IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 February 2023, accepted 23 February 2023, date of publication 8 March 2023, date of current version 14 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3253870

RESEARCH ARTICLE

Dynamic Downlink Interference Management in LEO Satellite Networks Without Direct Communications

JIHYEON YUN[1], TAEGUN AN[1], HAESUNG JO[1], BON-JUN KU[2], DAESUB OH[2], AND CHANGHEE JOO[1], (Senior Member, IEEE)

[1]Computer Science and Engineering, Korea University, Seoul 02841, South Korea
[2]Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, South Korea

Corresponding author: Changhee Joo (changhee@korea.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) Grant funded by the Korean Government through Ministry of Science and ICT (Information and Communications Technology) (Development of the spectrum sharing technology for non-GeoSynchronous Orbit (GSO) satellite system) under Grant 2021-0-00719.

ABSTRACT We investigate effective interference management for Low Earth Orbit (LEO) satellite networks that provide downlink services to ground users and share the same frequency spectrum range. Since there are multi-group LEO satellites with different constellation orbits, the ground users will experience time-varying interference due to the overlapping of main/side lobes of the satellite beams, which becomes even more challenging when the interfering satellites cannot communicate directly. To address the problem, we consider two LEO satellite groups that provide communication service in the same ground area, while competing for communication resources. We develop solutions that maximize the throughput and manage the time-varying interference under a certain level, without explicit message exchanges between the satellite groups. By exploiting statistical learning and deep reinforcement learning techniques, we develop learning-based resource allocation schemes and evaluate their performance through extensive simulations. We show their effectiveness under different reward settings and different interference managements, and demonstrate that a Deep Q-Network (DQN)-based scheme can achieve the close-to-optimal performance.

INDEX TERMS LEO satellite networks, interference management, spectrum sharing, Deep Q-Network.

I. INTRODUCTION

6G networks start emerging as the successor of 5G networks, and are expected to employ higher frequencies than 5G networks with substantially large capacity targeting microsecond latency. These will be essential to provide users satisfactory services through Internet-of-Things (IoT), super-intelligence, virtual and augmented reality (VR/AR), etc [1], [2], [3], [4]. Satellite communications play an important role in 6G networks since they can handle massive data traffic that is beyond the capacity under the traditional terrestrial-based communication systems. In particular, Low Earth Orbit (LEO) satellite communications attract much attention owing to low energy for arrangement and lower communication latency than other types of satellites. However, despite these advantages, LEO satellite networks face many challenges in providing seamless services, mainly due to limited frequency resources that are already occupied by other communication systems. Accordingly, an effective interference management for LEO satellite networks will be one of the key technology elements for 6G communication networks. It can be accomplished by various methods including exclusive spectrum allocation, multiple access techniques, and antenna techniques such as angular separation. In this work, we consider dynamic spectrum allocation methods, where competing LEOs share the same frequency spectrum without prior coordination.

We consider LEO satellite networks where each satellite provides downlink services to ground users. When an LEO satellite transmits signal to ground users located in the

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

main lobe area (or main beam area) of the satellite, the user may be also co-located in the main and/or side lobe area of other nearby satellites that share the same wireless spectrum. This is common in practice due to the scarce frequency resources for satellite communications [5], and causes significant interference, in which case the user cannot decode the received signal successfully.

The problem can be viewed as a long-term static situation, and we can attempt to find the best *fixed* allocation to maximize the performance. For example, in [6], the authors aim to achieve high throughput performance while mitigating the interference in LEO satellite networks in such setting. They exploit Upper Confidence Bound (UCB) algorithms for frequency allocation, which are based on simple computation of indexes and known to achieve the optimal performance asymptotically [7], [8], [9], [10]. There are also studies [11], [12] that address the time-varying interference problems in the satellite network using Deep Reinforcement Learning (DRL) techniques. By training neural network models, the authors find the best beam pattern and bandwidth allocation.

The aforementioned studies, however, are limited to static interference environments ignoring the short-term interference changes caused by the fast movement of LEO satellites, or assume one or zero interfering satellite. In practice, there are multiple groups of LEO satellites orbiting the earth, and their ground users experience severe time-varying interference [13]. The problem becomes even more challenging when the interfering satellites belong to a different communication system, and there is no direct coordination or communications due to several operational reasons such as security and cost. In such scenarios, without proper interference management, it is not possible to provide seamless communication services to users.

In this work, we consider dynamic interference environments where the satellites' movement following their orbits has a significant impact on the interference at the ground users over time. We assume that the satellites in different constellations belong to a different communication system and share spectrum resources without direct communications with each other, which is common in wireless networks, e.g., 802.11 and Bluetooth in unlicensed spectrum [14] and cognitive radio networks in licensed spectrum [15]. In such scenarios, we investigate how effectively one can allocate time-frequency channels and maximize the throughput performance while managing the interference from other satellites below a certain level, without the knowledge of channel allocation of other satellites nor statistical information in advance.

To this end, we adopt several UCB variants (Discounted UCB and Sliding-window UCB [16], [17]) that can take into account the time-varying property. We also consider Deep Q-Network, a deep reinforcement learning technique, that uses a deep convolutional neural network for stochastic search [18], [19]. We modify them accordingly for the frequency allocation policy in LEO satellite networks, and investigate how well they exploit the unused frequency channels under dynamic interference environments.

To the best of our knowledge, this is the first work that addresses the resource allocation problem under interference dynamics and investigates effective interference management schemes without direct coordination nor prior knowledge about the channel usage information of interfering satellites. We take the approaches of statistical learning and deep reinforcement learning by exploiting the UCB and DQN algorithms through problem reformulation, and develop the framework that can immediately constrain the collision rate. We evaluate the proposed schemes through extensive simulations with different settings of rewards, and show the effectiveness of the DQN, which can achieve close-to-optimal performance when configured accordingly.

The remainder of the paper is organized as follows. We first review related previous works in Section II. Then we describe the system model and formulate the problem in Section III. We develop three UCB-based schemes for frequency allocation in Section IV, and develop a DQN-based scheme through problem reformulation in Section V. In Section VI, we evaluate the performance of our schemes and evaluate their performances in different settings of reward and constraint. We finally conclude our paper in Section VII.

II. RELATED WORK

There have been many works that address the problem of the interference management in satellite networks. In large, they can be classified into two problems of Co-Channel Interference (CCI) and inter-satellite interference, depending on the sources that incur the interference. The CCI problem focuses on the resource allocation and interference management between the beams in multi-beam satellite system. The authors of [20] proposed high-throughput satellite (HTS) system architecture for geostationary satellites with multi-beam satellite communication system. In [21], the optimal frequency channel allocation for LEO satellite networks was investigated by adopting the Q-learning technique. In [22], the authors proposed an effective power allocation scheme using a deep reinforcement learning technique for Satellite Internet of Things (SIoT) systems under CCI and power supply constraints. On the other hand, the inter-satellite interference problem considers interference between different satellites. In [23], high-altitude platforms (HAPs), terrestrial relays (TRs), and LEO satellites are combined as the hierarchical resource allocation structure that allocates resources to different layers. In [24], a frequency allocation scheme using Non-Orthogonal Multiple Access (NOMA) was proposed in multi-layer LEO-GSO (Geosynchronous Orbit) satellite networks. In [25], the authors considered a satellite communication system with coexisting LEO and GSO, and addressed a joint beam-manage and power-allocation (JBMPA) problem by exploiting DQN to maximize Signal-to-Interference-plus-Noise Ratio (SINR).

Several versions of UCB algorithm have been also adopted for resource allocation problems in satellite networks. The

authors of [26] applied an UCB algorithm to the resource allocation in a space-air-ground integrated network, where LEO satellites provide backhaul connectivity to UAVs, and developed an optimal resource management scheme between UAVs and terrestrial base stations. Online task offloading in a space-air-ground network from a satellite to Internet-of-Things devices was investigated using UCB algorithms in [27]. Also in [28], the resource management scheme between LEO satellite constellations under jamming attack was developed by exploiting UCB algorithms. The authors of [28] addressed uplink transmissions to LEO satellites and manage the interference between the satellites from the users using UCB algorithms.

Different from aforementioned studies, we investigate the problem of downlink resource allocation in LEO satellite networks under dynamic inter-satellite interference, when there is no means of direct communications between different satellite constellations.

III. SYSTEM MODEL

We consider a downlink spectrum sharing system with multiple satellites, where the satellites share the same frequency spectrum range and provide downlink services to the users. There can be an interference if the service areas of multiple satellites overlap with each other and the satellites use the same frequency channel at the same time. We assume a time-slotted system. At each time, the satellite of interest, denoted by *control satellite*, makes the decision of downlink frequency resource allocation. We aim to maximize its throughput while managing the interference between the control satellite and other interfering satellites. Each interfering satellite operates separately and independently decides its frequency resource use. Since all the satellites move around the earth following their own orbits, the interference between the control satellite and those interfering satellites changes over time. We assume that the orbit information of the satellites might be available [29], but the information of the frequency channel used at each time is not likely to be shared. We consider the channel allocation problem of the control satellite under this dynamic interference environment.

We describe our model in detail. We assume that multiple satellites build a constellation, and within the same constellation, they travel at the same orbital velocity and manage their beam transmissions such that their service areas do not overlap with each other. This allows us to focus on the interference to the control satellite from the satellites of other constellation, denoted by *interfering satellites*.

Consider a ground area where several static users (or ground stations) are located as shown in Fig. 1. Both the control satellite and the interfering satellites go over the area following their own fixed orbit. We make two important assumptions on the spectrum usage of the interfering satellites: it is non-uniform over the frequency spectrum and geographically static. The non-uniform frequency usage can occur due to many different reasons, such as hardware

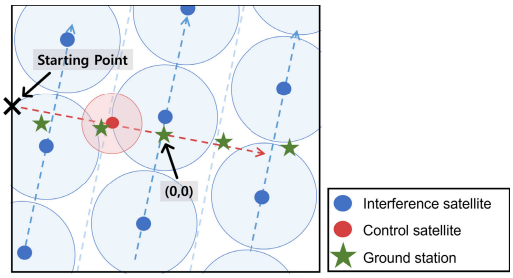


FIGURE 1. The orbit of the control satellite and its transmission range (red). We also illustrate the user locations (ground stations) by green stars, the constellation and orbits of interfering satellites, and their main-lobe beam range (blue). We also mark the coordinate origin and the starting point of the control satellite, respectively.

limitation, transmission power control, exclusive frequency allocation of different services, etc. The geographically static property can be observed when they provide seamless service to the ground users. Suppose that an interfering satellite in a location is associated with its users with a certain frequency usage pattern. As it moves and passes the location, the next satellite in the same constellation will come to the same location and succeed the service to the users, likely with the same frequency usage pattern. This result in a geographically-static frequency spectrum usage of the interfering satellites. With this, the control satellite will repetitively experience the same frequency usage pattern at the location, and thus can learn the pattern and avoid the interference.

We set the height[1] of the control satellite is set to 1000 km and that of the interfering satellites to 1414 km. The satellite velocity and the coverage area are set accordingly to their heights as in [29], revolving the earth in roughly 2 hours. We focus on our ground area, and once a satellite arrives at the area boundary following its orbit, then it reappears[2] at the other end and repeats the procedure. This not only provides a uniform interfering environment in the area, but also allows the control satellite to learn about the interference environment and to improve its decisions. For ease of exposition, we assume that all the satellites always set their beam steering perfectly. Also, for the control satellite, we assume that, at a time, it is associated with at most one ground user, which can experience an interference if the user is in the main lobe and/or side lobe areas of interfering satellites.

[1]According to the altitude, the satellites are classified into LEO at 200-2000 km, Medium Earth Orbit (MEO) at 2000-36000 km, and GEO at 36000 km. For a control satellite with higher altitude, our approach may not work, since, due to its slower movement, the users or the traffic request of the interfering satellites may change, and the traffic pattern is no longer geographically static.

[2]One can interpret the reappearance of the control satellite as the time-skipping to the next visit to the area, for ease of exposition. Or it can be considered as a virtualized environment such that the agent software that learns the interference and makes the resource allocation decision is separated from the physical satellite hardware. Once the control satellite arrives at the area boundary, the agent offs the satellite and moves through some communication means to another control satellite of the same orbit who is about to visit the area. In this case, the agent can make more repetitions and learn the interference pattern more quickly.

Time is slotted, and there are N equally-quantized channel blocks that are shared by all satellites. At each time, one frequency block can be used for the downlink service to a user. A satellite may provide its service to multiple users simultaneously, in which case it uses multiple blocks at a time. For ease of exposition, we assume that the control satellite has at most one associated user (other interfering satellites may use multiple blocks at a time), and thus it may select one frequency block to transmit a signal to the user or may give up the transmission. Extension to multiple users for the control satellite will be straightforward.

We explain the signal transmission procedure of the satellites as follows. There is the set $\mathcal{I}$ of interfering satellites and a control satellite, which share a set $\{1, 2, \ldots, N\}$ of frequency channel blocks.

1) At each time t, each interfering satellite $i \in \mathcal{I}$ selects frequency block j independently across blocks and satellites, to provide its service with probability p^{ij}. Interfering satellite i possibly chooses multiple blocks, and p^{ij}'s are unknown to the control satellite.
2) At the same time, the control satellite selects frequency block k for transmission. It may choose not-to-transmit at the time slot to avoid potential interference.
3) After the decisions, the signals are transmitted to associated users accordingly.
4) The ground user associated with the control satellite decodes the received signal during the time slot. If the received CINR (Carrier to Interference and Noise Ratio) is beyond a certain threshold γ, it successfully decodes the signal, and fails otherwise.
5) At the end of each time slot, the control satellite receives binary feedback about whether its transmission is successfully decoded or not. The feedback can be delivered to the satellite through direct uplink transmission or through a separate feeder-path transmission. We assume that the feedback is transmitted without error.

We compute the received CINR as $\frac{P_w}{P_u+P_n}$ where P_w is the power from the control satellite, P_u is the power sum from the interfering satellites, and P_n is the noise power. We make use of several design parameters which affect signal strength, e.g., antenna gains, attenuation, and effective isotropic radiated power (EIRP) [30]. The received signal is considered to be successfully decoded if the CINR is greater than a threshold γ. If the decoding fails, we say that a collision occurs and the frequency block at that time is wasted.

We formulate our problem. Let $\mathbf{E}_t^i = [e_t^{i1}, \cdots, e_t^{iN}]$ denote the vector of frequency block usage of interfering satellite $i \in \mathcal{I}$ at time t, where $e_t^{ij} = 1$ if satellite i transmits signal over frequency block j and $e_t^{ij} = 0$ otherwise. Each interfering satellite i transmits a signal at time t over frequency block j with probability p^{ij}. i.e., $e_t^{ij} = 1$ with probability p^{ij}. It satisfies $0 \le \sum_j e_t^{ij} \le N$ for each i and t. Let $\mathbf{P}^i = [p^{i1}, \cdots, p^{iN}]$ denote the transmission probability vector of interfering satellite i, which is fixed and unknown to the control satellite. On the other hand, the control satellite can select at most one frequency block at time t, and makes its decision based on its past experiences. Let $\mathbf{U}_t = [u_t^1, \cdots, u_t^N]$ denote the usage vector of the control satellite, where $u_t^j = 1$ if the control satellite transmits a signal over block j at time t and $u_t^j = 0$ otherwise. Note that it satisfies $\sum_j u_t^j \le 1$ and u_t^j can be 0 for all $j \in \{1, \cdots, N\}$ if it gives up the time slot.

Assume that the control satellite transmits a signal over frequency block j at time t and the user is located in the main/side lobe area of interfering satellite i that also uses block j, i.e., $e_t^{ij} = 1$ and $u_t^j = 1$ for some i and j. The user associated with the control satellite may and may not successfully decode the signal from the control satellite, which is determined by the received CINR. Let λ_t denote a binary for successful decoding at time t for the user associated the control satellite, which is set to

$$\lambda_t = \begin{cases} 1, & \text{if CINR} \ge \gamma, \\ 0, & \text{if CINR} < \gamma, \end{cases} \tag{1}$$

where γ is a fixed threshold. Our goal is to maximize the throughput performance of the control satellite while managing interference from interference satellites, which is

$$\begin{aligned} &\text{maximize} \quad \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\lambda_t\right], \\ &\text{such that} \quad \text{collision rate} \quad \le \bar{c}, \end{aligned} \tag{2}$$

where $\bar{c}$ is a constraint on the collision rate. The problem is challenging since there is *no direct information exchange between the control satellite and the interfering satellites*. The control satellite has to learn dynamic signal patterns quickly based on the location information of interfering satellites and users, and past experiences.

In this work, we explore several resource allocation strategies to solve (2) by taking different reinforcement learning methods. Also, we compare two different approaches to avoid the interference – giving a negative reward for a collision or directly constraining the collision rate. In the next sections, we describe our strategies and the reward settings.

IV. EXTENSION OF UCB ALGORITHM

At each time, the control satellite selects one frequency block aiming to maximize its throughput. Initially, it has no priori knowledge about which satellite will potentially interfere and which frequency blocks will be occupied. Considering the independent properties of block occupancy, it can be formulated as a Reinforcement Learning (RL) problem, in particular, a Multi-Armed Bandit (MAB) problem, in which the well-known Upper Confidence Bound (UCB) can be applied. However, the interference to the user of the control satellite changes over time according to the movement of the satellites, the original UCB algorithm cannot exploit this time-varying states and chooses one frequency block that achieves the best performance on average. To overcome this weakness, we also employ a couple of UCB variants,

i.e., Discounted UCB and Sliding-window UCB [17] that were suggested for non-stationary MAB problems. We also compare the performance of these UCB variants later in Section VI.

In the RL framework, the agent corresponding to the control satellite takes an action $\mathbf{U}_t$ at each time t, and gets feedback in the form of reward r_t at the end of the time slot. We design the reward value depending on whether the receiver successfully decodes the control satellite's signal or not, i.e.,

$$r_t = \begin{cases} 0, & \text{if } u_t^j = 0,\ \forall j \in [N], \\ \alpha, & \text{if CINR} \geq \gamma, \\ \beta, & \text{if CINR} < \gamma, \end{cases}$$

where $\alpha > 0$ and $\beta \leq 0$. That is, we set 0 reward for non-transmission, positive reward for a successful transmission, and non-positive reward for a collision.

Since an RL algorithm aims to maximize the return or the discounted reward sum in the infinite time horizon, the reward configuration will play the key role in managing interference. To this end, we examine different control approaches: indirect control on the collision rate by giving a negative reward, direct control by constraining the transmission when the collision rate is higher than $\bar{c}$, and both. Let $\mathbf{I}^{dc} \in \{0, 1\}$ denote whether we use direct control ($\mathbf{I}^{dc} = 1$) or not ($\mathbf{I}^{dc} = 0$).

From the perspective of the control satellite, let a_t denote the index of the chosen frequency block at time t, and let c_t denote the collision indicator due to the interference obtained from the feedback. We have $a_t = 0$ if none is chosen, and $a_t \in \{1, 2, \ldots, N\}$ otherwise. Depending on action a_t at time t and the existence of interference, we obtain the following outcome:

- If $a_t = 0$, then $\lambda_t = 0$ and $c_t = 0$.
- If $a_t \neq 0$, and
  - If $u_t^{a_t} = 0$ (this may happen when $\mathbf{I}^{dc} = 1$ and the collision constraint is violated, and thus the agent does not transmit), then $\lambda_t = 0$ and $c_t = 0$.
  - If $u_t^{a_t} = 1$ and received CINR $\geq \gamma$, then $\lambda_t = 1$ and $c_t = 0$.
  - If $u_t^{a_t} = 1$ and received CINR $< \gamma$, then $\lambda_t = 0$ and $c_t = 1$.

Combining with the signal transmission procedure in Section III, the policy evaluation procedure can be described as in Algorithm 1. Note that, under the direct control of collision rate, i.e., $\mathbf{I}^{dc} = 1$, we check the collision rate so far before transmission. To elaborate, after our non-zero decision a_t, if the collision rate is larger than the constraint $\bar{c}$, then we do not transmit (line 9 in Algorithm 1). At each iteration, we update the throughput and collision rate (lines 30-31 in Algorithm 1). Note that we define the collision rate as the transmission failure rate when there is an intention to transmit (i.e., $a_t \neq 0$) rather than the transmission failure rate when an actual transmission is made (i.e., $u_t^{a_t} = 1$). This is intended to decrease the collision rate when the satellite does not transmit due to the violation of the collision constraint. If we do not define the collision rate in this way, a high-collision rate beyond the threshold may never decrease.

Algorithm 1 Policy Evaluation Procedure

Input: policy π, collision constraint $\bar{c}$, control indicator $\mathbf{I}^{dc}$

1: **for** $t = 1, 2, \cdots$ **do**
2: User allocation for satellites
3: $\mathbf{E}_t^i$'s are determined using $\mathbf{P}^i$
4: $a_t \leftarrow$ decision under π
5: **if** $a_t = 0$ **then**
6: $u_t^j \leftarrow 0\ \forall j \in [N]$
7: **else**
8: **if** $\mathbf{I}^{dc} = 1$ and $collision(t-1) \geq \bar{c}$ **then**
9: $u_t^j \leftarrow 0\ \forall j \in [N]$
10: **else**
11: $u_t^{a_t} \leftarrow 1$ and $u_t^j \leftarrow 0\ \forall j \neq a_t$
12: **end if**
13: **end if**
14: Signal transmissions with $\mathbf{E}_t^i$ and $\mathbf{U}_t$
15: Satellites receive feedbacks from its users
16: **if** $a_t = 0$ **then**
17: $\lambda_t \leftarrow 0$ and $c_t \leftarrow 0$
18: **else**
19: **if** $u_t^{a_t} = 0$ **then**
20: $\lambda_t \leftarrow 0$ and $c_t \leftarrow 0$
21: **else**
22: **if** received CINR $\geq \gamma$ **then**
23: $\lambda_t \leftarrow 1$ and $c_t \leftarrow 0$
24: **else**
25: $\lambda_t \leftarrow 0$ and $c_t \leftarrow 1$
26: **end if**
27: **end if**
28: **end if**
29: Update internal parameters for π
30: $throughput(t) \leftarrow \frac{\sum_t \lambda_t}{t}$
31: $collision(t) \leftarrow \frac{\sum_t c_t}{\sum_t \mathbb{1}_{\{a_t \neq 0\}}}$
32: **end for**

Under UCB-based schemes, we can consider each frequency block as an arm; total $N + 1$ arms including no-transmission option. At each time t, an UCB-based scheme calculates the UCB index for each arm and selects the arm with the largest UCB index, which corresponds to line 4 in Algorithm 1. At the end of time, after the reward is known, we update internal parameters of the total reward and the number of selections, which is done at line 29 in Algorithm 1. The computation of the UCB index will be provided in the following.

A. ORIGINAL UCB

It has been shown that, in the finite-time horizon MAB problem, the UCB algorithm that selects the arm with the largest UCB index can achieve asymptotically optimal performance. Original UCB algorithm computes the index of

each arm a at time t as

$$\mathrm{UCB}_t^a = \eta_{t-1}^a + \sqrt{\frac{2\log t}{\tau_{t-1}^a}},$$
$$\eta_t^a = \frac{\sum_{z=1}^{t} r_z \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}}{\tau_t^a},$$

where η_t^a is the empirical average reward under arm a at time t, and $\tau_t^a = \sum_{z=1}^{t} \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}$ is the number of times when arm a was played until time t.

However, the original UCB algorithm is not suitable for our problem because it considers each reward equally regardless of how old it is. This time-agnostic behavior prevents it from taking into account the time-varying interference in our scenarios. To overcome the weakness, we adopt two UCB variants of Discounted UCB and Sliding-window UCB, which focus on near-future rewards and thus better capture the changes of time-varying environment.

B. DISCOUNTED UCB [17]

Discounted UCB computes the indexes by putting less weight to older rewards through discounting factor $0 < \zeta < 1$. Equations needed to calculate UCB indexes in Discounted UCB is as follows.

$$\overline{\mathrm{UCB}}_t^a = \overline{\eta}_{t-1}^a + \sqrt{\frac{\max(\overline{\eta}_{t-1}^a(1-\overline{\eta}_{t-1}^a), 0.002)\log \overline{\tau}_{t-1}}{\overline{\tau}_{t-1}^a}},$$
$$\overline{\eta}_t^a = \frac{\sum_{z=1}^{t} \zeta^{t-z} r_z \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}}{\overline{\tau}_t^a},$$

where $\overline{\tau}_t^a = \sum_{z=1}^{t} \zeta^{t-z} \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}$ and $\overline{\tau}_t = \sum_a \overline{\tau}_t^a$.

C. SLIDING-WINDOW UCB [17]

Another way to compute UCB indexes using recent rewards is applying a sliding-window. Let w be the size of sliding-window. Sliding-window UCB considers the rewards during recent w time slots to calculate the UCB index for each arm as

$$\widetilde{\mathrm{UCB}}_t^a = \widetilde{\eta}_{t-1}^a + \sqrt{\frac{2\log(\min(t,w))}{\widetilde{\tau}_{t-1}^a}},$$
$$\widetilde{\eta}_t^a = \frac{\sum_{z=t-w+1}^{t} r_z \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}}{\widetilde{\tau}_t^a},$$

where $\widetilde{\tau}_t^a = \sum_{z=t-w+1}^{t} \mathbb{1}_{\{a_z=a \text{ and } u_z^a=1\}}$.

Discounted UCB and Sliding-window UCB can forget some old information and focus on more recent rewards, and immediately achieve good performance if they are set appropriately. However, it is very challenging to adjust the discounting factor and the window size that yield good performance, which highly depends on the environment such as the changes of the interference frequency. We overcome the weakness of UCB variants, by formulating the problem as the more general MDP problem and taking deep reinforcement learning approach. In the next section, we model our problem as an MDP with dynamic interference states, and exploit the DQN algorithm.

V. DEEP REINFORCEMENT LEARNING APPROACH

We reconsider our problem of finding an optimal block to minimize interference from interfering satellites as an MDP. To this end, we first define the states and actions that are necessary to describe system dynamics, and then explain the DQN model and training method to efficiently learn the best frequency block allocation under the system dynamics.

A. MDP FORMULATION AND DQN

We first carefully design the state, action, reward, and discount factor. Then we introduce neural network model to maximize the cumulative rewards. We briefly overview its operations and loss function design.

- **State** s: Assuming that the location information of the satellites is available and there are M users on the ground area, we set the state as $(8+M)$ size vector that consists of the current location of the control satellite, the identifier of the user associated with the control satellite, and the location of all M users. The location is represented by (x, y) coordinates of the ground area. To stabilize DQN training, we normalize the state values scaled in $[-1, 1]$. Note that although it is not shown in Fig. 1, the control satellite also belongs to a constellation, and due to the satellites in its constellation, there is a time duration during which no user associates with the control satellite. These no-user-assigned states are treated as a terminal state.
- **Action** a**:** The action is the frequency block selected by the control satellite. We have $N+1$ actions including no transmission as in our UCB algorithms.
- **Reward** r**:** The reward is the evaluation of the action selected by the control satellite. In our case, it is the feedback from the user and depends on the decoding success or failure. We configure DQN with several reward value settings.
- **Discount factor** $\xi \in (0, 1]$**:** The discounting factor is the depreciation rate for reward when we accumulate it. A smaller discount factor implies quicker depreciation. We use a constant value $\xi = 0.9$ for DQN.

At each time t, given current state s_t, DQN selects an action a_t and obtains reward r_t as the feedback and accordingly the state changes from s_t to s_{t+1}. By repeating the action selections over time, DQN tries to learn the sequence of actions that yields the highest cumulative rewards. In the meantime, DQN approximately estimates the cumulative reward, called Q-value through a neural network. The introduction of neural network (behavior network), however, involves a couple of problems called temporal correlation between samples and non-stationary target. To mitigate the former problem, temporal correlation, DQN stores the sample experience (s_t, a_t, r_t, s_{t+1}) at each time t in a replay buffer D and trains the neural network through the experiences randomly chosen from the buffer. To address the latter problem, non-stationary target, it makes use of target network that has separate parameters from behavior network. Specifically, we train the behavior network by minimizing the

following loss function

$$L(\theta) = \mathbb{E}\Big[\big(r_t + \xi \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\big)^2\Big],$$

where the expectation is based on random sampling from the replay buffer. From the chosen experiences, we calculate the difference between the Q-value of the target network (θ^-) and that of the behavior network (θ), and then train the behavior network to minimize the difference. The target network parameter θ^- is an old copy of θ, which is periodically updated.

B. CONFIGURING DQN

We also develop a DQN scheme that can operate with and without the collision rate constraint. Basic computation of the collision rate is the same as in UCB algorithms. However, unlike UCB algorithms where one collision rate is computed, our DQN scheme maintains separate collision rate for each state, through which we intend for DQN not to violate the collision constraint. Note that the separate computation of collision rate can be also applied to our UCB algorithms. This, however, requires not only larger memory space but also long learning time, which substantially undermines the advantage of UCB algorithms – quick convergence. In contrast, DQN requires more sample experiences and it takes longer to complete the training and obtain convergent behaviors. Thus, the separate computation of collision rate is more suitable for DQN. Owing to the longer learning time, we will see later in Section VI that DQN indeed outperforms the UCB algorithms.

We train our DQN over 350, 000 steps and use ϵ-greedy as an exploration strategy which selects the estimated best action with probability $1 - \epsilon$ and a random action with probability ϵ. The ϵ decreases linearly from 0.95 to 0.05 during the first 280, 000 steps and then is fixed to 0 afterward. Further, we apply smooth update for the target network, where we set $\theta^- \leftarrow \upsilon\theta + (1 - \upsilon)\theta^-$ with weight $\upsilon = 0.005$.

For the neural networks that approximate Q-values, we set the input layer size to $(8 + M)$, the size of a state, and the output layer size to $N + 1$, the size of action space. There are two hidden layers, each of which has the size of 512. Each layer (except the output layer) uses ReLU as an activation function. We use the Adam optimizer, set the learning rate to 0.0003, and use the batch size of 1024 for stable learning of neural networks.

VI. SIMULATIONS

We evaluate the performance of our frequency allocation schemes through simulations. We consider a square ground with 600 km sides as shown in Fig. 1. The control satellite moves from left to right and the interfering satellites from bottom to top. Both are LEO satellites but with a different height as shown in Table 1. According to their height, they move at a different speed and have a different beam radius: the control satellite has a main beam radius of 50 km and the interfering satellites have a main beam radius of 75 km. For

TABLE 1. Simulation parameters [30].

Parameter	Symbol	Ctrl sat.	Int sat.
height (km)	h	1000	1414
downlink EIRP (dBW)	$EIRP_{down}$	-5	-5
Transmitter Gain (dBi)	G_T	2	2
Receiver Gain (dBi)	G_R	-	-
1st side lobe attenuation (dB)	ATT_1	-14	-14
2nd side lobe attenuation (dB)	ATT_2	-18	-18
the number of beams	b_n	1	1
beam radius (km)	b_r	50	75
noise temperature (K)	T	290	290
weather loss (dB)	l_w	0.5	0.5
scintillation loss (dB)	l_s	0.3	0.3

the interfering satellites, there are three orbits over the square ground, and interfering satellites travel following one of the orbits without overlapping their main lobe areas.

The following is the detailed movements for the interfering and control satellites. All the interfering satellites travel at the same speed and the main-beam point of each interfering satellite moves along a line $ax + by + c = 0$ on xy plane with $x, y \in [-300, 300]$. Let us number the orbits from the left. The interfering satellites on orbit 1 has $(a, b, c) = (-4, 1, -1000)$, those on orbit 2 has $(a, b, c) = (-4, 1, 0)$, and those on orbit 3 has $(a, b, c) = (-4, 1, 1000)$. On the other hand, the control satellite moves on the ground from left to right and the main-beam point of the control satellite follows the line $x + 4y = 0$. We also have 5 (fixed) ground users (or stations) that are located in $(-300, 60)$, $(-150, 30)$, $(0, 0)$, $(150, -30)$, and $(300, -60)$ over the xy plane respectively. The velocity of each satellite is set to $\sqrt{\frac{G*M}{h}}$ where $G = 6.6743 \cdot 10^{-11}(\mathrm{m}^3/\mathrm{kg} \cdot \mathrm{s}^2)$ is the gravitational Earth's constant, $M = 5.9742 \cdot 10^{24}(\mathrm{kg})$ is Earth's mass, and h is the height of the satellite [31]. The heights of the control satellite and the interfering satellites are set to 1000 km and 1414 km respectively.

For satellite communications, we consider the scintillation loss and the weather loss for atmospheric losses in the radio propagation. We also assume $N = 20$ blocks in the frequency band of 6.875GHz for downlink service and each block has bandwidth of 1.23MHz. All the frequency blocks are shared between the control and interfering satellites. The CINR received at the ground users can be computed as follows. Let P_T and P_R denote the signal power transmitted by the satellite and the signal power received at the ground user, respectively. They satisfy

$$P_R = P_T + G_R - l_w - l_s - l_{FSPL},$$
$$P_T = EIRP_{down} + G_T,$$

where l_{FSPL} is for free space path loss and other parameters are explained in Table 1. Letting P_S and P_I denote the power of wanted signal and the power of interference signal, respectively, we obtain the CINR as

$$CINR(dB) = 10\ \log_{10}(1 + \frac{P_S}{P_I + k \times T \times B}), \qquad (3)$$

where k is the Boltzmann constant and B is the transmission bandwidth. We refer other parameters for signal transmissions to [29].

We consider the first and second side lobes and ignore the subsequent side lobes for simplicity. We further assume that the first side lobe signal is attenuated by 14 dB with respect to the main lobe signal and the second side lobe signal by 18 dB due to the signal processing at the receiver. For the CINR threshold γ that determines successful transmission, if we set it too small, the first/second-side lobe interference may not affect signal decoding at all, and if we set it too large, any small side lobe signal can cause interference. It has to be set according to the level of coding and target block error rate [32]. In the simulations, we set $\gamma = 18$ for the case when the interference from a first or second-side lobe signal causes a collision, and additionally, consider $\gamma = 16$ for the case when the interference from a first side-lobe signal only causes a collision.

As the control satellite moves over its orbit, the user associated with the control satellite can be co-located in the main/side lobe areas of the interfering satellites. Depending on their frequency block allocation, the control satellite will experience different interference environments. We first consider a simple scenario where the interfering satellites on the same orbit have the same frequency block usage vector $\mathbf{P}^i$. Specifically, each interfering satellite has one of three different $\mathbf{P}^i$'s according to their orbit (remind that we consider three different orbits for interfering satellites). For each $\mathbf{P}^i$, we arbitrarily set probabilities for each frequency block[3] such that there are always a few blocks of mild interference at each time.

Fig. 2 shows the signal interference from the perspective of the control satellite. The control satellite moves on its orbit from the starting location, as marked in Fig. 1, at time 0. When it arrives at the right boundary in the ground area after about 2200 time slots (Note that we set duration of a time slot to 0.015 second.), it reappears at the starting location and repeats the movement. Note that as the control satellite moves, the ground user that it is associated with changes, which implies that the interfering environment changes too. The bottom figure of Fig. 2 shows the orbit number of interfering satellites whose main or side lobe overlaps with the main lobe of the control satellite at the ground user that the control satellite is currently associated with. We can observe that as the control satellite moves, the orbit of the interfering satellite changes too, and sometimes, multiple orbits are involved. There are some intervals when no orbit number is given, which are the time when the control satellite

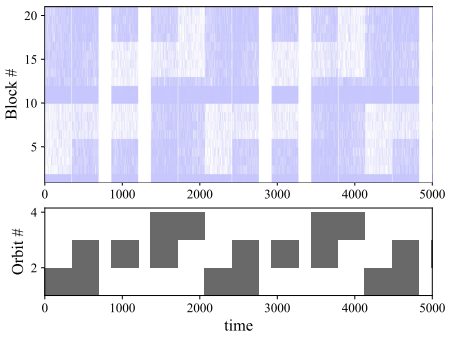


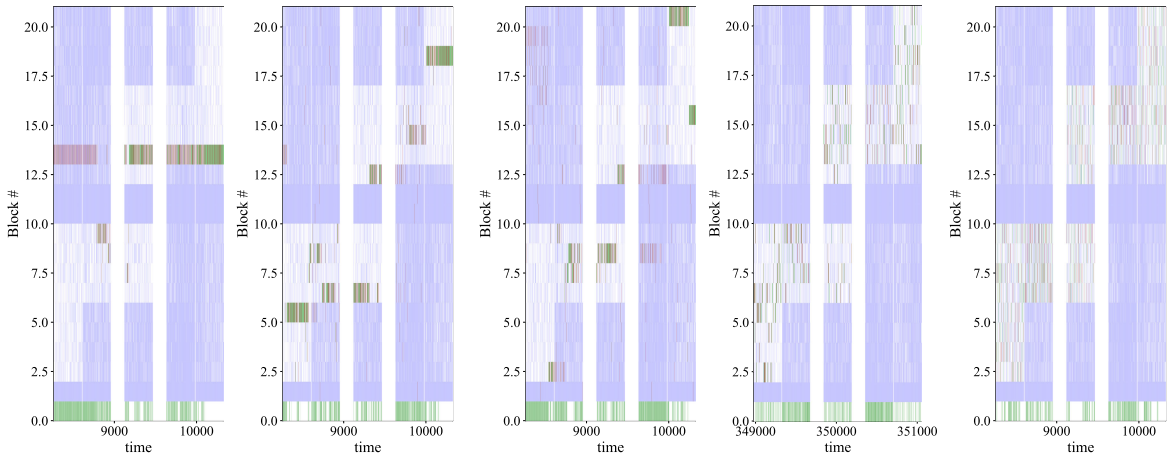
FIGURE 2. Frequency block usage and orbits of interfering satellites.

is not associated with any ground user.[4] The top figure of Fig. 2 illustrates the frequency block usage (or transmission) by the interfering satellites with an overlapped beam with the control satellite. If one of interfering satellites with the overlapped beam makes a transmission over block j at time t, we draw a bar between j and $j+1$ in y-axis at time t. Again, a vertical blank area denotes the time when the control satellite is not associated with any ground user. Fig. 2 clarifies which frequency blocks the control satellite has to choose at each time. For example, during (0, 350), the frequency blocks in range [1, 9] are less used than other blocks, and thus one of the blocks will be the best option.

We evaluate our proposed frequency allocation schemes that are based on Original UCB, Discounted UCB, Sliding-window UCB, and DQN, under different reward and constraint settings. In the case that the collision constraint is not directly applied (i.e., when $\mathbf{I}^{dc} = 0$), we manage the collision rate of the control satellite by applying a negative reward on a collision. Note that setting the reward values to optimize the performance is an interesting open problem. Several reward shaping techniques, e.g., potential-based reward shaping [33], adversarial inverse reinforcement learning [34], learning reward shaping [35], could be applied. However, optimizing the reward value is out of the scope of this paper, and we use hand-picked reward values of $(\alpha, \beta) = (1, -1), (10, -1), (1, -10)$, where a smaller β implies a heavier penalty and stronger refraining from transmitting. In the case that the collision constraint is directly applied (i.e., when $\mathbf{I}^{dc} = 1$), we use $(\alpha, \beta) = (1, 0)$ and the collision constraint $\bar{c} = 0.2$. We also attempt to apply the collision constraint indirectly and directly at the same time, in which case we use $(\alpha, \beta) = (10, -3)$ and $\bar{c} = 0.2$. As a result, we have total 5 different settings of $(\alpha, \beta, \psi) \in \{(1, -1, \cdot), (10, -1, \cdot), (1, -10, \cdot), (1, 0, 0.2), (10, -3, 0.2)\}$, where $\psi = \cdot$ implies no direct collision constraining, and $\psi = \bar{c}$ indicates direct collision constraining by $\bar{c}$.

[3] We set $\mathbf{P}^i = [1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 1, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$ for the interfering satellites on the leftmost orbit in Fig. 1, $\mathbf{P}^i = [1, 0.5, 0.5, 0.5, 0.5, 0.1, 0.1, 0.1, 0.1, 1, 1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.5, 0.5, 0.5, 0.5]$ for those on the middle orbit, and $\mathbf{P}^i = [1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1, 1, 0.5, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$ for those on the rightmost orbit.

[4] Although it is not shown in Fig. 1, there are other satellites that belong to the same constellation with the control satellite and they are assumed to be controlled to avoid any interference with the control satellite through direct message exchange. Each ground user is being served by one of these satellites. When the control satellite is serving no ground user, the other satellites in the same constellation cover all the ground user.



(a) Original UCB (b) Discounted UCB (c) Sliding-window UCB (d) DQN (e) Oracle

FIGURE 3. Frequency block selection when $(\alpha, \beta, \psi) = (10, -3, 0.2)$. A light blue bar between block $[x, x+1]$ denotes a transmission of interfering satellites on frequency block x, a red bar denotes a collision, and a green bar a successful transmission or no transmission of the control satellite.

As a baseline algorithm for performance comparison, we consider Oracle that has prior information about the channel statistics $\mathbf{P}^i$ of all the interfering satellites and knows which satellites interfere with the control satellite at each time slot. Let $\mathbf{C}_t$ be the set of satellites that interfere with the control satellite at time t. Oracle selects the frequency block that is least likely used (or most likely unused) by interference satellites in $\mathbf{C}_t$, i.e.,

$$\arg\max_j \prod_{i \in \mathbf{C}_t} (1 - p^{ij}).$$

If there is a tie, Oracle selects one block uniformly at random. For the Discounted UCB scheme and the Sliding-window UCB scheme, we empirically search for the best hyper-parameters of the discounting factor ζ and the window size w, and we set $\zeta = 0.99$ and $w = 500$, respectively. For the UCB-based schemes and Oracle, we run them about $10,000$ time slots for each simulation, and after each simulation, we measure throughput and collision rate computed as in lines 30 and 31 in Algorithm 1. For the DQN scheme, we run it for $350,000$ time slots and measure the throughput and collision rate using the last $70,000$ slots (when $\epsilon = 0$).

Fig. 3 illustrates the resource allocation of the control satellite for each scheme when $(\alpha, \beta, \psi) = (10, -3, 0.2)$. The transmissions of the control satellite are represented with the green or red bars. A bar between j and $j+1$ in y-axis represents the transmission of the control satellite using frequency block j at that time. Exceptionally, the bar between 0 and 1 in y-axis indicates the control satellite does not transmit any signals during the time. The green bars imply a successful transmission (the associated user successfully decoded the signal from the control satellite) or non-transmission, and the red bars indicate a collision. The light blue bars are transmissions of the interfering satellites as the top figure in Fig. 2. In Fig. 3, under all schemes, there are lots of green bars between 0 and 1 in y-axis, which means the control satellite frequently refrains from transmitting signals in order to satisfy the collision constraint. Furthermore, Original UCB experiences many collisions (i.e., many red bars) and cannot immediately adjust its transmission decision according to changes in the interfering environment. Discounted UCB and Sliding-window UCB select frequency blocks in a more diverse manner than Original UCB and experience slightly less collisions. DQN shows the most diverse frequency block selection[5] and also has the least collisions among the proposed schemes (i.e., among Original UCB, Discounted UCB, Sliding-window UCB, and DQN).

TABLE 2. Performance comparison of the proposed schemes and Oracle under the chosen $\mathbf{P}^i$'s.

Average throughput

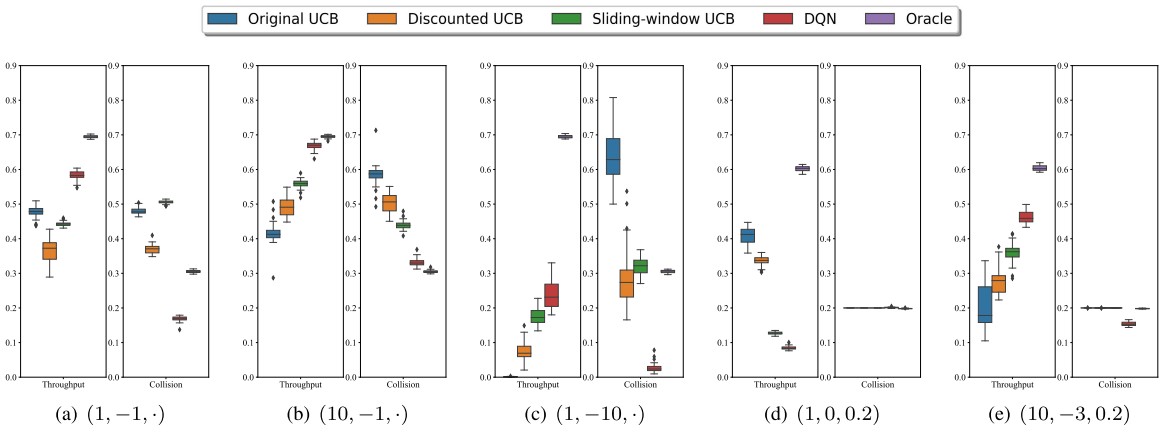
(α, β, ψ)	$(1, -1, \cdot)$	$(10, -1, \cdot)$	$(1, -10, \cdot)$	$(1, 0, 0.2)$	$(10, -3, 0.2)$
Original UCB	0.63	0.52	0	0.33	0.30
Discounted UCB	0.65	0.66	0.03	0.31	0.40
Sliding-window UCB	0.55	0.64	0.10	0.18	0.37
DQN	**0.70**	**0.70**	0.01	**0.46**	**0.47**
Oracle	0.70	0.70	0.70	0.51	0.51

Average collision rate

(α, β, ψ)	$(1, -1, \cdot)$	$(10, -1, \cdot)$	$(1, -10, \cdot)$	$(1, 0, 0.2)$	$(10, -3, 0.2)$
Original UCB	0.37	0.48	0.34	**0.20**	**0.20**
Discounted UCB	0.34	0.34	0.47	**0.20**	**0.20**
Sliding-window UCB	0.42	0.36	0.44	**0.20**	**0.20**
DQN	0.30	0.30	0	**0.19**	**0.19**
Oracle	0.30	0.30	0.30	0.20	0.20

We repeat the simulation 30 times for each scheme and setting (α, β, ψ), and evaluate schemes using average throughput and average collision rate. The results are summarized in Table 2. Average throughput results that show good performance compared to Oracle are displayed in bold and average collision rate results that satisfy the collision constraint are also marked in bold. Original UCB shows good performance when $(\alpha, \beta, \psi) = (1, -1, \cdot)$, and Discounted

[5]Since the DQN data are collected during the testing, its time is different from the others in Fig. 3.



(a) $(1, -1, \cdot)$ (b) $(10, -1, \cdot)$ (c) $(1, -10, \cdot)$ (d) $(1, 0, 0.2)$ (e) $(10, -3, 0.2)$

FIGURE 4. Comparison of throughput and collision rate of proposed schemes and Oracle, under randomized transmission probabilities $\mathbf{P}^i$'s. In each setting of (α, β, ψ), the left shows the throughput and the right shows average collision rate. Without directly applying the collision constraint, it is hard to limit the interference level, and the schemes suffer from high collision rate or low throughput.

UCB and Sliding-window UCB slightly outperform Original UCB with high positive reward $(\alpha, \beta, \psi) = (10, -1, \cdot)$. All the schemes, except Oracle that is not affected by the reward setting, achieve poor performance under high negative reward $(\alpha, \beta, \psi) = (1, -10, \cdot)$. We can also observe that high negative reward is not effective in lowering the collision rate. In contrast, the direct constraining on the collision rate (i.e., when $\psi = 0.2$) has an immediate impact on the collision rate, and all the schemes satisfy the collision rate under the constraint. Interestingly, most proposed schemes perform better when they used both direct and indirect control of the collision rate than when they use only direct control. That is, most proposed schemes achieve better performance under $(\alpha, \beta, \psi) = (10, -3, 0.2)$ than under $(\alpha, \beta, \psi) = (1, 0, 0.2)$. Among the proposed schemes, DQN achieves the highest throughput 0.47, which is comparable to the throughput of Oracle, 0.51.

Next we evaluate the proposed schemes under randomly generated transmission probability matrices, $\mathbf{P}^i$'s. To elaborate, we generate three different transmission probability matrices, each of which is used for transmissions of the interfering satellites following the same orbit. Let $\mathbf{P}_k = [p_k^1, \cdots, p_k^N]$ be the transmission probability vector of interfering satellites on an orbit $k \in \{1, 2, 3\}$. (i.e., $\mathbf{P}^i = \mathbf{P}_k$ if the interfering satellite i moves around the ground following the orbit k.) For each $\mathbf{P}_k$, among $N = 20$ elements, we randomly select 5 elements and give them random variables following a uniform distribution $U(0, 0.5)$, and give random variables following a uniform distribution $U(0, 1)$ to the remaining 15 elements.[6] Two different uniform distributions are used to ensure that frequency blocks with low probability of being used by the interference satellite are guaranteed so that the control satellite can obtain the evident performance gain by choosing the optimal block through learning. Except $\mathbf{P}^i$'s, other parameters for simulations remain same. Under this interference environment, we evaluate each scheme by running 30 simulations for each (α, β, ψ) and measure average throughput and collision rate.

[6]We fix $\mathbf{P}_1 = [0.84, 0.76, 0.16, 0.24, 0.51, 0.40, 0.31, 0.30, 0.48, 0.58, 0.18, 0.50, 0.28, 0.76, 0.62, 0.25, 0.91, 0.98, 0.81, 0.90]$, $\mathbf{P}_2 = [0.97, 0.48, 0.22, 0.26, 0.81, 0.55, 0.01, 0.72, 0.40, 0.82, 0.67, 0.001, 0.49, 0.87, 0.24, 0.50, 0.50, 0.05, 0.57, 0.24]$, and $\mathbf{P}_3 = [0.29, 0.07, 0.33, 0.92, 0.20, 0.80, 0.55, 0.14, 0.09, 0.80, 0.32, 0.24, 0.18, 0.82, 0.03, 0.98, 0.34, 0.42, 0.42, 0.13]$.

TABLE 3. Performance comparison of the proposed schemes and Oracle under randomized $\mathbf{P}^i$'s.

Average throughput

(α, β, ψ)	$(1, -1, \cdot)$	$(10, -1, \cdot)$	$(1, -10, \cdot)$	$(1, 0, 0.2)$	$(10, -3, 0.2)$
Original UCB	0.48	0.42	0	0.41	0.20
Discounted UCB	0.38	0.50	0.06	0.34	0.28
Sliding-window UCB	0.44	0.56	0.17	0.13	0.36
DQN	**0.58**	**0.67**	0.24	0.08	**0.46**
Oracle	0.70	0.70	0.70	0.60	0.60

Average collision rate

(α, β, ψ)	$(1, -1, \cdot)$	$(10, -1, \cdot)$	$(1, -10, \cdot)$	$(1, 0, 0.2)$	$(10, -3, 0.2)$
Original UCB	0.48	0.58	0.64	**0.20**	**0.20**
Discounted UCB	0.37	0.51	0.30	**0.20**	**0.20**
Sliding-window UCB	0.51	0.44	0.32	**0.20**	**0.20**
DQN	0.17	0.33	0.03	**0.20**	**0.15**
Oracle	0.31	0.31	0.31	0.20	0.19

TABLE 4. Performance evaluation with different CINR threshold $\gamma = 16$ under $(\alpha, \beta, \psi) = (10, -3, 0.2)$.

Scheme	Chosen $\mathbf{P}^i$'s		Randomized $\mathbf{P}^i$'s	
	Throughput	Collision	Throughput	Collision
Original UCB	0.60	**0.20**	0.39	**0.20**
Discounted UCB	**0.80**	**0.18**	0.64	**0.20**
Sliding-window UCB	**0.77**	**0.20**	0.62	**0.20**
DQN	**0.76**	**0.15**	**0.77**	**0.11**
Oracle	0.83	0.17	0.88	0.11

Table. 3 and Fig. 4 shows the performance of the schemes under the randomized $\mathbf{P}^i$'s. Table. 3 provides the average throughput and average collision rate for each scheme under different settings of (α, β, ψ) and Fig. 4 plots them through graphs. In Fig. 4, for each (α, β, ψ), the left graph represents the average throughput and the right graph represents the average collision rate. We observe that the results under randomized $\mathbf{P}^i$'s are similar to those under the arbitrarily chosen $\mathbf{P}^i$'s by comparing Table. 3 with Table 2. One exception is that DQN performs worse when $(\alpha, \beta, \psi) = (1, 0, 0.2)$. Nonetheless, DQN outperforms the

other UCB-based schemes in general, and achieves even smaller collision rate 0.15 than the constraint $\bar{c} = 0.2$ when $(\alpha, \beta, \psi) = (10, -3, 0.2)$, which can be some room for improvement. We also note that without direct constraint of the collision rate, it is hard to manage the interference below a certain level. Depending on schemes and the reward setting (α, β), the control satellite experiences a different collision rate. Finally, we remark the importance of the setting of (α, β, ψ) because it may have a significant impact on performance. Proper positive rewards for successful transmissions and negative rewards for collisions seem to be necessary to achieve high throughput performance along with direct control on the collision rate to manage the interference under a certain level.

Additionally, we consider different value for the CINR threshold. We set $\gamma = 16$ with which a first side-lobe signal of interfering satellite causes a collision at the control satellite, while a second side-lob signal cannot (multiple second side-lob signals can cause a collision). It can be achieved by advanced signal processing techniques. We evaluate the proposed schemes and Oracle when $(\alpha, \beta, \psi) = (10, -3, 0.2)$, both under the previous arbitrarily-chosen $\mathbf{P}^i$'s and randomized $\mathbf{P}^i$'s. The results are shown in Table. 4. We mark average throughput comparable to that of Oracle, and average collision rate satisfying the collision constraint in bold. As expected, we can observe that all schemes have higher average throughput when $\gamma = 16$ compared with those when $\gamma = 18$. Under the arbitrarily chosen $\mathbf{P}^i$'s, Discounted UCB, Sliding-window UCB, and DQN show good performance, and under the randomized $\mathbf{P}^i$'s, DQN has the best performance. Interestingly, DQN not only satisfies the collision constraint but also achieves a much lower collision rate than the constraint.

Overall, we observe that DQN achieves the highest performance without a prior knowledge about the statistic information of the interfering satellites while satisfying the interference constraint. However, it requires long training time to learn the Q-values. Discounted UCB and Sliding-window UCB can be preferred when we have a tight time budget for training, or they can be used as a preliminary scheme during the initial time period before completing the training for DQN.

VII. CONCLUSION

We consider multi-group LEO satellite networks that provide downlink services to ground users and share the scarce frequency resources. As LEO satellites move following different orbits, the ground users would suffer from time-varying interference due to main/side lobe effects of multiple satellites at the same time. We investigate effective interference management methods without direct communication between different satellite constellations. To elaborate, we develop learning-based frequency allocation schemes that aim to maximize the throughput while managing the dynamic interference. By exploiting statistical learning techniques of UCB variants and deep reinforcement learning techniques of DQN, we develop several learning-based schemes by revising them accordingly to provide direct constraints on the collision rate. We evaluate their performance through extensive simulations with different settings of rewards and ways of constraining the collision rate. In a nutshell, while it requires longer learning time, DQN achieves the best performance when the rewards are set appropriately.

It is interesting to extend our results to a scenario where the control satellite provides services to multiple users, and each user faces a different interference environment. The problem becomes more challenging when the users' attendance is dynamically determined. Another interesting direction is the interference between satellites belonging to the same constellation or the interference from terrestrial communication systems. The performance gain and loss incurred by the coordination between the system can be of great interest.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020.

[2] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar./Apr. 2021.

[3] 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.

[4] S. Kota and G. Giambene, "6G integrated non-terrestrial networks: Emerging technologies and challenges," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[5] J. P. Choi and V. W. S. Chan, "Resource management for advanced transmission antenna satellites," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1308–1321, Mar. 2009.

[6] J. Yun, T. An, H. Jo, B.-J. Ku, D. Oh, and C. Joo, "Downlink spectrum sharing of heterogeneous communication systems in LEO satellite networks," *J. Commun. Netw.*, vol. 24, no. 6, pp. 722–729, Dec. 2022.

[7] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002.

[8] S. Kang and C. Joo, "Low-complexity learning for dynamic spectrum access in multi-user multi-channel networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3267–3281, Nov. 2021.

[9] J. Yun, A. Eryilmaz, and C. Joo, "Remote tracking of dynamic sources under sublinear communication costs," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–6.

[10] D. Park, S. Kang, and C. Joo, "A learning-based distributed algorithm for scheduling in multi-hop wireless networks," *J. Commun. Netw.*, vol. 24, no. 1, pp. 99–110, Feb. 2022.

[11] Z. Lin, Z. Ni, L. Kuang, C. Jiang, and Z. Huang, "Dynamic beam pattern and bandwidth allocation based on multi-agent deep reinforcement learning for beam hopping satellite systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 3917–3930, Apr. 2022.

[12] S. Tang, Z. Pan, G. Hu, Y. Wu, and Y. Li, "Deep reinforcement learning-based resource allocation for satellite Internet of Things with diverse QoS guarantee," *Sensors*, vol. 22, no. 8, p. 2979, Apr. 2022.

[13] A. Yastrebova, I. Angervuori, N. Okati, M. Vehkapera, M. Hoyhtya, R. Wichman, and T. Riihonen, "Theoretical and simulation-based analysis of terrestrial interference to LEO satellite uplinks," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[14] L. H. Grokop and D. N. C. Tse, "Spectrum sharing between wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1401–1412, 2010.

[15] S. Kang, C. Joo, J. Lee, and N. B. Shroff, "Pricing for past channel state information in multi-channel cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 859–870, Apr. 2018.

[16] L. Kocsis and C. Szepesvári, "Discounted UCB," in *Proc. 2nd PASCAL Challenges Workshop*, vol. 2, Apr. 2006, pp. 51–134.
[17] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," 2008, *arXiv:0805.3415*.
[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[19] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[20] R. Zhang, Y. Ruan, Y. Li, and C. Liu, "Interference-aware radio resource management for cognitive high-throughput satellite systems," *Sensors*, vol. 20, no. 1, p. 197, Dec. 2019.
[21] F. Zheng, Z. Pi, Z. Zhou, and K. Wang, "LEO satellite channel allocation scheme based on reinforcement learning," *Mobile Inf. Syst.*, vol. 2020, pp. 1–10, Dec. 2020.
[22] B. Zhao, J. Liu, Z. Wei, and I. You, "A deep reinforcement learning based approach for energy-efficient channel allocation in satellite Internet of Things," *IEEE Access*, vol. 8, pp. 62197–62206, 2020.
[23] Y. Li, N. Deng, and W. Zhou, "A hierarchical approach to resource allocation in extensible multi-layer LEO-MSS," *IEEE Access*, vol. 8, pp. 18522–18537, 2020.
[24] R. Ge, D. Bian, J. Cheng, K. An, J. Hu, and G. Li, "Joint user pairing and power allocation for NOMA-based GEO and LEO satellite network," *IEEE Access*, vol. 9, pp. 93255–93266, 2021.
[25] X. Ding, Z. Ren, H. Lu, and G. Zhang, "Improving SINR via joint beam and power management for GEO and LEO spectrum-sharing satellite communication systems," *China Commun.*, vol. 19, no. 7, pp. 25–36, Jul. 2022.
[26] A. H. Arani, P. Hu, and Y. Zhu, "Re-envisioning space-air-ground integrated networks: Reinforcement learning for link optimization," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–7.
[27] H. Liao, Z. Zhou, and X. Zhao, "Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5250–5263, Apr. 2021.
[28] C. Han, A. Liu, L. Huo, H. Wang, X. Liang, and X. Tong, "Distributed resource management framework for IoS against malicious jamming," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8271–8286, Dec. 2021.
[29] *International Telecommunication Union*, document Recommendation ITU-R S.1328–2, Recommendation 1328-2, Geneva, Switzerland, 2002. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/s/R-REC-S.1328-2-200001-S!!PDF-E.pdf
[30] I. Leyva-Mayorga, B. Soret, M. Röper, D. Wübben, B. Matthiesen, A. Dekorsy, and P. Popovski, "LEO small-satellite constellations for 5G and beyond-5G communications," *IEEE Access*, vol. 8, pp. 184955–184964, 2020.
[31] B. G. Evans, *Satellite Communication Systems*, vol. 38. Edison, NJ, USA: IET, 1999.
[32] E. Chu, J. Yoon, and B. Jung, "A novel link-to-system mapping technique based on machine learning for 5G/IoT wireless networks," *Sensors*, vol. 19, no. 5, p. 1196, Mar. 2019.
[33] Y. Gao and F. Toni, "Potential based reward shaping for hierarchical reinforcement learning," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3504–3510.
[34] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adverserial inverse reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=rkHywl-A-
[35] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2020, pp. 15931–15941.

JIHYEON YUN received the M.S. degree from the School of Electrical and Computer Engineering (ECE), Ulsan National Institute of Science and Technology (UNIST), in 2019. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering (CSE), Korea University. Her research interests include remote estimation and sensor networks.

TAEGUN AN received the B.S. degree in computer science from the Ulsan National Institute of Science and Technology (UNIST). He is currently pursuing the Ph.D. degree with Korea University. His research interests include reinforcement learning (RL) and neural architecture search (NAS).

HAESUNG JO received the B.S. degree in computer engineering from Konkuk University, in 2021. He is currently pursuing the M.S. degree with Korea University. His research interests include machine learning and reinforcement learning.

BON-JUN KU received the Ph.D. degree from Chungbuk National University, in 2010. Since 1999, he has been with the Electronics and Telecommunications Research Institute (ETRI). He is currently a Principal Member of Research Staff with Satellite Communication Research Division, ETRI. His research interests include satellite communications systems, high-altitude platform station (HAPS), and wireless communication system engineering.

DAESUB OH received the Ph.D. degree from Jeonbuk National University, in 2014. He has been with the Electronics and Telecommunications Research Institute (ETRI), since 2000. His research interests include the broad areas of satellite communications, wireless communications, and spectrum management.

CHANGHEE JOO (Senior Member, IEEE) received the Ph.D. degree from Seoul National University, in 2005. He was with Purdue University and The Ohio State University, and then with the Korea University of Technology and Education (KoreaTech) and the Ulsan National Institute of Science and Technology (UNIST). Since 2019, he has been with Korea University. His research interests include the broad areas of networking and machine learning. He served as the Technical Program Committee Member for many primary conferences, including IEEE INFOCOM, ACM MobiHoc, IEEE WiOpt, and IEEE GLOBECOM. He was a recipient of the IEEE INFOCOM Best Paper Award, in 2008, the KICS Haedong Young Scholar Award, in 2014, the ICTC Best Paper Award, in 2015, and the GAMENETS Best Paper Award, in 2018. He was the General Co-Chair of the ACM MobiHoc, in 2022. He is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and IEEE/ACM TRANSACTIONS ON NETWORKING. He is a Division Editor of the *Journal of Communications and Networks*.