Trajectory-Aware UAV-Enabled WPT Networks Based Grid World DRL Approach

Sengly Muy, Vitou That
Department of Intelligent Energy and Industry
Chung-Ang University
Seoul, South Korea
muysengly@cau.ac.kr, vitou1707@cau.ac.kr

Jung-Ryun Lee *(Senior Member IEEE)*
School of Electrical and Electronics Engineering
Department of Intelligent Energy and Industry
Chung-Ang University
Seoul, South Korea
jrlee@cau.ac.kr

***Abstract*—In this research, we investigate a wireless power transfer (WPT) system involving an unmanned aerial vehicle (UAV) equipped with an array of antennas to wirelessly charge ground user (GU) devices. Our objective is to enhance the lowest GU energy levels by optimizing the UAV's trajectory, beam-forming strategy, and transmission power simultaneously. Since optimizing the lowest GU energy presents a challenging non-convex problem, we reformulate it as a discrete-time grid world problem. We propose a deep reinforcement learning (DRL) approach to optimize this problem by determining the UAV's movement direction, beam-forming angle, and transmit power level. We also integrate the water-filling algorithm with DRL to aid in determining the optimal hovering duration. Through simulations, we demonstrate that our approach significantly improves GU energy levels compared to the successive hover-and-fly algorithm while maintaining low computational complexity.**

***Index Terms*—UAV's trajectory, WPT, DRL, Water-filling algorithm.**

I. Introduction

Researchers are increasingly interested in the potential of radio-frequency energy harvesting through WPT for providing dependable energy sources to low-power Internet-of-Things (IoT) devices. Additionally, WPT offers the advantage of simultaneously charging multiple wireless devices, regardless of their mobility or intricate deployment, through the utilization of multiple beam-forming array antennas [1]. These array antennas, capable of multiple beam-forming, have demonstrated their utility in various applications, such as geostationary and aerospace telecommunications, owing to their performance and adaptability [2]. Nevertheless, in scenarios where infrastructure is disrupted, such as disaster areas, the existing infrastructure may not support WPT services for wireless IoT devices.

In recent times, UAVs have brought about a multitude of advantages through their applications, encompassing remote sensing, search and rescue missions, cargo delivery, security and surveillance, agricultural monitoring, and civil infrastructure inspections [3]. Furthermore, UAVs offer simplicity in deployment, cost-effectiveness, reduced maintenance requirements, and adaptability to diverse environmental conditions. Consequently, deploying UAVs to facilitate WPT services for wireless IoT devices emerges as a promising solution, especially in scenarios like disaster areas or compromised infrastructure. Nevertheless, the flight duration of UAVs is inherently constrained by their onboard battery capacity, necessitating the optimization of power consumption to extend their operational capabilities [4]. Nonetheless, conventional optimization techniques face challenges when attempting to optimize UAV trajectory planning while simultaneously considering system performance, as the problem must satisfy terrain and threat avoidance requirements, along with performance constraints.

Reinforcement learning (RL) algorithms have gained prominence in recent times within wireless networks, offering solutions to optimize various challenges and enhance system performance while adding intelligence to edge devices. RL operates by striving to maximize a reward function, achieved through trial-and-error interactions, in the quest to discover the most favorable decisions [5]. Q-learning (QL) represents a potent RL algorithm capable of learning the value of an action within a specific state, aiming to approach a solution close to global optimality [6]. Nevertheless, as the Q-table's data size (comprising state, action, and Q-value) in QL grows, it demands excessive memory during training, a phenomenon known as the curse of dimensionality. Conversely, deep Q-learning (DQL), a variant of DRL, employs a deep neural network (DNN) as a function approximator to handle high-dimensional raw input data [7]. DQL has demonstrated its effectiveness in addressing complex challenges [8], leading to its application in solving problems related to trajectory planning and resource allocation within UAV-assisted wireless networks [9].

II. Previous Work

Recently, numerous studies have been conducted to assess the efficiency of deploying UAV-based WPT systems. These studies focus on effectively managing and supervising UAV trajectories while ensuring optimal resource allocation. One such investigation, outlined in [10], examined a UAV-enabled

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC support program (IITP-2023-2018-0-01799) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20214000000280), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. RS-2023-00251105).

Mobile Edge Computing (MEC) system. In this scenario, a UAV initially charges IoT devices through WPT, after which each IoT device transmits collected data back to the UAV using energy harvested from it. In a separate study described in [11], researchers explored the deployment of a UAV-enabled WPT network for charging GUs through down-link communication. GUs can utilize the harvested energy to transmit independent data to the UAV via up-link channels, with the goal of maximizing the minimum up-link throughput within the UAV's limited flight time.

Additionally, [12] introduced a novel successive hover-and-fly algorithm aimed at optimizing the UAV's trajectory. This algorithm identifies specific locations for efficient energy transmission, with the UAV flying between these locations and hovering only at them for effective energy transfer. Furthermore, [13] delved into the maximization of energy harvesting for all GUs by optimizing a UAV's trajectory plan. In an idealized scenario, the authors disregarded the UAV's maximum velocity constraint and employed the Lagrange dual function to determine the best possible trajectory. Subsequently, they proposed an alternative successive hover-and-fly algorithm that utilizes successive convex programming optimization for practical trajectory design.

In recent times, machine learning (ML) techniques have found application in WPT systems to enhance their efficiency and performance. For instance, in [15], the author introduced an integrated approach combining block-chain and multi-agent DRL to tackle the optimization problem involving computation offloading, energy harvesting (EH), and optimal resource pricing. Similarly, [16] presented a deep Q-network (DQN) design to minimize the average age of information (AoI) among GUs by jointly optimizing the UAV's trajectory and the scheduling of information transmission and GUs' energy harvesting. Moreover, in [17], the author introduced a deep deterministic policy gradient (DDPG) approach with the objective of maximizing the volume of data offloaded to the UAV. In this particular system, the UAV harnesses energy from a base station and assists an edge server with computational tasks, showcasing the versatility of ML techniques in WPT systems.

Prior studies have primarily concentrated on deploying UAVs to optimize various aspects such as energy efficiency, energy transfer, communication coverage, or the cumulative data rate for GUs. In our research, we shift our focus to address the max-min problem concerning the residual energy of GUs. Specifically, we aim to maximize the minimum remaining battery capacity of a GU within the network. To elaborate, our approach entails deploying a UAV equipped with multiple-beam-forming array antennas in the network to wirelessly transfer energy and charge the batteries of ground-based IoT devices. In this network context, we formulate the optimization problem centered on maximizing the lowest GU energy, taking into account various control parameters of the UAV, including trajectory, hovering duration, beam-forming pattern, and transmit power.

III. System Model and Problem Formulation

A. System Model

We consider a UAV-supported WPT network, where K GUs are deployed in a cell with a radius of R and a charging period of T. The set of GUs is denoted as $\mathcal{K} = \{1, 2, ..., K\}$ and the available UAV flying time is expressed as $\mathcal{T} = (0, T]$. Moreover, we assume that all GUs are positioned on the ground at point $(x_k, y_k, 0), \forall k \in \mathcal{K}$. And, the UAV's location at time t is denoted as $(x(t), y(t), H), \forall t \in \mathcal{T}$ with a fixed altitude H and the maximum speed V. Therefore, the channel model from the UAV to the GU k can be expressed as,

$$\mathbf{h}_k = \sqrt{\beta_0 d_k^{-\alpha}}\mathbf{a}(\theta, \phi), \tag{1}$$

where α is the path-loss exponent, and β_0 is the channel power at the reference distance $d_0 = 1$m. d_k is the distance from the UAV to GU k. Furthermore, the maximum distance between the drone and the GUs can be calculated as

$$d_k \leq H \cos \Theta_{\max}, \tag{2}$$

where $\Theta_{\max}$ is the maximum elevation angle.
In this study, we consider a uniform planar array antenna installed at the bottom of the UAV, which generates multiple independent steered beams. Therefore, the effective channel gain from UAV to GU k is given by

$$\left|\mathbf{h}_k^H \boldsymbol{\upsilon}\right|^2 = \frac{\beta_0}{d_k^\alpha} \left|\mathbf{a}(\theta, \phi)\, \boldsymbol{\upsilon}\right|^2, \tag{3}$$

$\mathbf{a}(\theta, \phi)$ is the steering vector of elevation angle θ and azimuth angle ϕ for the LOS path. And, the received radio frequency power by GU k can be calculated as

$$Q_k = p\left|\mathbf{h}_k^H \boldsymbol{\upsilon}\right|^2 = \frac{p\beta_0 \left|\mathbf{E}_{\theta,\phi}\right|^2}{d_k^{\alpha/2}}, \tag{4}$$

where $\mathbf{E}(\theta, \phi) = \mathbf{a}(\theta, \phi)\, \boldsymbol{\upsilon}$ is the beam-forming pattern of the antenna array, and p is the transmit power of UAV. Then, we can formulate the total energy harvesting received by each GU k over the whole charging time t as

$$EH_k(t) = \int_0^t Q_k(\tau)\, d\tau. \tag{5}$$

B. Problem Formulation

Our study aims to maximize the minimum received energy of all GUs by jointly optimizing the trajectory, transmit power, and beam-forming pattern of the UAV. Therefore, we can formulate the optimization problem as

$$(\text{P1}): \max_{(x(t),y(t)),p(t),\mathbf{E}_{\theta,\phi}(t)} \min_{k\in\mathcal{K}} E_k(t) \tag{6}$$

$$\text{s.t.} \quad 0 \leq \phi \leq 2\pi, \tag{7}$$

$$0 \leq \theta \leq \Theta_{\max}, \tag{8}$$

$$0 \leq p(t) \leq P_{\max}, \tag{9}$$

$$E_{\text{UAV}}(t) \leq E_{\max}, \tag{10}$$

$$x^2(t) + y^2(t) \leq R^2, \tag{11}$$

$$\dot{x}^2(t) + \dot{y}^2(t) \leq V^2, \tag{12}$$

$$\forall t \in \mathcal{T}, \tag{13}$$

where $P_{\max}$, and $E_{\max}$ are the maximum transmit power of the UAV, and the maximum UAV battery level, respectively. And the constraint (12) represents the speed limit for UAV.

IV. Proposed DRL with the Water-Filling Algorithm

In this section, we present the DQL with the water-filling algorithm for UAV trajectory design and resource allocation.

A. DQL design

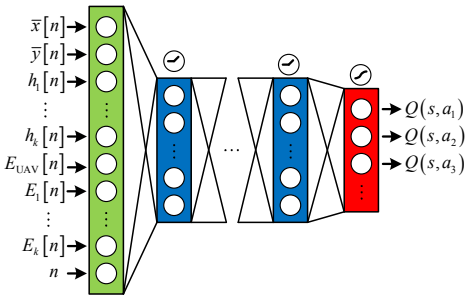


Fig. 1. The DQN design.

Because the DQL algorithm is more effective for low-dimensional discrete action spaces, we convert the problem (6) into discrete time-space by discretizing the whole charging duration into a finite number N of each time slot with duration $\delta = \frac{T}{N}$. It should be noticed that the duration δ is selected to be sufficiently small so that we may assume the UAV location remains roughly constant over each time slot n. We denote $x[n], y[n]$ as the position of UAV at time slot n, $p[n]$ as the UAV's transmit power at time slot n, and $\mathbf{E}_{\theta,\phi}[n]$ as the beam-forming pattern at time slot n, where $n \in \mathcal{N} = \{1, 2, ..., N\}$. Therefore, we can formulate the optimization problem as

$$(\text{P2}): \max_{(x[n],y[n]),p[n],\mathbf{E}_{\theta,\phi}[n]} \min_{k\in\mathcal{K}} E_k[n] \tag{14}$$

$$\text{s.t.} \quad 0 \le \phi \le 2\pi, \tag{15}$$

$$0 \le \theta \le \Theta_{\max}, \tag{16}$$

$$0 \le p[n] \le P_{\max}, \tag{17}$$

$$E_{\text{UAV}}[n] \le E_{\max}, \tag{18}$$

$$x^2[n] + y^2[n] \le R^2, \tag{19}$$

$$\Delta x^2[n] + \Delta y^2[n] \le V^2, \tag{20}$$

$$\forall n \in \mathcal{N}, \tag{21}$$

where $\Delta x^2[n] = (x[n] - x[n-1])^2$ and $\Delta y^2[n] = (y[n] - y[n-1])^2$.

To solve the optimization problem (P2) using DRL, we simplify (P2) via the grid world problem, which is well-known as the most basic and classic problem in reinforcement learning. To do so, we need to slice the location (x and y-axis) of the UAV into M^2 squares. We denote the location of the UAV in the grid world as $\bar{x}[n]$ and $\bar{y}[n]$, where $\bar{x}[n], \bar{y}[n] \in \left\{0, \frac{1}{M-1}2R, \frac{2}{M-1}2R, \ldots, 2R\right\}$. To find the UAV's trajectory in the grid world environment, we propose a design of a DQL model (state, action, and reward) as follows.

State: In our DRL design, the state is designed as

$$s[n] \in \{(\bar{x}[n], \bar{y}[n]), h_k[n], E_{\text{UAV}}[n], E_k[n], n\}_{k\in\mathcal{K}}, \tag{22}$$

where $h_k[n]$ is the channel gain from UAV to GU k at time slot n. $E_{\text{UAV}}[n]$ and $E_k[n]$ are the remaining battery of the UAV and the battery of ground devices k at time slot n, respectively.

Action: The direction of motion, transmit power, and beam-forming pattern of the UAV is controlled by a user, and the action of DRL is defined as

$$a[n] \in \{(\Delta x[n], \Delta y[n]), p[n], \boldsymbol{E}_{\theta,\phi}[n]\} \tag{23}$$

where $(\Delta x[n], \Delta y[n])$ is the direction of motion of the UAV in the grid world following x- and y-axis at time slot n, which is given by

$$\Delta x[n] \in \left\{\bar{x}[n] - \frac{1}{M-1}2R, \bar{x}[n], \bar{x}[n] + \frac{1}{M-1}2R\right\}, \tag{24}$$

and

$$\Delta y[n] \in \left\{\bar{y}[n] - \frac{1}{M-1}2R, \bar{y}[n], \bar{y}[n] + \frac{1}{M-1}2R\right\}. \tag{25}$$

Therefore, in the grid world, the UAV has $(A_q = 9)$ different ways to move, such as: up, down, left, right, up-left, up-right, down-left, down-right, and not move. The moving direction of the UAV can be illustrated in Fig. 2. $0 \le p[n] \le P_{\max}$ is the UAV's transmit power with P_q quantized level at time slot n. $\boldsymbol{E}_{\theta,\phi}[n]$ is the beam-forming pattern that is generated by E_q blocks of array antennas with elevation angle θ and azimuth angle ϕ at time slot n.

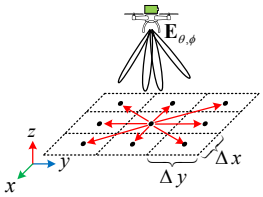


Fig. 2. The actions of UAV.

Reward: In DRL, the reward signal is used to evaluate how good an action is under a state. We notice that the lowest GU battery is the highest priority for the UAV to fly over or hover; therefore, we proposed a design of the reward function as

$$r[n] = \begin{cases} \sum_{k\in K} \bar{f}_{(x_k,y_k)}(x,y)\left(1 - \bar{E}_k[n]\right) & \text{satisfy constraints,} \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where $\bar{E}_k[n]$ is the normalized energy of the GU k. $\bar{f}_{(x_k,y_k)}(x,y)$ is the normalized probability density function (PDF) of multivariate normal distribution. Here, the group having GUs with low battery levels generates the highest reward area. Fig. 3 shows an example of the reward function with ten random GUs' locations and battery levels.

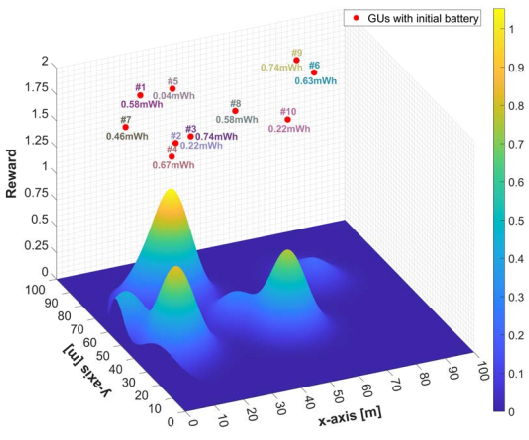


Fig. 3. The example of the reward function.

B. The water-filling algorithm

The water-filling algorithm is a process of determining equalization strategies for different system's performances. As the algorithm's name implies, water finds its level even when filled in one portion of a vessel with several openings due to Pascal's law. Thus, the water-filling algorithm is a practical algorithm to solve the max-min problems. To apply the water-filling algorithm to the proposed DRL, we first assume that there are P hovering locations, and denoting $p \in \mathcal{P} \triangleq \{1, 2, ..., P\}$ as the set of hovering points. Then the optimal hovering duration $\{T_1, T_2, ..., T_P\}$ can be calculated by the water-filling algorithm, where $h_{p,k}[n]$ is the channel from the p-th hovering point to the k-th GU at time slot n, and $(x, y, H)_p$ is the hovering point p. The pseudo-code for the application of the water-filling algorithm to the proposed DRL algorithm is given in Algorithm 1.

Algorithm 1 Water-filling algorithm.

output T_p
initialize:
    Randomly initialize the GUs' battery E_k
    Set $(x, y, H)_p = (x, y, H)_k$, $N = \frac{T}{\Delta}$, $T_p = 0$, E_{UAV}, $n = 1$
while ($n \leq N$ and $E_{\text{UAV}} \geq 0$)
    $k^* = \underset{k \in \mathcal{K}}{\text{argmin}}\ E_k[n]$
    $p^* = \underset{p \in \mathcal{P}}{\text{argmin}}\ h_{p,k^*}[n]$
    for k in LoS of $(x, y, H)_{p^*}$
        $E_k[n] \leftarrow E_k[n] + \eta P_{\max} \delta h_{p^*,k}[n]$
    end for
    $E_{\text{UAV}}[n] \leftarrow E_{\text{UAV}}[n] - P_{\max}\delta$
    $T_{p^*} \leftarrow T_{p^*} + \delta$
    $n \leftarrow n + 1$
end while

The complete algorithm of the proposed DQL with the water-filling technique is summarized in Algorithm 2.

Algorithm 2 DQL with the water-filling algorithm.

initialize
    Randomly initialize the location of GUs $(x, y)_k$
    Randomly initialize the Q-network $Q(s, a; \boldsymbol{w})$
    Randomly initialize the GUs' battery E_k
    Initialize the replay memory D with capacity C
    Initial the location of UAV to (R, R, H)
for each epoch **do**
    for each time slot n **do**
        Observe state $s[n]$
        Take action $a'[n]$ using ϵ-greedy policy
        if $v[n] = 0$ (Hovering)
            set $(x[n], y[n], H)$ as a hovering location
        end if
        Update next state $s[n+1]$, observe reward $r[n]$
        Store $(s[n], a[n], r[n], s[n+1])$ in replay memory D
    end for
    Find optimal hovering duration using **Algorithm 1**
    if the replay memory D is full **then**
        Train the DQN
    end if
end for

V. Performance Evaluation

In this section, we evaluate the performance of the proposed DQL with the water-filling algorithm in the context of the UAV-enabled WPT network with array antennas to transmit energy to the GUs. These are randomly deployed over the coverage area with random initial battery capacity determined using the uniform distribution. In our simulation, the UAV is equipped with 8×8 array antennas, which are divided into four 4×4 sub-array antennas. The departure location of the UAV is set as the center of the coverage area (R, R), and the flight altitude is fixed at a certain altitude of $H = 10m$ following the effectiveness of the WPT.

Moreover, in this study, we compare our proposed algorithm with the successive hover-and-fly algorithm and the static hovering scheme. In the static hovering scheme, the UAV hovers on top of each GU and moves from one hovering point to another instantly, where the hovering duration is calculated based on the water-filling algorithm in Algorithm 1. Table I provides a summary of the simulation settings that are utilized to set up the environment. The parameters linked to the DRL model are given in Table II.

Figure 4a shows the generated environment and the UAV's trajectory with flying and hovering duration for the finite time duration $T = 1200s$ and the maximum UAV transmission power $P_{\max} = 30$dBm. In this environment, the UAV finds the trajectory and hovering duration to transmit the energy for the GUs. The UAV hovers longer at the fifth hovering point than at other hovering points because our reward design gives the highest priority to hovering at the lowest-charged battery, GU #5 with 0.04 Wh. It is noted that the energy transmitted by the UAV to GUs is dependent not only on the position of the GUs

TABLE I
ENVIRONMENT SETUP.

Parameters	Value
Cell radius R	50m
Number of GUs K	10
Total flying time T	1200s
Time slot duration δ	1s
UAV's maximum speed V	10m/s
UAV's altitude H	10m
Pathloss exponent η	3.6
Rician fading gain	5dBm
Energy conversion factor η	50%
Number of array antenna	8×8
Array antenna block	4×4
GU's maximum battery	1Wh
UAV's maximum battery E_{UAV}	$\{1, 2, ..., 10\}$kWh
Elevation angle maximum	45 degree
Quantize x and y-axis M	1000

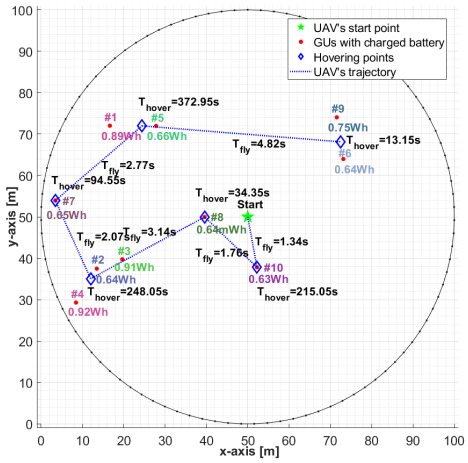
TABLE II
SIMULATION PARAMETERS FOR THE PROPOSED DRL.

Parameters	Value
Learning rate α	0.01
Number of hidden layer L	5 layers
ϵ-greedy ϵ	0.1
Discount factor γ	0.99
Replay memory size D	100
Optimizer	SGD

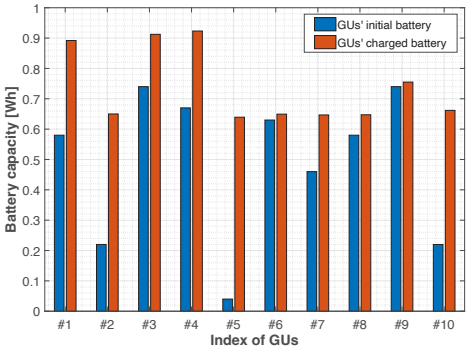
but also on the minimum battery level of GUs. In Figure 4b, the bar chart shows the battery charges of GUs at the initial stage and after being charged by the UAV. As seen in this figure, the proposed method maximizes the lowest GU battery charge, where GUs with the battery levels have received almost the same amount of energy from the UAV. From the result, we can verify that our proposed DQL with the water-filling algorithm can effectively solve the problem of maximizing the lowest GU energy by jointly optimizing the UAV's trajectory, beamforming pattern, and transmit power.

Figure 5a shows the minimum GU battery level versus the maximum transmit power $P_{\max}$ of the UAV for a constant flying time duration $T = 1200$s and UAV initial battery $E_{\max} = 10$kWh. The result shows that the minimum GU battery level increases exponentially for $P_{\max} \in \{20, 22, ..., 34\}$ dBm; otherwise, it remains stable after $P_{\max} = 34$dBm. It is clear that the result will remain stable, although we increase the UAV's transmit power because of the limit of the UAV's initial battery. The graph also shows that our proposed method outperforms the successive hover-and-fly trajectory algorithm and static hovering scheme. And Figure 5b shows that the proposed method and successive hover-and-fly algorithm achieve the same performance in terms of the remaining battery level of the UAV.

Furthermore, we evaluate the performance of our proposed DRL algorithm with varying number of GUs (5, 10, 15, 20, and 25) and the comparison is conducted under the same conditions, as shown in Figure 2. In this figure, the initial battery of the UAV remains at 6 kWh, and the maximum transmit power is fixed at 32 dBm. The result shows that the



(a)



(b)

Fig. 4. Simulation result of the proposed DRL with 10 GUs.

minimum battery level of the GUs decreases as the number of GUs increases due to the UAV's battery limitation, and our proposed DRL algorithm outperforms the consecutive hover-and-fly algorithm irrespective of the given number of GUs.

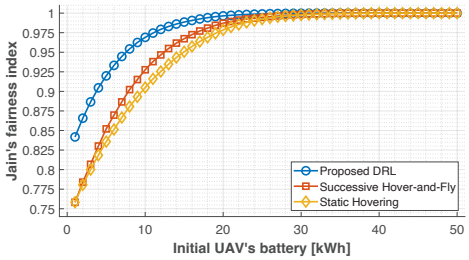
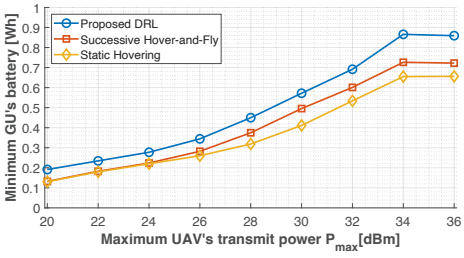
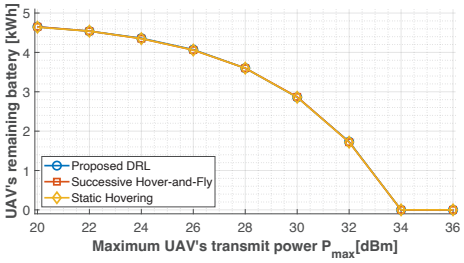


Fig. 7. Jain's fairness of GUs' battery.

In Figure 7, we evaluate and compare the fairness of the algorithms using Jain's fairness index under the same condition where the number of GUs is set to 25, the initial battery of the UAV is set to 6 kWh, and the maximum transmit power is fixed by 32 dBm. Here, Jain's fairness index. The graph indicates that increasing the initial battery of the UAV leads to an increase in Jain's fairness index to one. It is clear that increasing initial battery level of the UAV results



(a)



(b)

Fig. 5. The simulation result for different UAV's transmit powers.

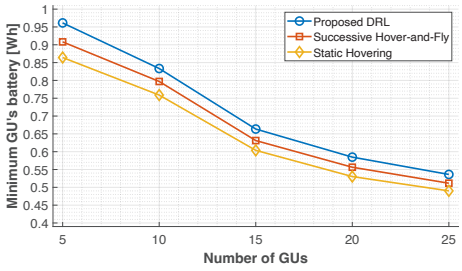


Fig. 6. Minimum GUs' battery vs. number of GUs.

in charging all GUs until their batteries are full, ensuring fair distribution of energy among them. Additionally, the result demonstrates that the proposed DRL algorithm outperforms the other schemes.

Table III summarizes the comparisons of the computational complexities of the time complexity analysis of the successive hover-and-fly and DQL with the water-filling algorithm.

TABLE III
TIME COMPLEXITY COMPARISON.

Algorithm	Operations
Successive hover-and-fly	1.21×10^{17}
The proposed DQL	4.05×10^{10}

VI. CONCLUSIONS

This paper studied jointly optimizing the UAV's trajectory, beamforming pattern, and transmit power to maximize the lowest GU energy, in which the UAV is adapted with the array antennas for delivering wireless energy to charge the GUs. To solve this problem, we first converted the problem into a discrete-time format and then transformed it again into a grid world problem. After that, we proposed a DQL design with the water-filling algorithm to find the optimal UAV trajectory, transmit power, and beamforming pattern. Simulation results reveal that the proposed algorithm significantly enhances the lowest GU energy received compared to the successive hover-and-fly algorithm with low-time computation.

REFERENCES

[1] Nariman, Med, Farid Shirinfar, Anna Papió Toda, Sudhakar Pamarti, Ahmadreza Rofougaran, and Franco De Flaviis. "A compact 60-GHz wireless power transfer system." IEEE Transactions on Microwave Theory and Techniques 64, no. 8 (2016): 2664-2677.

[2] Angeletti, Piero, and Marco Lisi. "Multimode beamforming networks for space applications." IEEE Antennas and Propagation Magazine 56, no. 1 (2014): 62-78.

[3] Hayat, Samira, Evşen Yanmaz, and Raheeb Muzaffar. "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint." IEEE Communications Surveys & Tutorials 18, no. 4 (2016): 2624-2661.

[4] Zeng, Yong, and Rui Zhang. "Energy-efficient UAV communication with trajectory optimization." IEEE Transactions on wireless communications 16, no. 6 (2017): 3747-3760.

[5] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[6] Watkins, Christopher JCH, and Peter Dayan. "\cal Q-learning." Machine learning 8, no. 3-4 (1992): 279-292.

[7] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).

[8] Huang, Hongji, Song Guo, Guan Gui, Zhen Yang, Jianhua Zhang, Hikmet Sari, and Fumiyuki Adachi. "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions." IEEE Wireless Communications 27, no. 1 (2019): 214-222.

[9] Bithas, Petros S., Emmanouel T. Michailidis, Nikolaos Nomikos, Demosthenes Vouyioukas, and Athanasios G. Kanatas. "A survey on machine-learning techniques for UAV-based communications." Sensors 19, no. 23 (2019): 5170.

[10] Du, Yao, Kun Yang, Kezhi Wang, Guopeng Zhang, Yizhe Zhao, and Dongwei Chen. "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems." IEEE Transactions on Vehicular Technology 68, no. 10 (2019): 10187-10200.

[11] Xie, Lifeng, Jie Xu, and Rui Zhang. "Throughput maximization for UAV-enabled wireless powered communication networks." IEEE Internet of Things Journal 6, no. 2 (2018): 1690-1703.

[12] Yuan, Xiaopeng, Tianyu Yang, Yulin Hu, Jie Xu, and Anke Schmeink. "Trajectory design for UAV-enabled multiuser wireless power transfer with nonlinear energy harvesting." IEEE Transactions on Wireless Communications 20, no. 2 (2020): 1105-1121.

[13] Xu, Jie, Yong Zeng, and Rui Zhang. "UAV-enabled wireless power transfer: Trajectory design and energy optimization." IEEE transactions on wireless communications 17, no. 8 (2018): 5092-5106.

[14] Melo, Francisco S. "Convergence of Q-learning: A simple proof." Institute Of Systems and Robotics, Tech. Rep (2001): 1-4.

[15] Seid, Abegaz Mohammed, Jianfeng Lu, Hayla Nahom Abishu, and Tewodros Alemu Ayall. "Blockchain-Enabled Task Offloading With Energy Harvesting in Multi-UAV-Assisted IoT Networks: A Multi-Agent DRL Approach." IEEE Journal on Selected Areas in Communications 40, no. 12 (2022): 3517-3532.

[16] Liu, Lingshan, Ke Xiong, Jie Cao, Yang Lu, Pingyi Fan, and Khaled Ben Letaief. "Average AoI minimization in UAV-assisted data collection with RF wireless power transfer: A deep reinforcement learning scheme." IEEE Internet of Things Journal 9, no. 7 (2021): 5216-5228.

[17] Zhang, Zhanpeng, Xinghuan Xie, Chen Xu, and Runze Wu. "Energy Harvesting-Based UAV-Assisted Vehicular Edge Computing: A Deep Reinforcement Learning Approach." In 2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops), pp. 199-204. IEEE, 2022.