IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 June 2024, accepted 28 July 2024, date of publication 5 August 2024, date of current version 21 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438930

RESEARCH ARTICLE

Satellite Communication Resource Scheduling Using a Dynamic Weight-Based Soft Actor Critic Reinforcement Learning

ZHIMIN QIAO[1], **WEIBO YANG**[2], **FENG LI**[1], **YONGWEI LI**[1], **AND YE ZHANG**[1]

[1]Department of Automation, Taiyuan Institute of Technology, Taiyuan 030008, China
[2]School of Automobile, Chang'an University, Xi'an 710064, China

Corresponding author: Zhimin Qiao (qiao.miracle@gmail.com)

This work was supported in part by the Fundamental Research Program of Shanxi Province under Grant 202303021222300 and Grant 202103021223352, in part by the 5th Discipline Leader Project of Taiyuan Institute of Technology, and in part by the Shanxi Province Department of Education University Science and Technology Innovation Project under Grant 2022L523.

ABSTRACT One of the key challenge faced by space-based network is how to maximize the demand for on-board resources for ground communication tasks, given the limited availability of satellite resources. For this challenge, firstly, we propose a joint state space of satellite task requirements and resource pools to obtain the global information of the environment, avoiding convergence to local optimal strategies. Secondly, we propose a new joint partitioning method for frequency and time resources, which avoids the fragmentation of the resource to the maximum extent. Thirdly, a new algorithm called dynamic weight based soft actor critic (DWSAC) is proposed, which enhances the update range when the actions taken by the agent significantly contribute to the improvement of system performance, otherwise weakens the update range, significantly improving the convergence efficiency and performance of the soft actor critic (SAC). The results show that the proposed model and algorithm have good practicability, which can make the average resource occupancy rate higher and the running cost lower.

INDEX TERMS Reinforcement learning, satellite resource scheduling, dynamic weight, soft actor critic.

I. INTRODUCTION

With the continuous growth of demand for satellite communication tasks, one of the major challenges faced by space-based network is how to achieve fast and efficient scheduling of satellite communication resource and maximize the satisfaction of the demand for satellite resources for ground communication tasks, given the limited availability of satellite resources [1]. The traditional fixed resource scheduling method will make the shortage of satellite communication resources increasingly serious. When facing non-uniform distribution of beam service requirements, this method finds it difficult to match the allocated resources with the resources required by the beam. Usually, there will be insufficient beam allocation resources for high service requirements and excess beam allocation resources for low service requirements, resulting in resource waste [2]. Therefore, designing reasonable and efficient communication resource scheduling methods for key system resources such as subcarriers, power, and beam hopping time slots can provide users with better communication services and reduce the cost of each bit of information in satellite-Internet communication, and it is of great significance for promoting the development of satellite communication technology.

A lot of research has been done on resources scheduling problem related to satellite communication system. In [3] designed a dynamic scheduling method that integrates beam coverage channels based on real-time location information of user terminals. This method can allocate channel resources in real time, thereby improving the utilization rate of spectrum. In [4] proposed a channel feedback optimization access mechanism and constructed a dynamic negotiation feedback model by introducing a humble incentive method. In [5] adopts a combination of GA and PSO to obtain the optimal reserved channel threshold, in order to reserve sufficient channel resources for various call types, service types,

The associate editor coordinating the review of this manuscript and approving it for publication was Ayaz Ahmad .

and terminal types. In [6] proposes a solution for multi-objective optimization problem aimed at minimizing system unmet capacity and transmission power. In [7] used convex optimization methods to optimize beam resource allocation. However, the above research mainly focuses on the problem of resource allocation at the user level within the beam, which is easy to waste the resource of each beam due to the spatial inhomogeneity and time variability of the service distribution. In [8], beams are uniformly clustered and the optimal clustering size is solved by convex optimization method, and then a joint power and beamhopping time-slot allocation algorithm is proposed to allocate resources to each cluster. In [9] studies the synergistic effect of hopping beam technology and NOMA technology in multi-beam satellite communication systems, establishes a joint scheduling model of hopping beam resources and power resource, and proposes an improved greedy algorithm to solve the model. However, with the increase of the number of beams and the number of users, the power consumption and computing power of the satellite are also strictly limited. Excessive constraints will cause the variables to be solved and the computational complexity to increase significantly.

With the development of RL technology [10] and the improvement of perception ability in satellite communication systems, more and more empirical data is being saved [11]. Deep reinforcement learning, due to its own characteristics, can effectively use this data to discover patterns and learning strategies. Therefore, the method based on reinforcement learning has been widely studied in task and resource scheduling [12], and is also suitable for resource scheduling in satellite communication. In [13] studies the application of reinforcement learning in satellite-terrestrial fusion networks. In [14] studies an asynchronous reinforcement learning model for subcarrier allocation in broadband cognitive radio network, taking into account whether secondary users can exchange information. In [15] adopts a deep Q learning scheme to jointly optimize cache, computation, and network resources in satellite-terrestrial network, improving resources utilization efficiency. In [16] models D2D devices as intelligent agents, models power scheduling problem as a Markov decision process, and then uses MADQN algorithm to train the optimal scheduling strategy. In [17] introduces the PDMA channel matrix to improve the channel resource multiplex rate and a reward threshold is introduced on the basis of the Q learning algorithm. In [18] proposes a power allocation method based on DRL to address high dynamic characteristics of low orbit satellites and the limitations of frequency and power resources. In [19] transforms the two-dimensional subcarrier and transmission power selection problem into a one-dimensional armed bandit problem, and proposed a distributed MAB based DQN algorithm. However, there are still some problems in these methods, such as the unreasonable design of task scheduling action space, the decrease in resource utilization caused by the unreasonable design of reward function, and the low convergence efficiency and poor convergence performance caused by the unreasonable algorithm design.

In this paper, we take the task requirements and satellite communication resources as the observation object. Firstly, we propose a joint state space of satellite task requirements and resource pools to obtain the global information of the environment, avoiding convergence to local optimal strategies. Secondly, we propose a new joint partitioning method for frequency and time resources, which avoids the fragmentation of the resource to the maximum extent. Finally, in order to overcome the drawbacks of traditional soft actor critic algorithm [20] that use fixed learning rate, the agent cannot dynamically adjust the learning rate based on the real-time reward changing over time, which affects the convergence rate and performance of the algorithm. We proposes a soft actor critic based on dynamic weight (DWSAC). It incorporates a dynamic weight mechanism to enhance the update range when the actions taken by the intelligent agent contribute to the improvement of system performance, otherwise reduce the update range.

The rest of the paper is organized as follows. Section II introduces the proposed DWSAC, which is used to solve satellite communication resource scheduling problem. Section III shows description of satellite communication resource scheduling based on Markov process. Section IV is simulation and analysis. Section V is conclusion.

II. SOFT ACTOR CRITIC ALGORITHM BASED ON DYNAMIC WEIGHT

Compared with DDPG, SAC uses a random policy and has some advantages over deterministic policy. Specifically, deterministic strategy refers to a strategy that only choose a best action for a state. However, in many problems, there may be more than one optimal action. In this case, a stochastic strategy can be considered, which can output the probability of each action in each state. The maximum entropy (ME) mechanism adopted by SAC is to not leave behind any useful action. The approach of using deterministic strategy in DDPG is to pick up the good one and discard the slightly inferior one, while the maximum entropy is to pick up everything and consider everything.

However, SAC adopts a fixed learning rate, and the agent cannot adjust the learning rate based on the real-time reward changing over time, which to some extent affects the convergence. Therefore, this paper introduces dynamic weight into the SAC and proposes DWSAC.

A. DYNAMIC WEIGHT

In order to introduce dynamic weight into the SAC, first of all, it is necessary to identify the valid and invalid information during the learning process. In the framework based on actor-critic, the large amount of information collected by an agent is sparse and delayed, which is ineffective for critic and therefore cannot efficiently obtain useful reward values in most cases. In addition, how to enable actor to learn action strategies from sparse reward values is another important problem that needs to be addressed, thus the algorithm's parameter update process needs to be modified. Due to the different characteristics of the parameters updating for actor and critic, it is necessary to set different updating weights respectively.

Firstly, the weight of critic network is introduced, and a ratio is set for the update of network parameter according to the reward value before and after the execution of the agent action, which is used to reflect the impact of the current action on the environment and improves the convergence rate. It should be noted that in the definition process of this ratio, if only the linear ratio of the current reward Υ_{cr} to the previous reward Υ_{pv} is used, it cannot accurately reflect the updated values. Therefore, the ratio is defined as follows:

$$\mathrm{R}_{c} = \begin{cases} 1 & \Upsilon_{\mathrm{pv}} = 0 \\ \exp\left(\Upsilon_{\mathrm{cr}} / \Upsilon_{\mathrm{pv}} - 1\right) & \Upsilon_{\mathrm{cr}} / \Upsilon_{\mathrm{pv}} > 1 \\ \Upsilon_{\mathrm{cr}} / \Upsilon_{\mathrm{pv}} & \Upsilon_{\mathrm{cr}} / \Upsilon_{\mathrm{pv}} \leq 1 \end{cases} \tag{1}$$

Next, the gradient weight can be assigned value using defined ratio. According to $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i), i \in \{1, 2\}$ for updating the critic network parameter in [21] and the Eq. (1), the update of critic network parameter can be described as follows:

$$\theta_i \leftarrow \theta_i - \varepsilon_c \cdot \min(\mathrm{R}_c, \xi_c) \cdot \hat{\nabla}_{\theta_i} J_Q(\theta_i)\ i \in \{1, 2\} \tag{2}$$

where θ_i is the parameter of critic network, ε_c is learning rate. ξ_c is the upper limit of the weight, used to prevent network oscillation when the gradient changes significantly during a single updating.

Similarly, for the weight of the actor network, it is found in [22] that if the current reward changes significantly compared to the previous reward, it will cause the update amplitude of actor to be too large, resulting in oscillation. To avoid this phenomenon, we propose a method to make the gradient weight of actor smoother, which can be defined as follows:

$$\mathrm{R}_a = 1 + \frac{\left|\Upsilon_{cr} - \Upsilon_{pv}\right|^2}{\Upsilon_{cr}^2 + \Upsilon_{pv}^2} \tag{3}$$

According to the formula $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ for updating the actor network parameter in [21] and the Eq. (3), the updating of actor network parameter can be described as follows:

$$\phi \leftarrow \phi - \varepsilon_a \cdot \min(\mathrm{R}_a, \xi_a) \cdot \hat{\nabla}_\phi J_\pi(\phi) \tag{4}$$

where ξ_a is a threshold used to avoid oscillation in actor network. After adopting this approach, effective resources can be efficiently utilized to accelerate the convergence of the network.

B. DESIGN OF SOFT ACTOR-CRITIC ALGORITHM BASED ON DYNAMIC WEIGHT

The SAC uses a function approximator to approximate the soft Q value and strategy, and optimizes the two networks using random gradient descent. The parameterized Q value function and strategy function are $Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ and $\pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)$, respectively, and their network parameters are θ and ϕ. Next, we will export update rules for these parameters.

The soft state value function $V(\mathbf{s}_t)$ can be defined as follows:

$$V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q(\mathbf{s}_t, \mathbf{a}_t) - \alpha \log \pi(\mathbf{a}_t \mid \mathbf{s}_t)\right] \tag{5}$$

The parameters of the soft Q valued function can be trained by minimizing the soft Bellman residuals, which can be described as follows:

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[V_{\bar{\theta}}(\mathbf{s}_{t+1})\right])\right)^2\right] \tag{6}$$

where the value function $V_{\bar{\theta}}(\mathbf{s}_{t+1})$ is an implicit parameterized form of the parameter θ of the soft Q value function of the Eq. (5), which is optimized with a random gradient and can be described as follows:

$$\begin{aligned} \hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathbf{a}_t, \mathbf{s}_t)\,(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) & - \left(r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\left(Q_{\bar{\theta}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right.\right. \\ & \left.\left. - \alpha \log\left(\pi_\phi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1})\right)\right)\right)) \end{aligned} \tag{7}$$

The updating utilizes the target soft Q value function with parameter $\bar{\theta}$.

For the strategy, SAC updates it with the Kullback-Leibler divergence, which can be described as follows:

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} D_{KL}\left(\pi'(\cdot \mid \mathbf{s}_t) \,\|\, \frac{\exp\left(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(\mathbf{s}_t, \cdot)\right)}{Z^{\pi_{\text{old}}}(\mathbf{s}_t)}\right) \tag{8}$$

where $Z^{\pi_{\text{old}}}(s_t)$ is the partition function used to normalize the distribution, which is usually difficult to handle and can usually be ignored.

The parameter θ of the strategy π can be learned by directly minimizing the expected Kulbach-Leibler divergence in the Eq. (8), which can be described as follows:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[\alpha \log\left(\pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)\right) - Q_\theta(\mathbf{s}_t, \mathbf{s}_t)\right]\right] \tag{9}$$

where α is a constant that determines the relative importance of the entropy term relative to the reward.

We use a re-parameterization method to minimize J_π, which is not only convenient but also has a lower variance estimation. To this end, a neural network is used to reparameterize the strategy, which can be described as follows:

$$\mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t) \tag{10}$$

where ϵ_t is input noise.

According to the Eq. (9) and Eq. (10), $J_\pi(\phi)$ can be rewritten as follows:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D},\epsilon_t\sim\mathcal{N}}\left[\alpha\log\pi_\phi\left(f_\phi\left(\epsilon_t;\mathbf{s}_t\right)\mid\mathbf{s}_t\right)\right.$$
$$\left. - Q_\theta\left(\mathbf{s}_t, f_\phi\left(\epsilon_t;\mathbf{s}_t\right)\right)\right] \tag{11}$$

where: π_ϕ is implicitly defined according to f_ϕ.

The gradient of Eq. (10) can be defined as follows:

$$\begin{aligned}&\hat{\nabla}_\phi J_\pi(\phi)\\&= \nabla_\phi\alpha\log\left(\pi_\phi\left(\mathbf{a}_t\mid\mathbf{s}_t\right)\right) + \left(\nabla_{\mathbf{a}_t}\alpha\log\left(\pi_\phi\left(\mathbf{a}_t\mid\mathbf{s}_t\right)\right)\right.\\&\quad\left. - \nabla_{\mathbf{a}_t}Q\left(\mathbf{s}_t,\mathbf{a}_t\right)\nabla_\phi f_\phi\left(\epsilon_t;\mathbf{s}_t\right)\right.\end{aligned} \tag{12}$$

where $\mathbf{a}_t$ is evaluated in $f_\phi(\epsilon_t;\mathbf{s}_t)$. This unbiased gradient estimation extends the policy gradient in the form of DDPG [24] to any easily processed random strategy.

The aforementioned algorithm learns the maximum entropy strategy under a given temperature, but in practical problems, the optimal temperature should be adjusted according to the specific problem. Therefore, setting a maximum entropy reinforcement learning objective and adaptively adjusting temperature has practical significance, where entropy is considered as a constraint in which the average entropy of the strategy is constrained, and the entropy varies in different states. The goal of the algorithm is to find a random strategy that maximizes the expected reward. In addition, this strategy minimizes the expected entropy constraint. It can be defined as follows:

$$\begin{aligned}&\max_{\pi_{0:T}}\ \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T} r\left(\mathbf{s}_t,\mathbf{a}_t\right)\right]\\&\text{s.t. }\ \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[-\log\left(\pi_t\left(\mathbf{a}_t\mid\mathbf{s}_t\right)\right)\right]\geq\mathcal{H}\quad\forall t\end{aligned} \tag{13}$$

where $\mathcal{H}$ is the minimum expected value of entropy. It is important to note that for a fully observable Markov decision process, the strategy for optimizing reward expectations is deterministic, so the constraint is usually strict and does not need to impose an upper limit on entropy.

Since the policy at time step t can only affects the future target value, a dynamic programming method can be used. Here we rewrite the goal in iteration-maximized form as follows:

$$\max_{\pi_0}\left(\mathbb{E}\left[r\left(\mathbf{s}_0,\mathbf{a}_0\right)\right] + \max_{\pi_1}\left(\mathbb{E}[\ldots] + \max_{\pi_T}\mathbb{E}\left[r\left(\mathbf{s}_T,\mathbf{a}_T\right)\right]\right)\right) \tag{14}$$

The Eq. (14) is constrained by entropy, and from the last time step, the constraint maximization problem is transformed into a dual problem, which can be defined as follows:

$$\begin{aligned}&\max_{\pi_T}\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[r\left(\mathbf{s}_T,\mathbf{a}_T\right)\right] = \min_{\alpha_T\geq 0}\max_{\pi_T}\mathbb{E}\left[r\left(\mathbf{s}_T,\mathbf{a}_T\right)\right.\\&\quad\left.-\alpha_T\log\pi\left(\mathbf{a}_T\mid\mathbf{s}_T\right)\right] - \alpha_T\mathcal{H}\end{aligned} \tag{15}$$

$$\mathbb{E}\left(\mathbf{s}_T,\mathbf{a}_T\right)\sim\rho_\pi\left[-\log\left(\pi_T\left(\mathbf{s}_T\mid\mathbf{s}_T\right)\right)\right]\geq\mathcal{H} \tag{16}$$

where α_T is a dual variable. Due to the fact that the objective is linear and the constraint (entropy) is a convex function in π_T, strong duality is also used here, which is closely related to the maximum entropy objective of the policy. The optimal policy is the maximum entropy strategy corresponding to temperature α_T : $\pi_T^*\left(\mathbf{a}_T\mid\mathbf{s}_T;\alpha_T\right)$. The solution of the optimal dual variable α_T^* can be defined as follows:

$$\arg\min_{\alpha_T}\mathbb{E}_{\mathbf{s}_t,\mathbf{a}_t\sim\pi_t^*}\left[-\alpha_T\log\pi_T^*\left(\mathbf{a}_T\mid\mathbf{s}_T;\alpha_T\right) - \alpha_T\mathcal{H}\right] \tag{17}$$

To simplify, the following equation takes advantage of the recursive definition of a soft Q value function:

$$\begin{aligned}&Q_t^*\left(\mathbf{s}_t,\mathbf{a}_t;\pi_{t+1:T}^*,\alpha_{t+1:T}^*\right)\\&= \mathbb{E}\left[r\left(\mathbf{s}_t,\mathbf{a}_t\right)\right] + \mathbb{E}_{\rho_\pi}\left[Q_{t+1}^*\left(\mathbf{s}_{t+1},\mathbf{a}_{t+1}\right)\right.\\&\quad\left.-\alpha_{t+1}^*\log\pi_{t+1}^*\left(\mathbf{a}_{t+1}\mid\mathbf{s}_{t+1}\right)\right]\end{aligned} \tag{18}$$

where $Q_T^*\left(\mathbf{s}_T,\mathbf{a}_T\right) = \mathbb{E}\left[r\left(\mathbf{s}_T,\mathbf{a}_T\right)\right]$. Using the duality problem again under the entropy constraints, the equation can be obtained as follows:

$$\begin{aligned}&\max_{\pi_{T-1}}\left(\mathbb{E}\left[r\left(\mathbf{s}_{T-1},\mathbf{a}_{T-1}\right)\right] + \max_{\pi_T}\mathbb{E}\left[r\left(\mathbf{s}_T,\mathbf{a}_T\right)\right]\right)\\&= \max_{\pi_{T-1}}\left(Q_{T-1}^*\left(\mathbf{s}_{T-1},\mathbf{a}_{T-1}\right) - \alpha_T\mathcal{H}\right)\\&= \min_{\alpha_{T-1}>0}\max_{\pi_{T-1}}\left(\mathbb{E}\left[Q_{T-1}^*\left(\mathbf{s}_{T-1},\mathbf{a}_{T-1}\right)\right]\right.\\&\quad\left.-\mathbb{E}\left[\alpha_{T-1}\log\pi\left(\mathbf{a}_{T-1}\mid\mathbf{s}_{T-1}\right)\right] - \alpha_{T-1}\mathcal{H}\right) + \alpha_T^*\mathcal{H}\end{aligned} \tag{19}$$

In this way, we can backtrack in time and recursively optimize the Eq. (15). It should be noted that the optimal strategy at time step t is a function of the dual variable α_t. Similarly, after solving Q_t^* and π_t^*, the optimal dual variable α_t^* can be solved, which can be defined as follows:

$$\alpha_t^* = \arg\min_{\alpha_t}\mathbb{E}_{\mathbf{a}_t\sim\pi_t^*}\left[-\alpha_t\log\pi_t^*\left(\mathbf{a}_t\mid\mathbf{s}_t;\alpha_t\right) - \alpha_t\overline{\mathcal{H}}\right] \tag{20}$$

The solution in the Eq. (20), as well as the update of the strategy and soft Q function described earlier, constitute the core of this algorithm. In theory, these variables and strategies are accurately solved recursively, and the maximum expected reward objective of the optimal entropy constraint in the Eq. (15) is optimized, but in reality, function approximators and random gradient descent is needed.

In this algorithm, two soft Q value functions are used to alleviate the positive bias of the strategy, which reduces the performance of value based methods [25]. Specifically, two soft Q value functions are parameterized with parameter θ_i, and they are trained independently to optimize $J_Q\left(\theta_i\right)$, and then the minimum value of the soft Q value is substituted into the Eq. (7) and Eq. (12) to solve the stochastic gradient and the policy gradient respectively.

In addition to the soft Q value function and strategy, learning α is also achieved by minimizing the dual objective in the Eq. (20). This can be achieved by approximating double gradient descent. Although it is impractical to completely optimize the original variables, under the assumption of convexity, truncated versions that perform incomplete optimization can be proven to converge [26]. Although

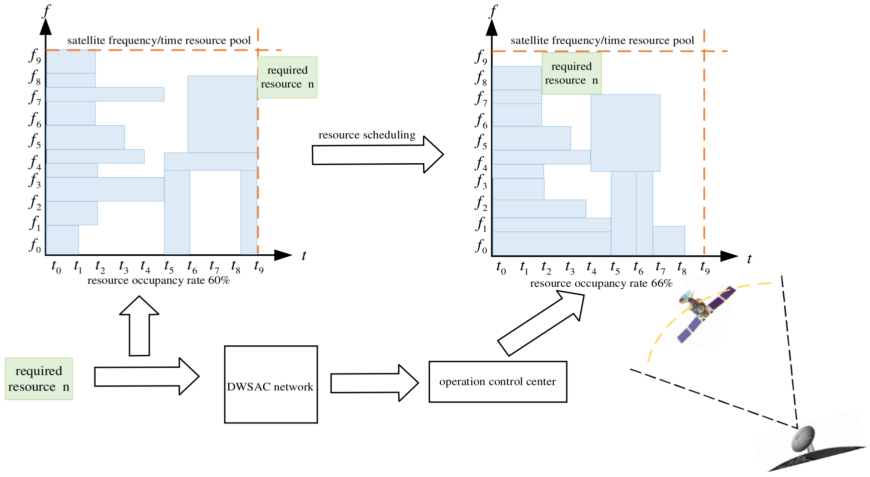

f
satellite frequency/time resource pool
f_9 f_8 f_7 f_6 f_5 f_4 f_3 f_2 f_1 f_0
required resource n
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t
resource occupancy rate 60%
resource scheduling
f
satellite frequency/time resource pool
required resource n
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t
resource occupancy rate 66%
required resource n
DWSAC network
operation control center

FIGURE 1. Satellite communication resource scheduling diagram.

these assumptions are not applicable to non-linear function approximators such as neural network, it has been found in practice that this method is still effective. Therefore, in order to calculate the gradient target of α, it can be defined as follows:

$$J(\alpha) = \mathbb{E}_{\mathbf{a}_t\sim\pi_t}\left[-\alpha\log\pi_t\left(\mathbf{a}_t \mid \mathbf{s}_t\right) - \alpha\overline{\mathcal{H}}\right] \tag{21}$$

Combined with the dynamic weights proposed in section II-A, this paper proposes the DWSAC. The pseudocode of the algorithm is given in Algorithm 1.

Algorithm 1 DWSAC Algorithm

```
Input: Initial parameters $\theta_1, \theta_2, \phi$.
Initializing parameters $\theta_1, \theta_2, \phi$
Initializing target network parameters $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$
Initializing an empty experience replay pool $\mathcal{D} \leftarrow \emptyset$
for each iteration do
    for each environmental step do
        Getting an action based on the current policy $\mathbf{a}_t \sim \pi_\phi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)$
        Getting the next state $\mathbf{s}_{t+1} \sim p\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right)$
        Saving to the experience replay pool $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r\left(\mathbf{s}_t, \mathbf{a}_t\right), \mathbf{s}_{t+1})\}$
    end for
    for each gradient updating step do
        Updating the parameter of the $Q$ value function $\theta_i \leftarrow \theta_i - \varepsilon_c \cdot \min(\mathrm{R}_c, \xi_c) \cdot \hat{\nabla}_{\theta_i} J_Q\left(\theta_i\right)$ for $i \in \{1, 2\}$
        Updating policy parameter $\phi \leftarrow \phi - \varepsilon_a \cdot \min(\mathrm{R}_a, \xi_a) \cdot \hat{\nabla}_\phi J_\pi(\phi)$
        Adjusting temperature $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$
        Updating target network parameter $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i$ for $i \in \{1, 2\}$
    end for
end for
Output: Parameters after training $\theta_1, \theta_2, \phi$.
```

III. DESCRIPTION OF SCRS BASED ON MARKOV PROCESS

Satellite communication resource scheduling (SCRS) includes time resource scheduling and frequency resource scheduling. The common resource scheduling process based on transparent satellite transponder is shown in Fig.1. By scheduling time and frequency reasonably and quickly, the utilization rate of satellite repeater resources can be improved under multiple constraints. Improving satellite resources occupancy rate through DWSAC network is shown in Fig.1. The essence of using RL methods to solve the SCRS is to find a set of better sorting sequences than using heuristic algorithms. In this paper, a satellite management center is modeled as an agent. The external environment includes task requirements and satellite communication resources. In addition, the main elements of RL based on DWSAC definition in the SCRS problem will be discussed as follows. The SCRS based on DWSAC is shown in Fig. 2.

1) state space

The satellite task requirements and resource pool state is $s_t = \left(s_{\mathrm{rp}}, s_{\mathrm{tl}}\right)_t$. s_{rp} is resource pool state at time t. s_{tl} is task list state at time t. When there are new task requirements, it is necessary to format the task list status s_{tl} and update the status of task list s_{tl}. s_{tl} is described as follows:

$$s_{\mathrm{tl}} = \{[\delta_1, \varphi_1(t_1), \varphi_2(f_1)], \cdots, [\delta_m, \varphi_1(t_m), \varphi_2(f_m)]\} \tag{22}$$

where, δ_m is the allocation status of the mth task in the satellite resource pool, t_m and f_m are the satellite time and frequency resources occupied by the task, respectively. $\varphi_1(t_m) = N \times t_m/\Delta t$ and $\varphi_2(f_m) = N \times f_m/\Delta f$ are the state reconstructions of t_m and f_m, respectively, so that they conform to the tensor size of the model inputing. t_m and f_m are both rephrased within $[1, N]$, while Δf and Δt represent the time and frequency resource ranges in the satellite resource pool, respectively. By dividing frequency resources and time resources into $N-1$ times in their respective dimensions, the satellite resource pool can be divided into $N \times N$ resource blocks. The

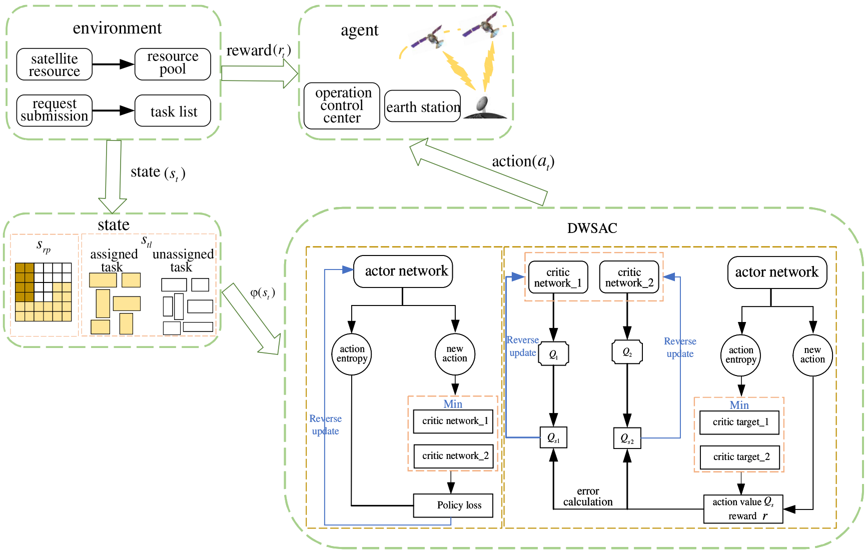


FIGURE 2. Overview diagram of satellite resource scheduling based on DWSAC method.

state matrix $\boldsymbol{s}_{\mathrm{rp}}$ is used to represent the occupancy of each resource block in the resource pool. The state matrix $\boldsymbol{s}_{\mathrm{rp}}$ is defined as follows:

$$\boldsymbol{s}_{\mathrm{rp}} = \begin{bmatrix} p_{(1,1)} & \cdots & p_{(1,n)} \\ \vdots & & \vdots \\ p_{(n,1)} & \cdots & p_{(n,n)} \end{bmatrix}_{N\times N} \tag{23}$$

where, $p_{(n,n)}$ is the occupancy indicator of the satellite resource pool in resource block (n, n).

2) action space

In the SCRS problem, the available action space $\boldsymbol{A}(\boldsymbol{s})$ is the decision space of the satellite control system, and action a_t is selected from $\boldsymbol{A}(\boldsymbol{s})$ based on the current state s_t at time t. The available operating space depends on the type of allocated resources and resource limitations, including priority action space $\boldsymbol{A}_{\mathrm{p}}$ and selection action space $\boldsymbol{A}_{\mathrm{c}}$. $\boldsymbol{A}(\boldsymbol{s})$ is defined as follows:

$$\boldsymbol{A}(\boldsymbol{s}) = \left\{ \left(A_{\mathrm{c}}(i), A_{\mathrm{p}}(j) \right) \mid 1 \leqslant i \leqslant m, j = 0, 1 \right\} \tag{24}$$

where, $\boldsymbol{a}_{\mathrm{c}} = A_{\mathrm{c}}(i)$ represents the task selection action, representing the task number selected in the list for this round; $\boldsymbol{a}_{\mathrm{p}} = A_{\mathrm{p}}(j)$ is the action of resource search priority.

3) reward function

For a task assignment of the same resource block size, the closer the resource occupancy rate is to the upper limit, the greater the reward value should be. Therefore, the reward function is defined as follows:

$$r = -\lg\left(1 - \frac{\sum_{m=1}^{M} \delta_m \times \varphi_1\left(t_m\right) \times \varphi_2\left(f_m\right)}{\Delta t \times \Delta f} + \varepsilon \right) \tag{25}$$

where, ε is a non-negative number used to avoid the occurrence of infinite values.

A. DESCRIPTION OF THE SIMULATION PLATFORM

With reference to the recommendation on LEO satellite parameter setting in 3GPP NTN TR38.811 [27], this paper assumes that LEO satellite works in Ka-band, satellite height is 600km, center frequency of downlink signal is 20GHz, total bandwidth is 400MHz, spectrum multiplexing between beams, and users are evenly distributed within the beam. All other parameters are shown in Table 1.

TABLE 1. Parameters setting of simulator.

Parameters type	Value
Satellite altitude	600 $[km]$
Satellite latitude and longitude	(30°E, 105°N)
Maximum beam power	30 $[W]$
Maximum beam bandwidth	40 $[MHz]$
Maximum number of transmission users in the channel	4
Number of single beam users	$1 \sim 50$
Individual user requirements	$0 \sim 20$ $[Mbit/s]$
Radiation gain of transmitting antenna	26 $[dBi]$
Radiation gain of receiving antenna	14 $[dBi]$
3dB angle	0.48°
Gaussian white noise power spectral density	-174 $[dBm/Hz]$

In addition, this paper uses two random generation methods to generate satellite task list datasets and trains deep reinforcement learning models. Firstly, in order to compare the performance of different scheduling methods, we adopt the zero waste task generation method. Secondly, in order to better match the actual situation of the satellite task list, we adopted the non-zero waste task generation method.

IV. NUMERICAL EXPERIMENTS

A. ALGORITHM PARAMETER SETTINGS

This paper compares three state-of-the-art deep reinforcement learning algorithms (SAC, DDPG, TD3) with the proposed DWSAC algorithm. The hyperparameter setting of DDPG is consistent with the reference [28], TD3 is consistent with the reference [29], and SAC is consistent with the

TABLE 2. Parameters setting of simulator.

Hyperparameter	Value
Optimizer	Adam
Value function parameters optimization learning rate	0.01
Strategy function parameters optimization learning rate	0.001
Discount rate (γ)	0.99
Experience cache size	$1e+6$
Number of samples per small batch	256
Entropy target	$-\dim(\mathcal{A})$
Activation function	ReLU
Target smoothing coefficient (τ)	0.005
Target update interval	1
Gradient step	1
Threshold for Actor updating weights	10
Threshold for Critic updating weights	10
Target network update frequency	300

reference [30]. The hyperparameter setting of the DWSAC proposed in this paper is shown in Table 2.

B. SIMULATION AND ANALYSIS IN SATELLITE RESOURCE SCHEDULING

Firstly, we analyze and evaluate the performance of the proposed DWSAC and the other three state-of-the-art algorithms in satellite resource scheduling based on the mean episode reward value. In Fig. 3 and Fig. 4, the convergence curves of reward value trained on the zero waste dataset and non-zero waste dataset are shown respectively. From the Fig. 3 and Fig. 4, it can be seen that although there are still significant fluctuation in the curve after 400 episodes, the trend of the curve is basically clear. DDPG receives the smallest reward, which means a lower utilization rate of satellite resources. Its curve has been fluctuating significantly throughout the entire training process, indicating the existence of value overestimation problem that lead to poor performance of reward values and difficulty in finding the optimal strategy. The average reward of TD3 and SAC is better than that of DDPG. Although the fluctuation amplitude of the curve has been alleviated, it is still very obvious, especially in TD3 due to the lack of target network mechanism and the non-uniformity of state value frequency in the experience cache, as well as the overestimation problem of the critic network, making it difficult to learn the optimal strategy. In addition, SAC has significant curve fluctuation due to its inability to efficiently evaluate the state values. The DWSAC proposed in this paper is significantly superior to the other three algorithms in terms of convergence speed, average reward, and curve stability, indicating that the dynamic weight update strategy can update the actor network and critic network with better quality based on value estimation, improving the efficiency and ability to find the optimal strategy, and enhancing robustness.

Secondly, based on zero waste and non-zero waste datasets, this paper conducted comparative experiments on task scheduling performance and running cost performance. As shown in Table 3 and Table 4, the effectiveness of the DWSAC is demonstrated, and the performance is compared with the other three algorithms.

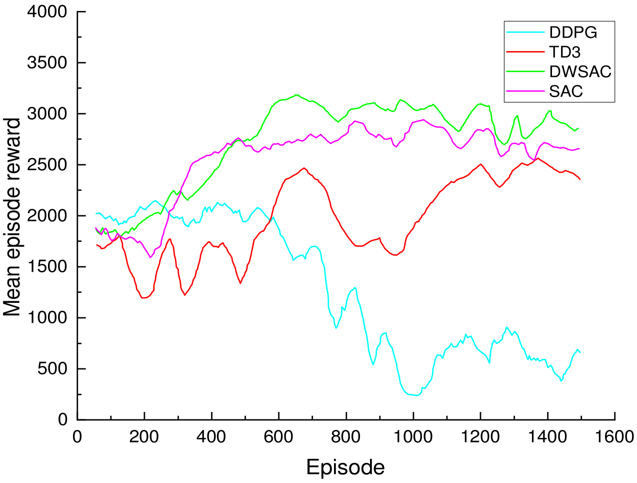


FIGURE 3. Mean episode reward at zero waste dataset.

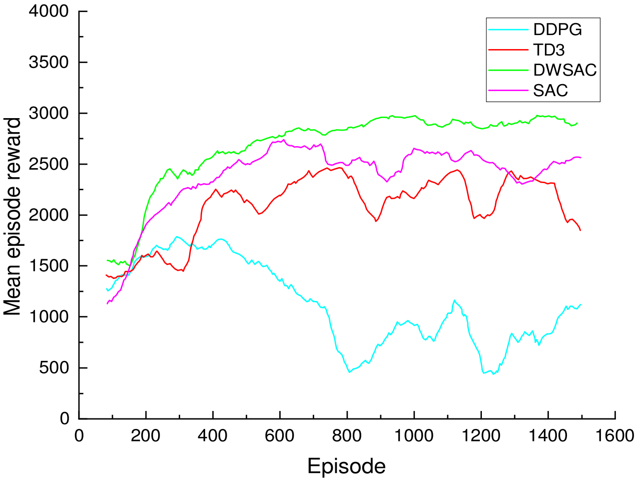


FIGURE 4. Mean episode reward at non-zero waste dataset.

From Table 3, it can be seen that compared with the three algorithms DDPG, TD3, and SAC, DWSAC showed significant improvements in average resource utilization rate (ARU) and running cost (RC) when tested on zero waste dataset. Compared with SAC, DWSAC can significantly improve ARU indicator within the allowable time limit, and with the increase of satellite missions, the performance improvement becomes more significant after adopting DWSAC. For example, when $M = 50$, the performance of DWSAC (96.7%) is significantly higher than that of SAC (94.9%), because the dynamic weight update strategy in DWSAC can update the network with better quality based on value estimation, improving the ability to find the optimal strategy. Especially in terms of time complexity, DWSAC (1.12%) can significantly reduce the running cost of the task scheduling process.

In order to evaluate the performance of DWSAC more comprehensively, this paper also conducts comparative experiments based on non-zero waste datasets. The

TABLE 3. Comparison of zero waste dataset.

Number of tasks	DDPG		TD3		SAC		DWSAC	
	$ARU/\%$	RC/s	$ARU/\%$	RC/s	$ARU/\%$	RC/s	$ARU/\%$	RC/s
$M = 20$	83.9	0.94	96.3	0.85	97.9	0.60	**98.1**	**0.51**
$M = 30$	87.1	1.11	94.6	0.91	95.4	0.82	**97.3**	**0.69**
$M = 40$	89.3	1.22	93.2	1.49	95.1	**0.98**	**95.9**	0.99
$M = 50$	92.6	1.30	92.8	1.56	94.9	1.49	**96.7**	**1.12**

TABLE 4. Comparison of non-zero waste dataset.

Number of tasks	DDPG		TD3		SAC		DWSAC	
	$ARU/\%$	RC/s	$ARU/\%$	RC/s	$ARU/\%$	RC/s	$ARU/\%$	RC/s
$M = 20$	76.9	1.03	84.9	0.63	85.7	0.59	**86.8**	**0.54**
$M = 30$	81.3	1.14	86.1	0.89	87.1	0.91	**88.3**	**0.81**
$M = 40$	83.1	1.23	87.2	1.21	88.1	1.14	**89.2**	**1.13**
$M = 50$	84.3	1.42	89.2	1.99	90.1	1.67	**92.6**	**1.32**

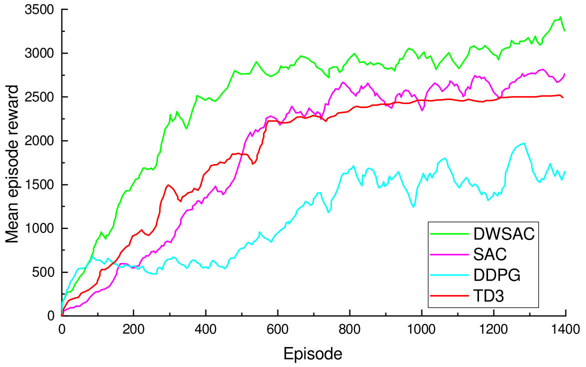


FIGURE 5. Mean episode reward of ***Hopper* − *v*2**.

experimental results are shown in Table 4, which are similar to those on the zero waste dataset. Since optimal allocation results cannot be obtained based on non-zero waste datasets, ARU can still be used to compare the performance of each algorithm. When $M = 50$, compared with DDPG, TD3, and SAC algorithms, DWSAC also achieved better performance in ARU (92.6%) and RC (1.32%).

C. SIMULATION AND ANALYSIS IN STANDARD CONTINUOUS CONTROL TASK

The simulation experiments in section IV-B shows that the DWSAC algorithm proposed in this paper has good performance in solving satellite resource scheduling problem. In order to further test whether the algorithm still has good performance when the control task changes, that is, the universality of the algorithm, six challenging continuous control tasks are selected from the OpenAI Gym standard test set and the Humanoid standard test set to test the proposed DWSAC and its comparative algorithms. These testing tasks specifically include $Hopper - v2$, $Walker2d - v2$, $HalfCheetah - v2$, $Ant - v2$, $Humanoid - v2$, and $Humanoid(rllab)$. The average reward curves for each standard testing task are shown in Fig.5 - Fig.10, respectively.

Fig.5 - Fig.10 shows the average reward curves of four algorithms, DWSAC, SAC, TD3, and DDPG, trained in different continuous control tasks. In each task, the four algorithms used for testing are evaluated once after each complete training to calculate the average reward. From the average reward curves of the Fig.5 - Fig.10, it can be seen that both in terms of algorithm learning convergence speed and final performance, DWSAC performs similarly to the three state-of-the-art algorithms on simpler tasks. However, on more difficult tasks, the performance of the other three algorithms lags behind DWSAC significantly. For example, DDPG has made little progress on $Ant - v2$, $Humanoid - v2$, and $Humanoid(rllab)$, especially in the latter two tasks. TD3 has also made little progress on $Humanoid - v2$ and $Humanoid(rllab)$. In terms of the stationarity of the curve in the Fig.5 - Fig.10, the proposed DWSAC is slightly better

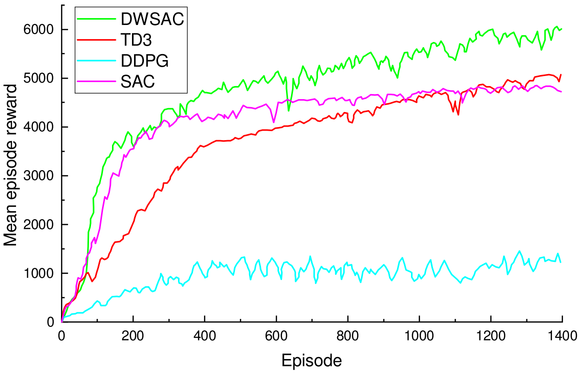


FIGURE 6. Mean episode reward of ***Walker*2*d* − *v*2**.

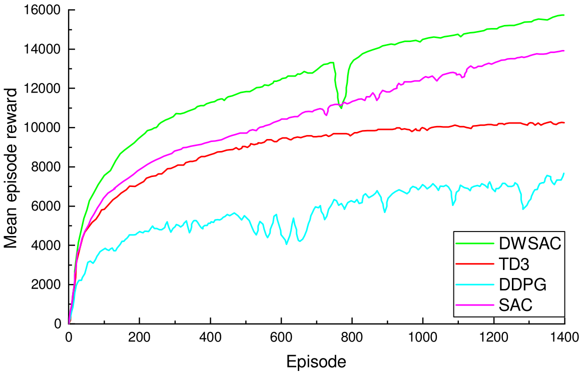


FIGURE 7. Mean episode reward of ***HalfCheetah* − *v*2**.

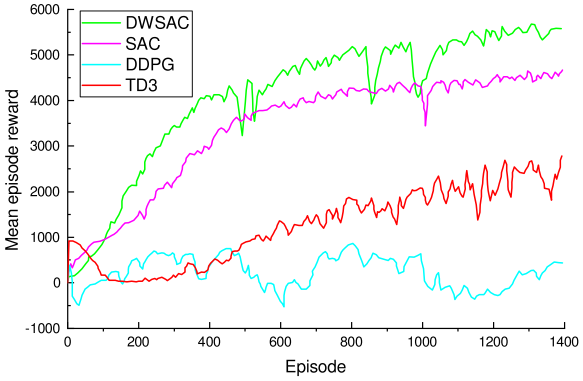


FIGURE 8. Mean episode reward of *Ant* − *v*2.

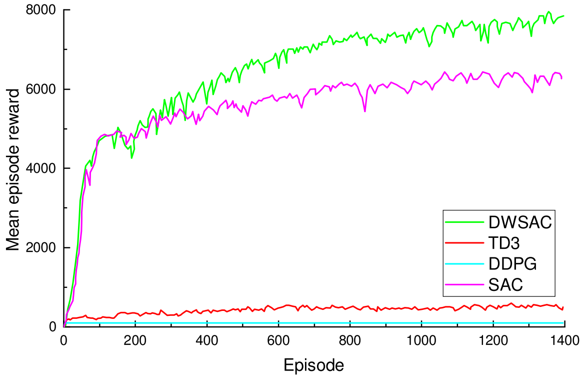


FIGURE 9. Mean episode reward of *Humanoid* − *v*2.

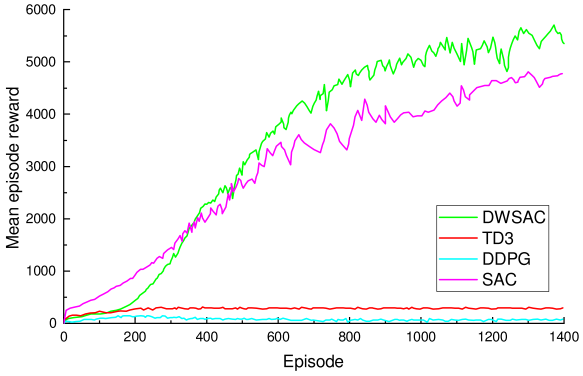


FIGURE 10. Mean episode reward of *Humanoid*(*rllab*).

than SAC and far better than TD3 and DDPG, indicating that the DWSAC has better robustness. From the Fig.5 - Fig.10, it can also be seen that the proposed algorithm has a higher learning rate than the original SAC algorithm. In addition, the quantitative results obtained by using DWSAC in this simulation are also better than the algorithms in [31], [32], and [33], indicating that DWSAC's learning efficiency and final performance on these standard tasks exceed several state-of-the-art algorithms.

V. CONCLUSION

With the increasing demand for satellite communication tasks, limited satellite resources cannot meet the needs of all users at the same time. At this time, meeting the needs as much as possible through satellite resource scheduling has become a key challenge. This paper adopts deep reinforcement learning method. Firstly, a joint state space of satellite task requirements and resource pool is proposed to obtain global information of the environment, avoiding convergence to a local optimal strategy. Secondly, a new joint partitioning method for frequency and time resources is proposed, which avoids the fragmentation of the resource to the maximum extent. Finally, DWSAC is proposed to overcome the shortcoming that SAC adopts a fixed learning rate, which makes it impossible for the agent to dynamically adjust the learning rate according to the change of the instant reward with time step. The dynamic weight mechanism added in the algorithm enhances the update range when the action taken by the agent is conducive to the improvement of system performance, otherwise it reduces the update range, which significantly improves the convergence efficiency and convergence performance of the algorithm.

In terms of simulation results, firstly, the proposed model and algorithm are used to simulate on the zero waste task dataset and the non-zero waste dataset respectively. The simulation data show that the proposed method shows the significant improvements in average resource utilization rate (ARU) and running cost (RC). Secondly, four algorithms were used to simulate six different difficulty tasks in OpenAI Gym and Humanoid. It can be seen from the average reward curve of the six tasks that DWSAC is superior to DDPG, TD3 and SAC in terms of learning efficiency, robustness and final performance.

In the future, we hope to treat each satellite as an independent agent and adopt multi-agent reinforcement learning method to solve the satellite resource scheduling problem. This method not only effectively reduces computational complexity, but also has strong scalability and is easier to learn the optimal strategy.

REFERENCES

[1] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T. X. Vu, and G. Goussetis, "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 70–109, 1st Quart., 2021.

[2] N. Chukhno, O. Chukhno, S. Pizzi, A. Molinaro, A. Iera, and G. Araniti, "Approaching 6G use case requirements with multicasting," *IEEE Commun. Mag.*, vol. 61, no. 5, pp. 144–150, May 2023.

[3] D. Yanan and P. Wenzhen, "Dynamic channel assignment algorithm for fusion beam coverage in GMR-1 satellite mobile communication system," *Mobile phone Lett.*, vol. 44, no. 9, pp. 40–57, 2021.

[4] D. Yi and C. Xinyu, "A feedback optimization access mechanism for satellite channel," *China Space Sci. Technol.*, vol. 40, no. 5, pp. 99–110, 2021.

[5] G. Jiani, *Research on Channel Allocation Strategies for Satellite Mobile Communication System Adapting to High Speed Terminal*. Beijing, China: Beijing Univ. Posts Telecommun., 2017, pp. 15–20.

[6] C. N. Efrem and A. D. Panagopoulos, "Dynamic energy-efficient power allocation in multibeam satellite systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 228–231, Feb. 2020.

[7] Q. Xue, X. Fang, M. Xiao, S. Mumtaz, and J. Rodriguez, "Beam management for millimeter-wave beamspace MU-MIMO systems," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 205–217, Jan. 2019.

[8] L. Wang, C. Zhang, D. Qu, and G. Zhang, "Resource allocation for beam-hopping user downlinks in multi-beam satellite system," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 925–929.

[9] A. Wang, L. Lei, E. Lagunas, S. Chatzinotas, A. I. P. Neira, and B. Ottersten, "Joint beam-hopping scheduling and power allocation in NOMA-assisted satellite systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[10] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Found. Trends Mach. Learn.*, vol. 16, no. 1, pp. 1–118, 2023.

[11] P. Yue, J. An, J. Zhang, J. Ye, G. Pan, S. Wang, P. Xiao, and L. Hanzo, "Low Earth orbit satellite security and reliability: Issues, solutions, and the road ahead," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1604–1652, 3rd Quart., 2023.

[12] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 1–14, Jul. 2023.

[13] X. Wang and H. Kong, "Q-learning based relay selection strategy for hybrid satellite-terrestrial cooperative transmission," *J. Appl. Sci.*, vol. 39, no. 2, pp. 250–260, 2021.

[14] Y. Zhou, F. Zhou, Y. Wu, R. Q. Hu, and Y. Wang, "Subcarrier assignment schemes based on Q-learning in wideband cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1168–1172, Jan. 2020.

[15] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.

[16] Y. Qiu, Z. Ji, Y. Zhu, G. Meng, and G. Xie, "Joint mode selection and power adaptation for D2D communication with reinforcement learning," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–6.

[17] F.-T. Chen, M. Huang, and Y.-F. Jin, "Resource allocation algorithm for low earth orbit satellites oriented to user demand," *J. Comput. Appl.*, vol. 44, no. 4, p. 1242, 2023.

[18] Z. Huaming and L. Qiang, "Downlink power allocation scheme for LEO satellites based on deep reinforcement learning," *J. Univ. Chin. Acad. Sci.*, vol. 39, no. 4, p. 543, 2022.

[19] Z. Tian, J. Wang, J. Wang, and J. Song, "Distributed NOMA-based multi-armed bandit approach for channel access in cognitive radio networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1112–1115, Aug. 2019.

[20] Q. Yang and R. Parasuraman, "A strategy-oriented Bayesian soft actor-critic model," *Proc. Comput. Sci.*, vol. 220, pp. 561–566, Aug. 2023.

[21] X. Tang, B. Huang, T. Liu, and X. Lin, "Highway decision-making and motion planning for autonomous driving via soft actor-critic," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4706–4717, May 2022.

[22] X. Zhao, S. Ding, Y. An, and W. Jia, "Applications of asynchronous deep reinforcement learning based on dynamic updating weights," *Int. J. Speech Technol.*, vol. 49, no. 2, pp. 581–591, Feb. 2019.

[23] J. Jiang, Z. Zhang, C. Xu, Z. Yu, and Y. Peng, "One forward is enough for neural network training via likelihood ratio method," 2023, *arXiv:2305.08960*.

[24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[25] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[26] H. Hasselt, "Double Q-learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[27] J. Ssimbwa, B. Lim, J.-H. Lee, and Y.-C. Ko, "A survey on robust modulation requirements for the next generation personal satellite communications," *Frontiers Commun. Netw.*, vol. 3, May 2022, Art. no. 850781.

[28] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[29] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," 2018, *arXiv:1811.12560*.

[30] Y. Zhang, Y. Zhou, H. Lu, and H. Fujita, "Traffic network flow prediction using parallel training for deep convolutional neural networks on spark cloud," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7369–7380, Dec. 2020.

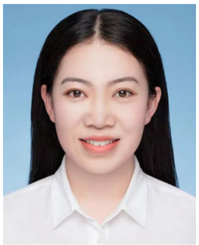
[31] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1329–1338.

[32] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, "Q-Prop: Sample-efficient policy gradient with an off-policy critic," 2016, *arXiv:1611.02247*.

[33] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–12.

ZHIMIN QIAO received the B.E. degree in automation from the North University of China, Taiyuan, China, in 2012, the M.S. degree in control engineering from Northeastern University, Shenyang, China, in 2015, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2021. He is currently a Lecturer with Taiyuan Institute of Technology, Taiyuan. His research interests include satellite resource scheduling, multi-agent reinforcement learning, and evolutionary computation.

WEIBO YANG received the Ph.D. degree in automation from Xi'an Jiaotong University, Xi'an, China. He is currently working as an Assistant Professor with Chang'an University. His research interests include combinatorial optimization, vehicle routing problems, and evolutionary computation.



FENG LI received the B.S. degree from Taiyuan Institute of Technology, Taiyuan, China, in 2011, and the M.S. degree in control science and engineering and the Ph.D. degree in control science and engineering from Tianjin University, Tianjin, China, in 2014 and 2021, respectively. She is currently working as an Associate Professor with Taiyuan Institute of Technology. Her research interests include process parameter detection, industrial process tomography, and deep learning.

YONGWEI LI received the B.S. degree from the North University of China, Taiyuan, China, in 2013, the M.S. degree in systems engineering from Northeasten University, Shenyang, China, in 2015, and the Ph.D. degree in instrument science and technology from the North University of China, in 2021. He is currently working as an Associate Professor with Taiyuan Institute of Technology, Taiyuan. His research interests include advanced sensors and test control systems.

YE ZHANG received the B.S. degree in electrical engineering from China University of Mining and Technology, China, in 2011, the M.S. degree from Tianjin University, China, in 2014, and the Ph.D. degree in electrical engineering from Taiyuan University of Technology, Taiyuan, China, in 2023. She is currently a Lecturer with Taiyuan Institute of Technology, Taiyuan. Her research interests include power quality analysis and control of power converters for renewable energy systems.