample j for calculating the loss value. And variable β represents the hyperparameter, which is determined by empirical value.

Then, the weights of DDQN are updated according to the calculated gradients. Moreover, we also need to update the priorities of samples in the SumTree structure. For the sample, the greater the TD-error, the better the effect of training the DDQN model. Here, we use TD-error value to represent the priority of the sample. We get the TD-error value of the sample and update its priority by

$$\text{TD} - \text{error}_j = y_j - Q(s_j, a_j; \theta), \tag{32}$$

$$p(j) = \text{TD-error}_j. \tag{33}$$

The ground control centre has rich computing resources and can directly obtain the real-time state information of BSs and satellites. In the process of communication, BSs and satellites send their real-time state information to the ground control centre for updating their state information. The agent is able to obtain the latest state information of STINs from the ground control centre in real time. In our research, we deploy PSDDQN algorithm in the ground control centre. The specific details of PSDDQN is described in the following:

5. Experimental Results and Analysis

We firstly introduce the experimental environment. Then, we validate PSDDQN algorithm in the simulation data and analysis the experimental results.

5.1. Experimental Environment. The communication system of STINs in this paper is mainly composed of 120 LEO satellites and 6 BSs with satellite gateways. The index of virtual BS is 0. And we use Walker delta model to construct satellite constellation by satellite tool kit (STK). The satellite constellation has 12 orbits, and each orbit has 10 satellites. The inclination and height of each orbit are 45 degrees and 550 km, respectively. Moreover, every satellite is equipped with same antenna. The specific parameters of simulation are described by Table 1.

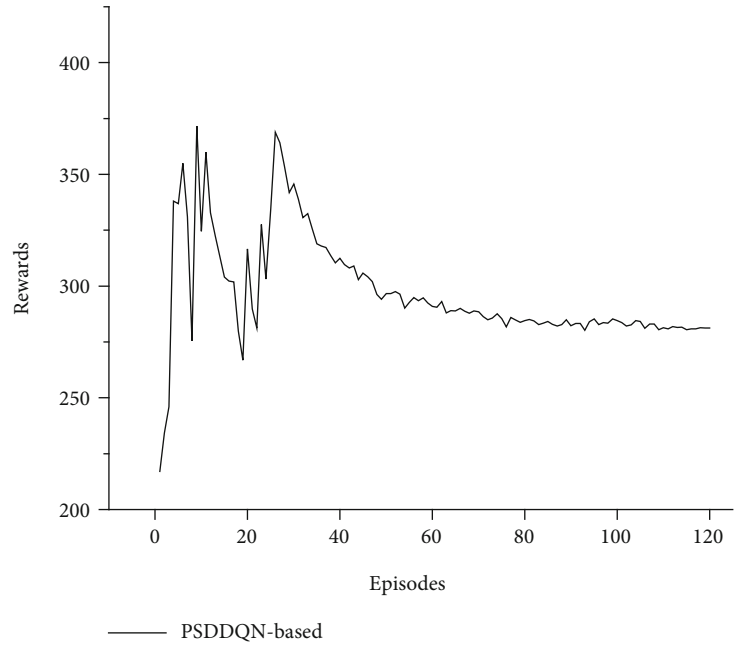

Figure 6: Rewards of PSDDQN with different episodes.

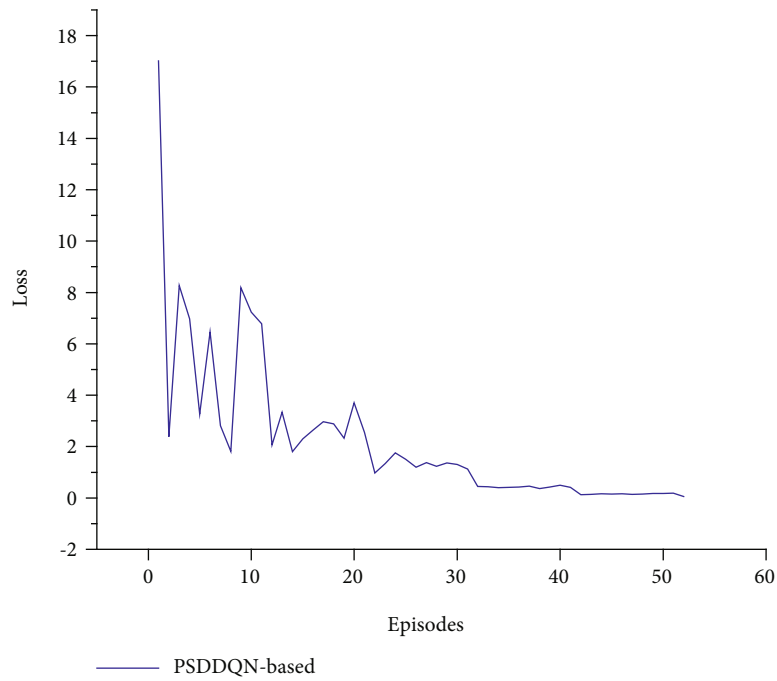

Figure 7: Loss value of PSDDQN with different episodes.

Considering that the estimated CSI is inaccurate and there is feedback propagation delay during the information transmission, the CSI information obtained by BSs is imperfect. Therefore, it is inaccurate to use the obtained CSI information to calculate the data transmission rate between satellites and BSs. To solve this problem, in practice, we use the estimated CSI value and the error value which follows circular symmetric complex Gaussian distribution to approximate the CSI value.

We conduct experiments on a computer with Win 10 OS, 16G RAM, and 3.2 GHz CPU. The programming language used is Python, and we select TensorFlow framework to construct PSDDQN model. Additionally, each BS contains 4 channels in this paper. Therefore, each BS can connect to 4 satellites simultaneously. Considering that the number of BSs is 6, the fully connected layer of PSDDQN has 24 neurons. Moreover, the two independent network structures of PSDDQN are same. And each neural network

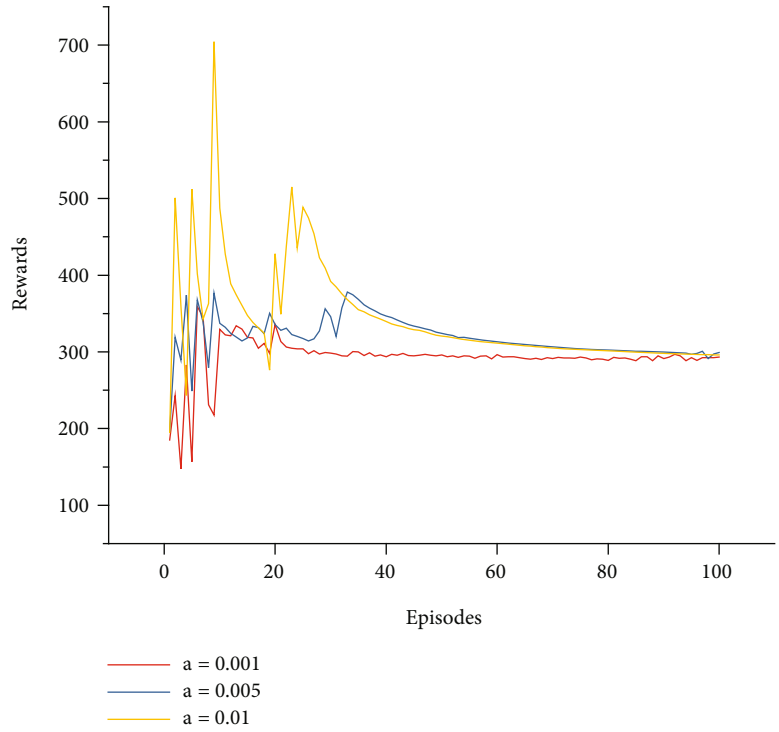

Figure 8: Rewards of PSDDQN with different learning rates.

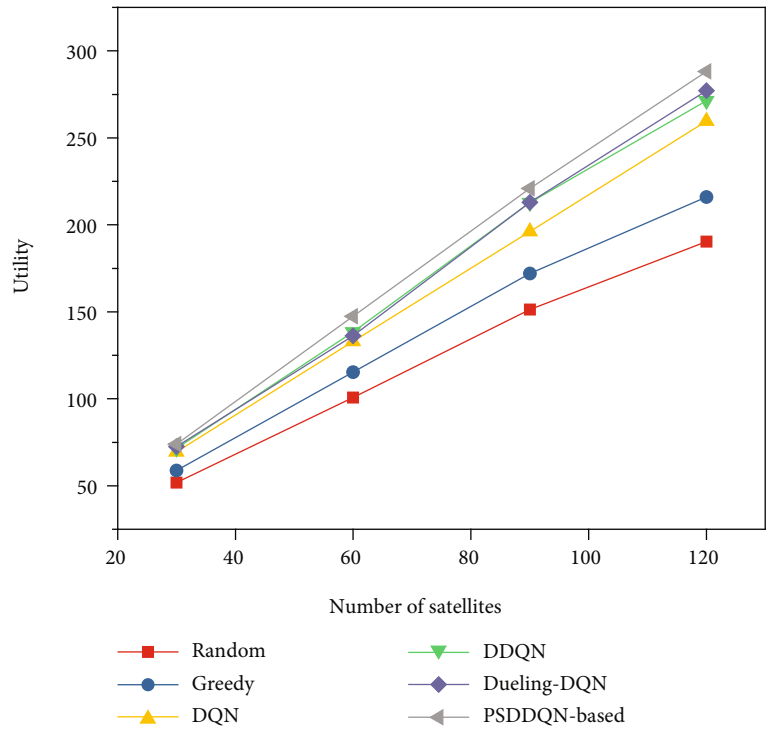

Figure 9: System utility with different number of satellites.

structure consists of the input layer, two hidden layers, and the output layer. In addition, the number of neurons in the input layer, hidden layer, and output layer is 156, 64 and 32, and 24, respectively. The specific parameters of PSDDQN are given by Table 2.

Considering that most operations in PSDDQN are dot product operations, we use FLOPS to represent its computation time. Because we use the full connection way to build the neural network, the FLOPs in the first layer, the second layer, and the third layer are $(2*156\text{-}1) * 64$, $(2 * 64\text{-}1) *$

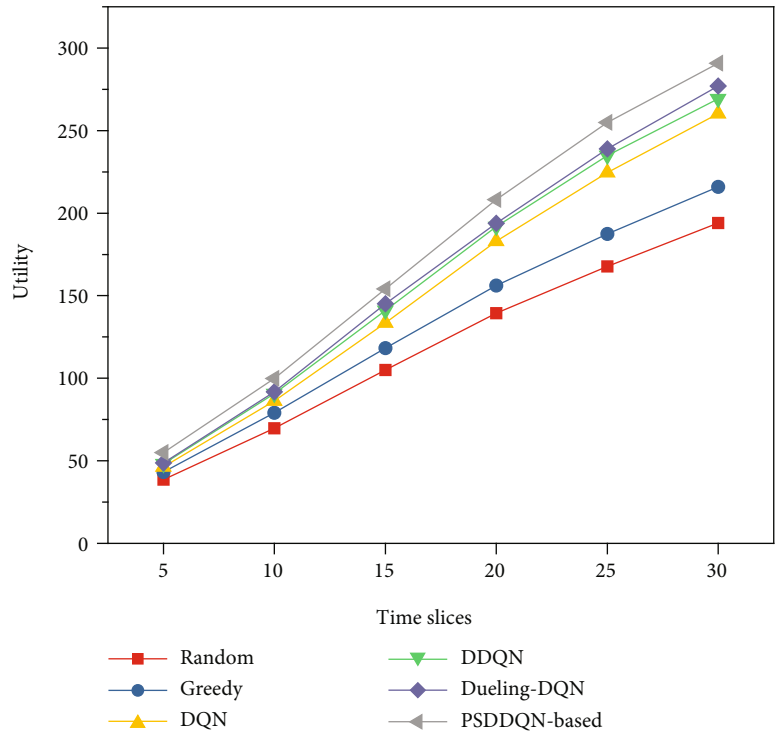

Figure 10: System utility with different time slices.

32, and (2 ∗ 32-1) ∗ 24, respectively. In addition, the activation function used in the first and second layers is ReLU; the required FLOPS of ReLU in these two layers are 64 and 32. The activation function used in the third layer is Softmax; the required FLOPS of Softmax in the third layer is 24 ∗ 4. Therefore, the total required FLOPs to execute one sample is 25672. Considering that the batch size we used is 32, the total FLOPS of PSDDQN is 25672 ∗ 32. For the parameters of PSDDQN, the parameters required in the first layer, the second layer, and the third layer are 156 ∗ 64+64, 64 ∗ 32+32, and 32 ∗ 24+24. Therefore, the total number of parameters in PSDDQN is 12920.

The weight of utility ω and storage cost ξ in Equation (20) is set to 0.7 and 0.3, respectively. The utility of unit data offloaded by the satellite is 1. Moreover, the duration of each time slice is 3.5 minutes, and there are 30 time slices in total.

5.2. Experimental Result Analysis. We firstly verify the convergence of PSDDQN. Then, we compare PSDDQN algorithm with random allocation algorithm, greedy allocation algorithm, DQN [37], DDQN [43], and Dueling-DQN [44], respectively. Finally, we show the experimental results and give the discussion.

Random allocation algorithm: for the satellite, one BS is randomly selected from the BSs within its coverage. Then, the satellite is associated with the selected BS, and a channel is randomly selected from the selected BS and assigned to the satellite.

Greedy allocation algorithm: for the satellite, the BS with the largest amount of offloaded data is selected from the BSs within its coverage and associated with the satellite. If the BS has no free channels, the suboptimal BS is selected.

5.2.1. Evaluation of Reinforcement Learning Model. Figure 6 shows the rewards of PSDDQN algorithm with different episodes. We note from Figure 6 that in the beginning, the rewards obtained by PSDDQN algorithm fluctuate sharply. The reason is that the parameters of PSDDQN algorithm are initialized randomly. As the number of episodes increases, the fluctuation of rewards obtained decreases gradually and finally tends to be stable. It is obvious from Figure 6 that PSDDQN algorithm converges at about 50 episodes.

Figure 7 plots the loss value of PSDDQN algorithm with different episodes. We observe that the loss value of PSDDQN is large in the beginning. With the increasing of episodes, the loss value of PSDDQN decreases gradually. Finally, the loss value tends to be stable. In addition, from Figure 7, we know that PSDDQN algorithm learns useful knowledge from state transition information.

From Figures 6 and 7, we can see that PSDDQN algorithm converges at about 50 episodes. The convergence speed of PSDDQN in STINs is relatively fast. In the training process, the satellite selects one BS from the BSs within its coverage as an action. Because the communication time between satellites and BSs is very short, most of state transition information is from virtual BS to virtual BS. Therefore, when state transition is from virtual BS to virtual BS, the satellite can only select virtual BS as an effective action. In PSDDQN, we directly set that the satellite at this time can only connect to virtual BS, which effectively reduces the convergence time of PSDDQN. This explains why PSDDQN converges fast.

When training PSDDQN algorithm, we use gradient descent method to update its parameters. If the learning rate

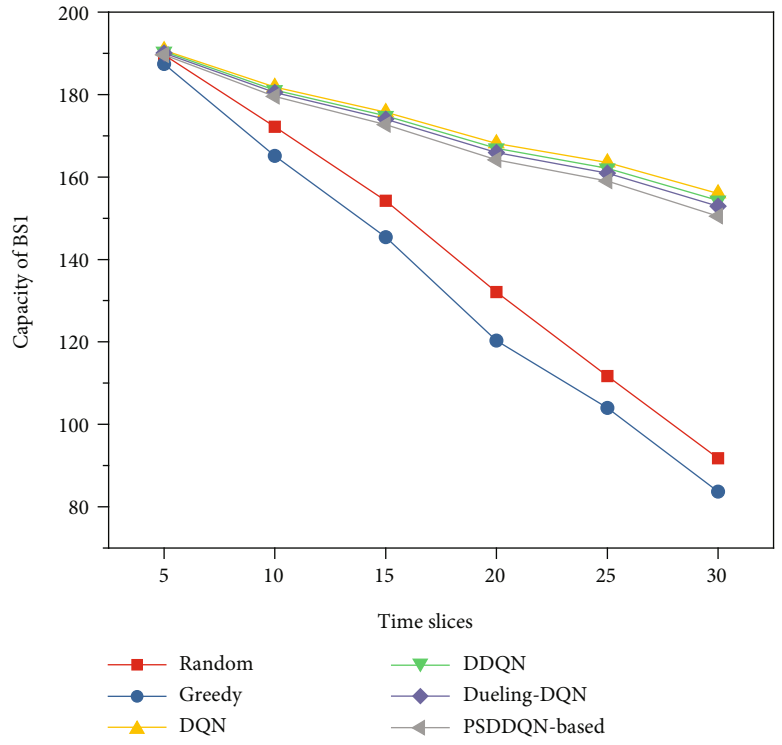

Figure 11: Capacity change of BS1.

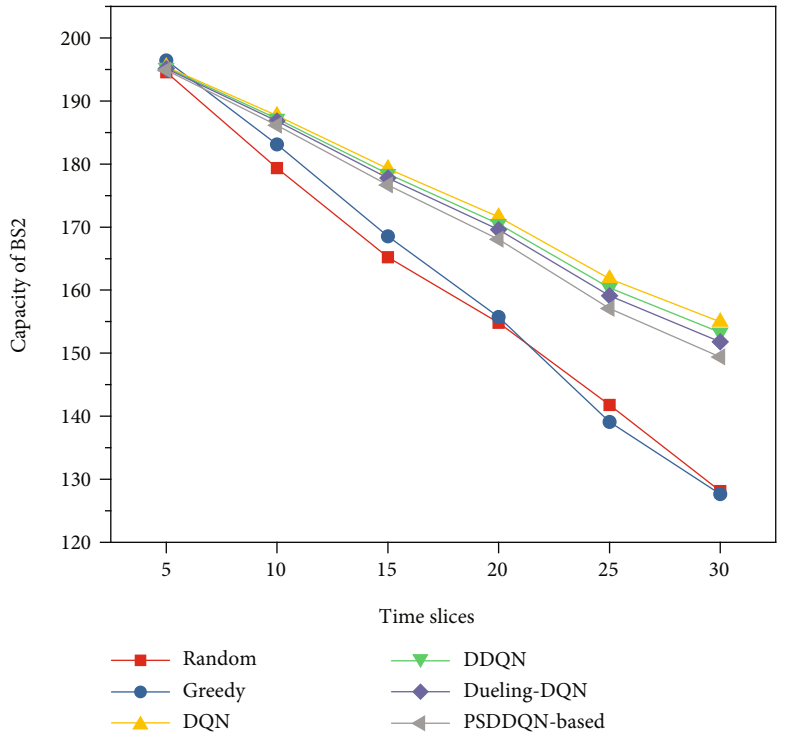

Figure 12: Capacity change of BS2.

is large, the step of parameter update is large. The convergence speed of PSDDQN is fast, but its performance fluctuates sharply in the convergence process. In addition, PSDDQN often cannot find the optimal solution. If the learning rate is small, the step of parameter update is small. And the convergence speed of PSDDQN is slow. Therefore, setting an appropriate learning rate in the experiment is very important for the convergence of PSDDQN.

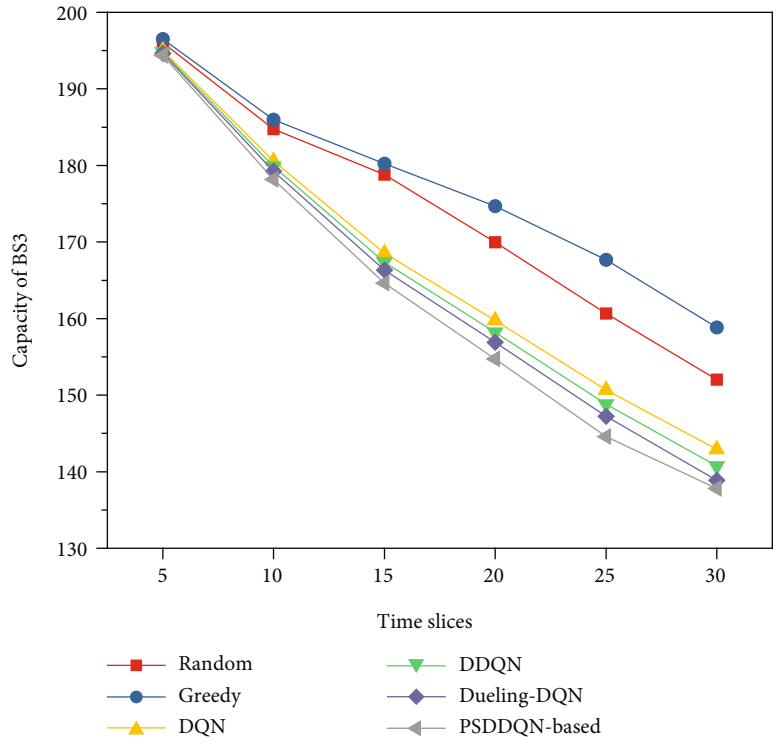

Figure 13: Capacity change of BS3.

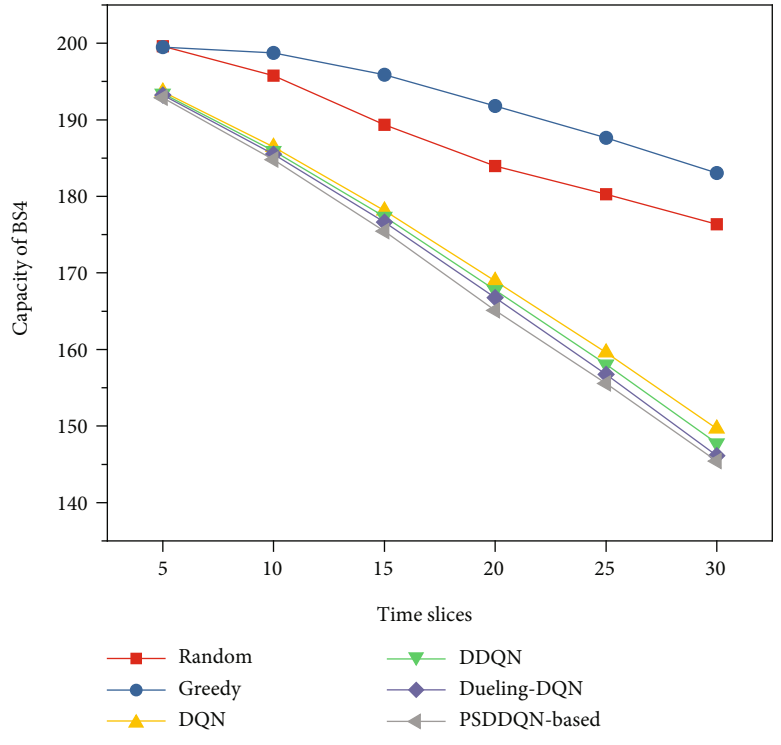

Figure 14: Capacity change of BS4.

Figure 8 depicts the rewards obtained by PSDDQN with different learning rates. With the increasing of episodes, PSDDQN algorithm with different learning rates can converge to a steady state. In addition, we also find that when $\alpha = 0.01$, the fluctuation of rewards obtained by PSDDQN is relatively large. However, if the value of α is too large, the agent often obtains the local optimal result rather than the optimal solution. When $\alpha = 0.001$, the convergence

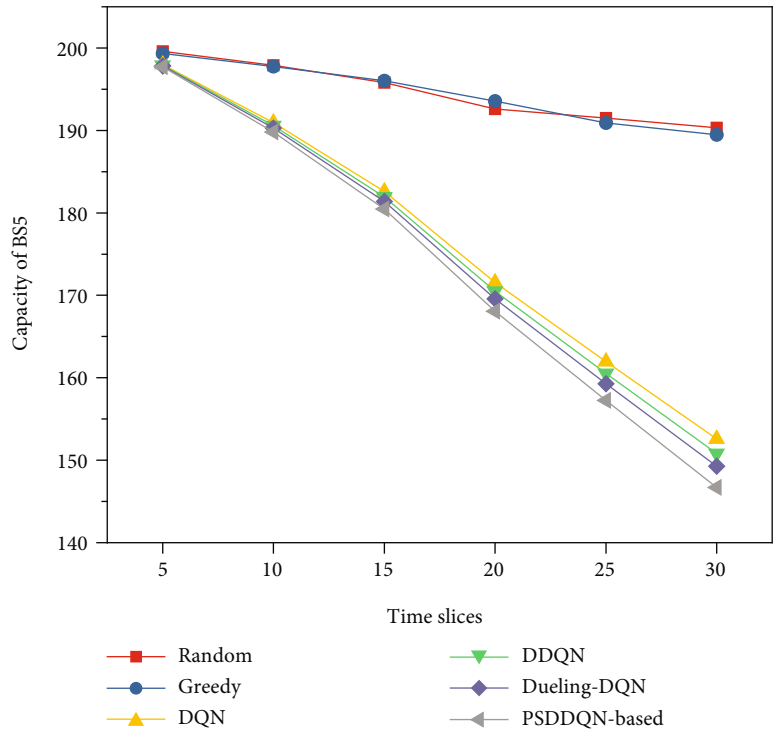

Figure 15: Capacity change of BS5.

speed of PSDDQN is slower than that of $\alpha = 0.005$. Therefore, in this paper, we set α to be 0.005 to train PSDDQN.

5.2.2. Evaluation of System Utility and Capacity Performance of BSs. We compare the system utility obtained by different algorithms with different number satellites in Figure 9. The increases of satellites number lead to the gradually increases of utility for all algorithms. This is because that the more satellites exist, the more connections between satellites and BSs in a time slice. Therefore, the amount of data offloaded from satellites to BSs increases, which increases the system utility. Moreover, we also find that the utility obtained by random allocation algorithm is the least, and the utility obtained by PSDDQN is the largest. For random allocation algorithm, it randomly assigns BSs and channels to satellites without considering the channel state and their communication time. As a result, the utility obtained by random allocation algorithm is the least. For greedy allocation algorithm, satellites prefer to associate with BSs that can offload more data. Therefore, greedy allocation algorithm achieves better performance than random allocation algorithm.

Furthermore, the performance of RL-based methods (e.g., DQN, DDQN, and Dueling-DQN) is similar. The reasons are as follows. First, RL-based algorithms consider the load of BSs in joint satellite association and channel allocation, so BSs can store offloaded data at a lower cost. Second, when offloading data, they not only consider the utility generated by satellites in the current time slice but also consider the utility generated by satellites in the subsequent time slices. Therefore, RL-based algorithms obtain more utility than random allocation algorithm and greedy allocation algorithm. In addition, DQN, DDQN, and Dueling-DQN algorithms use uniform sampling method to train their models. And PSDDQN adopts priority sampling method to train its model, which enhance the learning efficiency of uniform sampling. This explains why PSDDQN obtains the best performance.

In Figure 10, we compare the system utility obtained by all algorithms. We from Figure 10 note that with the increasing of time slices, the utility of all algorithms shows a gradual upward trend. And PSDDQN algorithm has the greatest utility, and random allocation algorithm has the least utility. Moreover, we also find that in the beginning, each algorithm obtains almost the same utility. However, with the increasing of time slices, the utility obtained by different algorithms is different. The main reason is that in satellite scheduling and resource allocation, RL-based algorithms pay more attention to the expected cumulative utility rather than the immediate utility. Therefore, they get better performance. In contrast, random allocation algorithm and greedy allocation algorithm only consider the utility generated by satellites in the current time slice and ignore the utility generated by satellites in the subsequent time slices.

Moreover, we can also see that for RL-based algorithms, the performance of PSDDQN is the best and that of DQN is the worst. The reason is that DQN is trained by a neural network structure, leading to the over estimation of the Q value. In addition, the training process of DQN is unstable, while DDQN adopts two independent network structures to select the action and calculate the Q value, which alleviate the over estimation problem. This explains why DDQN performs better than DQN. When calculating the Q value, Dueling-DQN pays more attention to the action with large advantage value. Thus, it achieves better results than DDQN. For

PSDDQN, it not only adopts two independent network structures to calculate the Q value but also uses priority sampling method to sample samples with larger TD-error value for training model. Therefore, compared with DQN, DDQN, and Dueling-DQN algorithms, PSDDQN algorithm gets better performance.

Here, we show the capacity change of BSs. In STINs, BS0 is virtual BS, and its capacity does not change at any time. So we do not show the change of its capacity.

Figures 11–15 show the capacity of BSs of different algorithms with different time slices. When the time slices increase, the remaining capacity of BSs decreases gradually. The reason is that with the increasing of time slices, more satellites can offload data to BSs. In addition, we find that the remaining capacity of different BSs is different. Particularly, we observe from Figures 11 and 12 that random allocation algorithm and greedy allocation algorithm offload most of data to BS1 and BS2, leading to the significant reduction in the capacity of BS1 and BS2. The remaining capacities of BS1 and BS2 are 90 and 128, respectively. From Figures 13–15, we also note that random allocation algorithm and greedy allocation algorithm offload less data to BS3, BS4, and BS5. The remaining capacities of BS3, BS4, and BS5 are 158, 185, and 195, respectively. Therefore, we know that random allocation algorithm and greedy allocation algorithm cause the capacities of BSs to be used unevenly.

Furthermore, we find that when RL-based algorithms are adopted, the remaining capacities of BS1, BS2, BS3, BS4, and BS5 are 160, 155, 140, 145, and 150, respectively. These algorithms basically ensure that the capacity of each BS is used evenly. When satellite association and channel allocation are performed, greedy allocation algorithm only considers the amount of offload data and ignores the load of BSs, which causes the capacity of some BSs to be overused and the storage cost of some BSs becomes larger. In contrast, RL-based algorithms consider not only the amount of offloaded data but also the capacity of BSs. Therefore, the results obtained by RL-based algorithms are better than those of greedy allocation algorithm and random allocation algorithm. In addition, considering that PSDDQN offloads more data to BSs, it consumes more capacity compared with other RL algorithms.

6. Conclusion

In our research, we investigated the joint satellite scheduling and resource allocation in STINs. We added a virtual BS to solve the problem that satellites cannot connect to BSs and reconstructed the communication model between satellites and BSs. Then, we formulated the joint satellite association and channel allocation as a joint optimization problem about utility and proposed PSDDQN algorithm to obtain the optimal strategy. When assigning appropriate BSs and channels to satellites, PSDDQN algorithm also considers the load balance of BSs. The simulation results demonstrate that PSDDQN obtains the maximum utility generated by offloaded data and achieves the load balance of BSs.

For some applications, data generated by satellites need to be transmitted to the terrestrial networks in real time. However, most satellites cannot communicate with BSs during the time of a snapshot. Therefore, it is a challenge to transmit the real-time data generated by satellites to the terrestrial networks. Considering that the satellites connected to BSs can be used as gateways, in the future work, we mainly study the traffic scheduling of satellites and try to design a traffic scheduling scheme for reasonably and dynamically transmitting the data of satellites to these gateways. Finally, these gateways are used to transmit these data to the terrestrial networks.

Data Availability

The simulation data used to support the findings of this study are available from the corresponding author upon request.

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 National Science Foundation of China (No. 61772385).

References

[1] S. Cioni, R. D. Gaudenzi, O. D. R. Herrero, and N. Girault, "On the satellite role in the era of 5G massive machine type communications," *IEEE Network*, vol. 32, no. 5, pp. 54–61, 2018.

[2] M. Shaat, E. Lagunas, A. I. Perez-Neira, and S. Chatzinotas, "Integrated terrestrial-satellite wireless backhauling: resource management and benefits for 5G," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 39–47, 2018.

[3] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47–62, 2019.

[4] L. Kuang, X. Chen, C. Jiang, H. Zhang, and S. Wu, "Radio resource management in future terrestrial-satellite communication networks," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 81–87, 2017.

[5] W. Wang, S. Zhao, Y. Zheng, and Y. Li, "Resource allocation method of cognitive satellite terrestrial networks under non-ideal spectrum sensing," *IEEE Access*, vol. 7, pp. 7957–7964, 2019.

[6] W. Wang, J. Wei, S. Zhao, Y. Li, and Y. Zheng, "Energy efficiency resource allocation based on spectrum-power tradeoff in distributed satellite cluster network," *Wireless Networks*, vol. 26, no. 6, pp. 4389–4402, 2020.

[7] C. C. Lin, N. W. Su, D. J. Deng, and I. Tsai, "Resource allocation of simultaneous wireless information and power transmission of multi-beam solar power satellites in space-terrestrial integrated networks for 6G wireless systems," *Wireless Networks*, vol. 26, no. 6, pp. 4095–4107, 2020.

[8] Y. Hu, M. Chen, and W. Saad, "Joint access and backhaul resource management in satellite-drone networks: a

competitive market approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3908–3923, 2020.

[9] Y. Chen, J. Li, W. Chen, Z. Lin, and B. Vucetic, "Joint user association and resource allocation in the downlink of heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5701–5706, 2016.

[10] A. Khalili, S. Akhlaghi, H. Tabassum, and D. W. K. Ng, "Joint user association and resource allocation in the uplink of heterogeneous networks," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 804–808, 2020.

[11] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[12] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[13] J. Qiu, J. Lyu, and L. Fu, "Placement optimization of aerial base stations with deep reinforcement learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC),*, pp. 1–6, Dublin, Ireland, 2020.

[14] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 263–278, 2020.

[15] C. Jiang and X. Zhu, "Reinforcement learning based capacity management in multi-layer satellite networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4685–4699, 2020.

[16] H.-X. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 131–141, 2021.

[17] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3507–3523, 2021.

[18] M. Feng, S. Mao, and T. Jiang, "Joint frame design, resource allocation and user association for massive MIMO heterogeneous networks with wireless backhaul," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1937–1950, 2018.

[19] R. Liu, Q. Chen, G. Yu, and G. Y. Li, "Joint user association and resource allocation for multi-band millimeter-wave heterogeneous networks," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8502–8516, 2019.

[20] J. S. Liu, C. H. R. Lin, and Y. C. Hu, "Joint resource allocation, user association, and power control for 5G LTE-based heterogeneous networks," *IEEE Access*, vol. 8, pp. 122654–122672, 2020.

[21] Z. Cheng, N. Chen, B. Liu et al., "Joint user association and resource allocation in HetNets based on user mobility prediction," *Computer Networks*, vol. 177, p. 107312, 2020.

[22] M. Zalghout, A. Khalil, M. Crussiére, S. Abdul-Nabi, and J.-F. Hélard, "Context-aware and priority-based user association and resource allocation in heterogeneous wireless networks," *Computer Networks*, vol. 149, pp. 76–92, 2019.

[23] X. Zhu, C. Jiang, L. Kuang, N. Ge, and J. Lu, "Energy efficient resource allocation in cloud based integrated terrestrial-satellite networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, 2018.

[24] P. Zuo, T. Peng, W. Linghu, and W. Wang, "Resource allocation for cognitive satellite communications downlink," *IEEE Access*, vol. 6, pp. 75192–75205, 2018.

[25] T. Mai, H. Yao, F. Li, X. Xu, Y. Jing, and Z. Ji, "Computing resource allocation in LEO satellites system: a Stackelberg game approach," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 919–924, Tangier, Morocco, 2019.

[26] B. Deng, C. Jiang, and S. Guo, "Energy minimization of resource allocation in cloud-based satellite communication networks," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2353–2356, 2019.

[27] S. M. Shahid, Y. T. Seyoum, S. H. Won, and S. Kwon, "Load balancing for 5G integrated satellite-terrestrial networks," *IEEE Access*, vol. 8, pp. 132144–132156, 2020.

[28] Z. Ji, S. Wu, C. Jiang, D. Hu, and W. Wang, "Energy-efficient data offloading for multi-cell satellite-terrestrial networks," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2265–2269, 2020.

[29] R. Deng, B. Di, H. Zhang, and L. Song, "Ultra-dense LEO satellite constellation design for global coverage in terrestrial-satellite networks," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, 2020.

[30] L. He and K. He, "Efficient memory-bounded optimal detection for GSM-MIMO systems," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–12, 2022.

[31] L. He and K. He, "Towards optimally efficient search with deep learning for large-scale MIMO systems," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–14, 2022.

[32] X. Lai, "Outdated access point selection for mobile edge computing with cochannel interference," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–10, 2022.

[33] S. Tang and L. Chen, "Computational intelligence and deep learning for next-generation edge-enabled industrial IoT," *IEEE Transactions on Network Science and Engineering*, vol. -PP, no. 99, pp. 1–15, 2022.

[34] L. Chen, R. Zhao, K. He, Z. Zhao, and L. Fan, "Intelligent ubiquitous computing for future UAV-enabled MEC network systems," *Cluster Computing*, vol. 2021, no. 25, pp. 1–10, 2021.

[35] L. Chen, "Physical-layer security on mobile edge computing for emerging cyber physical systems," *Computer Communications*, vol. PP, no. 99, pp. 1–9, 2022.

[36] X. Liu, Z. Qin, and Y. Gao, "Resource allocation for edge computing in IoT networks via reinforcement learning," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, 2019.

[37] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11158–11168, 2019.

[38] F. Tang, Y. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G HetNet," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 12, pp. 2773–2782, 2020.

[39] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6380–6391, 2020.

[40] W. Hou, H. Wen, H. Song, W. Lei, and W. Zhang, "Multiagent deep reinforcement learning for task offloading and resource

allocation in Cybertwin-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16256–16268, 2021.

[41] P. Luong, F. Gagnon, L. N. Tran, and F. Labeau, "Deep reinforcement learning-based resource allocation in cooperative UAV-assisted wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7610–7625, 2021.

[42] R. Fdhila, T. M. Hamdani, and A. M. Alimi, "A multi objective particles swarm optimization algorithm for solving the routing pico-satellites problem," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1402–1407, Seoul, Korea (South), 2012.

[43] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4005–4018, 2019.

[44] Z. Liu, X. Chen, Y. Chen, and Z. Li, "Deep reinforcement learning based dynamic resource allocation in 5G ultra-dense networks," in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 168–174, Tianjin, China, 2019.