Towards Beam Hopping and Power Allocation in Multi-Beam Satellite Systems With Parameterized Reinforcement Learning

Yongyi Ran, *Member, IEEE*, Feng Tan, Shuangwu Chen, *Member, IEEE*, Jizhao Lei, and Jiangtao Luo, *Senior Member, IEEE*

***Abstract*—The simultaneous optimisation of beam hopping and power allocation is a crucial technique for enhancing the performance of Multi-Beam Satellite (MBS) systems. However, the previous joint optimisation approaches cannot well handle with the issues of high-dimensional state space and discrete-continuous hybrid action space. In this paper, we propose a joint optimization approach based on parameterized reinforcement learning to simultaneously regulate beam hopping and power allocation for MBS systems (called DeepMBS). In DeepMBS, a multi-objective problem is firstly formulated to optimize system throughput and energy efficiency. Then, the optimization problem is modelled as a Markov Decision Process (MDP), and the original deep Q-network is extended with a parameterized action space to simultaneously determine the beam hopping (discrete action) and power allocation (continuous action). In addition, we design an empirical filtering mechanism to enhance the performance of DeepMBS. Finally, the results of extensive experiments demonstrate that the proposed DeepMBS can gain a better performance in terms of throughput and energy efficiency compared to the baseline algorithms. Furthermore, the proposed DeepMBS (EFM) algorithm demonstrates superior accuracy and sensitivity in capturing changes of communication demands.**

***Index Terms*—Multi-beam satellite, deep reinforcement learning, parameterized action space, beam hopping, power allocation.**

I. Introduction

Satellite communication systems serve as a complement to terrestrial communication systems, playing a vital role in 6 G space-air-ground network [1], [2], [3]. Specially, multi-beam satellites have the characteristics of wide coverage, large communication capacity and flexible resource scheduling [1]. Whereas, the traffic in the coverage area is geographically non-uniform and time-varying. Beam Hopping (BH) is a key technology in MBS to solve this problem [4]. However, if beams are illuminated with the same power, the provided transmission capacity will still fail to match the uneven traffic among beams [5]. Also, the on-board power of a satellite is extremely scarce. Therefore, the power consumption of beams also should be well regulated along with BH.

However, it is challenging to jointly regulate beam hopping and the limited on-board power to meet the uneven transmission demands in MBS. First, not only the throughput but also the energy efficiency should be maximized to overcome the scarcity of power and avoid energy waste. Second, this joint paradigm will increase the dimension of system state space and action space, thus resulting in "curse of dimensionality". Third, such joint paradigm lead to a discrete-continuous hybrid action space [6], where the action variable of beam hopping is discrete while that of power allocation is continuous. Approximating the continuous action by an finite discrete set will lose the natural structure of the continuous action, while relaxing the discrete action into a continuous set will significantly increase the complexity of the action space.

Many research has been devoted to optimising beam hopping and power allocation in MBS to improve the performance of MBS systems. The works [7] and [8] have been devoted to the study of the beam hopping problem and the power allocation problem of MBS, respectively. Takahashi et al. [8] has proposed a new mathematical model for balancing the relationship between beam pointing and beam power to improve the utilisation of satellite communication resources, demonstrating the necessity and advantages of joint optimisation scheme of beam pointing and transmit power over unidirectional optimisation scheme, but the paper did not consider the beam hopping problem of MBS. Based on this, Wang et al. [9] proposed a joint optimization algorithm to control power allocation, beam scheduling, and terminal-timeslot assignment for the coexisted BH-NOMA systems, and developed a bounding scheme to tightly gauge the global optimum. Du et al. [10] developed a Deep Reinforcement Learning (DRL) based strategy combining power allocation and beam hopping (PABH), which improves power utilization and satellite throughput to a certain extent, and proves that power allocation is feasible. However, the previous non-learning-based methods are difficult to well capture the dynamics of the MBS systems, while the existing DRL-based methods cannot well deal with the issue of hybrid action space.

To address the above issues, we propose a novel joint optimization algorithm to concurrently regulate beam-hopping and power allocation for MBS based on parameterized DRL, named DeepMBS. To maximize the total throughput as well as the power efficiency, a multi-objective problem is firstly formulated. Then, to alleviate "curse of dimensionality" and well catch the dynamics of the MBS, a Markov Decision Process (MDP) is modeled for each beam and the optimization problem is then solved by using DRL in a polling paradigm. In order to tackle the issue of discrete-continuous hybrid action space, we extend to the original DRL (exactly deep Q-network (DQN) [7]) with the parameterized action space, which can simultaneously determine the beam hopping (discrete action) and power allocation (continuous action) without approximation or relaxation. In addition, in order to speed up the convergence of DeepMBS, an experience filtering mechanism is designed to store more suitable experiences for algorithm training. The experimental results illustrate that DeepMBS can outperform the baseline algorithms [7], [9], [10] in terms of throughput and energy efficiency in MBS. Furthermore, the proposed DeepMBS (EFM) algorithm demonstrates superior accuracy and sensitivity in capturing the dynamic changes of communication demands.

II. System Model and Problem Formulation

A. System Model

The paper investigates the multi-beam satellite downlink in the Ka-band, where the satellite is equipped with phased array antennas capable of simultaneously providing up to K beams. Assume that the reference

Manuscript received 21 December 2023; revised 14 March 2024; accepted 6 April 2024. Date of publication 8 May 2024; date of current version 19 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U23A20275, Grant 62003067, Grant 62101525, and Grant 62171072, and in part by the Natural Science Foundation of Chongqing under Grant cstc2021jcyj-msxmX0586. The review of this article was coordinated by Dr. Hongliang Zhang. *(Corresponding authors: Feng Tan; Jiangtao Luo.)*

Yongyi Ran, Feng Tan, and Jiangtao Luo are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 404100, China (e-mail: ranyy@cqupt.edu.cn; 2016215052@stu.cqupt.edu.cn; luojt@cqupt.eud.cn).

Shuangwu Chen is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: chensw@ustc.edu.cn).

Jizhao Lei is with the China Satellite Network Group Company, Ltd., Chongqing 401147, China (e-mail: xdljz2000@163.com).

Digital Object Identifier 10.1109/TVT.2024.3395509

geosynchronous orbit (GEO) MBS system is time-slotted, the index of time slots is t, and the duration of each time slot is d. The set of satellite beams is denoted as $\mathcal{K} = \{k|k = 1, 2, \ldots, K\}$. Herein the coverage area is assumed to be fixed within a time slot, and can be divided into a set of cells, represented as $\mathcal{N} = \{n|n = 1, 2, \ldots, N\}$ $(K \ll N)$. The K beams provide transmission service to N cells in a time-division multiplexing manner, and one beam can only serve one cell at a time. Full Frequency Reuse (FFR) is used between beams with the same bandwidth B_c.

1) Traffic Model: Due to the time division multiplexing mechanism, the satellite holds a buffer for each cell to cache the data packets to be transmitted. Specifically, it is assumed that the number of packets newly requested by cell n during t-th time slot is ρ_t^n. The number of packets in the buffer to be transmitted for cell n during t-th time slot is ϕ_t^n, so the matrix of the number of packets to be transmitted for all cells during t-th time slot is $\mathbf{\Phi}_t = [\phi_t^n | n \in \mathcal{N}]$. Given the limited satellite payload capacity and the timeliness of data services, we assume that the satellite only store the data packets requested in the most recent z_{th} time slots and discard the earlier data.

2) Channel Model: According to ITU-R S.672-4 [10], the downlink loss matrix $\mathbf{H} = \{h^{k,n} \mid k \in \mathcal{K}, n \in \mathcal{N}\}$ from the on-board transmitter to the user receiver can be calculated as

$$\mathbf{H}=\mathbf{\Theta} \cdot \mathbf{G_u} \cdot \mathbf{G_B} \tag{1}$$

where $\mathbf{\Theta} = diag\{\sigma_1, \sigma_2, \ldots, \sigma_N\}$ indicates the channel gain matrix, $\mathbf{G_B} = \{g_{k,n}^b \mid k \in \mathcal{K}, n \in \mathcal{N}\}$ denotes the transmit antenna gain matrix from the beam k to cell n, and $\mathbf{G_U} = \mathrm{diag}\{g_1^u, g_2^u, \ldots, g_N^u\}$ stands for the receive antenna gain matrix of the corresponding N cells.

3) Transmission Model: If cell n is illuminated by beam k, the signal-to-noise ratio from beam k to cell n is

$$\Gamma^{k,n} = \frac{h^{k,n} \cdot P_k}{B_c N_0 + \sum_{i \in \mathcal{K}, i \neq n} h^{i,n} \cdot P_i} \tag{2}$$

where $h^{k,n} \in \mathbf{H}$ denotes the loss from beam k to cell n, P_k is the transmit power of beam k, N_0 is the power spectral density of noise. According to the DVB-S2 standard, the channel capacity can be expressed as

$$C_t^{k,n} = x_t^{k,n} \cdot B_c \cdot f_{DVB}(\Gamma_t^{k,n}) \tag{3}$$

where $x_t^{k,n}$ denotes whether the beam k illuminates the cell n, if yes, $x_t^{k,n} = 1$, otherwise, $x_t^{k,n} = 0$. Also, $\sum_{n \in \mathcal{N}} x_t^{k,n} = 1, \forall k \in \mathcal{K}$. f_{DVB} is the performance mapping function [10]. Then, we can obtain the amount of data actually transmitted to cell n during time slot t:

$$\wp_t^n = \min\{C_t^n d, \phi_t^n\} \tag{4}$$

where $C_t^n = C_t^{k,n}$, $k = \arg_k\{x_t^{k,n} = 1\}$.

B. Problem Formulation

The objective of DeepMBS is to enhance system throughput and energy efficiency by controlling the beam illumination cell (i.e., n) and the transmission power of beams (i.e., P_k). Its objective function can be expressed as follows:

$$\text{opt. } P1 = \max \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \wp_t^n$$

$$P2 = \max \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{\sum_{n \in \mathcal{N}} \wp_t^n}{\sum_{k \in \mathcal{K}} P_{k,t} \cdot d}$$

$$s.t.\ \ \mathrm{C1}: \sum\nolimits_{k \in \mathcal{K}} P_k \leq P_{tot}$$

$$\mathrm{C2}: P_{\min} \leq P_k \leq P_{\max}$$

$$\mathrm{C3}: \sum\nolimits_{n \in \mathcal{N}} x_t^{k,n} = 1, \forall k \in \mathcal{K} \tag{5}$$

where $\mathcal{T}$ is the set of time slots, $P1$ is to maximize the throughput, and $P2$ is to maximize the energy efficiency. The constraint $C1$ implies that power consumption of all beams should be less than the total transmit power, $C2$ means that the power of each beam must be in the range of $[P_{\min}, P_{\max}]$, $C3$ denotes that one beam can only serve one cell at a time.

III. The Proposed DeepMBS Algorithm

In this section, the proposed DeepMBS based on parameterized Deep Reinforcement Learning is described in detail.

A. Markov Decision Process Model

The joint beam-hopping and power allocation in MBS systems can be considered as a sequential decision problem and characterised as a discrete-time Markov-Decision Process (MDP) [4]. In order to alleviate the "curse of dimensionality", we treat each beam as a DRL agent, named beam agent. Specifically, for any beam agent k (for simplicity, the subscript k is omitted), the primary elements of MDP, including the state space $\mathbf{S}$, action space $\mathbf{A}$, and reward function r can be defined as follows.

State Space $\mathbf{S}$*:* In this paper, the states of the MBS systems consists of two parts: the matrix of the number of packets $\mathbf{\Phi}$ and the downlink loss matrix $\mathbf{H}$ of the MBS. Then, the sate vector s can be defined as follows, and all possible s constitute the state space $\mathbf{S}$.

$$s = (\mathbf{\Phi}, \mathbf{H}). \tag{6}$$

Action Space $\mathbf{A}$*:* In DeepMBS, beam-hopping and power allocation will be decided concurrently, so the action vector of the MBS a can be expressed as follows, and all possible a constitute the action space $\mathbf{A}$.

$$a = (n, P \,|\, n \in \mathcal{N}, P \in \mathcal{P}) \tag{7}$$

where n means that the cell n is illuminated by the corresponding beam k during time slot t, P denotes the transmission power of the corresponding beam k, and $\mathcal{P} = [P_{\min}, P_{\max}]$. It is noted that n is discrete while P is continuous, thus the action space is discrete-continuous hybrid.

Reward Function r_t*:* In DeepMBS, the optimization objective is to maximize the system throughput and the energy efficiency. Therefore, the reward function of beam agent k can be defined as

$$r_t = \lambda_1 \wp_t^n + \lambda_2 \wp_t^n / (P_{k,t} \cdot d) - \lambda_3 \eta \tag{8}$$

where λ_1, λ_2 and λ_3 are normalized weight factors, $n = \arg_n\{x_t^{k,n} = 1\}$, and $\eta = P_k / \sum_{k \in \mathcal{K}}$, P_k is a penalty factor for the beam agent k when the constraint $C1$ in formula (5) is not satisfied.

B. DeepMBS Based on Parameterized DRL

As described above, the action space of jointly regulating beam hopping and power allocation for MBS systems is a discrete-continuous hybrid action space [6], where the action variable of beam hopping is discrete while that of power allocation is continuous. Approximating the continuous action by an finite discrete set will lose the natural structure of the continuous action, while relaxing the discrete action into a continuous set will significantly increase the complexity of the action space. In addition, the original DRL approaches cannot be directly used to handle with the hybrid action space. For example, the control

problems with discrete action space are usually solved by applying the Deep Q-Network algorithms [7], while policy-based methods [6] are usually applied to deal with the control problems with continuous action space.

Parameterized Action Space $\mathbf{A}'$*:* In this paper, in order to address the above problem, we introduce the parameterized action space into our proposed DeepBMS without any relaxation and approximation. Then, the parameterized action vector a' is defined as:

$$a' = (n, P_n | P_n \in \mathcal{P} \text{ for all } n \in \mathcal{N}) \tag{9}$$

where a high level action n is firstly chosen from a discrete set $\mathcal{N}$, and all possible a' constitute the action space $\mathbf{A}'$. Upon choosing n, a low level parameter $P_n \in \mathcal{P}$, which is associated with the n-th high level action, is then selected. It is noted that here P_n is a continuous set for all $n \in \mathcal{N}$.

Parameterized DRL: In the context of parameterized action space, the action value function can be rewritten as $Q(s_t, a'_t) = Q(s_t, n_t, P_{n_t})$, where $s_t \in \mathbf{S}$, $n_t \in \mathcal{N}$, $P_{n_t} \in \mathcal{P}$. Then, the Bellman equation can be rewritten as

$$\begin{aligned} &Q\left(s_t, n_t, P_{n_t}\right) \\ &= \underset{r_t, s_{t+1}}{\mathrm{E}}\left[r_t + \gamma \max_{n \in \mathcal{N}} \sup_{P_n \in \mathcal{P}} Q\left(s_{t+1}, n, P_n\right) | s_t = s\right] \end{aligned} \tag{10}$$

where γ is the discount factor.

For a given n and Q function, we can find that

$$P_n^Q(s) = \arg\sup_{P_n \in \mathcal{P}} Q(s, n, P_n) \tag{11}$$

is a function of the sate s. Therefore, we can approximate $Q(s_t, n_t, P_{n_t})$ by using a deep neural network $Q(s_t, n_t, P_{n_t}; \omega)$, where ω is its network weights. Similarly, we can approximate $P_{n_t}^Q(s_t)$ by using a deterministic policy network $P_{n_t}(s_t; \theta_t)$, where θ stands for its network weights. The reference for the neural network structure of this algorithm is [11].

Just like the training procedure of DQN [12], the network weights ω can be learned by minimizing the mean-square Bellman error via gradient descent. The target value y_t can be defined as

$$y_t = r(s_t, a'_t) + \gamma \max_{n \in \mathcal{N}} Q\left(s_{t+1}, n, P_n\left(s_{t+1}; \theta_t\right); \omega^-\right) \tag{12}$$

where ω^- is the network weights of the target network $Q' = Q(s_t, n_t, P_{n_t}; \omega^-)$. Then, the loss function can be defined as

$$\ell_t^Q(\omega) = \frac{1}{2}\left[Q\left(s_t, n_t, P_{n_t}; \omega\right) - y_t\right]^2 \tag{13}$$

$$\ell_t^\Theta(\theta) = -\sum_{n \in \mathcal{N}} Q\left(s_t, n, P_n\left(s_t; \theta\right); \omega_t\right) \tag{14}$$

Finally, ω_t and θ_t can be updated via the gradients $\nabla_\omega \ell_t^Q(\omega_t)$ and $\nabla_\theta \ell_t^\Theta(\theta_t)$, respectively:

$$\begin{aligned} \omega_{t+1} &\leftarrow \omega_t - \alpha_t \nabla_\omega \ell_t^Q\left(\omega_t\right) \\ \theta_{t+1} &\leftarrow \theta_t - \beta_t \nabla_\theta \ell_t^\Theta\left(\theta_t\right) \end{aligned} \tag{15}$$

where α_t and β_t denote the learning rate when updating ω_t and θ_t, respectively.

Other Tricks: 1) To reduce the solution space, we adopt the single-agent polling multiplexing mechanism, which has been proven to have superior performance in dealing with beam-hopping problems in our previous work [4]. 2) To further improve the performance of the algorithm, we add an empirical filtering mechanism (EFM) in DeepMBS, named DeepMBS(EFM). That is, during the training process of the algorithm, the experiences with low learning value (LLE) are filtered out and the experiences with high learning value (HLE) are stored in the replay buffer. LLE is defined as an experience where an agent receives a low reward value and discrete actions are repeated over a period of time. The details of the proposed DeepMBS is shown in Algorithm 1.

C. Complexity Analysis

This section discusses the time complexity of the DeepMBS(EFM), which is determined by the neural network structure as well as the sizes of the state space and action space. The DeepMBS(EFM) involves two neural networks, denoted as $P(\theta)$ and $Q(\omega)$. Suppose $P(\theta)$ and $Q(\omega)$ consist of $\mathrm{L_P}$ and $\mathrm{L_Q}$ fully connected layers, respectively. Taking into account the bias terms adding in the fully connected layers, the time complexity is calculated as follows:

$$\begin{aligned} &\sigma(\mathrm{s})\zeta_0^{\mathrm{P}} + 2 \times \sum_{\mathrm{l}=0}^{\mathrm{L_P}-1} \zeta_{\mathrm{l}}^{\mathrm{P}} \zeta_{\mathrm{l}+1}^{\mathrm{P}} + \sigma(\mathrm{N})\sigma(\mathrm{P})\zeta_{\mathrm{L_P}}^{\mathrm{P}} \\ &\quad + (\sigma(\mathrm{s}) + \sigma(\mathrm{N})\sigma(\mathrm{P}))\zeta_0^{\mathrm{Q}} + 2 \times \sum_{\mathrm{l}=0}^{\mathrm{L_Q}-1} \zeta_{\mathrm{l}}^{\mathrm{Q}} \zeta_{\mathrm{l}+1}^{\mathrm{Q}} + \sigma(\mathrm{N})\zeta_{\mathrm{L_P}}^{\mathrm{P}} \\ &= \mathrm{O}\left(\sigma(\mathrm{s})\zeta_0^{\mathrm{P}} + \sum_{\mathrm{l}=0}^{\mathrm{L_P}-1} \zeta_{\mathrm{l}}^{\mathrm{P}} \zeta_{\mathrm{l}+1}^{\mathrm{P}} + \sigma(\mathrm{N})\sigma(\mathrm{P})\zeta_{\mathrm{L_P}}^{\mathrm{P}}\right. \\ &\quad \left. + (\sigma(\mathrm{s}) + \sigma(\mathrm{N})\sigma(\mathrm{P}))\zeta_0^{\mathrm{Q}} + \sum_{\mathrm{l}=0}^{\mathrm{L_Q}-1} \zeta_{\mathrm{l}}^{\mathrm{Q}} \zeta_{\mathrm{l}+1}^{\mathrm{Q}} + \sigma(\mathrm{N})\zeta_{\mathrm{L_P}}^{\mathrm{P}}\right) \end{aligned} \tag{16}$$

where $\zeta_{\mathrm{l}}^{\mathrm{P}}$ and $\zeta_{\mathrm{l}}^{\mathrm{Q}}$ represent the number of neurons in the l_{th} layer, $\mathrm{l} = 0$ denotes the input layer, and $\mathrm{l} = \mathrm{L_P}$ or $\mathrm{l} = \mathrm{L_Q}$ indicates the output layer. $\sigma(\mathrm{s})$ represents the dimensionality of the state vector, $\sigma(\mathrm{N})$ donates the number of discrete action variables and $\sigma(\mathrm{P})$ is the number of continuous action variables. If the neural network structure is fixed, the time complexity can be simplified as follows:

$$\mathrm{O}(\sigma(\mathbf{s}) + \sigma(\mathrm{N})\mathcal{N}(\mathrm{P}) + \sigma(\mathrm{N})) \tag{17}$$

which entirely depends on the dimensions of the state vector and action vector.

IV. Evaluation

In this section, we describe the setup of our experiments and analyze the experimental results.

A. Experiment Setup

We implemented a Ka-band MBS system using Python for simulation experiments. In this system, a multi-beam satellite operates in geostationary orbit at an altitude of 35786 km. The total bandwidth allocated to the satellite is 500 MHz, and the total available transmit power of the satellite is 34.5 dBw. The maximum transmit power per beam is 28 dBw, and the minimum transmit power per beam is 21 dBw. The satellite's multi-beam antenna has 7 beams, with an antenna aperture of 0.25 m. There are a total of 30 cells within the satellite coverage area. The ground receiving antenna gain is 42.1 dBi, and the free space loss is 209.6 dB (the same setting is used in [12]). Then, DeepMBS(EFM) employs a two-layer fully connected feedforward neural network to approximate $Q(s, n, P_n; \omega)$, with each layer containing 128 and 64 neurons, respectively, and using "ReLU" as the activation function. A three-layer fully connected feedforward neural network is utilized to

Algorithm 1: The DeepMBS Algorithm.

The Training Process

Input: Initialize exploration parameter ε, minibatch size B, replay buffer $\Re$, a probability distribution ξ, network weights ω_0 and θ_0.

1: **for** $t = 1, 2, \ldots, T$ **do**
- Receive initial observation state s_t.
- Compute action parameters $P_{n_t} \leftarrow P_{n_t}(s_t; \theta_t)$.
- Select action $a'_t = (n_t, P_{n_t})$ according to the ε-greedy policy
- $a'_t = \begin{cases} \text{randomly selected from } \mathcal{N}, \text{ probability}=\varepsilon \\ n_t = \arg\max_{n_t \in \mathcal{N}} Q(s_t, n_t, P_{n_t}; \omega_t), \text{ otherwise} \end{cases}$
- Take action a'_t, observe reward r_t and the next state s_{t+1}.
- Determine the value of the experience and store the HLE $[s_t, a'_t, r_t, s_{t+1}]$ into $\Re$.
- Sample B transitions $\{s_b, a'_b, r_b, s_{b+1}\}_{b \in B}$ randomly from $\Re$.
- Calculate the target value y_b according to the formula (12).
- Calculate the loss functions and gradients according to the formula (13) and (14).
- Update the ω_t and θ_t according to formula (15).

2: **end for**

The Polling Decision-Making Process

3: **for** beam $k = 1$ to K **do**
- Receive initial observation state s_t^k.
- $a_t'^k = (n_t^k, P_{n_t^k})$ is calculated through the well-trained DeepMBS model and get next-state s_t^{k+1}.

4: **end for**
- Obtain the joint beam-hopping and power allocation strategy $a' = [a'_1, a'_2, \ldots, a'_K]$ for MBS systems.

approximate $P_n(s; \theta)$, with each layer consisting of 128, 128, and 64 neurons, respectively. To ensure the output values fall within a symmetric interval, "tanh" is used as the activation function for the output layer. Additionally, the settings of other key parameters are as follows: initial learning rates $\alpha_t = \beta_t = 0.0001$, mini-batch size $B = 128$, discount factor=0.99, Replay memory capacity $\Re = 100000$.

B. Baseline Algorithms

To verify the performance gain of our proposed DeepMBS algorithm in terms of packet loss rate, throughput, queuing delay, and energy efficiency defined in formula (5)-$P2$, we compare DeepMBS with the following baseline algorithms:

- Random Algorithm: This approach randomly determines the illuminated cells and transmit power.
- Greedy Algorithm [9]: This approach selects K cells with the highest communication demand and evenly allocates the transmit power among the K cells.
- Genetic Algorithm [7]: In this method, cells and transmission power are selected after G rounds of cross-mutation.
- DQN(EFM) [10]: This approach uses Deep Q-Network with empirical filtering mechanism (EFM) to derive the optimal joint beam hopping and power allocation strategy, where the continuous transmission power P is discretized.

C. Performance Comparison

Convergence Analysis: Fig. 1 shows the convergence results of our proposed DeepMBS and DeepMBS with EFM, i.e., DeepMBS(EFM).

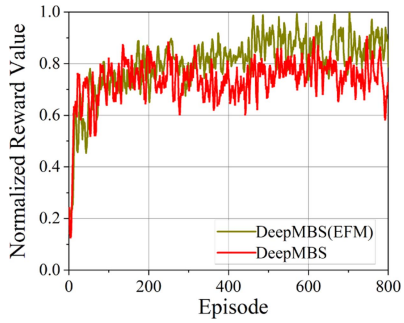

Fig. 1. The normalized reward value.

In our experiments, the algorithms are iterated 3,600 times in each episode. Both DeepMBS and DeepMBS(EFM) gradually become converged after 200 episodes. In addition, it can be found that DeepMBS(EFM) can achieve a better average reward than DeepMBS. The reason is that EFM filters out the invalid LLEs from the experience, and the remaining HLEs are more conducive to learning.

Performance Comparison: Firstly, as illustrated in Fig. 2(a)–(c), the packet loss rate, system throughput and average queuing delay of almost all algorithms show an increasing trend when the total communication demand increases from 1,800 Mbps to 3,600 Mbps. In the case of low communication demand, except for the Random algorithm, the other four algorithms tend to have the same performance. This is because the performance of MBS is less dependent on resource scheduling schemes under lower communication demand. However, DeepMBS(EFM) outperforms the four baseline algorithms when the communication demand increases and exceeds 3,000 Mbps. In particular, when the communication demand is 3,600 Mbps, the throughput of DeepMBS(EFM) is 14.3% higher than that of DQN(EFM) algorithm (the second best) and 57.6% higher than that of Random algorithm (the worst). The potential reason is that the DRL-based algorithms DeepMBS(EFM) and DQN(EFM) can better catch the system dynamics and more emphasize the long-term cumulative gains compared to other baseline algorithms. In addition, DeepMBS(EFM) can accurately control power allocation with its parameterized action space.

Secondly, the results of the energy efficiency at different communication demands are shown in Fig. 2(d). It can be found that DeepMBS(EFM) can achieve relatively stable energy efficiency of about 0.72 Mbit/J under different communication demands. Specifically, when the communication demand is 1,800 Mbps, DeepMBS(EFM) can improve the energy efficiency by about 6.9% and 38.9% with respect to DQN(EFM) (the second best) and Random algorithms (the worst), respectively. This is because that DeepMBS(EFM) with parameterized action space can achieve a more accurate power allocation, while the other baseline algorithms discretize the continuous action (i.e., power allocation) and lose the natural structure of the continuous action. Whereas, when the communication demand increases and approaches the maximum capacity of the MBS, almost every beam k needs to work in full power mode, so almost all algorithms have high energy efficiency in this case.

D. Performance Analysis of DeepMBS

In this section, we conducted more experiments to investigate the efficacy of our proposed algorithm with different demand patterns and optimization objectives.

In order to assess the generalization ability of DeepMBS (EFM), we utilize two communication demand patterns with distinct characteristics to drive our experiments. The "**Type 1 demand pattern**" has relatively

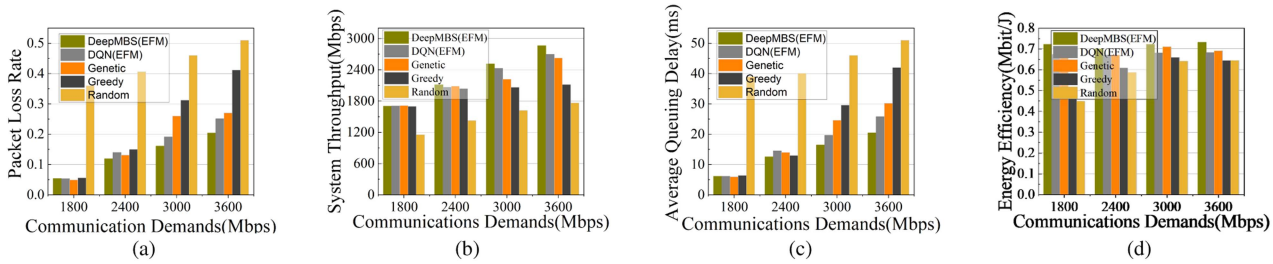

Fig. 2. Performance comparison in terms of packet loss rate, system throughput, avg. queuing delay and energy efficiency. (a) The packet loss rate. (b) The system throughput. (c) The avg. queuing delay. (d) The energy efficiency.

slow and slight fluctuations (the communication demand of a single cell follows a normal distribution with a mean of $e = D_{total}N$ and a variance of $\sigma_1^2 = 10$, where D_{total} represents the total communication demand and N represents the number of cells), making them easy to be predicted in advance. In contrast, the "**Type 2 demand pattern**" demonstrates dramatical fluctuations (the communication demand of a single cell follows a normal distribution with a mean of $e = D_{total}N$ and a variance of $\sigma_2^2 = 40$). Under this pattern, the communication demands may change suddenly and drastically, which requires the proposed algorithm to adjust in time and flexibly.

In addition, to verify the necessity for joint optimization of throughput and energy efficiency, we adapt the reward function with different optimization objectives (i.e., 1) energy efficiency, 2) system throughput or 3) energy efficiency and system throughput) for the DeepMBS(EFM) algorithm. The combinations of different algorithms, communication demand patterns, and reward functions are listed as follows:

- *Baseline:* DeepMBS(EFM) + "Type 1 demand pattern" + "joint optimization of system throughput and energy efficiency".
- *Plan A:* DeepMBS(EFM) + "Type 1 demand pattern" + "optimization of energy efficiency".
- *Plan B:* DeepMBS(EFM) + "Type 1 demand pattern" + "optimization of system throughput".
- *Plan C:* DeepMBS(EFM) + "Type 2 demand pattern" + "joint optimization of system throughput and energy efficiency".
- *Plan D:* DQN(EFM) + "Type 2 demand pattern" + "joint optimization of system throughput and energy efficiency".

where Baseline, Plan A and Plan B are compared to verify the necessity for joint optimization of throughput and energy efficiency, while Baseline, Plan C and Plan D are compared to illustrate the generalization ability of DeepMBS(EFM).

The experimental results of the aforementioned schemes are illustrated in Fig. 3. Firstly, it is evident that different communication demand patterns do affect the performance of the proposed algorithm. DeepMBS(EFM) performs worse under "Type 2 demand pattern" than "Type 1 demand pattern". This discrepancy arises because when communication demand undergoes significant fluctuations, the DeepMBS(EFM) algorithm may not be able to fully adapt to such changes, resulting in a slight decrease in performance. Nonetheless, the overall performance of DeepMBS(EFM) under "Type 2 demand pattern" still outperforms that of DQN(EFM). This shows that DeepMBS(EFM) can more accurately and sensitively capture the dynamic changes of communication demand, and have better generalization ability than DQN(EFM).

Furthermore, we find that different reward functions would have different effects on the performance of the proposed algorithm. Specifically, when the reward function incorporates system throughput and energy efficiency simultaneously (i.e., λ_1=λ_2= 0.5, Baseline), the system achieves a maximum throughput of 2865.62 Mbps and a peak energy efficiency of 0.72 Mbit/J. In contrast, when the reward function

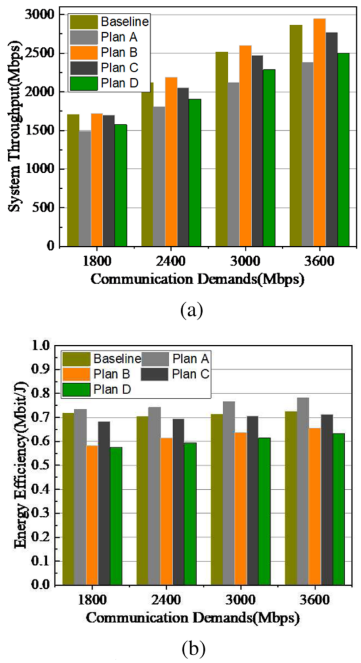

Fig. 3. Performance comparison in terms of system throughput and energy efficiency. (a) The system throughput. (b) The energy efficiency.

is solely based on energy efficiency (i.e., λ_1= 0, Plan A), the system achieves a maximum throughput of 2383.24 Mbps and the highest energy efficiency of 0.78 Mbit/J. Conversely, with the reward function solely focusing on system throughput (i.e., λ_2= 0, Plan B), the system attains a maximum throughput of 2944.83 Mbps and a peak energy efficiency of 0.65 Mbit/J. That is to say, the Baseline sacrifices 0.06 Mbit/J of energy efficiency to achieve a higher throughput of 482.38 Mbps than Plan A. In contrast, compared to Plan B, the Baseline trades 79.21 Mbps of throughput for an improvement of 0.07 Mbit/J in energy efficiency. Through the comprehensive analysis, it is clear that integrating both system throughput and energy efficiency in the reward function can get better overall performance. This verifies the necessity of conducting multi-objective optimization.

V. Conclusion

In this paper, we propose a novel joint optimization algorithm based on parameterized DRL to concurrently regulate beam hopping and power allocation in MBS systems, named DeepMBS. In DeepMBS, a multi-objective problem is firstly formulated to optimize system throughput and energy efficiency. Then, the optimization problem is modelled as a MDP, and the original deep Q-network is extended with

a parameterized action space to simultaneously determine the beam hopping (discrete action) and power allocation (continuous action). Finally, we carry out extensive experiments by combining three types of reward functions with two different communication demand patterns, and the results illustrate that the proposed DeepMBS can improve the system throughput by 14.3% to 57.6% and improve the energy efficiency by 6.9% to 38.9% compared to the baseline algorithms. In addition, the proposed DeepMBS(EFM) can more accurately and sensitively capture the dynamic changes of communication demands.

REFERENCES

[1] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "Joint HAP access and LEO satellite backhaul in 6G: Matching game-based approaches," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1147–1159, Apr. 2021.

[2] M. Xu et al., "EPViSA: Efficient auction design for real-time physical-virtual synchronization in the human-centric metaverse," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 3, pp. 694–709, Mar. 2024.

[3] L. He, J. Li, Y. Wang, J. Zheng, and L. He, "Balancing total energy consumption and mean makespan in data offloading for space-air-ground integrated networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 209–222, Jan. 2024.

[4] G. Xu, F. Tan, Y. Ran, Y. Zhao, and J. Luo, "Joint beam-hopping scheduling and coverage control in multibeam satellite systems," *IEEE Wireless Commun. Lett.*, vol. 12, no. 2, pp. 267–271, Feb. 2023.

[5] G. Maral, M. Bousquet, and Z. Sun, *Satellite Communications Systems: Systems, Techniques and Technology*. Hoboken, NJ, USA: Wiley, 2020.

[6] V. D. Tuong, N.-N. Dao, W. Noh, and S. Cho, "Deep reinforcement learning-based hierarchical time division duplexing control for dense wireless and mobile networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7135–7150, Nov. 2021.

[7] L. Wang, X. Hu, S. Ma, S. Xu, and W. Wang, "Dynamic beam hopping of multi-beam satellite based on genetic algorithm," in *Proc. IEEE Intl.Conf Parallel Distrib. Process. with Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Soc. Comput. Netw.*, 2020, pp. 1364–1370.

[8] M. Takahashi, Y. Kawamoto, N. Kato, A. Miura, and M. Toyoshima, "Adaptive power resource allocation with multi-beam directivity control in high-throughput satellite communication systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1248–1251, Aug. 2019.

[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.*, 2021, pp. 1–6.

[10] X. Du, X. Hu, Y. Wang, and W. Wang, "Dynamic resource allocation for beam hopping satellites communication system: An exploration," in *Proc. IEEE Int. Conf. Trust, Secur.. Privacy Comput. Commun.*, 2022, pp. 1296–1301.

[11] Y. Ran, X. Zhou, H. Hu, and Y. Wen, "Optimizing data centre energy efficiency via event-driven deep reinforcement learning," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1296–1309, Mar./Apr. 2023.

[12] X. Hu, Y. Zhang, X. Liao, Z. Liu, W. Wang, and F. M. Ghannouchi, "Dynamic beam hopping method based on multi-objective deep reinforcement learning for next generation satellite broadband systems," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 630–646, Sep. 2020.