Handover for Multi-Beam LEO Satellite Networks: A Multi-Objective Reinforcement Learning Method

Yang Sun, *Member, IEEE*, Yuqing Zhai, Wenjun Wu, *Member, IEEE*, Pengbo Si, *Senior Member, IEEE*, and Fei Richard Yu, *Fellow, IEEE*

***Abstract*— In multi-beam low-earth orbit (LEO) satellite networks, frequent handovers between intra-satellite and inter-satellite beams are inevitable. In this letter, we design a beam handover strategy based on the multi-objective reinforcement learning (MORL) method to achieve seamless and effective handover between multiple beams of LEO satellites. We first model the handover optimization problem of the multi-beam LEO satellite networks as a multi-objective optimization (MOO) problem to jointly maximize throughput, minimize the handover frequency, and keep the network load balanced. On this basis, we convert the MOO problem into a multi-objective Markov decision process (MOMDP), and utilize an MORL method, called multi-objective deep Q-learning network (MODQN), to learn and achieve the optimal solution. Simulation results show the effectiveness and superiority of the proposed handover scheme.**

***Index Terms*— Multi-beam LEO satellite networks, beam handover, multi-objective reinforcement learning.**

I. Introduction

As a powerful supplement to the terrestrial networks, low-earth orbit (LEO) satellite networks can transcend the terrain limitations to achieve universal coverage around the world and have gradually attracted widespread attention [1]. The dynamic changes in LEO satellite network topology lead to the frequent triggering of handovers not only between intra-satellite beams but also between inter-satellite beams [2], [3]. Frequent beam handover will significantly increase service interruptions and affect the user's service experience.

Recently, most existing handover schemes mainly focused on inter-satellite handovers. Several researchers utilized the graph theory to optimize the handover decisions of LEO satellites [4], [5], [6]. These graph-based handover schemes generally abstracted the coverage relationship between satellites and users into a weighted directed graph, and found the shortest path to determine the user's optimal handover sequence with the aim of throughput maximization [4], quality of service (QoS) improvement [5], and network load balancing [6]. However, the graph construction of the above handover schemes was often based on a large amount of global information, which can't adapt to the dynamics of the LEO networks. With the development of artificial intelligence, deep reinforcement learning (DRL) has also been applied to handover optimization [7], [8], [9], [10]. Wang et al. simultaneously considered multiple handover factors and proposed a DRL-based handover scheme to make the handover decisions [7]. Yang et al. designed a deep Q-learning network (DQN) framework of adaptive learning rate with momentum to optimize the multi-metric handover problem [8]. Lee et al. developed a novel DRL-based handover protocol that simplified the handover process to minimize access delays and collision rates [9]. Liu et al. proposed a multi-agent successive hysteretic deep Q-learning algorithm to determine the handover decisions for LEO mega-constellations [10]. However, the above DRL-based handover schemes generally used the single-objective DRL algorithm based on the scalar reward of several performance indicators to find the optimal handover strategy for the LEO satellite networks with a known preference. These customized handover schemes are unable to clearly distinguish the trade-offs between multiple objectives, and can't be widely applied to different preference scenarios of LEO satellite networks.

In this letter, we comprehensively consider various performance metrics and propose an intelligent beam handover scheme based on the multi-objective reinforcement learning (MORL) method. The handover problem of multi-beam LEO satellite networks is modeled as a multi-objective optimization (MOO) problem with the aim of maximizing the throughput, and minimizing the handover frequency while keeping load balancing of networks. To optimize the multiple conflicting objectives and make the optimal handover decisions, we transform the MOO problem into a multi-objective Markov decision process (MOMDP) and propose a multi-objective deep Q-learning network (MODQN) algorithm to solve it. Simulation results show that the proposed handover scheme can achieve more effective and robust performance compared to other comparison schemes.

II. System Model and Problem Formulation

A. System Model

Here we consider a multi-beam LEO satellite network in which a LEO constellation with L satellites serving I users, as shown in Fig.1. The set of LEO satellites is denoted as $\mathbb{L} = \{1, 2, \ldots l, \ldots, L\}$. Each satellite has V beams, the set of which is denoted as $\mathbb{V} = \{1, 2, \ldots, v, \ldots, V\}$. For

Received 25 August 2024; revised 24 September 2024; accepted 26 September 2024. Date of publication 30 September 2024; date of current version 12 December 2024. This work was funded in part by the Natural Science Foundation of Beijing Municipality under Grant L212003, in part by the National Natural Science Foundation of China under Grant 62001011 and 62371014, in part by the Natural Science Foundation of Beijing Municipality under Grant L211002 and 4222002. The associate editor coordinating the review of this letter and approving it for publication was S. A. Tegos. *(Corresponding author: Wenjun Wu.)*

Yang Sun, Yuqing Zhai, Wenjun Wu, and Pengbo Si are with the School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China (e-mail: sunyang@bjut.edu.cn; zhaiyuqing@emails.bjut.edu.cn; wenjunwu@bjut.edu.cn; sipengbo@bjut.edu.cn).

Fei Richard Yu is with the Department of System and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Digital Object Identifier 10.1109/LCOMM.2024.3470890

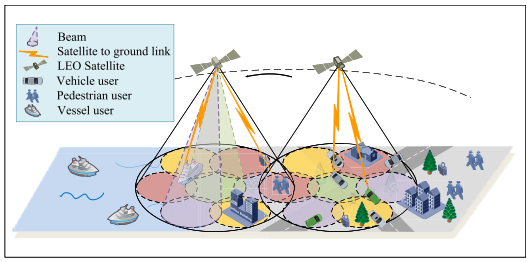

Fig. 1. System model of multi-beam LEO satellite networks.

convenience, we introduce a new set $\mathbb{K} = \mathbb{L} \times \mathbb{V}$ to define all the beams in the network, and the two-dimensional vector $\kappa = (l, v) \in \mathbb{K}$ denotes the v-th beam of satellite l. The set of users is denoted as $\mathbb{I} = \{1, 2, \ldots, i, \cdots, I\}$. We use a binary indicator $u_{i,l,v}(t)$ to express whether user i is connected with beam (l, v) or not. When user i requests to access beam (l, v) at time slot t, $u_{i,l,v}(t) = 1$, and otherwise, $u_{i,l,v}(t) = 0$. $\mathbf{u}_i(t) = [u_{i,1,1}(t), \ldots, u_{i,l,v}(t), \ldots, u_{i,L,V}(t)]$ represents the access vector between all beams and user i. Specifically, we use $\rho_i(t) = l \in \mathbb{L}$ and $\delta_i(t) = v \in \mathbb{V}$ to denote the satellite and the corresponding beam that user i accesses at time slot t, respectively.

The channel gain $G_{i,l,v}(t)$ between user i and beam (l, v) at time slot t is mainly composed of path loss, atmospheric fading and Rician small-scale fading, and can be expressed as

$$G_{i,l,v}(t) = \left(\frac{\xi}{4\pi d_{i,l,v}(t) f_c}\right)^2 \cdot A(d_{i,l,v}(t)) \cdot \eta, \tag{1}$$

where ξ and f_c are speeds of light and carrier frequency, η is Rician small-scale fading, $d_{i,l,v}(t)$ is the distance between satellite and user. $A(d_{i,l,v}(t)) = 10^{\left(\frac{3 d_{i,l,v}(t) \chi}{10 h_{l,v}(t)}\right)}$ is atmospheric fading [5], where χ is the attenuation through clouds and rain, $h_{l,v}(t)$ is the altitude of the satellite.

The signal-to-noise ratio (SNR) between user i and beam (l, v) is

$$\gamma_{i,l,v}(t) = \frac{p_{i,l,v} \cdot G_{i,l,v}(t)}{\sigma^2}, \tag{2}$$

where $p_{i,l,v}$ is the transmit power between user i and beam (l, v), σ^2 is the white Gaussian noise power.

According to Shannon formula, the transmission rate from beam (l, v) to user i in time slot t is given by

$$c_{i,l,v}(t) = \frac{B}{N_{l,v}(t)} \log_2(1 + \gamma_{i,l,v}(t)), \tag{3}$$

where B is the bandwidth and $N_{l,v}(t)$ is the number of users currently being served by beam (l, v).

Due to the high-speed movement of satellites, there are a large number of handovers that occur between intra-satellite and inter-satellite beams. The different types of handover result in different handover costs, which are defined as follows

$$\Psi_i(t) = \begin{cases} 0, & \rho_i(t) = \rho_i(t-1) \text{ and } \delta_i(t) = \delta_i(t-1) \\ \varphi_1, & \rho_i(t) = \rho_i(t-1) \text{ and } \delta_i(t) \neq \delta_i(t-1), \\ \varphi_2, & \rho_i(t) \neq \rho_i(t-1) \end{cases} \tag{4}$$

where φ_1 is the handover cost between beams within the same satellite, φ_2 is the handover cost between beams of different satellites, and we have $0 < \varphi_1 < \varphi_2$.

B. Problem Formulation

To maximize the throughput, minimize the number of handover events, and balance the loads among the beams, the handover optimization problem in multi-beam LEO satellite networks can be formulated as a dynamic MOO problem. We define P_1 as the maximization of long-term throughput for all users, P_2 as the minimization of handover cost, and P_3 as the minimization of the difference between the maximal and minimal throughput of beams which represents the degree of load balancing of the network. The handover optimization problem can be formulated as follows:

$$\begin{aligned} \text{opt. } P_1 &= \max_{\{\mathbf{u}_i(t)\}} \sum_{t=0}^{T-1} \sum_{(l,v)\in\mathbb{K}} \sum_{i\in\mathbb{I}} u_{i,l,v}(t) c_{i,l,v}(t) \\ P_2 &= \min_{\{\mathbf{u}_i(t)\}} \sum_{t=0}^{T-1} \sum_{(l,v)\in\mathbb{K}} \sum_{i\in\mathbb{I}} \Psi_i(t) \\ P_3 &= \min_{\{\mathbf{u}_i(t)\}} \sum_{t=0}^{T-1} \left(\begin{array}{l} \max\limits_{(l,v)\in\mathbb{K}} \sum\limits_{i\in\mathbb{I}} u_{i,l,v}(t) c_{i,l,v}(t) - \\ \min\limits_{(l,v)\in\mathbb{K}} \sum\limits_{i\in\mathbb{I}} u_{i,l,v}(t) c_{i,l,v}(t) \end{array} \right), \\ \text{s.t. } C1: & \; u_{i,l,v}(t) \in \{0,1\}, \quad \forall i \in \mathbb{I}, \forall (l,v) \in \mathbb{K} \\ C2: & \sum_{(l,v)\in\mathbb{K}} u_{i,l,v}(t) = 1, \quad \forall i \in \mathbb{I}, \forall (l,v) \in \mathbb{K} \end{aligned} \tag{5}$$

where the constraints $C1$ and $C2$ denote each user should select and can only select no more than one beam for access. Due to the dynamics of the networks, seeking the optimal handover strategy for long-term benefits based on traditional optimization methods may result in significant computational costs. MORL is committed to pursuing the simultaneous optimization of multiple conflicting objectives and has unique advantages in solving the dynamic MOO problems.

III. Handover Scheme Based on Multi-Objective Deep Q-Learning Network

In this section, we first transform the multi-objective handover optimization problem into an MOMDP model, and utilize the MORL method, called MODQN algorithm, to learn and approach the optimal handover decisions.

A. MOMDP Model

The principal structure of MORL is MOMDP, which can be described by a 3-tuple $\left\langle \mathbb{S}, \mathbb{A}, \overrightarrow{R} \right\rangle$, where $\mathbb{S}$, $\mathbb{A}$ and $\overrightarrow{R}$ are the state space, action space and reward vector, respectively.

1) State Space: Here we treat each user as an agent, each user i can locally observe the state information from the environment. The state $s_t^i \in \mathbb{S}$ can be defined as

$$s_t^i = (\mathbf{u}_i(t), \mathbf{G}_i(t), \mathbf{\Gamma}(t), \mathbf{N}(t)), \tag{6}$$

where $\mathbf{G}_i(t) = [G_{i,1,1}(t), \ldots, G_{i,l,v}(t), \ldots, G_{i,L,V}(t)]$ denotes the channel quality between all beams and user i, $\mathbf{\Gamma}(t) = [\Gamma_{1,1}(t), \ldots, \Gamma_{l,v}(t), \ldots, \Gamma_{L,V}(t)]$ denotes the location of all beams, and $\mathbf{N}(t) = [N_{1,1}(t), \ldots, N_{l,v}(t), \ldots, N_{L,V}(t)]$ denotes the number of users served by each beam.

2) Action Space: We use the access vector to represent the action $a_t^i \in \mathbb{A}$ of user i at time slot t, which can be defined as

$$a_t^i = \left\{ u_{i,1,1}, \ldots, u_{i,l,v}, \ldots, u_{i,L,V} \,\middle|\, \begin{array}{l} \sum\limits_{(l,v)\in\mathbb{K}} u_{i,l,v} = 1, \\ u_{i,l,v} \in \{0,1\} \end{array} \right\}. \tag{7}$$

3) Reward: Unlike the single-objective DQN, MODQN uses a vector of rewards $\overrightarrow{R}_t^i \in \overrightarrow{R}$ instead of a scalarization function with respect to the three objectives, which can be defined as

$$\overrightarrow{R}_t^i = \left[r_{1,t}^i, r_{2,t}^i, r_{3,t}^i \right], \tag{8}$$

where $r_{1,t}^i$, $r_{2,t}^i$, $r_{3,t}^i$ are the rewards of throughput, handover cost and load balancing, respectively.

To improve the quality of user data services as much as possible, the throughput reward of user i is defined as

$$r_{1,t}^i = \sum_{(l,v)\in\mathbb{K}} u_{i,l,v}(t) \cdot c_{i,l,v}(t). \tag{9}$$

Frequent handovers may lead to significant signaling interactions and transmission interruptions, so the handover cost reward of user i is defined as

$$r_{2,t}^i = \begin{cases} 0, & \rho_i(t) = \rho_i(t-1) \text{ and } \delta_i(t) = \delta_i(t-1) \\ -\varphi_1, & \rho_i(t) = \rho_i(t-1) \text{ and } \delta_i(t) \neq \delta_i(t-1). \\ -\varphi_2, & \rho_i(t) \neq \rho_i(t-1) \end{cases} \tag{10}$$

To achieve the fairness scheduling of the beams, here we describe the load balancing performance of the networks by using the difference between the maximal and minimal throughput of beams, which is defined as follows

$$r_{3,t}^i = -\frac{1}{I} \left(\begin{array}{l} \max\limits_{(l,v)\in\mathbb{K}} \sum\limits_{i\in\mathbb{I}} u_{i,l,v}(t) c_{i,l,v}(t) - \\ \min\limits_{(l,v)\in\mathbb{K}} \sum\limits_{i\in\mathbb{I}} u_{i,l,v}(t) c_{i,l,v}(t) \end{array} \right). \tag{11}$$

B. MODQN

MODQN is a generalization of standard single-objective DQN and is suitable to optimize multiple conflicting objectives. The main idea of MODQN is to establish a separate DQN network for each objective and find the optimal solution by scalarizing the action-value functions of multiple DQNs.

The framework of the proposed MODQN is shown in Fig. 2. In MODQN, the agent comprises three parallel training DQN networks, each has an action-value function $Q(s_t, a_t)$ that represents the expected discounted reward corresponding to each objective. The action-value vector of three DQN networks can be expressed as $\overrightarrow{Q}(s_t, a_t) = [Q_1(s_t, a_t), Q_2(s_t, a_t), Q_3(s_t, a_t)]$. Since the optimization problem has multiple conflicting objectives, MODQN needs to find the optimal action that maximizes the benefit between multiple objectives in each step. Here we use the linear scalarization method [11] to make the agent choose a single action to perform. Let $\Omega = [\omega_1, \omega_2, \omega_3]$ be the weight vector of the optimization objectives, we use a scalarized action-value

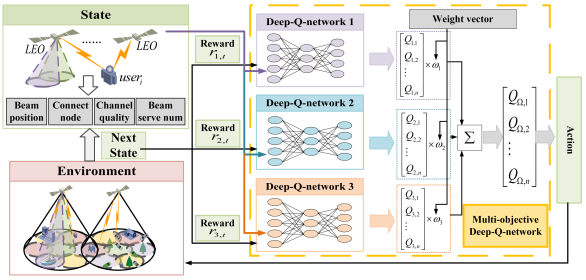

Fig. 2. The framework of the proposed MODQN.

$\zeta(s_t, a_t)$ to score the action of multiple objectives, which can be calculated by

$$\zeta(s_t, a_t) = \Omega^{\mathrm{T}} \bullet \overrightarrow{Q}(s_t, a_t) = \sum_{j=1}^{3} \omega_j Q_j(s_t, a_t). \tag{12}$$

In each step, the agent can select the action a_t^i by using the ϵ-greedy strategy:

$$a_t^i = \arg\max_{a\in\mathbb{A}} \zeta(s_t^i, a). \tag{13}$$

When the agent performs an action in MODQN, the environment will return a reward vector $\overrightarrow{R}_t^i = \left[r_{1,t}^i, r_{2,t}^i, r_{3,t}^i \right]$. The agent stores $\left(s_t^i, a_t^i, r_{j,t}^i, s_{t+1}^i \right)$ of each objective j in the replay memory individually. The DQN network of each objective is trained and optimized independently. Wherein, the training process of each DQN involves two neural networks, i.e., the evaluation network and the target network. The action-value function of evaluation network $Q_j(s_t, a_t; \theta_j)$ is updated every iteration, while $Q_j\left(s_t, a_t; \theta_j^-\right)$ of the target network is updated periodically at specified episodes, where θ_j, θ_j^- are learnable parameters of evaluation and target network, respectively. In each iteration, each DQN is trained by randomly sampling a batch from the corresponding replay memory.

The update formula for the network parameter θ_j under each objective is as follows

$$\theta_j = \theta_j + \alpha \left(r_{j,t} + \beta \max_{a_t'} Q_j\left(s_t', a_t'; \theta_j^-\right) - Q_j\left(s_t, a_t; \theta_j\right) \right) \nabla Q_j\left(s_t, a_t; \theta_j\right). \tag{14}$$

where α is the learning rate, and β is the discount factor.

The loss function can be calculated by using the mean square error after the DQN network has been trained, which is defined as follows

$$L(\theta_j) = E\left[\left(y_{j,t} - Q_j\left(s_t, a_t; \theta_j\right) \right)^2 \right], \tag{15}$$

where $y_{j,t}$ is the temporal difference (TD) target and is expressed as

$$y_{j,t} = \begin{cases} r_{j,t}, & \text{if } t+1 \text{ is terminal} \\ r_{j,t} + \beta \max\limits_{a_{t+1}\in A} Q_j^-\left(s_{t+1}, a_{t+1}; \theta_j^-\right), & \text{else} \end{cases}. \tag{16}$$

We employ the Adam optimization algorithm to train each DQN network to minimize the loss function. The detailed process of the proposed MODQN is presented in Algorithm 1.

Algorithm 1 MODQN Algorithm

Initialize three DQN networks and replay memory $M_1 M_2 M_3$.
for each episode h **do**
  for each step t **do**
    Select action a_t^i with ϵ-greedy strategy;
    Update next state s_{t+1}^i and obtain reward vector $\overrightarrow{R}_t^i$;
    for each objective j **do**
      Store $\left(s_t^i, a_t^i, r_{j,t}^i, s_{t+1}^i\right)$ in replay memory M_j.
    end for
    for each objective j **do**
      Sample a batch from the replay memory;
      Calculate the loss function $L\left(\theta_{j,t}\right)$;
      Train the evaluation and target neural networks by using the Adam optimizer;
      Update θ_j and θ_j^- according to Eq.(14).
    end for
  end for
end for

C. Complexity Analysis

Here we provide a comparison of the computational complexity between traditional DQN and MODQN for solving MOO problems. Traditional DQN generally transforms the MOO problem into a single-objective optimization problem by using a scalar reward in the MDP model. The computational complexity of traditional DQN training is $\mathcal{O}\left(TH\sum_{z=1}^{Z-1} x_z x_{z+1}\right)$, where T, H represent the numbers of time steps and training episodes, Z is the number of neural network layers, x_z is the input size of the z-th layer. When the preference among objectives changes frequently, traditional DQN needs to be retrained many times which will cause a large amount of time and resource consumption. In contrast, MODQN optimizes multiple objectives simultaneously and can adapt to different preferences through one training. The computational complexity can be expressed as $\mathcal{O}\left(JTH\sum_{z=1}^{Z-1} x_z x_{z+1}\right)$, where J is the number of objectives.

IV. Simulation Result and Analysis

In this section, we evaluate the performance of the proposed handover scheme through extensive simulations using the Python 3.8 platform. We construct an LEO constellation based on System Tool Kit (STK), each satellite runs in the given orbit at a fixed speed. Multiple users are randomly distributed within a 200km×90km rectangular area centered on $(40°N, 116°E)$, and each user's movement adopts the random wandering mode. To verify the performance of the proposed MODQN algorithm under different preferences and network conditions, here we set a default simulation scenario in which the numbers of users and satellites are 100 and 4, and the speeds of the users and satellites are 30km/h and 7.4km/s. The neural networks of the three objectives in MODQN have the same structure which consists of three hidden layers with 100, 50, and 50 neurons, respectively. The tanh function is adopted as the activation function. The detailed parameters are listed in Table I. Moreover, several handover schemes are considered as the benchmarks for performance comparison with the proposed MODQN algorithm: *RSS_max* selects the beam with the best channel quality; *DQN_throughput* uses the traditional DQN that only maximizes throughput; *DQN_scalar* uses the traditional DQN with a weighted reward of three objectives [7].

TABLE I
Simulation Parameters

Parameters	Values	Parameters	Values
h	780 km	p	2W
V	7	f_c	20 GHz
B	500 MHz	σ^2	-174 dBm/Hz
φ	20 dB [5]	χ	0.05 dB/km [5]
Length of time slot	1s	α	0.01
Duration of each episode	10s	β	0.9
Batch size	128	Episode	9000

TABLE II
Rewards Under Different Weights

Weight	$\sum r_1$	$\sum r_2$	$\sum r_3$	$\sum\sum r$
$[1,1,1]$	29.89	-7.89	-2.51	19.49
$[1,0,0]$	30.13	-7.28	-4.49	18.36
$[0,1,0]$	29.17	-6.34	-2.46	20.37
$[0,0,1]$	28.88	-8.62	-1.48	18.78
$[0.5,0.3,0.2]$	32.96	-7.66	-2.91	22.39
$[0.5,0.2,0.3]$	30.79	-8.78	-2.50	19.51
$[0.3,0.5,0.2]$	28.62	-5.85	-2.35	20.42
$[0.2,0.5,0.3]$	28.18	-6.53	-2.40	19.25
$[0.4,0.4,0.2]$	30.37	-7.10	-3.18	20.09
$[0.4,0.2,0.4]$	29.06	-10.52	-1.45	17.09
$[0.2,0.4,0.4]$	28.62	-7.82	-2.21	18.59

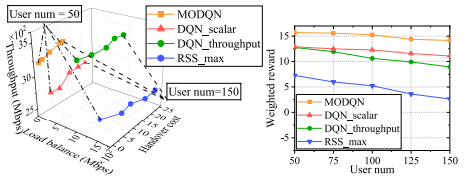

Fig. 3. Performance comparison under different user numbers. (a) Detailed performance of three objectives. (b) Weighted reward.

To evaluate the impact of the preferences among objectives, we compare the rewards corresponding to each objective under different weights in Table II. Specifically, $\sum r_1$, $\sum r_2$, $\sum r_3$ represent the rewards of throughput, handover cost and load balancing, respectively. $\sum\sum r$ is the sum of rewards for the three objectives. $[\omega_1, \omega_2, \omega_3] = [1,1,1]$ is used as the baseline for the evaluation. As shown in Table II, when $[\omega_1, \omega_2, \omega_3] = [0.5, 0.3, 0.2]$, the trade-off among three objectives is well balanced and the overall performance is significantly improved.

Based on the default simulation scenario, we set the weight to $[\omega_1, \omega_2, \omega_3] = [0.5, 0.3, 0.2]$ and give a comprehensive performance evaluation of the proposed algorithm under different network conditions by changing a single network parameter at a time. Fig. 3 and Fig. 4 give the performance comparisons under different numbers of users and satellites. Fig. 3(a) and Fig. 4(a) provide three-dimensional diagrams of detailed performances of three objectives under different numbers of users and satellites. Clearly, as the number of users or satellites rises, both throughput and handover cost increase, and load

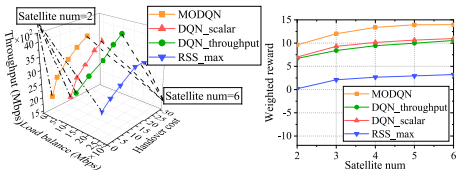

Fig. 4. Performance comparison under different satellite numbers. (a) Detailed performance of three objectives. (b) Weighted reward.

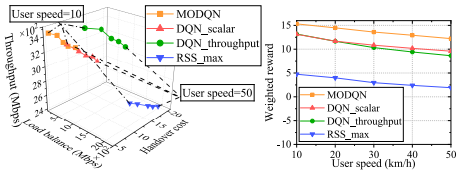

Fig. 5. Performance comparison under user speeds. (a) Detailed performance of three objectives. (b) Weighted reward.

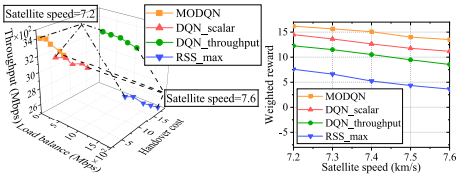

Fig. 6. Performance comparison under different satellite speeds. (a) Detailed performance of three objectives. (b) Weighted reward.

balancing performance deteriorates. Specifically, compared with *RSS_max* and *DQN_throughput* that support a single handover criterion, *DQN_scalar* and the proposed MODQN can achieve more effective and balanced improvements in the performance of all three objectives. To assess the overall performance, we further compare the weighted sum of the average rewards for the three objectives which is referred to as the weighted reward for brevity. It can be observed from Fig. 3(b) and Fig. 4(b) that the weighted reward decreases as the number of users increases, while the weighted reward increases as the number of satellites increases. No matter how the number of users or satellites changes, the weighted reward of the proposed MODQN is notably superior to other comparison schemes.

Fig. 5 and Fig. 6 show the performance comparisons under different user speeds and satellite speeds. We can infer from Fig. 5(a) and Fig. 6(a) that the increase in user and satellite speeds can accelerate the changes in the coverage of satellite beams and increase the possibility of handover occurrences between satellite beams, which leads to a reduction of the performance of the three objectives. As shown in Fig. 5(b) and Fig. 6(b), the weighted reward decreases with the increase of user and satellite speeds. It can be seen that the detailed performance and the weighted reward of three objectives of the proposed MODQN are better than those of *RSS_max* and *DQN_scalar*. Although *DQN_throughput* is slightly better than the proposed MODQN in terms of throughput, but underperforms in handover cost and load balancing. The proposed MODQN can always achieve better performance compared with other comparison schemes under different network conditions, which also verifies the robustness and effectiveness of the proposed algorithm.

V. Conclusion

In this letter, an intelligent beam handover strategy based on the MODQN algorithm was proposed for multi-beam LEO satellite networks to reduce the handover frequency and improve the network performance. We modeled the handover problem as an MOO problem and transformed it into an MOMDP model. We utilized the MODQN algorithm with the scalarization method to learn and achieve optimal handover decisions. Simulation results showed that the proposed scheme can better optimize and balance multiple objectives, and can adapt to different performance preference scenarios with lower computational and storage resource costs. It should be noted that the proposed MODQN algorithm can also be suitable for the handover optimization of terrestrial networks or space networks. In the future, we expect to combine MODQN with distributed learning to further explore handover strategies with low complexity for large-constellation LEO networks.

References

[1] X. Lin, S. Cioni, G. Charbit, N. Chuberre, S. Hellsten, and J.-F. Boutillon, "On the path to 6G: Embracing the next wave of low earth orbit satellite access," *IEEE Commun. Mag.*, vol. 59, no. 12, pp. 36–42, Dec. 2021.

[2] J. Li et al., "Collaborative ground-space communications via evolutionary multi-objective deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, early access, Sep. 12, 2024, doi: 10.1109/JSAC.2024.3459029.

[3] J. Zhu, Y. Sun, and M. Peng, "Beam management in low earth orbit satellite networks with random traffic arrival and time-varying topology," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13352–13367, Sep. 2024, doi: 10.1109/TVT.2024.3393924.

[4] X. Lv, S. Wu, A. Li, J. Jiao, N. Zhang, and Q. Zhang, "A weighted graph-based handover strategy for aeronautical traffic in LEO SatCom networks," *IEEE Netw. Lett.*, vol. 4, no. 3, pp. 132–136, Sep. 2022.

[5] M. Hozayen, T. Darwish, G. K. Kurt, and H. Yanikomeroglu, "A graph-based customizable handover framework for LEO satellite networks," in *Proc. IEEE Globecom Workshops*, Dec. 2022, pp. 868–873.

[6] H. Chen, G. Nie, and H. Tian, "A multi-slot load balancing scheme for LEO satellite communication handover target selection," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2024, pp. 1–6.

[7] J. Wang, W. Mu, Y. Liu, L. Guo, S. Zhang, and G. Gui, "Deep reinforcement learning-based satellite handover scheme for satellite communications," in *Proc. 13th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Changsha, China, Dec. 2021, pp. 1–6.

[8] J. Yang, Z. Xiao, H. Cui, J. Zhao, G. Jiang, and Z. Han, "DQN-ALrM-based intelligent handover method for satellite-ground integrated network," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 4, pp. 977–990, Aug. 2023.

[9] J.-H. Lee, C. Park, S. Park, and A. F. Molisch, "Handover protocol learning for LEO satellite networks: Access delay and collision minimization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7624–7637, Jul. 2024.

[10] H. Liu, Y. Wang, P. Li, and J. Cheng, "A multi-agent deep reinforcement learning based handover scheme for mega-constellation under dynamic propagation conditions," *IEEE Trans. Wireless Commun.*, early access, Jun. 6, 2024, doi: 10.1109/TWC.2024.3407358.

[11] T. Tajmajer, "Modular multi-objective deep reinforcement learning with decision values," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2018, pp. 85–93.