2022 International Wireless Communications and Mobile Computing (IWCMC) | 978-1-6654-6749-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWCMC55113.2022.9824467

Double Deep Q-learning Based Satellite Spectrum/Code Resource Scheduling with Multi-constraint

[1]Zixian Chen, [1]Xiang Chen*, and [2]Chong-Yung Chi
[1]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
[2]Department of Electrical Engineering, Institute of Communications Engineering, NTHU, Hsinchu 30013, Taiwan
Email: chenzx29@mail2.sysu.edu.cn, chenxiang@mail.sysu.edu.cn, cychi@ee.nthu.edu.tw

***Abstract*—For multi-user satellite Internet of Things (IoT) systems operating at lower signal-to-noise ratio, spread spectrum techniques are usually used to combat narrowband interference. In addition, the communication performance in the spread spectrum system depends on the anti-jamming ability of the spreading codes (SCs). Therefore, how to design the SCs scheduling strategies under users' requirements and resource constraints has become a crucial problem for satellite IoT systems. In this paper, communication rewards and scheduling delays are introduced as gauges to measure the scheduling performance of the satellite gateway station control center (SGSCC). Specifically, SGSCC must efficiently and effectively allocate limited available SCs over terminal gateways under request at each transmission time slot. The SCs scheduling problem is formulated as a Markov Decision Process (MDP) along with the observed environments composed of resource status and user request status. Then a deep reinforcement learning scheduling algorithm is devised by embedding the idea of Long Short-Term Memory (LSTM) in the standard Double Deep Q-learning (DDQN). Simulation results show that the proposed algorithm can achieve much better performance than traditional algorithms in terms of communication rewards and scheduling delays. Finally, we draw some conclusions.**

***Index Terms*—Quality of Service, Multi-constraint Scheduling, Satellite IoT, Spread Spectrum, Double Deep Q-learning.**

I. Introduction

In recent years, the communication systems of satellite IoT have gradually become research hotspots, and the demands for satellite services are also increasing. With the prosperity of satellite services, an effective strategy is needed to allocate limited satellite communication resources over a large number of satellite IoT terminals [1].

Traditionally, communication resources in satellite systems are usually allocated in the time domain, the frequency domain, or the space domain. For example, satellite remote sensing needs to rank the importance of different communication tasks in the time dimension [2]. In the frequency dimension, the quality of service for users is affected by the allocation of available bandwidth resources [3]. In the space dimension, the power allocation beamforming scheme is designed to reduce the interference between beams [4]. For satellite IoT, Direct Sequence-Code Division Multiple Access (DS-CDMA) systems allow users to share bandwidth and operate at lower signal-to-noise ratio (SNR) [5].

On the other hand, due to the dense distribution of Geosynchronous Earth Orbit (GEO) satellite orbits, the satellite systems are inevitably interfered by adjacent satellites [6]. DS-CDMA systems cannot work well at lower signal-to-interference ratio (SIR) in the presence of uncertain non-cooperative external interferences. Fortunately, the Eigen-based SCs design framework for CDMA (ECDMA) satellite systems is proposed in [7]. Based on spectrum shaping, the ECDMA system calculates the SC of different SIR combined with the feature analysis of external interferences. The terminal gateways (TGs) use different SCs which can resist the interference of adjacent satellites with different SIR levels. Because the TGs lack the information about the availability status of the SCs in the SGSCC and the latter cannot predict when the former will require the SCs, the former can only actively send the request to the latter through random access procedure [8]. Therefore, the SC allocation methods reported in the existing literatures are no longer applicable in the scheduling of Eigen-based SCs with different SIR levels.

In the satellite IoT system, the TGs receive and process the data of the terminal sensors [9]. Usually, TGs suffer from different degrees of interference for the collection of information from various sensors. As a result, TGs have different requirements for Quality of Service (QoS) [10], which in our work can be expressed as different requirements for SIR values and scheduling delays. Effective SC scheduling must not only meet the TGs' requirements for SC levels (defined by SIR values), but also ensure that the SC assignments of TGs are through within the respective expected scheduling delays.

To solve this problem, we regard the scheduling problem as a multi-dimensional knapsack problem [11], where each dimension means that SCs with a certain level are available to be assigned to the TGs. At the same time, the scheduling task can be formulated as a Markov Decision Process (MDP). A deep reinforcement learning scheduling algorithm is designed that embeds the information of TGs and Long Short-Term Memory (LSTM) in the standard Double Deep Q-learning (DDQN). Simulation results demonstrate the effectiveness of the proposed scheduling algorithm to solve the SC allocation problem under multiple constraints.

The rest of the paper is organized as following: In Section II, we construct the SCs scheduling model. The proposed

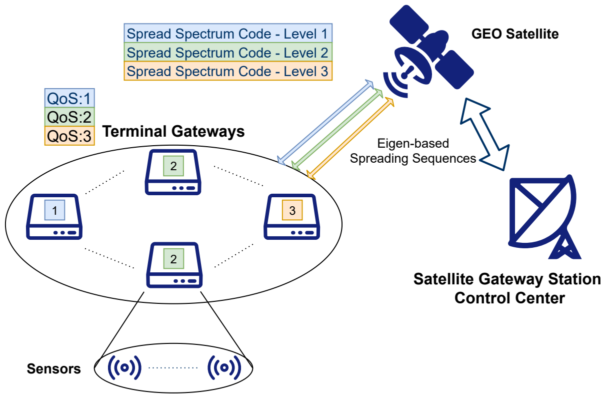
Spread Spectrum Code - Level 1
Spread Spectrum Code - Level 2
Spread Spectrum Code - Level 3
GEO Satellite
QoS:1
QoS:2
QoS:3
Terminal Gateways
Eigen-based
Spreading Sequences
1
2
3
2
Satellite Gateway Station
Control Center
Sensors

Fig. 1. System model

scheduling algorithm based on DDQN is presented in Section III. In Section IV, some simulation results and analysis are provided to demonstrate the efficacy of the proposed scheduling algorithm and comparison with some existing benchmark algorithms. Fianlly, we will make a summary and future outlook in Section V.

II. System model and problem formulation

A. System model

We consider a GEO Eigen-based SCs satellite system as depicted in Fig. 1. Within the coverage of a GEO satellite, some TGs using Eigen-based SCs are interfered by adjacent satellites. The TGs denoted by $s \in \mathcal{S} = \{0, 1, 2, ..., N-1\}$ receive and process the information data from sensors (such as gas monitoring, water monitoring, soil monitoring, etc.).

The system makes a scheduling decision at the beginning of each discrete time slot $t \in \{0, 1, 2, ..., T-1\}$. At the beginning of each time slot, the random access of one of the N TGs will reach the SGSCC to apply for the SCs used in the next transmission time slot. The SGSCC needs to process the random access of the TG one by one in sequence (i.e., assign an SC or reject assignment) and each scheduling decision needs to be completed within one random access time slot.

Assume that the SGSCC can handle K (bounded by N) random access messages (the SC requests) for each transmission time slot, subdivided into K random access time slots. The total number of scheduling discrete time slots is defined as $T = K\bar{T}$, where $\bar{T}$ is the number of transmission time slots for one episode. For SGSCC, it is a challenge to assign SCs to TGs through an optimal strategy when the number of assignable SCs is limited. It is not only necessary to allocate SCs according to the importance of the current TG, but also to reserve resources for more important TGs that may appear in the coming time slots.

B. Eigen-based spreading code model

We assume that there are M Eigen-based SCs $\mathbf{C}(t) = \{c_0(t), c_1(t), ..., c_{M-1}(t)\}$ for each transmission time slot can be allocated by SGSCC. Through the operation of the interference signal matrix in [7], we can obtain Eigen-based SCs with different SIR values. Without loss of generality, with 3dB as the division interval, all the M SCs are divided into three levels $c_m(t) \in \{1, 2, 3\}$ according to their respective SIR values. The higher the level of SC (denoting higher SIR value) used by the TG, the less affected by the interference of adjacent satellites.

The availability status of the M SCs is defined as $\mathbf{D}(t) = \{d_0(t), d_1(t), ..., d_{M-1}(t)\}$ and $d_m(t) \in \{0, 1\}$. Status 0 indicates that the corresponding SC is available in the next transmission time slot. If the SGSCC assigns the mth SC c_m to a TG, then the corresponding availability status $d_m(t) = 1$, implying that $c_m(t)$ has been occupied in the next transmission time slot and cannot be allocated to other TGs. Surely, to reserve the SC of the next transmission time slot for the subsequent TGs, the SGSCC may refuse the SC request in the current random access time slot, thus without updating the availability status. If there are still random accesses pending for the next transmission time slot, the SGSCC can then process the next random access of the TGs.

The SCs allocation at each transmission time slot starts with the availability status of all SCs reset to '0'. In addition, the interference of adjacent satellites may change after several transmission time slots, so it is necessary to recalculate the Eigen-based SCs. Then the SIR values (along with levels) and the number of the assignable SCs will be updated, indicating that the resources allocated by the SGSCC have been changed.

C. Quality of Service (QoS) model

The QoS level for the TG depends on the SIR value of the required SC when interfered by adjacent satellites. Therefore, the QoS levels of TGs are defined as $\mathbf{Q}(t) = \{q_0(t), q_1(t), ..., q_{N-1}(t)\}$ where $q_n(t) \in \{1, 2, 3\}$ for all n. With the same transmit power, each TG can use the SC with a level higher than or equal to its QoS level, meanwhile receiving a communication reward based on its QoS level.

The adaptive transmission capabilities of TGs are defined as $\mathbf{E}(t) = \{e_0(t), e_1(t), ..., e_{N-1}(t)\}$ where $e_n(t) \in \{0, 1\}$ for all n. If $e_n(t) = 1$, it means no SCs with suitable level available, however, the corresponding TG can use SC with a level one lower than its QoS level, but the communication reward will be reduced by a factor $\beta \in [0, 1]$, depending on the cost of adaptive adjustment.

Provided that more important TGs require SCs with higher SIR values (higher level) to use, and the transmitted services get a higher communication reward, depending on the corresponding QoS level. The communication reward of assigning the lth SC to the kth TG can be expressed as:

$$t\_r_k(t) = \begin{cases} q_k(t), & \text{if } \mathbb{A} \text{ is true} \\ \beta q_k(t), & \text{if } \mathbb{B} \text{ is true} \\ 0, & \text{else} \end{cases} \tag{1}$$

$$\mathbb{A} : c_l(t) \geq q_k(t) \text{ and } d_l(t) = 0$$
$$\mathbb{B} : c_l(t) = q_k(t) - 1,\ e_k(t) = 1 \text{ and } d_l(t) = 0$$

The compatibility setting between the SCs and TGs is generally in accord with the actual satellite IoT system, which can avoid the problem of no available SCs for higher-level TGs and some idle lower-level SCs.

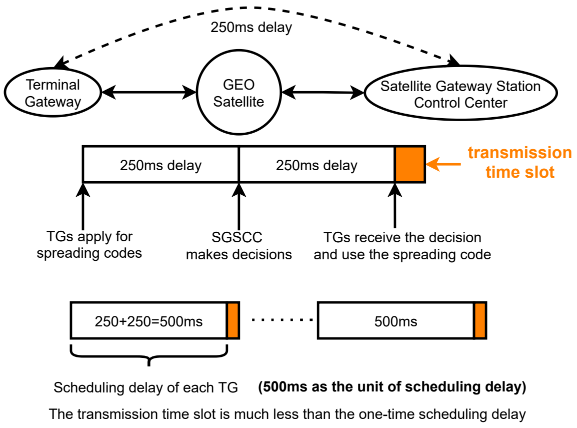
250ms delay
Terminal Gateway
GEO Satellite
Satellite Gateway Station Control Center
250ms delay
250ms delay
transmission time slot
TGs apply for spreading codes
SGSCC makes decisions
TGs receive the decision and use the spreading code
250+250=500ms
500ms
Scheduling delay of each TG
(500ms as the unit of scheduling delay)
The transmission time slot is much less than the one-time scheduling delay

Fig. 2. Scheduling delay model

D. Scheduling delay model and problem formulation

The scheduling delay for each TG is defined as the time elapsed since the latest SC assignment time. Since the SC request information needs to be forwarded by the GEO satellite, this will bring about a communication delay of about 250ms (approximately 250ms-270ms for a round trip) [12], as shown in Fig. 2.

According to the communication delay of 250ms, there will be a scheduling delay of 500ms when the SGSCC allocates a SC. We assume that a transmission time slot is much less than a 500ms scheduling delay [13]. Therefore the unit of scheduling delay can be defined as 500ms. The TGs need to apply for the SCs through random access 500ms earlier than one transmission time slot in order to use the allocated SC after the scheduling delay. Among all the K TGs with successful random access, SGSCC will sequentially decide whether to assign SCs to achieve the lowest "average scheduling delay".

The scheduling delay of the nth TG is defined as $\delta_n(t) = t - u_n(t)$, $\delta_n(t) \in \mathbb{N}$, where $u_n(t)$ denotes the time of the last successful scheduling for the nth TG. When the nth TG is scheduled and allocated a SC, its scheduling delay is reset to 0. Otherwise, it is increased by 1, indicating that it is not scheduled in the next transmission time slot (and thus the scheduling delay is increased by 500ms). Accordingly, the evolution of $\delta_n(t)$ is defined in (2) below, and $\bar{\delta}_n(t)$ denote the normalized scheduling delay for the nth TG as defined in (3), where $\delta_{n\,\max}$ is the permission threshold for the scheduling delay. Thus, the normalized scheduling delays of TGs are defined as $\boldsymbol{\Delta}(t) = \{\bar{\delta}_0(t), \bar{\delta}_1(t), ..., \bar{\delta}_{N-1}(t)\}$. The normalization of scheduling delays can unify the scheduling delays of all TGs to the same scale to facilitate subsequent processing.

$$\delta_n(t+1) = \begin{cases} 1, & \text{if scheduling is successful,} \\ \delta_n(t)+1, & \text{else.} \end{cases} \quad (2)$$

$$\bar{\delta}_n(t) = (\delta_n(t)/\delta_{n\,\max}) \in [0,1] \quad (3)$$

Our objective is to maximize the communication reward per transmission time slot (cf. (4)) and minimize the normalized scheduling delays of all TGs (cf. (5)). Therefore, a scheduling algorithm is proposed that can take into account the scheduling delays and QoS levels for each TG and the status of assignable SCs. Specifically, we consider the following two scheduling problems:

- Optimization problem 1 $(OP1)$

$$\textbf{Maxmize}: \lim_{\bar{T}\to\infty} \frac{1}{\bar{T}K}[\sum_{t=0}^{\bar{T}-1}\sum_{k=0}^{K-1} t\_r_k(Kt+k)] \quad (4)$$

- Optimization problem 2 $(OP2)$

$$\textbf{Minimize}: \lim_{\bar{T}\to\infty} \frac{1}{\bar{T}N}[\sum_{t=0}^{\bar{T}-1}\sum_{n=0}^{N-1} \bar{\delta}_n(Kt)\cdot q_n(Kt)] \quad (5)$$

III. Proposed algorithm based on DDQN

In this section, we reformulate $OP1$ and $OP2$ into a finite Markov decision process (MDP). The $OP1$ given by (4) tries to maximize the average communication reward per transmission time slot, while the $OP2$ given by (5) tries to minimize the average normalized scheduling delay over all the TGs. Therefore, it is necessary that the effective scheduling consider these two problems simultaneously at the current random access time slot, and decide whether an SC with suitable level is assigned or not. It is also needed to somehow predict higher-reward or higher scheduling delay TGs in future time slots and reserve SCs based on historical scheduling experience, which is a complex problem. Based on the original DDQN, the environment status is divided into the resource status and request status of SCs in order to handle these problems at the same time.

A. Markov decision process formulation

The whole system can be modeled as a MDP, consisting of state set $\mathcal{S}$, action set $\mathcal{A}$, transition probability $P(\cdot|\cdot)$, reward R and discount factor $0 < \gamma < 1$, collectively denoted as $(\mathbf{S}(t), A(t), R(t{+}1), P(\mathbf{S}_{t+1}|\mathbf{S}_t), \gamma)$. At the beginning of each time slot, the SGSCC generates and executes $A(t)$ based on the current state $\mathbf{S}(t)$. Then the environment status varies, then yields the corresponding reward $R(t+1)$ and change to next state $\mathbf{S}(t+1)$ along with $P(\mathbf{S}_{t+1}|\mathbf{S}_t)$. Details are as follows:

1) **State**: $\mathbf{S}(t) \triangleq \{\mathbf{S}_1(t), \mathcal{H}(t)\}$ detailed as follows:
   - Spreading Code Resource Status $\mathbf{S}_1(t)$: $\mathbf{S}_1(t) \triangleq \{\mathbf{C}(t), \mathbf{D}(t)\}$, in which the levels $c_m(t) \in \{1,2,3\}$ and the availability status $d_m(t) \in \{0,1\}$ of M SCs.
   - Terminal Gateway Request Status $\mathbf{S}_2(t)$: $\mathbf{S}_2(t) \triangleq \{q_n(t), e_n(t), \bar{\delta}_n(t)\}$, indicates the request status of the nth TG, in which the QoS level $q_n(t) \in \mathbf{Q}(t)$, transmission capability $e_n(t) \in \mathbf{E}(t)$ and normalized scheduling delay $\bar{\delta}_n(t) \in \boldsymbol{\Delta}(t)$.

   Since TGs request status is partially observable within one transmission time slot (only K TGs' requests can be observed sequentially), the request status $\mathbf{S}_2(t)$ is collected into short-term history memory $\mathcal{H}(t) \triangleq \{\mathbf{S}_2(t-J), ..., \mathbf{S}_2(t)\}$ to extract temporal features and predict the request status of TGs in the coming time slots, where J is the number of historical random access collected.

2) **Action**: $A(t) \in \{0, 1, 2, 3\}$, where action 0 means no SC assignment to the TG of the current time slot; action $1, 2, 3$ stands for the corresponding level of SC assigned to the TG of the current time slot.

3) **Reward Function**: $R(t+1)$, the obtained reward after the execution of the action $A(t)$ at the current time t is given by:

$$R(t+1) = t_{-}r_k(t) - \bar{\delta}_n(t) \cdot q_n(t). \qquad (6)$$

The first term and second term in (6) are the communication reward (owing to the SC allocation to the TG) and the penalty due to the refusal of the SC allocation to the TG, respectively, for the current time slot; the latter is zero if a SC is allocated to the TG. Such reward setting allows the algorithm to take into account of both the QoS levels and the scheduling delays of the TGs.

The objective of the optimization is to find an optimal strategy π^*, which can maximize the expectation of long-term cumulative discount reward defined as $E[\sum_{t=0}^{T} \gamma^t R_{t+1}]$.

B. DDQN algorithm design

Model-based MDP requires the information of the transition probability $P(\mathbf{S}_{t+1}|\mathbf{S}_t)$ of each state, which is impossible to obtain in advance. With reinforcement learning (RL), the SGSCC can explore optimal policies directly through actual interaction with the environment with no need of $P(\mathbf{S}_{t+1}|\mathbf{S}_t)$. In other words, $P(\mathbf{S}_{t+1}|\mathbf{S}_t)$ has been blindly learned.

Deep Q-learning (DQN) is derived from the combination of Q-learning and deep network. The neural network is fitted to a Q-learning median lookup table, by inputting the state of the environment to get a value corresponding to each action. To update the values of $\boldsymbol{\theta}$ at the tth step, the DQN-based algorithm combining Q-learning and DNNs has two different Q-functions, i.e., an evaluation network $Q(\boldsymbol{s}, a|\boldsymbol{\theta})$ and a target network $Q_t(\boldsymbol{s}, a|\boldsymbol{\theta}_t)$. Nevertheless, the DQN-based algorithm may cause a large deviation in its model due to overestimating the value of target network Q_t. To avoid the overestimation, we propose an improved DDQN-based algorithm based on the DQN algorithm, which decouples the action selection and calculation of the value of Q-target. The difference between the two Q-functions is minimized by following the experience replay, where a loss function is used and it is defined as:

$$L(\boldsymbol{\theta}) = \sum_{\boldsymbol{s},a,r',\boldsymbol{s}'} (Y_{target}^{DDQN} - Q(\boldsymbol{s}, a|\boldsymbol{\theta}))^2, \qquad (7)$$

with $Y_{target}^{DDQN} = r' + \gamma Q_t(\boldsymbol{s}', \arg\max_{a'} Q(\boldsymbol{s}', a'|\boldsymbol{\theta})|\boldsymbol{\theta}_t)$, and r' denotes the reward from state $\boldsymbol{s}$ to $\boldsymbol{s}'$ via action a.

The details of the proposed algorithm are shown as Algorithm 1, and the network architecture of the proposed algorithm is shown in Fig. 3. The improved DDQN algorithm divides the input into two parts: SC resource status $\mathbf{S}_1(t)$ and TG request short-term history memory $\mathcal{H}(t)$. The one-hot $(1, K)$ input state indicates the number of random access TGs processed for the next transmission time slot.

Algorithm 1: DDQN of Spreading Code Scheduling

```
Input: No. of episodes $Eps$, no. of time slots $T$, no.
       of SCs $M$, no. of TGs $N$, no. of applications
       $K$, no. of requests collected $J$, batch size $b$,
       experience memory size $B$, networks update
       interval $W$, learning rate $\alpha$.
Output: Target network $Q_t$ with parameters $\boldsymbol{\theta}_t$.
Initialize experience memory $\mathcal{B} = \varnothing$ and short-term
  history memory $\mathcal{H} = \varnothing$.
Initialize evaluation network $Q$ and target network $Q_t$
  with random weights $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_t$, respectively.
Randomly initialize $\mathbf{C}(0)$, $\mathbf{Q}(0)$ and $\mathbf{E}(0)$; Set all
  $\bar{\delta}_n(0) = 0$ and $d_m(0) = 0$.
for episode=1 to $Eps$ do
    for $t$=1 to $J$ do
        Select random actions to interact with the
          environment, collect $\mathbf{S}_2$(t) to update
          $\mathcal{H}(t) \triangleq \{\mathbf{S}_2(t-J), ..., \mathbf{S}_2(t)\}$.
    end
    for $t$=$J+1$ to $T$ do
        According to the $\epsilon - greedy$ policy: With
          probability $\epsilon \in [0, 1]$, randomly select action
          $A(t) \in \{0, 1, 2, 3\}$; otherwise select
          $A(t) = \arg\max_{A(t)} Q(\mathbf{S}(t), A(t))$;
        Execute $A(t)$ to obtain the reward $R(t+1)$
          and $\mathbf{S}(t+1)$;
        Collect $\mathbf{S}_2(t+1)$ to update $\mathcal{H}(t+1)$ and store
          ($\mathbf{S}(t)$, $A(t)$, $R(t+1)$, $\mathbf{S}(t+1)$) into $\mathcal{B}$;
        if $t > J + b$ then
            Randomly sample a mini-batch from $\mathcal{B}$ with
              the corresponding $\mathcal{H}(t)$, and construct $b$
              sets of: ($\mathbf{S}(t)$, $A(t)$, $R(t+1)$, $\mathbf{S}(t+1)$);
            Set $Y_i = R_i(t+1) + \gamma Q_t(\mathbf{S}_i(t+1), \arg\max_{A_i(t+1)} Q(\mathbf{S}_i(t+1), A_i(t+1)))$;
            Calculate loss:
              $L(\boldsymbol{\theta}) = \frac{1}{b}\sum_{i=0}^{b-1}(Y_i - Q(\mathbf{S}_i(t), A_i(t)))^2$;
            Update $Q$ parameter vector
              $\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$;
            Update $\boldsymbol{\theta}_t$ by $\boldsymbol{\theta}$ for every $W$ random access
              time slots.
        end
    end
end
```

The LSTM network [14] that consolidates useful information from long-term inputs is suitable for feature extraction of temporally sequential information. To predict future requests and decide whether to assign SC to the current TG, we use the LSTM network to process the short-term history memory $\mathcal{H}(t)$. The DDQN algorithm inputs the two sets of states into the linear network and the LSTM respectively to extract the corresponding features. Then gathers them in the set of hidden components to extract high-dimensional information. Finally,

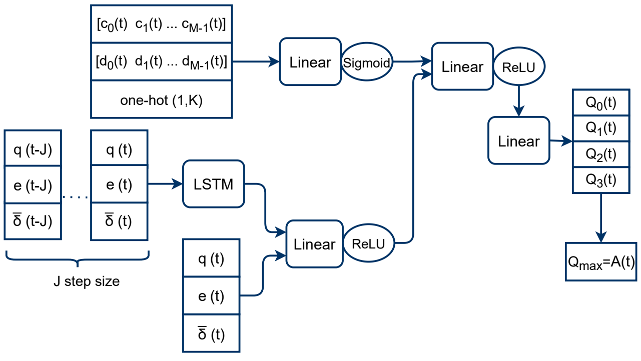

Fig. 3. Network architecture of the DDQN

the value of each action is output through the output layer. Then the algorithm selects the action with the highest value Q_{max} in the output layer to execute.

IV. Simulation Results

In this section, we present some simulation results for different scenarios to verify the efficacy of the proposed scheduling algorithm (Algorithm 1). The evaluation network and the target network both use the network architecture as shown in Fig. 3. We set the sigmoid and ReLU as the activation function of the input linear layer and LSTM respectively, and the number of neurons in the hidden layer is 128. At the same time, the initial value of the network's weight is selected according to the normal distribution with zero mean and the variance of 0.1. The network optimizer and loss function are based on Adam and mean square error (MSE) respectively. Other hyper parameters settings are listed in TABLE I.

TABLE I
Hyper Parameters

γ	α	ϵ	β	Eps	M	N
0.95	0.001	$1 \rightarrow 0.1$	0.25	300	4	45
W	B	b	K	J	$\bar{T}$	T
40	4000	32	8	8	100	800

The levels $\mathbf{C}(t)$ of the M assignable SCs of SGSCC are selected from $\{1, 2, 3\}$ with probabilities $[0.20, 0.45, 0.35]$. For each TG, its QoS level $q_n(t)$ is selected from $\{1, 2, 3\}$ with probabilities $[0.50, 0.35, 0.15]$, while the transmission capacity of 50% the TGs is set to $e_n(t) = 1$. At the same time, the scheduling delays $\delta_{n\,\max}$ of all TGs are randomly selected from $\{10, 15, 20, 25\}$, where the unit is 500ms.

For the performance comparison, the proposed scheduling algorithm and three commonly used algorithms are evaluated under the same constraints as follows:

- *Random policy*: randomly assign SCs or reject assignments;
- *Standard policy*: only assign SCs to TGs strictly according to their QoS levels;
- *ZeroWait policy*: allocate SCs in the order of the TGs with successful random access.

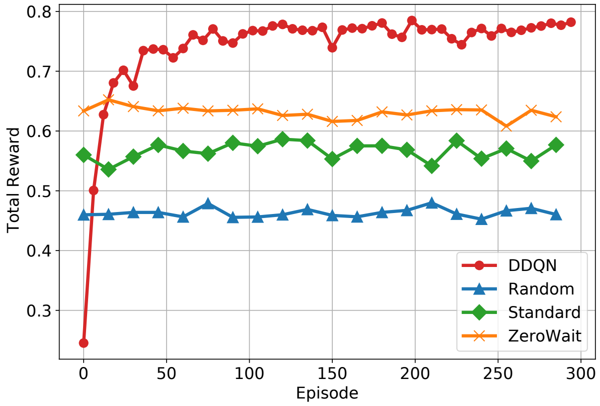

Fig. 4. Average communication reward per transmission time slot.

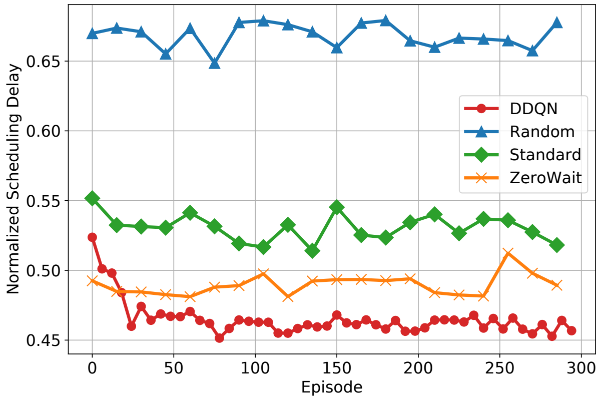

Fig. 5. Average normalized scheduling delays per transmission time slot.

First, we evaluate the performance of the *DDQN policy* in the scheduling environment parameters given in TABLE I. Fig. 4 and Fig. 5 show the convergence results of the four algorithms over 300 epsilons. *DDQN policy* performs better than all the other scheduling policies in terms of average average communication reward and average normalized scheduling delay. *Standard policy* and *ZeroWait policy* cannot achieve high performance because only one (but not both) of the communication reward and the scheduling delay is considered. As a result, some higher-level SCs are left idle by *Standard policy*, while all spreading codes are allocated prematurely by *ZeroWait policy*.

In order to show the effect of the number (K) of random access TGs in each transmission time slot on the performance of SC allocation. The simulation is also performed for the four algorithms for different values of K over 10 independent runs, and the obtained results (normalized scheduling delay and total communication reward) of DDQN policy are the averages over the last 10 episodes of the training results.

Figure 6 shows that as K increase, the normalized scheduling delays associated with *DDQN policy*, *Random policy* and *Standard policy* decrease simply because the number of selectable TGs increases in each transmission time slot.

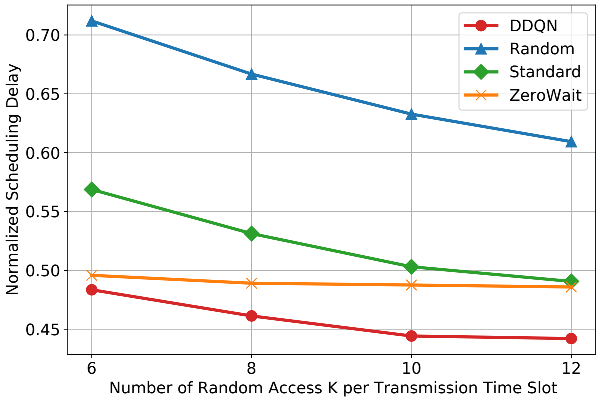

Fig. 6. The effect of different K on the normalized average scheduling delay.

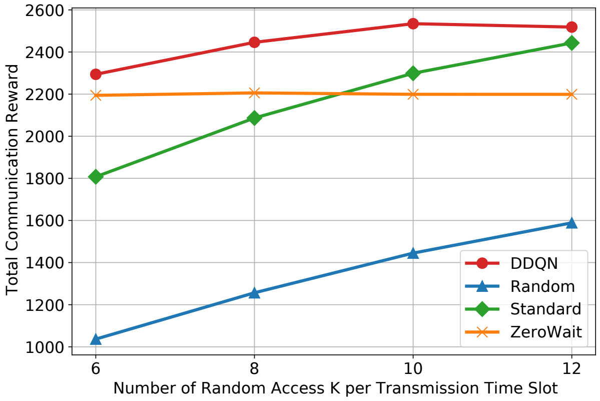

Fig. 7. The effect of different K on the total communication reward in a typical episode.

However, due to the premature assignment of all available SCs by *ZeroWait policy*, the corresponding normalized scheduling delay only slightly decreases.

The total communication reward for the four policies for a typical episode is shown in Fig. 7. One can observe from this figure, that as K increases, *DDQN policy* can assign more higher level SCs to TGs (i.e., high-level TGs), thereby yielding higher total communication reward. This observation also applies to *Random policy* and *Standard policy* owing to simple scheduling operation for high-level TGs, while the total communication reward of *ZeroWait policy* almost remains unchanged.

V. Conclusion

We have presented a scheduling DDQN policy which is implemented by the proposed scheduling algorithm (Algorithm 1), for efficient resource (SCs) allocation fulfilled by the SGSCC of satellite IoT systems. The proposed algorithm is designed by considering the scheduling delay minimization and the transmission award maximization at the same time, thereby providing a feasible solution to users' requirements and practical resource constraints. Simulation results have been provided to demonstrate that the proposed algorithm achieves much better performance than traditional algorithms and exhibits more efficient scheduling capabilities. In future work, we will consider the scenario that TGs share SCs in tacit agreement so that the system will become more intelligent, together with how to reduce the collision caused by the TGs that use the same SC will be the crux direction.

Acknowledgment

The work is supported partly by the National Key Research and Development Program of China (No.2019YFE0196400), Guangdong R&D Project in Key Areas under Grant (2019B010158001, 2019B010156004), IndustryUniversity-Research Cooperation Project in Zhuhai (No.ZH22017001200072PWC), and Ministry of Science and Technology in Taiwan (MOST 110-2221-E007-031).

References

[1] W. J. Choi, D. W. Lee, J. W. Eun and J. H. Lee, "Theoretical interpretation of interference arising between closely spaced dual polarized geostationary satellites." *Journal of Information and Communication Convergence Engineering*, vol. 19, no. 3, 2021, pp. 131–135.

[2] B. Ren, J. Liu, Z. Li, J. Wu and Z. Yao, "Satellite requirement preference driven TT&C resources scheduling algorithm for time sensitive missions," in *Proc. 2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)*, 2020, pp. 15–19.

[3] X. Liu, K. Xu, F. Wu and J. Wu, "A beam-dominating frequency resource allocation and scheduling scheme for multi-beam satellite system," in *Proc. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, 2021, pp. 532–535.

[4] D. Peng, A. Bandi, Y. Li, S. Chatzinotas and B. Ottersten, "Hybrid beamforming, user scheduling, and resource allocation for integrated terrestrial-satellite communication." *IEEE Trans. Vehicular Technology*, vol. 70, no. 9, pp. 8868–8882, Sept. 2021.

[5] Y. Huang and A. O. Fapejuwo, "Integrated call admission control and packet scheduling for multimedia direct sequence code division multiple access (DS-CDMA) wireless networks," in *Proc. IEEE 60th Vehicular Technology Conference*, Fall 2004, vol. 4, pp. 2673–2677.

[6] A. D. Panagopoulos, M. P. Anastasopoulos and P. G. Cottis, "Error performance of satellite links interfered by two adjacent satellites," *IEEE Antennas and Wireless Propagation Letters*, vol. 6, pp. 364–367, 2007.

[7] N. Gu, L. Kuang, X. Chen, Z. Ni and J. Lu, "An eigen-based spreading sequences design framework for CDMA satellite systems," in *Proc. 2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, pp. 1–6.

[8] Q. Wang, G. Ren, S. Gao and K. Wu, "A framework of non-orthogonal slotted aloha (NOSA) protocol for TDMA-based random multiple access in IoT-oriented satellite networks," *IEEE Access*, vol. 6, pp. 77542–77553, 2018.

[9] J. Vankka, "Performance of satellite gateway over geostationary satellite links," in *Proc. 2013 IEEE Military Communications Conference*, 2013, pp. 289–292.

[10] A. J. Roumeliotis, C. I. Kourogiorgas and A. D. Panagopoulos, "An optimized simple strategy for capacity allocation in satellite systems with smart gateway diversity," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4668–4674, Sept. 2021.

[11] H. Kellerer, U. Pferschy and D. Pisinger, "Multidimensional knapsack problems," *Knapsack Problems*, 2004, pp. 285–316.

[12] X. Hu, X. Luan, S. Ren and J. Wu,"Propagation delays computation in GEO multi-beam satellite communications system," in *Proc. 2012 International Conference on Systems and Informatics (ICSAI2012)*, 2012, pp. 1631–1634.

[13] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge and C. -X. Wang, "5G embraces satellites for 6G ubiquitous IoT: Basic models for integrated satellite terrestrial networks." *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14399-14417, Sept. 2021.

[14] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, 2014, 27.