future internet

MDPI

Article

Deep Reinforcement Learning Evolution Algorithm for Dynamic Antenna Control in Multi-Cell Configuration HAPS System

Siyuan Yang [1,*], Mondher Bouazizi [2], Tomoaki Ohtsuki [2], Yohei Shibata [3], Wataru Takabatake [3], Kenji Hoshino [3] and Atsushi Nagate [3]

[1] Graduate School of Science and Technology, Keio University, Yokohama 223-8522, Japan
[2] Department of Information and Computer Science, Faculty of Science and Technology, Keio University, Yokohama 223-8522, Japan; bouazizi@ohtsuki.ics.keio.ac.jp (M.B.); ohtsuki@ics.keio.ac.jp (T.O.)
[3] SoftBank Corp. Technology Research Laboratory, Tokyo 135-0064, Japan; yohei.shibata02@g.softbank.co.jp (Y.S.); wataru.takabatake@g.softbank.co.jp (W.T.); kenji01.hoshino@g.softbank.co.jp (K.H.); atsushi.nagate@g.softbank.co.jp (A.N.)
* Correspondence: yang@ohtsuki.ics.keio.ac.jp

Abstract: In this paper, we propose a novel Deep Reinforcement Learning Evolution Algorithm (DRLEA) method to control the antenna parameters of the High-Altitude Platform Station (HAPS) mobile to reduce the number of low-throughput users. Considering the random movement of the HAPS caused by the winds, the throughput of the users might decrease. Therefore, we propose a method that can dynamically adjust the antenna parameters based on the throughput of the users in the coverage area to reduce the number of low-throughput users by improving the users' throughput. Different from other model-based reinforcement learning methods, such as the Deep $\mathcal{Q}$ Network (DQN), the proposed method combines the Evolution Algorithm (EA) with Reinforcement Learning (RL) to avoid the sub-optimal solutions in each state. Moreover, we consider non-uniform user distribution scenarios, which are common in the real world, rather than ideal uniform user distribution scenarios. To evaluate the proposed method, we do the simulations under four different real user distribution scenarios and compare the proposed method with the conventional EA and RL methods. The simulation results show that the proposed method effectively reduces the number of low throughput users after the HAPS moves.

Keywords: HAPS; antenna control; reinforcement learning; evolution algorithm

Citation: Yang, S.; Bouazizi, M.; Ohtsuki, T.; Shibata, Y.; Takabatake, W.; Hoshino, K.; Nagate, A. Deep Reinforcement Learning Evolution Algorithm for Dynamic Antenna Control in Multi-Cell Configuration HAPS System. *Future Internet* **2023**, *15*, 34. https://doi.org/10.3390/fi15010034

Academic Editors: Kien Nguyen, Mikio Hasegawa, Hiroo Sekiya and Kentaro Ishizu

Received: 3 December 2022
Revised: 3 January 2023
Accepted: 9 January 2023
Published: 12 January 2023

1. Introduction

High platform station (HAPS) provides extremely broad coverage regions and a powerful line-of-sight (LoS) connectivity to terrestrial user equipment (UE) at the ground. As early as the 1990s, HAPS began to be paid attention to and studied through numerous research perspectives [1].

Compared with Geostationary Earth Orbit (GEO) satellites that are orbiting at a height of about 36,000 km and Low Earth Orbit (LEO) satellites that are orbiting at a height of about 1200 km, HAPS operates at the stratosphere at heights between 20 and 50 km. Therefore, the Round Trip Time (RTT) of HAPS is much faster than that of GEO and LEO satellites. Furthermore, since HAPS is relatively close to the ground, the power density is approximately one million times that of a GEO satellite and approximately ten thousand times that of a LEO satellite, allowing HAPS to provide high-quality communication services to existing mobile devices [2]. In addition, compared to other systems such as Starlink, which operates at an altitude of 340 km to 550 km [3], HAPS is much less expensive in terms of both the launch and the communication costs. Nonetheless, it does not "pollute" the upper layers of the atmosphere with the space waste it creates.

It is recognized as one of the hot topics for Beyond 5G (B5G) and 6G mobile communications [4–8]. With the necessary advanced materials and technological leaps, HAPS has been discussed as a viable technique.

With their potential and with the decrease in the cost of the technology behind it, they are expected to be massively deployed for consumer usage in the coming years as candidates for cellular coverage to provide service or to augment the capacity of other broadband service providers [9].

S. Karapantazis et al. [5] and A. K. Widiawan et al. [10] summarized the essential technical aspects of HAPS systems and the current and potential applications of HAPS. In [11], the authors studied the potential HAPS system architectures and deployment strategies in order to achieve global connectivity.

In [12], White et al. studied the possibility of using HAPS to provide high data rate communications simultaneously to a number of trains in motion. According to the estimated and tracked Direction-of-Arrival (DoA) at the HAPS, they can control the parameters of the antenna array at the HAPS to transmit the beam to the UEs.

2. Related Work

However, due to wind pressure, it is difficult for HAPS to remain stationary. Thus, the degradation of the users' throughput and handovers to UEs' end happened [13,14] after the HAPS coverage shifting. This kind of quasi-stationary state seriously impacts the performance of the communication system [15]. Dessouky et al. [16,17] researched the problem of maximization of coverage through optimization of the parameters of the HAPS antenna arrays, and proposed an optimized way to minimize both the coverage gaps between cells and the excessive cell overlap. Yasser et al. [18] studied the influence of handover performance when the HAPS is moved or rotated by winds. He et al. [19] examined the swing state modeling of the cellular coverage geometry model and the influence of swing on handover. Many studies [11,20–28] on antenna control of HAPS proposed employing antenna control methods to prevent interference between surrounding cells and HAPSs to alleviate the decrease in received signal power caused by HAPS shifting or rotation. Kenji et al. [20] proposed a beamforming method to reduce the impact of the degradation of system capacity caused by handover between two cells. Florin et al. [21] analyzed the concentric circular antenna array (CCAA) and proposed a Genetic Algorithm (GA) to minimize the maximum side-lobe level (SLL). In [22], Sun et al. further developed the discrete cuckoo search algorithm (IDCSA) used to reduce the maximum SLL under the constraint of a particular half-power bandwidth. To increase the system capacity, Dib et al. [23] also researched SLL reduction. In contrast to existing approaches, they proposed a Symbiotic Organism Search (SOS) algorithm. The SOS algorithm requires no tuning of parameters, which makes it an attractive optimization method. In [24], the particle swarm optimization (PSO) GA is used for reducing the SLL to improve the carrier-to-interference ratio (CIR). However, in high-dimensional space, PSO is easy to enter a local optimum, such as other GAs, and the iterative process' convergence rate is low. These limitations motivate us to develop a new method with high convergence and better throughput performance.

With the rapid development of deep learning techniques, reinforcement learning (RL) is widely used in various fields including in 5G and B5G [29–31]. F. B. Mismar et al. in [32] used Deep Q-Network (DQN) for online learning on how to maximize the users' signal-to-interference plus noise ratio (SINR) and sum-rate capacity. The authors design a binary encoding for performing multiple relevant actions at once in the DQN structure. A. Rkhami et al. in [33] used the RL method to solve the virtual network embedding problem (VNEP) in 5G and B5G. The authors considered that the conventional Deep Reinforcement Learning (DRL) usually obtains the sub-optimal solutions of VNEP, which leads to inefficient utilization of the resources and increases the cost of the allocation process. Thus, they proposed a relational graph convolutional neural network (GCNN) combined with DRL to automatically learn how to improve the quality of VNEP heuristics. In [34], the authors proposed a deep learning integrated RL, which combined deep learning (DL)

and RL. The DL is used for preparing the optimized beamforming codebook and the RL is used for selecting the best beam out of the optimized beamforming codebook based on the user movements.

According to those early works, we can find that the DRL technique can train the neural network with feedback from the environment. Thus, such as solving the beamforming problem in [32] that using the DRL solution does not require the CSI to find the SINR-optimal beamforming vector. Moreover, different from model-free RL methods that need a huge optimal solution searching overhead for solving a complex problem, the DRL approach uses the DNN to predict the optimal solution without searching overhead. The DRL solution can be trained by the SINR feedback from users. Based on this motivation, we study the DRL and propose a novel DRL approach.

Contributions

The movement and rotation of the HAPS due to wind pressure can cause the cell range to shift, which in turn causes the degradation of users' throughput and handovers between cells [13,14]. With the development of Global Positioning System (GPS) technology, HAPS can control itself to recover its original position state according to GPS positioning technology and thus restore the user signal quality. That being said, the position and rotation state of HAPS will always fluctuate within a certain range due to the unpredictable wind direction and wind force. However, we can improve throughput with beam steering. Thus, to solve the degradation of received power at UEs, we propose a Deep Reinforcement Learning Evolution (DRLEA) method. In our previous work [35], we proposed a Fuzzy $\mathcal{Q}$-learning method. This method used Fuzzy logic in the model-free RL method named $\mathcal{Q}$-learning to control multiple searches in a single training step. Compared with the $\mathcal{Q}$-learning, the proposed Fuzzy $\mathcal{Q}$-learning method has a lower cost of action searching. However, we found that the throughput performance of the proposed Fuzzy $\mathcal{Q}$-learning method under non-uniform user distribution scenarios needs to be improved. Therefore, we proposed a DRLEA method for dynamic antenna control in the HAPS system to reduce the number of low-throughput users.

The proposed DRLEA considers that each iteration in the conventional DRL is a new generation. Starting from the first generation (the first iteration), DRLEA searches for the optimal solution in the current generation and trains the DNN to learn this evolutionary process. The DRLEA records the best result of the current generation as the historical optimal solution for guiding the next generation. If the current generation cannot find better solutions to reduce low throughput users, i.e. users whose throughput is lower than the median throughput prior to antenna parameters adjustment, the mutation happens (randomly selecting an action). The process is repeated until the specified execution time is reached. Compared with conventional RL methods, such as $\mathcal{Q}$-learning-based and DQN-based methods, the proposed method can avoid the sub-optimal solution as much as possible. This is because, in every iteration a random initial set of parameters is used, and is then optimized, leading to less chances of falling into the local optimal.

In addition, the performance of user throughput is closely related to the user's SINR and bandwidth. We can improve the user's SINR by gradually adjusting the antenna parameters based on the user's feedback. As for the user's dedicated bandwidth, it is only a function of the coverage of the antenna array in which the user is located as well as the number of users within that coverage area. Therefore, we do not necessarily need to know the channel state information (CSI) to design the beamforming matrix. Compared with obtaining an accurate CSI, obtaining the user location information, such as using the GPS technique, is less costly in terms of resources and time. It is nonetheless easier and computationally less expensive to adjust the antenna array parameters to account for the changes to the footprint of the users' locations than that of their CSI.

Moreover, we implement three conventional methods, PSO, $\mathcal{Q}$-learning, and DQN as benchmarks to evaluate the proposed DRLEA method and show that the proposed DRLEA is still reliable and efficient. This paper's contributions can be outlined as follows:

- We propose a novel DRLEA method that addresses the problem of dynamic control of the HAPS antenna parameters to decrease the number of users with low throughput. The proposed method combined the EA and DRL to avoid sub-optimal solutions.
- We design a new loss function that includes not only $\mathcal{Q}$-value of the predicted optimal action, but also the historical optimal solutions obtained from previous training.
- Considering the random movement of HAPS caused by wind, we use the user's throughput as a reward, which includes the users' location information. Thus, with the same user distribution scenario and after training, the proposed method can quickly improve the users' throughput under different types of HAPS movements. Even if the HAPS randomly moves again, the proposed approach can still reduce the number of low-throughput users.

The key notations used in this article are listed in Table 1.

Table 1. Key Notations.

M	The number of HAPSs
N	The number of antenna arrays in each HAPS
K	The number of users
θ_{3dB}	The vertical beam half power beam width (HPBW)
ϕ_{3dB}	The horizontal beam HPBW
ϕ_{tilt}	The horizontal tilt
θ_{tilt}	The vertical tilt
p	The transition probability
γ	signal-to-interference plus noise ratio
α	Reward discount factor
β	Learning rate
$\mathcal{S}$	The state space
$\mathcal{A}$	The action space
A	The action of the HAPS
$a_{i,l}$	The l-th action of the i-th antenna array
s_t	The state at the time t
A_j	The j-th action
$\mathcal{R}(s_t, A_j)$	The reward with the (s_t, A_j)
$\mathcal{Q}(s_t, A_j)$	The $\mathcal{Q}$-value with the (s_t, A_j)
$\mathcal{F}$	The neural network
$\mathcal{D}$	The experience reply memory
$\mathcal{X}$	The randomly generated antenna parameters
E	The number of epochs
T	The number of steps in each epoch
P	The number of particles
$gbest$	The optimal solution
$pbest$	The sub-optimal solution

3. System Model

3.1. Model of HAPS

In Figure 1, we show a typical HAPS communication system model. We think about the scenario of M HAPSs serving multiple users in the mMIMO mmWave networks. In more detail, each HAPS is equipped with N antenna arrays to generate N beams. M HAPSs assist the BS to serve multiple users. HAPSs relay the signal transmitted from the BS to users to improve the throughput of users.

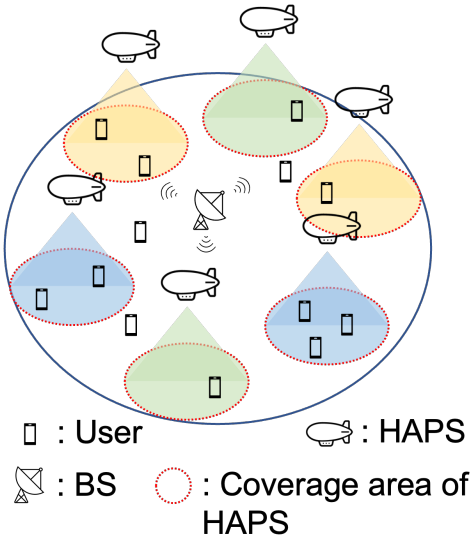

Figure 1. A HAPS system model.

Moreover, we also consider the wind-caused HAPS movement, as shown in Figure 2. Figure 2a shows the HAPS with shifting and Figure 2b shows the HAPS with rotation. Whatever shifting or rotation, it will cause the degradation of users' throughput.

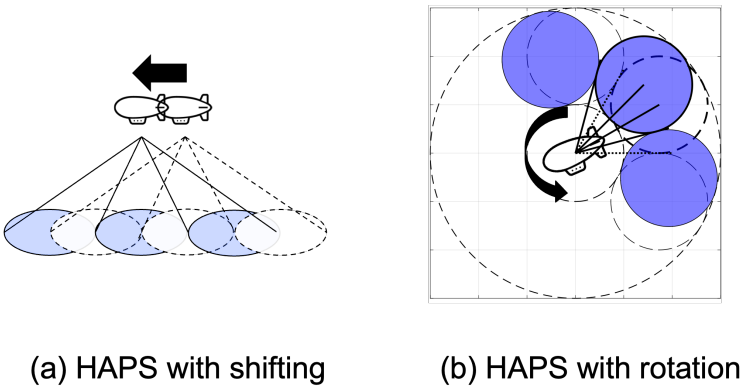

Figure 2. Movement scenarios.

3.2. Antenna Pattern

Planar patch antennas [36] used in each antenna array are considered in this paper. To obtain the vertical and horizontal antenna gains, for each antenna array, four antenna parameters are considered: Figure 3a shows the vertical beam half power beam width (HPBW) θ_{3dB} and the horizontal beam HPBW ϕ_{3dB}, Figure 3b shows the vertical tilt θ_{tilt} and the horizontal tilt ϕ_{tilt}. In Figure 3b, $\Delta\phi_{tilt}$ and $\Delta\theta_{tilt}$ denote the changing of ϕ_{tilt} and θ_{tilt}, respectively. From there, we could derive the expression of the horizontal or vertical antenna gains for an angle Ψ from the main beam direction as [37]:

$$G_h(\Psi) = G_v(\Psi) \begin{cases} -3(\Psi/\Psi_b)^2, & (0^\circ \le \Psi \le \Psi_1) \\ L_N, & (\Psi_1 < \Psi \le \Psi_2) \\ X - 60\log_{10}(\Psi), & (\Psi_2 < \Psi \le \Psi_3) \\ L_F, & (\Psi_3 < \Psi \le 90^\circ) \end{cases} \quad (1)$$

where Ψ_b is one-half the 3 dB beamwidth in the plane of interest, $\Psi_1 = \Psi_b\sqrt{-L_N/3}$, $\Psi_2 = 3.745\Psi_b$, $X = L_N + 60\log_{10}(\Psi_2)$, and $\Psi_3 = 10^{(X-L_F)/60}$ [37], L_N denotes the near-in-side-lobe level, and L_F denotes the far-side-lobe level.

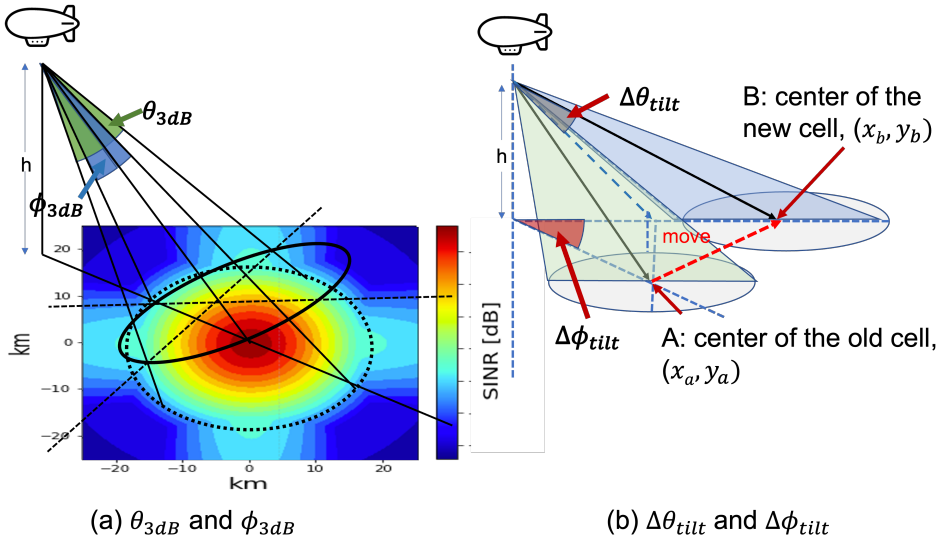

Figure 3. Antenna parameters.

Therefore, we could express the combined gain G as:

$$G = \max(G_v + G_h, L_F) + G_p, \tag{2}$$

where G_v and G_h are the vertical and horizontal antenna gains, respectively, and G_p denotes the maximum antenna gain as shown in Equation (3).

$$G_p = 10\log_{10}(\frac{{B_w}^2}{\theta_{3dB}\phi_{3dB}}) + G, \tag{3}$$

where B_w denotes the beamwidth. Figure 4 shows an example of an antenna pattern for vertical or horizontal polarization.

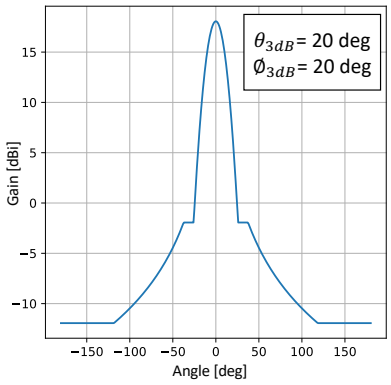

Figure 4. Example of an antenna pattern for vertical or horizontal polarization.

3.3. Problem Formulation

To reduce the number of users whose with low throughput, we formulate the dynamic antenna control problem into maximizing the throughput of users. The throughput $\mathcal{T}$ of a user can be obtained by the following equation:

$$\mathcal{T} = b \times \log_2(1 + \gamma), \tag{4}$$

where γ denotes the SINR at the user, $b = B/K$ denotes the bandwidth assigned to the user, B denotes the bandwidth of each antenna array and K denotes the number of users in an antenna array coverage.

As obvious from Equation (4) and the x-z plane in Figure 5, we can know that the rising of the throughput with the increase in the SINR is very slow when the bandwidth is very small. Thus, considering making users' bandwidth as large as possible is necessary, particularly in the non-uniform user distribution scenario. Compared with the conventional methods, which reduce the SLL [24] or improve the SINR at users [35], this formulation not only improves the SINR at the user's end, but also improves the bandwidth at the user.

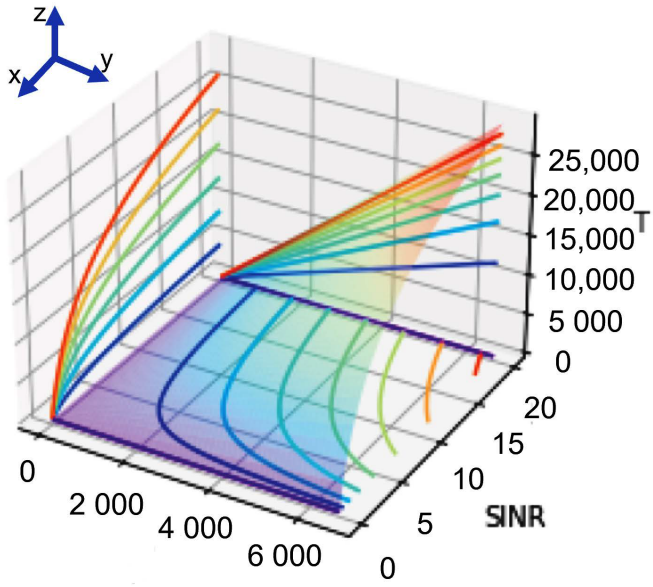

Figure 5. Throughput.

4. The Proposed DRLEA Approach

In this section, we present three algorithms that have been used as benchmarks and the proposed DRLEA algorithms for reducing the number of low-throughput users by dynamic antenna control. In the current paper, we address the case of a single HAPS movement. Our objective is to control only the HAPS in question (which has supposedly moved) to reduce the number of low-throughput users in the area it was serving before it moved. This means that no overlap or exchange of areas with surrounding HAPS will occur.

Therefore, we can simplify the system model as follows. We consider $M-1$ fixed-position HAPS around the HAPS in question that is moved due to wind pressure. Our target is to optimize the antenna parameters of this HAPS to reduce the number of users with low throughput. No handovers between the different HAPS is accounted for, and no new users are introduced to the coverage area of the HAPS in question.

4.1. Markov Decision Process

We define the state at time t as s_t, and the selected action under the state s_t is A_j. Moreover, we consider using the four antenna parameters [ϕ_{tilt}, θ_{tilt}, ϕ_{3dB}, θ_{3dB}] at time t as the state s_t. The action $A_j \in \mathcal{A}$ is defined as the change of one set of antenna parameters, where $\mathcal{A}$ denotes the action set of the HAPS. To reduce the computational complexity, we use discrete antenna parameters to reduce the number of actions. Moreover, to perform the all actions at once, we design the action mapping list as shown in Table 2. We assume that the number of antenna parameters of each antenna array is $P = 4$, and the number of values of each antenna parameter is $V = 3$. Thus, the number of actions of each antenna array is $L = V^P$, and the number of actions of a HAPS is $J = L^N$. $A_1(a_{(1,1)}, \ldots, a_{(N,1)})$ in Table 2 indicates that antenna array 1 to antenna array N performs action index 0.

Table 2. Actions mapping list.

Actions of the HAPS	**Actions of the Antenna Array *i***	**Action Index**	ϕ_{3dB}	θ_{3dB}	ϕ_{tilt}	θ_{tilt}
$A_1(a_{(1,1)}, \ldots, a_{(N,1)})$	$a_{(i,1)}(0,0,0,0)$	0	$+\Delta\phi_{3dB}$	$+\Delta\theta_{3dB}$	$+\Delta\phi_{tilt}$	$+\Delta\theta_{tilt}$
$A_2(a_{(1,1)}, \ldots, a_{(N-1,2)})$	$a_{(i,2)}(0,0,0,1)$					
⋮	⋮	1	$-\Delta\phi_{3dB}$	$-\Delta\theta_{3dB}$	$-\Delta\phi_{tilt}$	$-\Delta\theta_{tilt}$
$A_{J-1}(a_{(1,L)}, \ldots, a_{(N,L-1)})$	$a_{(i,L-1)}(2,2,2,1)$	2	0 °	0 °	0 °	0 °
$A_J(a_{(1,L)}, \ldots, a_{(N,L)})$	$a_{(i,L)}(2,2,2,2)$					

The reward $\mathcal{R}(s_t, A)$ with the state s_t and the action A is represented by the Equation (5).

$$\mathcal{R}(s_t, A) = \frac{\sum_{i,j}^{\frac{K}{2}} \mathcal{T}_i^{'} - \mathcal{T}_j}{\frac{K}{2}}, \tag{5}$$

where K denotes the number of users, $\sum_{i}^{\frac{K}{2}} \mathcal{T}_i^{'}$ denotes the sum of the throughput of the 50 percent users with the least throughput after performing the selected action A, and $\sum_{j}^{\frac{K}{2}} \mathcal{T}_j$ denotes the sum of the throughput of the 50 percent users with the least throughput under the initial state. Thus, the antenna parameters control problem can be represented by a Markov Decision Process (MDP): $\mathcal{M} \triangleq (\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \alpha)$, where $\mathcal{S}$ denotes an infinite state space, p denotes the transition probability that characterizes the stochastic evolution of states in time, with the collection of probability distributions over the state space $\mathcal{S}$, and $\alpha \in [0, 1)$ is the reward discount factor.

The goal is to find a deterministic optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$, such that:

$$\pi^* := \arg\max_{\pi \in \Phi} \mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t r(s_t, \pi(s_t))\right], \tag{6}$$

where Φ is the set of all admissible deterministic policies. At time step t, HAPS selects an action simultaneously based on the policies π. The $\mathcal{Q}$ function is shown as follows:

$$\mathcal{Q}^{\pi}(s, A) := \mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \mathcal{R}(s_t, \pi(s_t)) \mid s_0 = s, A_0 = A\right]. \tag{7}$$

Thus, the optimal policy π^* can be obtained by:

$$\begin{aligned} \pi^*(s) &= \arg\max_{A} Q^*(s, A) \\ &= \mathbb{E}_{s' \sim p(\cdot|s,A)}\left[\mathcal{R}(s, A) + \max_{A'} \alpha \mathcal{Q}^*(s', A')\right] \end{aligned} \tag{8}$$

where s' denotes the next state, $\max_{A'} \mathcal{Q}^*(s', A')$ denotes performing the action A' that can obtain the maximum $\mathcal{Q}$ value under the state s'.

4.2. Conventional Methods

In this subsection, we will describe three methods that serve as benchmarks to solve the MDP.

4.2.1. $\mathcal{Q}$-Learning

$\mathcal{Q}$-learning is a kind of classical RL algorithm [38]. To search the optimal $\mathcal{Q}$ function, $\mathcal{Q}$-learning builds a $\mathcal{Q}$ table to record the sum of existing $\mathcal{Q}$ value $\mathcal{Q}(s, A)$ of the action A for the current state s. The $\mathcal{Q}$ value is obtained by Equation (9).

$$\mathcal{Q}(s, A) \leftarrow \mathrm{Q}(s, A) + \beta\left[\mathcal{R}(s, A) + \alpha \max_j \mathcal{Q}(s', A_j) - \mathcal{Q}(s, A)\right]. \tag{9}$$

Here, β denotes the learning rate and $\max_j \mathcal{Q}(s', A_j)$ denotes performing the action A_j that can obtain the maximum $\mathcal{Q}$ value under the state s'. The details of the $\mathcal{Q}$-learning algorithm is shown in Algorithm 1.

Algorithm 1 $\mathcal{Q} - learning$ method for HAPS antenna control.

Require: s_t, α, ϵ, $\mathcal{A}$.
Ensure: $\mathcal{Q}(s_t, A_1), \mathcal{Q}(s_t, A_2), \ldots, \mathcal{Q}(s_t, A_J)$.

```
Build a Q table QT.
for epoch = 0 to E do
    Initialize the antenna parameters.
    for step t = 0 to T do
        Obtain the antenna parameters as state s_t.
        Randomly generates r in the range of (0,1).
        if r > ε then
            Randomly select an action A ∈ A
        else
            A = arg max_j QT(s_t, A_j)
        end if
        Perform the action A.
        Get reward R(s_t, A) by Equation (5).
        Update s_t to s_{t+1}.
        Update the Q table QT:
        QT(s_t, A) = QT(s_t, A) + Q(s_t, A)
    end for
end for
```

4.2.2. DQN

DQN is another classical RL [39]. It calculates the $\mathcal{Q}$ value of each action based on the reward returned from the environment states and uses the DNN instead of $\mathcal{Q}$ table in $\mathcal{Q}$-learning method to predict the optimal $\mathcal{Q}$ value. Therefore, we can obtain the $\mathcal{Q}$ value with the state s and the action A based on Equation (10).

$$\mathcal{Q}(s, A) \leftarrow \mathcal{R}(s, A) + \alpha \max_j \mathcal{Q}(s', A_j). \tag{10}$$

Different from $\mathcal{Q}$-learning searching $\mathcal{Q}$ table to obtain the optimal solution, the DQN method builds a deep neural network to learn the $\mathcal{Q}$ value of each possible action corresponding to the input environment state. The details of the proposed DQN-based antenna control method are shown in Algorithm 2.

Algorithm 2 DQN-based method for HAPS antenna control.

Require: s_t, α, ϵ, $\mathcal{A}$.
Ensure: $\mathcal{Q}(s_t, A_1), \mathcal{Q}(s_t, A_2), \ldots, \mathcal{Q}(s_t, A_J)$.

```
Initialize main network F(Θ) and target network F(Θ').
Initialize experience reply memory D.
for epoch = 0 to E do
    Initialize the antenna parameters.
    for step t = 0 to T do
        Obtain the antenna parameters as state s_t.
        Randomly generates r in the range of (0,1).
        if r > ε then
            Randomly select an action A ∈ A
        else
            A = arg max_j Q(s_t, A_j; Θ)
        end if
        Perform the action A.
        Get reward R(s_t, A) by Equation (5).
        Update s_t to s_{t+1}.
        Store (s_t, A, R(s_t, A), s_{t+1}) in the D.
        Get the output of the main network F(Θ): Q(s_t, A) = F(s_t, A; Θ).
        Generate target Q value:

        Q'(s_{t+1}, A') = max F(s_{t+1}; Θ').
        Update the main network F(Θ) to minimize the loss function:
```

$$\mathcal{L}(\Theta) = MSE(\mathcal{Q}(s_t, A), \mathcal{R}(s_t, A) + \alpha \mathcal{Q}'(s_{t+1}, A')) \tag{11}$$

```
    end for
    Update the target network: F(Θ') ← F(Θ).
end for
```

In Algorithm 2, E denotes the maximum number of epochs, T denotes the maximum number of steps, and Θ denotes the weights of the deep neural network. We build two DQNs, one is the main network used for evaluating the $\mathcal{Q}$ value of the action obtained by the ϵ-greedy policy. This main network is trained during each step to estimate the approximate optimal action in the current state. The target network is updated with a copy of the latest learned parameters of the main network after each epoch. In other words, using a separate target network helps keep runaway bias from dominating the system numerically causing the estimated $\mathcal{Q}$ values to diverge. Thus, using two DQNs instead of only one DQN can avoid the DQN algorithm to overestimate the true rewards [40]. We calculate the reward $\mathcal{R}(s_t, A)$ with the state s_t and the action A by Equation (5). In each step, we select an action based on the ϵ-greedy method, and store the current state s_t, the selected action A, the reward $\mathcal{R}(s_t, A)$, and the next state s_{t+1} into the experience replay memory $\mathcal{D}$ for training the main network $\mathcal{F}(\Theta)$. Next, we train the main network $\mathcal{F}(\Theta)$ to minimize the loss function $\mathcal{L}(\Theta)$, as shown in Equation (11). After enough training, we input the current state s_t into the main network, then we can observed the predicted $\mathcal{Q}$ value of all actions under the state s_t. Thus, the optimal action $A = \arg\max_j \mathcal{Q}(s_t, A_j; \Theta)$ can be obtained.

4.2.3. PSO

In this paper, we modify the PSO antenna control method in [24] to reduce the number of low throughput users, as shown in Algorithm 3.

Algorithm 3 PSO based algorithm for HAPS antenna control.

Initialize the particles X and V.
for $i = 0$ to E **do**
  for $k = 0$ to P **do**
    Update V:

$$V_i^k = V_i^{k-1} + \omega_1 * rand1_i^k * (pbest_i^k - X_i^{k-1}) + \\ + \omega_2 * rand2_i^k * (gbest_i^k - X_i^{k-1}), \tag{12}$$

    Update X:

$$X_{i+1}^k = X_i^k + V_i^k \tag{13}$$

    Calculate $\mathcal{R}$ by Equation (5) according to X_{i+1}^k
    if $\mathcal{R}_{i+1}^k \geq \mathcal{R}_i^k$ **then** then
      $pbest_i^k \leftarrow X_{i+1}^k$
    end if
    if $\mathcal{R}_{i+1}^k \geq \mathcal{R}_{i+1}$ of other particles **then**
      $gbest \leftarrow X_{i+1}^k$
    end if
  end for
end for
Output the best antenna parameters *gbest*

Where X_i^k denotes the antenna parameters of the k-th particle in the i-th iteration, V_i^k denotes the k-th particle's velocity (the change of antenna parameters) in the i-th iteration, $pbest_i^k$ denotes the best X of k-th particle until the i-th iteration, $gbest$ denotes the best X of all particles in all the iterations, ω_1 and ω_2 denote two independent learning rates, $rand_1$ and $rand_2$ are two independent random numbers for increasing randomness. The PSO algorithm randomly generates P particles to search the optimal antenna parameters X in each iteration and record it into $pbest$. After each iteration, if the $pbest$ of the current iteration is larger than the previous $pbest$, record it into $gbest$ as the optimal solution.

4.3. Deep Reinforcement Learning Evolution Algorithm

In the DQN method, the $\mathcal{Q}$ value of each action is calculated based on the reward returned from the environment states. During the repeating of the experimental process, DQN can learn how to adjust antenna parameters to reduce the number of low-throughput users. Same with DQN approach, we obtain the $\mathcal{Q}$ value by using the Equation (10). However, using the DQN method for the HAPS system is difficult to search for the optimal solution. We use Figure 6 to explain the reason.

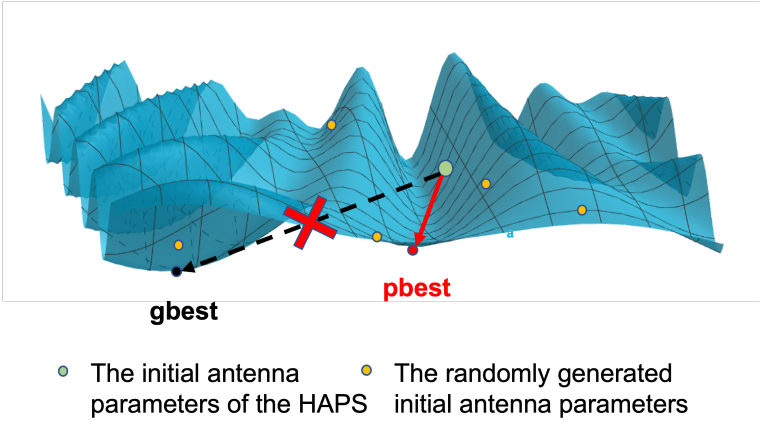

Figure 6. An example of how to find the optimal solution.

In Figure 6, the 'pbest' is one of the sub-optimal solutions, the 'gbest' is the optimal solution. The DQN agent will adjust the antenna parameters step-by-step to find the optimal solution by using gradient descent. Nonetheless, if the initial state is located in the green point shown in Figure 6, the DQN agent cannot obtain the optimal solution even using the Epsilon-greedy method for action selection. To address this problem, we design a novel DRLEA algorithm as shown in Algorithm 4. The workflow of the proposed method is shown in Figure 7.

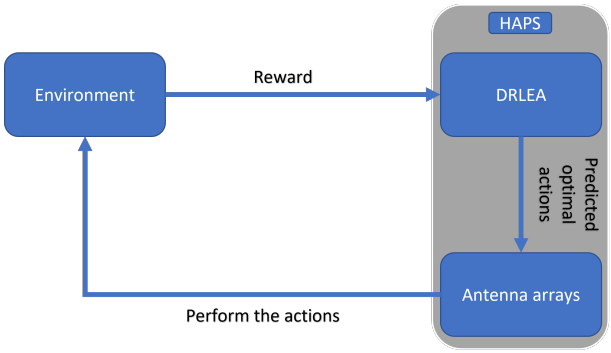

Figure 7. The pipeline of the proposed DRLEA method.

Different from the DQN method with the same initial state at the beginning of each training epoch, the DRLEA will randomly generate a different initial state for each training epoch, such as the yellow points shown in Figure 6. For each epoch, the DRLEA performs many steps to search for the 'optimal' solution (actually is a sub-optimal solution). After a training epoch, the DRLEA compares the 'optimal' solution with the historical optimal solution and then keeps the better one. The details of the proposed DRLEA are shown in Algorithm 4.

Same with Algorithm 2, in Algorithm 4, we build two DNNs, the main network and the target network. We calculate the reward $\mathcal{R}(s_t, A)$ with the state s_t and the action A by Equation (5). In step t, we first select an action A to maximize the $\mathcal{Q}$-value. If the reward $\mathcal{R}(s_t, A)$ is lower than 0 or the reward of the previous step, the DRLEA will re-select and perform a random action from $\mathcal{A}$. Next, we train the main network $\mathcal{F}(\Theta)$ to minimize the loss function $\mathcal{L}(\Theta)$, as shown in Equation (14). In Equation (14), with the increase in the epochs, the influence of the current optimal solution is increased. After enough training, we input the current state s_t into the main network; then, we can obtain the predicted $\mathcal{Q}$ value of all actions under the state s_t. Thus, the optimal action $A = \arg\max_j \mathcal{Q}(s_t, A_j; \Theta)$ can be obtained. After E iterations, the optimal antenna parameters are recorded in *gbest*.

Algorithm 4 Deep Reinforcement Learning Evolution Algorithm

Require: s_t, γ, ϵ, actions $[a_1, \ldots, a_l]$.
Ensure: $\mathcal{Q}(s_t, a_1), \mathcal{Q}(s_t, a_2), \ldots, \mathcal{Q}(s_t, a_J)$.

Initialize main network $\mathcal{F}(\Theta)$ and target network $\mathcal{F}(\Theta^{'})$.
Initialize experience reply memory $\mathcal{D}$.
Initialize $\mathcal{X}$ with random matrices.
$\mathcal{X} = (x_0, \ldots, x_i, \ldots, x_E)$ denotes the E different randomly generated antenna parameters.
$x_i = [\phi_{3dB}, \phi_{tilt}, \theta_{3dB}, \theta_{tilt}]$.
Initialize *pbest* and *gbest* with zero matrices.
for $i = 0$ to E **do**
    Initialize the antenna parameters.
    Initialize $s_t = x_e$, *pbest*.
    for step $t = 0$ to T **do**
        $A = \arg\max_j \mathcal{Q}(s_t, A_j; \Theta)$
        Perform the action A.
        Get reward $\mathcal{R}(s_t, A)$ by Equation (5).
        if $\mathcal{R}(s_t, A) \leq \max(0, \mathcal{R}(s_{t-1}, A^{t-1}))$ **then**
            Randomly select an action $A \in \mathcal{A}$
            Perform the action A.
        end if
        if $\mathcal{R}(s_t, A) \geq \mathcal{R}(s_{t-1}, A^{t-1})$ **then**
            Update *pbest*: $pbest = pbest + A$
        end if
        Update s_{t+1}: $s_{t+1} = s_t + A$.
        Store $(s_t, A, \mathcal{R}(s_t, A), s_{t+1})$ in the $\mathcal{D}$.
        Get the output of the main network $\mathcal{F}(\Theta)$:
        $\mathcal{Q}(s_t, A) = \mathcal{F}(s_t, A; \Theta)$.
        Generate target $\mathcal{Q}$ value:
        $\mathcal{Q}^{'}(s_{t+1}, A^{t+1}) = \max \mathcal{F}(s_{t+1}; \Theta^{'})$.
        Update the main network $\mathcal{F}(\Theta)$ to minimize the loss function:

$$\mathcal{L}(\Theta) = \mathrm{MSE}\Bigg(\mathcal{Q}(s_t, A), \mathcal{R}(s_t, A) + \frac{\alpha}{i+1}\mathcal{Q}^{'}\left(s_{t+1}, A^{t+1}\right) + \left(\alpha - \frac{\alpha}{i+1}\right) * \left(1 - \frac{|s_t - gbest|}{|gbest|}\right)\Bigg) \tag{14}$$

    end for
    if $\mathcal{R}(s_t, A) \geq \mathcal{R}_{\text{best}}$ **then**
        $\mathcal{R}_{\text{best}} = \mathcal{R}(s_t, A)$
        $gbest \leftarrow pbest$
    end if
    Update the target network: $\mathcal{F}(\Theta^{'}) \leftarrow \mathcal{F}(\Theta)$.
end for

5. Simulation Results

5.1. Simulation Setting

To evaluate our proposed methods, we generate the four different non-uniform UEs distribution datasets obtained by [41]. The four different user distributions are Tokyo, Osaka, Sendai, and Nagoya, as shown in Figure 8.

We assume that the HAPS works at a height of 20 km. Each HAPS can cover an area within a 20 km radius. The transmit power and bandwidth of each antenna array are 43 dBm and 20 Mhz, respectively. We consider that the transmission frequency is 2 GHz. Considering the interference between HAPSs, we set 18 HAPSs to surround 1 HAPS.

We set the updated value of horizontal tilt, horizontal HPBW, vertical tilt, and vertical HPBW to 20 deg, 4 deg, 10 deg, and 4 deg, respectively. We use python to implement all simulation programs.

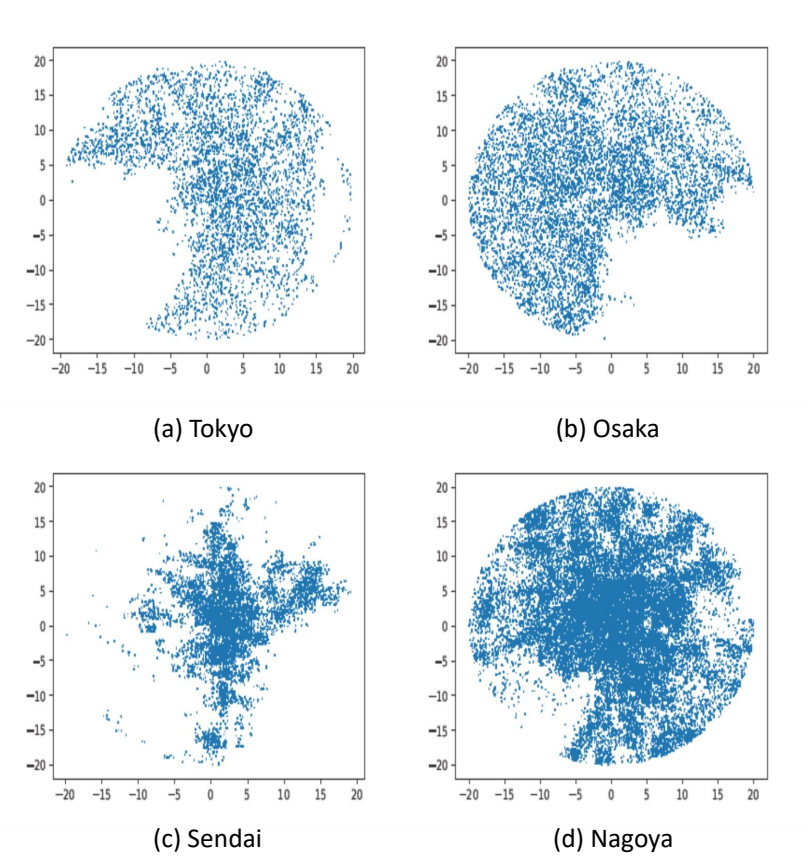

Figure 8. The four user distribution scenarios: (**a**) Tokyo, (**b**) Osaka, (**c**) Sendai, and (**d**) Nagoya.

We set 100 epochs and 100 steps in each epoch, in the three RL-based approaches ($\mathcal{Q}$-learning, DQN, and DRLEA). In the DQN and DRLEA, we set the learning rate of the neural network as 0.0001. The discount factor α is 0.65 and the learning rate β for $\mathcal{Q}$ value calculation is 0.75 in the three RL-based approaches. In the PSO approach, the number of iterations is 100, the number of particles is 100, and the $\omega_1 = 0.5$ and $\omega_2 = 0.5$.

5.2. CDF of UE Throughput Performance

We train all the methods under the scenario where HAPS is rotated by 30 degrees, respectively, under four different user distributions. Figure 9 shows the cumulative distribution function (CDF) of the users' throughput under the rotation scenario.

In all the scenarios, the proposed DRLEA reduces the number of low-throughput users in the throughput range of $[0.0, 16.0]$ kbps. In the case of Tokyo, the proposed algorithm achieves the best CDF performance in the throughput range of $[2.5, 16.0]$ kbps. In the case of Osaka, the proposed method achieves the best performance in the throughput range of $[5.3, 16.0]$ kbps. In the case of Sendai, the proposed method achieves the best performance in the throughput range of $[2.0, 11.8]$ kbps. In the case of Nagoya, the proposed method achieves the best performance in the throughput range of $[1.3, 5.0]$ kbps and $[6.0, 16.0]$ kbps.

In the $\mathcal{Q}$-learning, the Fuzzy $\mathcal{Q}$-learning, and the DQN, which use the simple ϵ-greedy method for searching the optimal solution, are difficult to escape the local optimal. Even if we use the ϵ-greedy method to randomly select an action, the result is still close to the local optimal solution. However, in the proposed method, we consider using the strategy of EA.

This strategy can make the result escape the local optimal by randomly setting the different initial states at the beginning of each epoch.

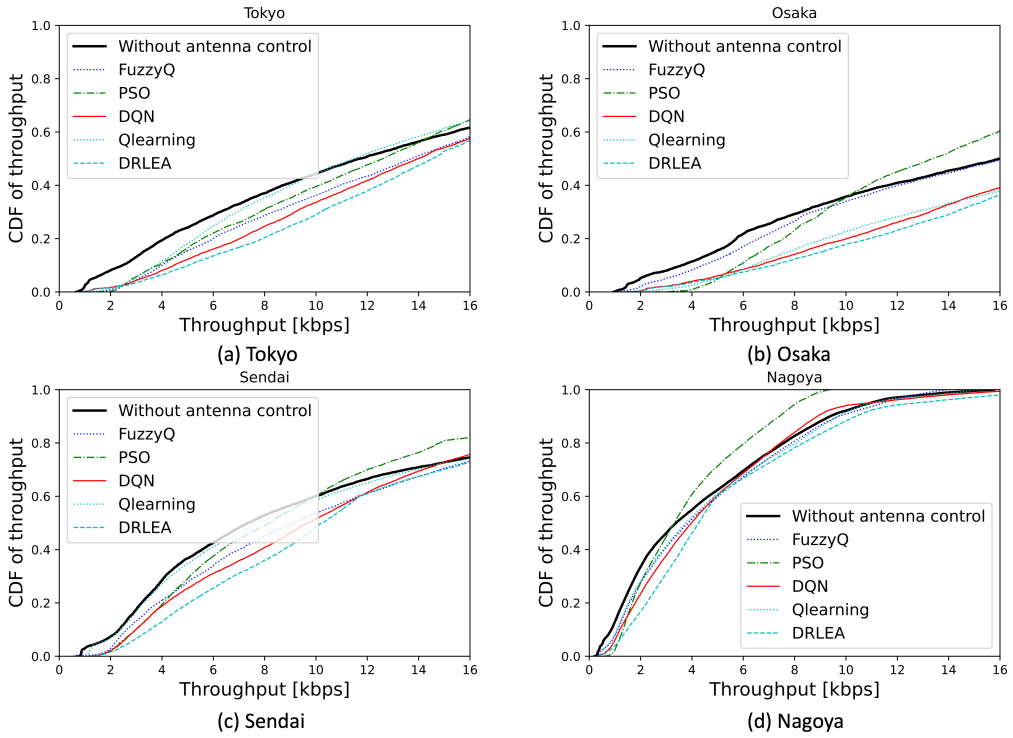

Figure 9. CDF of UE throughput performance under the HAPS with rotation.

Similarly, the PSO method also uses the same strategy. In Figure 9, we can find that in the $[0.0, 2.0]$ kbps throughput range, the PSO method has comparable or even better CDF of throughput performance than the other methods. In particular, in Figure 9b, PSO achieves the best CDF performance in the $[0.0, 4.0]$ kbps throughput range. However, due to the high-dimensional space, the number of particles we set in our experiments is not enough for PSO to search for the optimal solution. It is still easy to fall into the local optimum.

Unlike PSO, which only targets the current global optimal solution and local optimal solution, the proposed method trains the neural network to adapt itself to avoid falling into the local optimal in a high-dimensional space.

Moreover, we evaluate the proposed method under the shifting scenario. Here, we use the DRLEA and DQN methods, which are trained under the rotation case, and the other three methods are trained under the shifting scenario. As shown in Figure 10, we can find that the DQN and the proposed DRLEA without retraining achieve comparable throughput performance to that of Q-learning and Fuzzy Q-learning with training in the HAPS left shift of the 5 km case. Since throughput is closely related to the user distribution, DNNs trained under the same user distribution know how to adjust the antenna parameters to reduce the number of low throughput users, even if HAPS moves again.

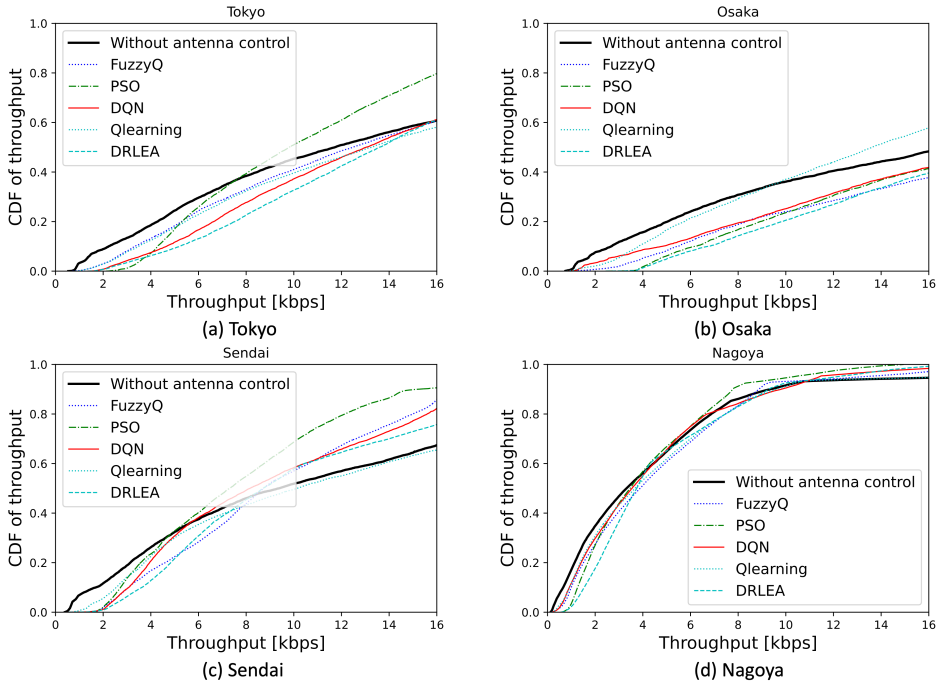

Figure 10. CDF of UE throughput performance under the HAPS with shifting.

6. Conclusions

In this paper, we addressed the problem of the reduction of the number of users with low throughput caused by the movement of HAPS. To do so, a novel method named DRLEA was proposed. Different from the PSO and conventional RL methods, the proposed DRLEA method can adjust the antenna parameters without any searching overhead. We used the throughput of users for the reward calculation instead of using the received SINR. Using throughput for reward calculation will increase the computational overhead but it can make the DRLEA learn not only the SINR of users but also the location information of that. In other words, the proposed DRLEA method can improve the user's SINR and bandwidth at the same time. Moreover, the proposed DRLEA combined EA with DRL to avoid sub-optimal solutions. A novel loss function was designed to train the DNN with the historical optimal solution to avoid the sub-optimal solutions.

Through simulations, we demonstrated that the proposed approach clearly improves the throughput of the users at the lower end of the spectrum. Compared with approaches such as the PSO and the Q-learning ones, the proposed DRLEA method achieves the best throughput performance under all the user distribution cases. Moreover, to prove the good robustness of the proposed DRLEA method, we use the proposed DRLEA, which is trained in the HAPS with a rotation scenario to control the antenna parameters under the shifting scenario. Through simulations, we demonstrated that, compared with the other three methods, which are trained under the HAPS with shifting, the proposed DRLEA method can achieve comparable throughput performances even without re-training.

Author Contributions: Conceptualization, S.Y.; methodology, S.Y.; validation, S.Y., M.B. and T.O.; formal analysis, S.Y. and M.B.; writing—original draft preparation, S.Y. and M.B.; writing—review and editing, S.Y., M.B.,T.O., Y.S., W.T., K.H. and A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JST SPRING, Grant Number JPMJSP2123. These research results were obtained from the commissioned research (No. 05701) by the National Institute of Information and Communications Technology (NICT), Japan.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DRLEA	Deep Reinforcement Learning Evolution Algorithm
HAPS	High-Altitude Platform Station
DNN	Deep Neural Network
DQN	Deep $\mathcal{Q}$ Network
GA	Genetic Algorithm
UEs	User Equipments
GEO	Geostationary Earth Orbit
LEO	Low Earth Orbit
RTT	Round Trip Time
LoS	Line-of-Sight
BS	Base Station
IoT	Internet of Things
B5G	Beyond 5G
DoA	Direction-of-Arrival
CCAA	Concentric Circular Antenna Array
IDCSA	Discrete Cuckoo Search Algorithm
SLL	Side-Lobe Level
SOS	Symbiotic Organism Search
PSO	Particle Swarm Optimization
CIR	Carrier-to-Interference Ratio
CDF	Cumulative Distribution Function
CSI	Channel State Information
VNEP	Virtual Network Embedding Problem
GCNN	Graph Convolutional Neural Network
RL	Reinforcement Learning
DL	Deep Learning
DRL	Deep Reinforcement Learning
MIMO	Multiple-Input Multiple-Output
mmWave	MillimeterWave
MSE	Mean Squared Error

References

1. Tozer, T.; Grace, D. High-Altitude Platforms for Wireless Communications. *Electron. Commun. Eng. J. (ECEJ)* **2001**, *13*, 127–137. [CrossRef]
2. Why SoftBank Is Looking to the Stratosphere. Softbank. Available online: https://www.softbank.jp/en/sbnews/entry/20190826_01 (accessed on 26 August 2019).
3. Tereza, P.; Elizabeth, H. Starlink Satellites: Everything You Need to Know about the Controversial Internet Megaconstellation. SPACE. Available online: https://www.space.com/spacex-starlink-satellites.html (accessed on 23 November 2022).
4. Gao, N.; Jin, S.; Li, X.; Matthaiou, M. Aerial RIS-Assisted High Altitude Platform Communications. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 2096–2100. [CrossRef]
5. Karapantazis, S.; Pavlidou, F. Broadband communications via high-altitude platforms: A survey. *IEEE Commun. Surv. Tutorials* **2005**, *7*, 2–31. [CrossRef]
6. Ding, C.; Wang, J.B.; Zhang, H.; Lin, M.; Li, G.Y. Joint Optimization of Transmission and Computation Resources for Satellite and High Altitude Platform Assisted Edge Computing. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 1362–1377. [CrossRef]
7. Ye, J.; Dang, S.; Shihada, B.; Alouini, M.S. Space-Air-Ground Integrated Networks: Outage Performance Analysis. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7897–7912. [CrossRef]
8. Jeon, H.B.; Park, S.H.; Park, J.; Huang, K.; Chae, C.B. An Energy-efficient Aerial Backhaul System with Reconfigurable Intelligent Surface. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6478–6494. [CrossRef]
9. Zhou, D.; Gao, S.; Liu, R.; Gao, F.; Guizani, M. Overview of development and regulatory aspects of high altitude platform system. *Intell. Converg. Networks* **2020**, *1*, 58–78. [CrossRef]
10. Widiawan, A.K.; Tafazolli, R. High Altitude Platform Station (HAPS): A Review of New Infrastructure Development for Future Wireless Communications. *Wirel. Pers. Commun.* **2007**, *42*, 387–404. [CrossRef]

11. Mohammed, A.; Mehmood, A.; Pavlidou, F.N.; Mohorcic, M. The Role of High-Altitude Platforms (HAPs) in the Global Wireless Connectivity. *Proc. IEEE* **2011**, *99*, 1939–1953. [CrossRef]
12. White, G.P.; Zakharov, Y.V. Data Communications to Trains From High-Altitude Platforms. *IEEE Trans. Veh. Technol.* **2007**, *56*, 2253–2266. [CrossRef]
13. El-Jabu, B.; Steele, R. Effect of positional instability of an aerial platform on its CDMA performance. In Proceedings of the Gateway to 21st Century Communications Village, VTC 1999-Fall, IEEE VTS 50th Vehicular Technology Conference (Cat. No.99CH36324), Amsterdam, The Netherlands, 19–22 September 1999; Volume 5, pp. 2471–2475. [CrossRef]
14. Thornton, J.; Grace, D. Effect of Lateral Displacement of a High-Altitude Platform on Cellular Interference and Handover. *Trans. Wireless. Commun.* **2005**, *4*, 1483–1490. [CrossRef]
15. Panfeng, H.; Naiping, C.; Shuyan, N. Coverage model of multi beam antenna from high altitude platform in the swing state. In Proceedings of the 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 26–28 August 2016; pp. 750–753. [CrossRef]
16. Dessouky, M.I.; Sharshar, H.A.; Albagory, Y.A. Geometrical Analysis of High Altitude Platforms Cellular Footprint. *Prog. Electromagn. Res.-Pier* **2007**, *67*, 263–274. [CrossRef]
17. Dessouky, M.; Nofal, M.; Sharshar, H.; Albagory, Y. Optimization of Beams Directions for High Altitude Platforms Cellular Communications Design. In Proceedings of the Twenty Third National Radio Science Conference (NRSC'2006), Menouf, Egypt, 14–16 March 2006; pp. 1–8. [CrossRef]
18. Albagory, Y.; Nofal, M.; Ghoneim, A. Handover Performance of Unstable-Yaw Stratospheric High-Altitude Stations. *Wirel. Pers. Commun.* **2015**, *84*, 2651–2663. [CrossRef]
19. He, P.; Cheng, N.; Cui, J. Handover performance analysis of cellular communication system from high altitude platform in the swing state. In Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, China, 13–15 August 2016; pp. 407–411. [CrossRef]
20. Hoshino, K.; Sudo, S.; Ohta, Y. A Study on Antenna Beamforming Method Considering Movement of Solar Plane in HAPS System. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5. [CrossRef]
21. Enache, F.; Depărăţeanu, D.; Popescu, F. Optimal design of circular antenna array using genetic algorithms. In Proceedings of the 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, Romania, 29 June–1 July 2017; pp. 1–6. [CrossRef]
22. Sun, G.; Liu, Y.; Chen, Z.; Zhang, Y.; Wang, A.; Liang, S. Thinning of Concentric Circular Antenna Arrays Using Improved Discrete Cuckoo Search Algorithm. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6. [CrossRef]
23. Dib, N. Design of planar concentric circular antenna arrays with reduced side lobe level using symbiotic organisms search. *Neural Comput. Appl.* **2018**, *30*, 3859–3868. [CrossRef]
24. Ismaiel, A.; Elsaidy, E.; Albagory, Y.; Atallah, H.; Abdel-Rahman, A.; Sallam, T. Performance Improvement of High Altitude Platform Using Concentric Circular Antenna Array Based on Particle Swarm Optimization. *AEU—Int. J. Electron. Commun.* **2018**, *91*, 85–90. [CrossRef]
25. Mandal, D.; Kumar, H.; Ghoshal, S.; Kar, R. Thinned Concentric Circular Antenna Array Synthesis using Particle Swarm Optimization. *Procedia Technol.* **2012**, *6*, 848–855.
26. Arum, S.C.; Grace, D.; Mitchell, P.D.; Zakaria, M.D. Beam-Pointing Algorithm for Contiguous High-Altitude Platform Cell Formation for Extended Coverage. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5. [CrossRef]
27. Grace, D.; Thornton, J.; Chen, G.; White, G.; Tozer, T. Improving the system capacity of broadband services using multiple high-altitude platforms. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 700–709. [CrossRef]
28. Thornton, J.; Grace, D.; Capstick, M.; Tozer, T. Optimizing an array of antennas for cellular coverage from a high altitude platform. *IEEE Trans. Wirel. Commun.* **2003**, *2*, 484–492. [CrossRef]
29. Zhang, L.; Lai, S.; Xia, J.; Gao, C.; Fan, D.; Ou, J. Deep reinforcement learning based IRS-assisted mobile edge computing under physical-layer security. *Phys. Commun.* **2022**, *55*, 101896. [CrossRef]
30. Zhan, W.; Huang, B.; Huang, A.; Jiang, N.; Lee, J. Offline Reinforcement Learning with Realizability and Single-policy Concentrability. In Proceedings of the Proceedings of Machine Learning Research, London, UK, 2–5 July 2022; Loh, P.L., Raginsky, M., Eds.; JMLR: Cambridge, MA, USA, 2022; Volume 178, pp. 2730–2775..
31. Lazaridou, A.; Baroni, M. Emergent Multi-Agent Communication in the Deep Learning Era. *arXiv* **2020**, arXiv:2006.02419.
32. Mismar, F.B.; Evans, B.L.; Alkhateeb, A. Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination. *IEEE Trans. Commun.* **2020**, *68*, 1581–1592. [CrossRef]
33. Rkhami, A.; Hadjadj-Aoul, Y.; Outtagarts, A. Learn to improve: A novel deep reinforcement learning approach for beyond 5G network slicing. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6. [CrossRef]
34. Eappen, G.; Cosmas, J.; T, S.; A, R.; Nilavalan, R.; Thomas, J. Deep learning integrated reinforcement learning for adaptive beamforming in B5G networks. *IET Commun.* **2022**, *16*, 2454–2466.

35. Wada, K.; Yang, S.; Bouazizi, M.; Ohtsuki, T.; Shibata, Y.; Takabatake, W.; Hoshino, K.; Nagate, A. Dynamic Antenna Control for HAPS Using Fuzzy Q-Learning in Multi-Cell Configuration. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1–6. [CrossRef]
36. *Minimum Performance Characteristics and Operational Conditions for High Altitude Platform Stations Providing IMT-2000 in the Bands 1885–1980 MHz, 2010–2025 MHz and 2110–2170 MHz in Regions 1 and 3 and 1885–1980 MHz and 2110–2160 MHz in Region 2, document ITU-R M.1456*; International Telecommunications Union: Geneva, Switzerland, 2000.
37. Shibata, Y.; Kanazawa, N.; Konishi, M.; Hoshino, K.; Ohta, Y.; Nagate, A. System Design of Gigabit HAPS Mobile Communications. *IEEE Access* **2020**, *8*, 157995–158007; Erratum in *IEEE Access* **2021**, *9*, 618. [CrossRef]
38. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-Learning Algorithms: A Comprehensive Classification and Applications. *IEEE Access* **2019**, *7*, 133653–133667. [CrossRef]
39. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
40. Hasselt, H.v.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, Phoenix, AZ, USA, 12–17 February 2016; AAAI Press: Honolulu, HI, USA, 2016; pp. 2094–2100.
41. Dynamic Population Data. Agoop Corporation. Available online: https://www.agoop.co.jp/service/dynamic-population-data/ (accessed on 6 December 2021).