Cluster Hopping in Multi-beam Satellite Communication Systems Using Deep Reinforcement Learning

Shruti Sharma[1], Sangmin Han[2], Jaehyup Seong[2] and Wonjae Shin[3]
[1]*Department of Electrical and Computer Engineering, Ajou University, Suwon, 16499, South Korea*
[2]*Department of Artificial Intelligence Convergence Network, Ajou University, Suwon, 16499, South Korea*
[3]*School of Electrical Engineering, Korea University, Seoul, 02841, South Korea*
{shruti, hsm960622, john12234}@ajou.ac.kr, wjshin@korea.ac.kr

***Abstract*—Beam hopping method has become a promising and significant technology in the next generation of high throughput satellite (HTS) systems. Using deep reinforcement learning (DRL), this research suggests a unique method for improving precoded cluster hopping in multi-beam geostationary orbit (GEO) satellite communication systems. Choosing the optimum beam clusters for transmission based on the current channel circumstances, our goal is to increase the effectiveness and capacity of satellite communication systems. Our simulation results based on DRL yields better performance when compared to other techniques.**

***Index Terms*—geostationary orbit, deep reinforcement learning, satellite networks, beam hopping, precoding.**

I. Introduction

Multi-beam geostationary orbit (GEO) satellite communication systems play a critical role in providing global coverage and high-capacity data transmission for various applications. To meet the increasing demand for data services, optimizing the capacity allocation and resource management in these systems is crucial. One approach that has gained significant attention is the concept of beam hopping, which allows flexible resource allocation. Beam hopping, in combination with precoding techniques, enhances system performance by adapting to varying traffic patterns and user demands [1]. It offers the potential to improve system capacity, optimize spectral efficiency, and ensure efficient utilization of satellite resources.

Advantages of cluster hopping (CH) have been the subject of several studies. In order to enhance system performance and capacity allocation, authors presented a cluster hopping strategy mixed with precoding techniques [1]. Comparing them to more established techniques, they showed improved capacity and higher user demand satisfaction. So as to hasten the optimization process. On the other hand, deep reinforcement learning (DRL), which combines the strengths of deep learning and reinforcement learning, has become a popular strategy in machine learning. The contributions of this work are as follows:

- The DRL method is adapted to maximize the minimum ratio between the offered capacity and the requested demand among the cluster.
- We also compare our proposed method with respect to the heuristic approach and linear programming method.

II. System Model

In our study, we analyze a multi-beam satellite system with a total of N_b beams. At any given time instance, only a subset of F beams can be concurrently activated. We define the illumination ratio as $\frac{F}{N_b}$. An illumination ratio of $1/4$, for example, means that 25% of the total number of beams are lighted. It is assumed that all beams operate in the same spectrum and utilize full-frequency reuse. The hopping window is divided into several time slots, and we try to match the illumination pattern average capacity to the specified demand. In a specific snapshot and cluster, the received signal vector for the k active beams in cluster i is represented as y_k^i which is expressed as

$$y_k^i = (\mathbf{h}_k^i)^{\mathsf{T}}\mathbf{v}_k^i s_k^i + \sum_{j\neq k}(\mathbf{h}_k^i)^{\mathsf{T}}\mathbf{v}_j^i s_j^i + \sum_{u\neq i}\sum_{j}(\mathbf{h}_k^i)^{\mathsf{T}}\mathbf{v}_j^u s_j^u + z_k^i, \tag{1}$$

where $\mathbf{h}_k^i = [h_1^i, \cdots, h_K^i]^{\mathsf{T}} \in \mathbb{C}^{K\times 1}$ denotes is the channel vector for k-th beam in i-th cluster. $\mathbf{v}_k^i \in \mathbb{C}^{K\times 1}$ being the precoding vector for k-th beam in i-th cluster. z_k represents the additive Gaussian zero-mean unit-variance noise. To reduce interference, we use minimal mean square error (MMSE) precoding [2]. The received signal-to-interference-plus-noise ratio (SINR) for the k-th beam in the i-th cluster can be expressed as

$$\gamma_k^i = \frac{|(\mathbf{h}_k^i)^{\mathsf{H}}\mathbf{v}_k|^2}{\sum_{j\neq k}|(\mathbf{h}_k^i)^{\mathsf{H}}\mathbf{v}_j|^2 + \sum_{u\neq i}\sum_j |(\mathbf{h}_k^i)^{\mathsf{H}}\mathbf{v}_j|^2 + (\sigma_k^i)^2}. \tag{2}$$

Therefore, the achievable capacity to cluster i is expressed as

$$C_i = W f_{\mathsf{DVB}}(\gamma_k^i), \tag{3}$$

where the f_{DVB} is the mapping function based on the digital video broadcasting [3]. Specifically, we aim to achieve approximate equality between the requested demand and offered capacity of each beam and cluster, $\forall i \in \{1, \cdots N_b\}$ and $\forall k \in \{1, \cdots N_c\}$, respectively. The illumination design optimization

is conducted at the hopping window level. Thus, we adjust the cluster demand to $\hat{D}_i = T_h D_i$ [bits/hopping window] in which T_h indicates the hopping window consisting of N_s time-slots. Also, we adjust the time slot-based cluster capacity to $\hat{C}_i = T_s C_i$ [bits/time-slot] in which T_s denotes the time slot. Therefore, the effective capacity at the hopping window level can be given by $\hat{R}_i = \sum_{t=1}^{N_s} u_t[i]\hat{C}_i$ [bits/hopping window], where $u_t[i]$ is a binary number for the i-th cluster.

The objective is to maximize the minimum ratio between the offered capacity and the requested demand among the cluster/beams and can be expressed as follows:

$$\mathcal{P}_1: \max_{\mathbf{u}_1,\cdots,\mathbf{u}_{N_s}} \min \frac{\hat{R}_i}{\hat{D}_i}$$

$$\text{s.t.} \quad \sum_{i=1}^{N_c} u_t[i] \leq F', \tag{4a}$$

$$\mathbf{u}_t^{\mathsf{T}} \mathbf{A} \mathbf{u}_t = 0, \ t = 1, \ldots, N_s, \tag{4b}$$

$$u_t[i] \in \{0,1\}, \ \forall i, \ t = 1, \cdots N_s, \tag{4c}$$

$$\sum_{i=1}^{N_c} u_t[i] P_i \leq P_T, \tag{4d}$$

$$P_i \leq P_{\mathsf{max}}. \tag{4e}$$

The vector $\mathbf{u}_1, \cdots, \mathbf{u}_{N_s}$ represents the optimization variables, where $\mathbf{u}_t$ is a binary vector of size $N_c \times 1$. The positions of 1's in $\mathbf{u}_t$ indicate the indices of the illuminated clusters.

The constraint (4a) guarantees that the overall number of active beams at a particular time slot t equals F', which is the preset number of active clusters. This equation limits the total number of active beams not to exceed the number of active clusters accessible during a specific time period. (4b) enforces that active clusters are not adjacent to each other using the matrix $\mathbf{A} \in \{0,1\}^{N_c \times N_c}$, which is a square symmetric matrix, i.e., $A_{i,j} = A_{j,i}$. If $A_{i,j} = 1$, it indicates that cluster i is adjacent to cluster j. This constraint ensures that adjacent clusters cannot be simultaneously active, promoting better interference management and resource allocation. (4c) states that the elements of the vector $\mathbf{u}_t$ should be binary, meaning they can only take on the values of 0 or 1. (4d) and (4e) reflects the relationship between the power levels P_i and $u_t[i]$ ensuring that the selected beam powers do not exceed the allocated total power budget P_T and that individual beam powers stay within the limit P_{max}. The above optimization problem $\mathcal{P}_1$ is solved by turning it into a maximization problem with the help of an additional slack variable ϕ and expressed as

$$\mathcal{P}_2: \max_{\mathbf{u}_1,\cdots,\mathbf{u}_{N_s},\phi} \phi$$

$$\text{s.t.} \quad \frac{\hat{R}^i}{\hat{D}^i} \geq \phi, \tag{5a}$$

(4a), (4b), (4c), (4d), (4e).

III. Heuristic and DRL Based Solution

A. Heuristic solution

For heuristic method, we select the cluster with the highest demand-to-capacity ratio and activate it if it satisfies the constraints. Then update it by recalculating the demand-to-capacity ratio for the remaining clusters and repeat the selection process until the desired number of clusters is activated or the constraints are met.

Algorithm 1: CH algorithm using DRL

1. Initialize the environment with parameters such as demand and system constraints.
2. Initialize the random weights θ and target network θ^-.
3. Create a DQN Agent with a Q-network, optimizer, and loss function.
4. Train the agent by selecting actions based on the current state.
5. Store the experience tuples (s, a, r, s').
6. Update the Q-network parameters using the agent's memory and experiences.
7. Record episode rewards, episode numbers, epsilon.
8. Sample the mini-batch of (s, a, r, s') from D.
9. Train the Adam optimizer.
10. Evaluate the performance of the offered capacity.
11. End.

B. DRL based solution

In this section, we present an overview of the fundamental concepts in DRL, including state, action value, and reward. Here state represents $s_t^i = \{\hat{D}^i, \hat{C}^i, \hat{R}^i, F'\}$, action refers to the decision based on the binary variable expressed as $a_t = \{u_t[1], u_t[2], \cdots, u_t[i] \mid u_t[i] \in \{0,1\}, \forall i\}$, and reward is the feedback indicating the minimum ratio between the offered capacity and the requested demand. We propose an algorithm based on DRL for optimizing the illumination pattern in a multibeam GEO satellite networks. CH algorithm using the DRL process is illustrated in Algorithm 1.

The algorithm consists of several steps, including state initialization, action selection, state transition, reward calculation, and model training using a memory buffer and an Adam optimizer. We use a deep neural network (DNN) in this context to estimate the ideal action-value function as

$$Q^*(s_t^i, a_t) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} \,\middle|\, s_t^i = s, a_t = a, \pi\right], \tag{6}$$

where γ is discount factor. The training process involves updating the Q-network using experiences from the environment, utilizing techniques like experience replay to improve learning stability. Experience replay is employed, where past experiences are stored in a replay memory buffer and randomly sampled for training, enhancing learning efficiency and stability. During each training step, a batch of experience items, here we call it a mini-batch, are randomly sampled from replay memory and the target value is calculated through the target network Q based on the Bellman equation [4]. The target value is calculated as

$$y_t = r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}^i, a_{t+1}; \theta^-). \tag{7}$$

TABLE I
SIMULATION PARAMETERS

Parameters	Value
Satellite longitude	13°E (GEO)
Satellite total power, P_T	6000 W
Beam radiation pattern	Provided by ESA
Downlink carrier frequency	19.5 GHz
Roll-off factor	20%
Hopping Window	256
Duration of a time-slot	1.3 ms
User link bandwidth, BW	500 MHz
Learning Rate	0.0001
Training Epochs	2000
Replay Start Size	1000
Discount Factor	0.9
Initial Exploration Rate	0.5
Final Exploration Rate	0.01
Activation Function	Relu

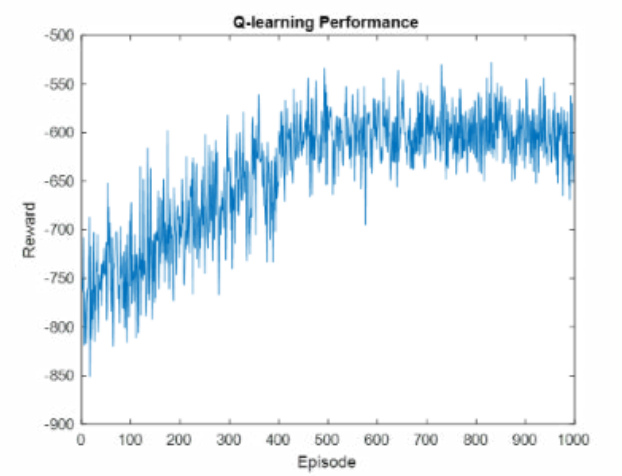

Fig. 1. Reward versus the number of episodes.

Based on the target value in $\mathcal{P}_2$, the loss $L(\theta)$ of the current network is calculated as

$$L(\theta_t) = \mathbb{E}_{(s,a,r,s')}\left[\left(y_t - Q\left(s_t^i, a_t; \theta_t\right)\right)^2\right]. \quad (8)$$

IV. SIMULATION RESULTS AND DISCUSSION

In this section, we evaluate the proposed CH snap-shot selection and illumination period optimization results. The simulation parameters are listed in Table 1. A total of 2000 training epochs are used. The structure of the Q-Network has three fully connected layers.

Fig.1. shows how the rewards obtained by the agent change as the number of episodes increases. This figure helps us understand the learning progress and convergence of the algorithm, allowing us to analyze the effectiveness of the applied DRL technique. In Fig. 2, the performance of the proposed DRL scheme is compared with various schemes. The simulation results reveal that linear programming (LP) and greedy algorithms averagely satisfy the cluster demand by 87% and 89%, respectively. The performance of the proposed DRL framework for demand satisfaction is by 93.6% on average, demonstrating its superiority in terms of cluster demand matching compared to that of benchmark schemes. Therefore,

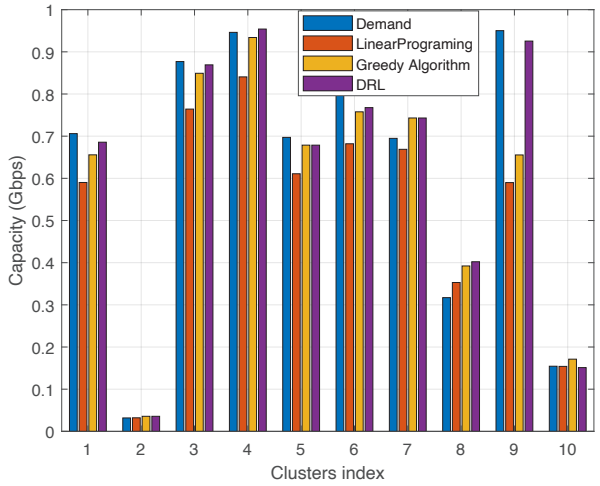

Fig. 2. Demand vs. offered capacity at the cluster level.

it is concluded that DRL presents a powerful technique for handling the intricate demands of precoded cluster hopping, outperforming LP and greedy methods in the process. DRL might have a higher training complexity but can lead to efficient decision-making once trained. Greedy algorithms and LP could be computationally efficient for smaller instances but might lack optimally guarantee. While greedy algorithms make locally optimal decisions, they may not guarantee global, LP can be limited by memory and time for larger instances.

V. CONCLUSTION

We have proposed a DRL method for enhancing beam-hopping and precoded in multi-beam GEO satellite communication systems. We have compared the effectiveness of our DRL-based approach greedy algorithm through extensive simulations. This demonstrates how DRL has the potential to be an effective solution for increasing multi-beam satellite communication systems' efficiency and performance.

ACKNOWLEDGEMENT

This work was supported in part by the National Research Foundation of Korea (NRF) grants (No.2021R1A4A1030775, No.2022R1A2C4002065) and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grants (No.2021-0-00260, No.2021-0-00467, and No.2022-0-00704).

REFERENCES

[1] M. G. Kibria *et al.*, "Precoded cluster hopping in multi-beam high throughput satellite systems," in *Proc. 2019 IEEE Glob. Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[2] C. B. Peel *et al.*, "A vector-perturbation technique for near-capacity multiantenna multiuser communication-Part I: Channel inversion and regularization," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 195–202, 2005.

[3] "Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 2: DVB-S2 extensions (DVB-S2x)," Document ETSI EN 302 307-2.

[4] X. Hu *et al.*, "A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1612–1615, 2018.