electronics

MDPI

Article

Deep Reinforcement Learning-Driven UAV Data Collection Path Planning: A Study on Minimizing AoI

Hesong Huang [1,*], Yang Li [1], Ge Song [2] and Wendong Gai [1]

[1] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China; lyang@sdust.edu.cn (Y.L.); gwd2011@sdust.edu.cn (W.G.)
[2] College of Electrical and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China; songge@sdust.edu.cn
* Correspondence: huang_sust@sdust.edu.cn

Abstract: As a highly efficient and flexible data collection device, Unmanned Aerial Vehicles (UAVs) have gained widespread application because of the continuous proliferation of Internet of Things (IoT). Addressing the high demands for timeliness in practical communication scenarios, this paper investigates multi-UAV collaborative path planning, focusing on the minimization of weighted average Age of Information (AoI) for IoT devices. To address this challenge, the multi-agent twin delayed deep deterministic policy gradient with dual experience pools and particle swarm optimization (DP-MATD3) algorithm is presented. The objective is to train multiple UAVs to autonomously search for optimal paths, minimizing the AoI. Firstly, considering the relatively slow learning speed and susceptibility to local minima of neural network algorithms, an improved particle swarm optimization (PSO) algorithm is utilized for parameter optimization of the multi-agent twin delayed deep deterministic policy gradient (MATD3) neural network. Secondly, with the introduction of the dual experience pools mechanism, the efficiency of network training is significantly improved. Experimental results show DP-MATD3 outperforms MATD3 in average weighted AoI. The weighted average AoI is reduced by 33.3% and 27.5% for UAV flight speeds of v = 5 m/s and v = 10 m/s, respectively.

Keywords: UAV; path planning; data collection; deep reinforcement learning; particle swarm optimization; dual experience pools

Citation: Huang, H.; Li, Y.; Song, G.; Gai, W. Deep Reinforcement Learning-Driven UAV Data Collection Path Planning: A Study on Minimizing AoI. *Electronics* **2024**, *13*, 1871. https://doi.org/10.3390/electronics13101871

Academic Editor: Shiho Kim

Received: 20 April 2024
Revised: 2 May 2024
Accepted: 5 May 2024
Published: 10 May 2024

1. Introduction

In the current age of digital revolution, the advancement of Internet of Things (IoT) has exerted profound influences on across various sectors of society [1–3]. The widespread deployment of sensor networks has enabled a continuous influx of real-time data into systems [4], encompassing parameters such as temperature, humidity, and light intensity, among others. These data sources hold a pivotal significance in supporting real-time management systems, including traffic supervision, industrial control, and so forth [5–7]. In this context, the freshness of data collection becomes paramount for ensuring the quality of informed decision-making. Nevertheless, with the continuous growth of data scale and complexity, the freshness of data collection becomes a crucial consideration in guaranteeing the quality of real-time decision-making. In order to evaluate data freshness, Age of Information (AoI) is adopted as an evaluation index [8,9].

Currently, AoI has gained popularity as a metric to assess data timeliness in IoT networks [10–15]. It underscores the "freshness" of data within target nodes. In reference [10], a balance between high reliability and information freshness was achieved through fine-tuning jointly encoded packets. Simultaneous transmission over multiple sub-channels effectively reduced AoI. Chen et al. [11] introduced an algorithm from a charging perspective, addressing the optimization of information delay in wireless-powered edge networks, concurrently reducing both average and maximum peak AoI. Pu et al. [12] proposed a

novel AoI-bounded scheduling algorithm, ensuring that the peak age of AoI for each data package is maintained within a finite range. To optimize average AoI, Zhao et al. [13] studied the impact of interference on information delay in large-scale wireless networks and presented a novel method. The research revealed similarities in channel access probabilities and differences in packet arrival rates, particularly under low node density conditions. Zhou and Saad [14] introduced a threshold-based optimization strategy to minimize average AoI, achieving near-optimal performance in noisy channels through a low-complexity suboptimal strategy. Gu et al. [15] assessed the performance of average peak AoI in IoT systems by introducing AoI metrics, and evaluating coverage and underlying schemes. These studies primarily conduct theoretical analyses of AoI performance or adopt conventional approaches to minimize AoI through resource allocation optimization.

When facing challenges such as limited transmission power and unavailable uplink channels in the IoT, one widely recognized solution is the utilization of Unmanned Aerial Vehicles (UAVs) for auxiliary remote sensing. Their high maneuverability, ease of deployment, and capability to cover hard-to-reach areas make them efficient data collectors, significantly enhancing the efficiency of data acquisition in IoT [16,17]. Despite the advantages, UAV-assisted IoT networks encounter problems stemming from restricted energy and the consequential effects of UAV trajectory on both the effectiveness of data acquisition and power consumption. Additionally, the uplink transmission time and power consumption of IoT devices are influenced by the choice of hover points. Hence, the focus of this work lies in investigating the problem of UAVs path planning to minimize AoI, thereby enabling IoT to collect data in a timely and effective manner.

1.1. Related Works

This section revisits some research on UAV path planning related to AoI.

In reference [18], for minimizing the average AoI, a dynamic programming and ant colony algorithm are employed to decompose this problem into energy transmission and data collection time sequence allocation. Gao et al. [19] employed an end-to-end strategy to successfully enhance data collection efficiency and reduce the AoI. For the purpose of minimizing AoI, Xiong et al. [20] utilized GA to optimize the data collection process. Lu et al. [21] utilized mobile unmanned vehicles to cooperate with UAVs in data collection, thereby achieving a balance between information freshness and energy replenishment, leading to a reduction in AoI. Liu and Zheng [22] utilized continuous approximation and a genetic algorithm to optimize UAVs' speed and path, as well as AoI and onboard energy of the monitoring area data, thereby reducing task completion time.

However, the aforementioned works often employ heuristic algorithms to address trajectory optimization problems, facing challenges such as insufficient adaptability, susceptibility to local optima, and high computational complexity. These issues typically hinder their ability to adapt to increasingly complex and scalable wireless networks. As artificial intelligence continues to advance, deep reinforcement learning (DRL) could enable UAVs to make intelligent localized decisions and accomplish tasks. Leveraging DRL algorithms and trained models, UAVs can autonomously and rapidly optimize flight trajectories through interaction with the environment. Hu et al. [23] proposed the compound CA2C algorithm to address path design problems resulting from UAVs' collaborative perception and transmission, thereby reducing the AoI. Zhou et al. [24] introduced the A-TP algorithm, which employs DRL to optimize UAV paths, thereby enhancing the efficiency of IoT data collection. Abd-Elmagid et al. [25] proposed an improved DRL algorithm to address the AoI of different processes. Peng et al. [26] employed the double deep Q-learning network algorithm to intelligently plan paths for UAVs, aiming to reduce UAVs' energy consumption, thereby enhancing information freshness. Yin et al. [27] focused on UAV-assisted vehicular communication, using a deep Q network to optimize transmission power and offloading ratios, minimizing AoI. Chen et al. [28] conducted research on resource allocation for AoI perception using online DRL.

1.2. Contributions

This paper investigates the path planning problem for UAV-assisted IoT network data collection and introduces a DRL algorithm to achieve intelligent decision-making for UAV flight trajectories, thereby reducing the level of AoI. The key contributions are outlined in the following manner.

(1) To minimize AoI, it is transformed into a Markov game process, and the multi-agent twin delayed deep deterministic policy gradient (MATD3) algorithm is designed to optimize the UAV flight trajectory.

(2) To address the relatively slow learning speed and susceptibility to local minima issues associated with neural network algorithms, we applied an improved Particle Swarm Optimization (PSO) algorithm for the parameter optimization of the MATD3 neural network. Additionally, to further expedite the training speed of the network, we incorporated an additional experience replay buffer to store superior experience information. This enhancement facilitates a quicker convergence of the network towards the optimal policy.

(3) Through a comparison with the traditional MATD3 algorithm, the proposed approach was evaluated. Simulation results indicate that it effectively reduces AoI, demonstrating the effectiveness of the designed algorithm.

The remaining parts of this work are structured as enumerated below. The system model and the multi-UAV path planning algorithm based on DP-MATD3 are introduced in Sections 1 and 3. Simulation analysis was conducted in Section 4. The conclusion is provided in Section 5.

2. System Model

2.1. Network Model

In the scenario of UAVs assisting in data collection for the IoT, the UAVs perform data acquisition and transmission, as illustrated in Figure 1. Assuming within the service range, there are M UAVs, and their electricity is E, and N IoT devices are randomly distributed, with each device having a task size of l_n, where $M \in \mathcal{M} = \{1, 2, \ldots, M\}$ and $N \in \mathcal{N} = \{1, 2, \ldots, N\}$.

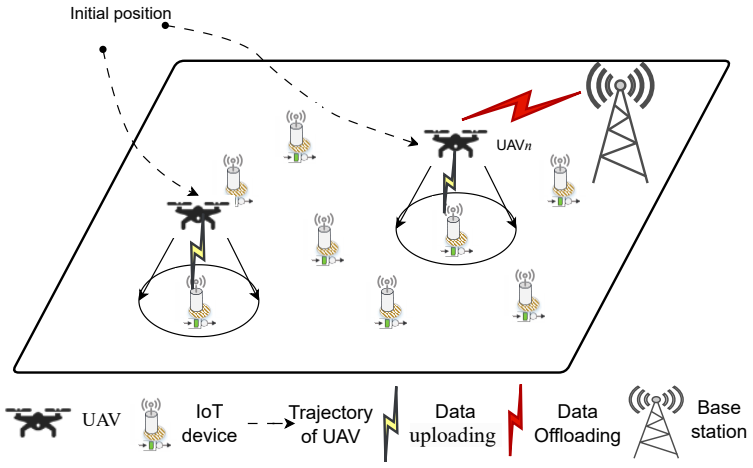

Figure 1. A scene of UAV-assisted data collection.

Then, a model is established to represent the connectivity between UAVs and IoT devices, utilizing the decision variable $\chi_m^n(t) \in \{0, 1\}$ to indicate the connection status between mth UAV and nth IoT device at time t. When $\chi_m^n(t) = 1$, it signifies that mth UAV is accessing nth IoT device; otherwise, it denotes no communication link between them. It is assumed that in each time slot, the communication between IoT devices and UAVs is a one-to-one correspondence. Therefore, constraints $\sum_n \chi_m^n(t) \leq 1$ indicate that each UAV can communicate with only one IoT device at any given time. Additionally, to enhance

collaboration among multiple UAVs, any nth IoT device can be serviced by at most one UAV at the same time, represented by the constraint $\sum_m \chi_m^n(t) \leq 1$.

In addition, considering that multiple UAVs are serving the same area, it is essential to address the issue of collisions between UAVs. The distance constraint is denoted as follows

$$\|q_{m1}(t) - q_{m2}(t)\| \geq o_{\min}, \forall t \in [0, T], \forall m_1, m_2 \in M \tag{1}$$

here, $q_m(t)$ represents the two-dimensional position of UAV $m \in \mathcal{M}$ while flying at a constant altitude of H meters, and o_{min} is the minimum inter-UAV collision distance.

2.2. Data Collection Model

In practical scenarios, given the typically low power of IoT devices, their long-range communication capabilities are notably constrained. To address this challenge, UAVs need to reach the communication zone of these devices and stay within the range for some time to facilitate efficient data collection. Drawing on experimental findings regarding channel gain from references [29,30], it is observed that in moderate altitudes, communication primarily relies on Line-of-Sight (LoS) propagation. Consequently, the channel gain $g_m^n(t)$ can be approximated as [31]

$$g_m^n(t) = h_0 d_m^{-2}(n) = \frac{h_0}{\|q_m(t) - c(n)\|^2} \tag{2}$$

where $q_m(t)$ and $c(n)$ denote the location of mth UAV at time t and nth IoT device, respectively. h_0 denotes the channel gain when $d = 1$. The wireless transmission rate is denoted as:

$$r_m^n(t) = B \log_2\left(1 + \frac{P_{up} g_m^n(t)}{\sigma^2}\right) \tag{3}$$

In order to collect data from the nth IoT device, it is assumed that the UAV must hover above the IoT device for a minimum duration. Therefore, the delay in information transmission T_m^n needs to meet the following constraint:

$$T_m^n \geq \frac{l_n}{r_m^n} \tag{4}$$

2.3. Energy-Consuming Model

UAV energy consumption includes two parts: flight energy consumption and communication energy consumption. The power consumption $P_m(v)$ of the mth UAV when moving horizontally at the speed v can be expressed as follows

$$P_m(v) = \omega_p r_p M \times \lambda_p + \frac{1}{2}\rho C_{D_0} A_e v^3 + \frac{\pi}{4} n_p c_p \rho C_{D_0} \omega_p^3 r_p^4 \left[1 + 3\left(\frac{v}{\omega_p r_p}\right)^2\right] \tag{5}$$

where M and A_e represent the weight and the frontal area of the UAV, respectively, ω_p is the angular velocity, C_{D_0} denotes the drag coefficient, n_p signifies the number of blades, r_p indicates the rotor disk radius, c_p is the blade chord, and ρ represents the air density. The total energy consumption of the mth UAV at the hovering position is given by

$$E_{q,m} = T_m^n (P_m(0) + P_{c,m}) \tag{6}$$

where $P_m(0)$ represents the hover power and $P_{c,m}$ is the communication power of the mth UAV. Therefore, the energy consumption in time slot k is given by

$$E_{u,m}(k) = \frac{L_{k,m}}{v} P_m(v) + c_{k,m} E_{q,m} \tag{7}$$

where $L_{k,m}$ represents the flight distance of the mth UAV in time slot k, and $c_{k,m}$ represents whether the mth UAV collects data in time slot k, which can be expressed as

$$c_{k,m} = \begin{cases} 1, \text{UAV data collection} \\ 0, \text{otherwise} \end{cases} \tag{8}$$

The battery remaining power $E_{k,m}$ of mth UAV in time slot k is as follows:

$$E_{k,m} = E - \sum_{i=0}^{k} E_{u,m}(i) \tag{9}$$

2.4. AoI

To guarantee timely data collection, AoI is a critical performance index. As shown in Figure 2, commencing from the initial value A_0, AoI progressively accumulates at a constant rate of 1 until the reception of an update.

The following equation represents the AoI of nth IoT device at time t:

$$A_n(t) = \begin{cases} t - o(t), \{t, \chi_m^n(t) = 1\} \\ A_n(t-1) + 1, others \end{cases} \tag{10}$$

where $o(t)$ signifies the timestamp of data generation. It is worth noting that when an IoT device has no data stored or has already been collected, the AoI is 0.

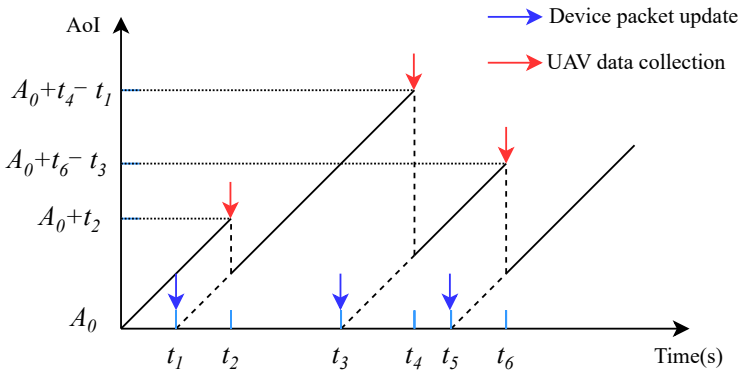

Figure 2. The change process of AoI of IoT devices over time.

2.5. Problem Modeling

This paper aims to design a collaborative path-planning method for multiple UAVs, to minimize the collected AoI. This process encompasses two distinct stages: the flight stage, where UAVs move, and the hovering collection stage, where UAVs hover to collect information from IoT devices. Therefore, the AoI collected by UAVs $A_n(t)$ can be divided into: the AoI of the information itself at the beginning of transmission, and the delay in information transmission T_m^n, which is the time it takes for information to transmit from nth IoT device to mth UAV. Accordingly, the optimization task is briefly outlined as:

$$\min_{q,\chi} \frac{1}{N} \sum_{n=1}^{N} A_n(t) \tag{11}$$

$$\text{s.t. } \|q_{m1}(t) - q_{m2}(t)\| \geq o_{\min}, \forall t \in [0, T], \forall m_1, m_2 \in M, \tag{12}$$

$$\sum_{n=1}^{N} \chi_m^n(t) \leq 1, \forall m \in M \tag{13}$$

$$\sum_{m=1}^{M} \chi_m^n(t) \leq 1, \forall n \in N \tag{14}$$

$$E_{k,m} > 0 \tag{15}$$

3. DP-MATD3-Based Multi-UAV Path Planning Algorithm

3.1. Markov Game Process for the Given Problem

UAVs need to plan paths collaboratively in IoT data collection, and their location and movement behavior will affect each other, forming a multi-agent cooperation problem. Therefore, this problem can be modeled and solved using Markov games.

(1) State Space

In time slot k, the position of mth UAV is denoted as $q_{k,m}$, the remaining energy of mth UAV is $E_{k,m}$; the distance measured by the rangefinder is $D_{k,m} = \left\{d_{k,m}^1, \ldots, d_{k,m}^f\right\}$, where f is the number of rangefinders equipped on the UAV, the position of the IoT devices is $c = \{c(1), \ldots, c(N)\}$, the pending transmission task of the IoT device is $w_k = \left\{w_{k,1}, \ldots, w_{k,N}\right\}$, and the AoI is $A_k = \left\{A_{k,1}, \ldots, A_{k,N}\right\}$. Therefore, the system state can be defined as follows:

$$S_k = (s_{k,1}, \ldots, s_{k,M}) \tag{16}$$

where $s_{k,m}, m \in [1, M]$ represents the state of the m-th UAV and can be expressed as:

$$s_{k,m} = \left\{q_{k,m}, E_{k,m}, D_{k,m}, c, w_k, A_k\right\} \tag{17}$$

(2) Action Space

For the collaborative scenario of multiple UAVs, the joint action of the UAVs is represented as follows:

$$\mathrm{A}_k = (a_{k,1}, \ldots, a_{k,M}) \tag{18}$$

where $a_{k,m}, m \in [1, M]$ is the decision made by mth UAV in time slot k. Using $\beta_{k,m} \in [0, 2\pi]$ to denote the flying angle of mth UAV, the decision $a_{k,m}$ is expressed as follows:

$$a_{k,m} = \beta_{k,m} \tag{19}$$

Therefore, in time slot k+1, the position of the mth UAV $q_{k+1,m}$ can be expressed as follows

$$q_{k+1,m} = (x_{k,m} + vcos\beta_{k,m}, y_{k,m} + vsin\beta_{k,m}, H) \tag{20}$$

(3) Reward Function

In DRL, to minimize the AoI of data and ensure real-time accuracy, the reward function can be designed as a function of AoI:

$$r_{k,a\_m} = \begin{cases} R - A_{k,n}, \chi_m^n(t) = 1 \\ 0, \text{ others} \end{cases} \tag{21}$$

where R is a positive constant, $A_{k,n}$ is the AoI value when mth UAV chooses to communicate with the nth IoT device. As the AoI value decreases, the reward value increases, indicating that the UAV tends to arrive at that device sooner to reduce the AoI. When the UAV is in flight and not communicating with IoT devices, the reward value is set to 0.

To ensure that IoT devices are not accessed repeatedly, the reward function can be designed as follows:

$$r_{k,c\_m} = \begin{cases} -g_c, \chi_m^n(t) = 1, n \in O \\ 0, \text{ others} \end{cases} \tag{22}$$

where g_c is a positive constant, and O is the set of nodes that UAVs have visited. If mth UAV visits a node it has visited in the past, it will receive a penalty, as repeated visits to the same node are discouraged.

To prevent collisions between UAVs during path planning, the reward function can be designed as:

$$r_{k,ij} = \begin{cases} -b, d_{i,j} < 2r_p & i,j \in M \\ 0, \text{ others} \end{cases} \tag{23}$$

where b denotes a positive constant, $d_{i,j}$ is the distance between two UAVs, and r_p is the rotor radius.

To avoid collisions, the reward function can be designed as follows:

$$r_{k,b\_m} = \begin{cases} -b_{\text{blk}}, d_{m,blk} < 2r_p \\ 0, \text{ others} \end{cases} \tag{24}$$

where b_{blk} is a positive constant, and $d_{m,blk}$ is the distance between the UAV and the obstacle.

Simultaneously, to ensure that the battery power of the UAV remains above zero during operation, the reward function is expressed as follows:

$$r_{k,e\_m} = \begin{cases} -g_e, E_{k,m} <= 0 \\ 0, \quad E_{k,m} > 0 \end{cases} \tag{25}$$

where g_e also indicates a positive constant and $E_{k,m}$ denotes the battery power of mth UAV.

Therefore, the cumulative reward r_k is formulated as follows:

$$r_k = \sum_{m=1}^{M} \sum_{n=1}^{N} \left(r_{k,a\_m} + r_{k,c\_m} \right) + \sum_{i=1}^{M} \sum_{j \neq i}^{M} r_{k,ij} + \sum_{m=1}^{M} \left(r_{k,b\_m} + r_{k,e\_m} \right) \tag{26}$$

3.2. DP-MATD3 Algorithm: Concept and Workflow

Traditional reinforcement learning algorithms based on value and policy encounter limitations when addressing the problem of multi-UAV trajectory planning. Value-based algorithms focus on learning the value functions of states or actions but often overlook the details of policies, making it challenging to adequately consider interactions and cooperation among agents in complex multi-agent collaborative scenarios, thus impacting system performance. The policy-based algorithms directly learn policy functions. However, in multi-agent problems, the policy space is typically large, leading to high computational and sample complexities.

To overcome these limitations, the MATD3 algorithm based on the Actor-Critic framework is utilized. It integrates the learning of policy functions and value functions, considering both the importance of policies and the information from value functions. The solution to this multi-agent stochastic game process is achieved through centralized training and decentralized execution, providing a comprehensive approach to address the challenges in multi-UAV path planning.

(1) TD3 Algorithm

The TD3 algorithm is designed to address reinforcement learning problems in continuous action spaces, building upon the foundation of the DDPG algorithm. It introduces several improvements to mitigate issues present in the DDPG algorithm, including the problem of overestimating Q-values. Through the incorporation of twin critic networks, delayed updates, and policy noise, TD3 aims to enhance learning efficiency and stability compared to DDPG.

By precisely defining the boundaries of each action, the TD3 algorithm ensures the rationality of actions. In contrast to the DQN algorithm, TD3 exhibits greater flexibility in handling multidimensional variables, cleverly optimizing multiple actions simultaneously. Through techniques such as batch sampling and neural network estimation, TD3 obtains a

probability distribution that encompasses multiple actions, allowing it to derive solutions with associated actions.

TD3's core structure comprises an Actor and two Critic networks. Each of them comprises two sub-networks: an online network for real-time decision-making and a target network for stable training. To store the accumulated experience during training, TD3 utilizes a Replay Buffer. When the replay buffer reaches full capacity, TD3 clears the oldest experiences to make room for the latest ones, helping to break the correlation between experiences in small batches.

Compared to the DDPG algorithm, the TD3 algorithm significantly demonstrates advantages in the following three aspects:

(1) Dual Critic Networks

TD3 introduces two critic networks, effectively reducing the overestimation of Q-values.

(2) Delay Update

By implementing a delayed mechanism for updating the target network, TD3 reduces the frequency of updating the target networks, which helps to slow down the learning process of the algorithm, making it more stable and preventing inaccurate Q-values from prematurely affecting the policy network.

(3) Policy Noise

TD3 introduces a target policy network and adds noise to its output to enhance the exploratory nature of the policy. This mechanism effectively improves the algorithm's exploration capability, particularly demonstrating superior performance in complex environments.

(2) MATD3 Algorithm

The MATD3 algorithm is an extension of the TD3 algorithm, combining the theoretical framework of DRL with the concept of multi-agent cooperation. It is suitable for scenarios where multiple UAVs collaborate to plan paths to minimize performance metrics such as information age. During training, the Critic network functions can obtain information from all agents. Each agent i, utilizing the information obtained, can learn two centralized evaluation functions $Q^{\pi}_{i,\theta_{1,2}}(s, a_1, \ldots, a_N)$. To address the issue of overestimated Q-values, when calculating the target Q-value, the minimum evaluation network is chosen for computation:

$$y_i = r_i + \gamma \min_{j=1,2} Q^{\pi}_{i,\theta_j}\left(s', a'_l, \ldots, a'_M\right) \tag{27}$$

In addition, to ensure the exploratory and robust nature of the learned policy, clipped Gaussian noise is added to the output, as outlined in Equation (21):

$$a'_i = \mathrm{clip}\left[\mu_{\theta'_i}\left(s'_i\right) + \mathrm{clip}(N(0,\delta), -c, c), a_{\mathrm{low}}, a_{\mathrm{high}}\right] \tag{28}$$

The Critic network is updated at a higher frequency, with the Actor network updated every d times after the Critic network update. This is conducted to ensure that the updates to the Actor network are more targeted and effective, as the Critic network has been updated multiple times before updating the Actor network, providing more accurate value function estimates. Additionally, the target networks also adopt a delayed updating strategy, enhancing the stability of the algorithm.

(3) DP-MATD3 Algorithm

The MATD3 algorithm is a fusion of reinforcement learning and neural networks, respectively, using their deep learning advantages and parametric representation ability. In the construction and parameter determination process of neural networks, the key lies in determining the network's structure and weights. However, the learning speed of neural network algorithms is relatively slow, and they are prone to getting stuck in local minima, posing challenges for MATD3.

The Critic network plays a central role in MATD3, responsible for estimating the value of actions in the environment. This process involves a significant amount of weight adjustments, and the slow learning and susceptibility to local minima in neural network

algorithms make the optimization of weights particularly challenging. To overcome these challenges, the MATD3 algorithm introduces the PSO algorithm, which has advantages such as fast convergence, simplicity, and global search capability. The network performance of the MATD3 algorithm can be enhanced by combining the PSO algorithm with the Critic network and leveraging its global search capability to train its weights.

In the PSO algorithm, each particle represents a possible solution, where its position denotes the values of the Critic network parameters, and its velocity indicates the direction and stride of the particle's movement. The particle updates its speed and position by comparing its individual and group best position, namely p_{best} and g_{best}. The update process involves the following steps:

(1) Initialize Particle Swarm

Firstly, G particles in the D-dimensional search space are initialized, where $X_g = (x_{g1}, \ldots x_{gD})$ represents the position of the g-th particle and is a set of weights and biases for the Critic network, and $V_g = (V_{g1}, \ldots, V_{gD})$ represents its speed. These particles are randomly distributed in the parameter space.

(2) Compute Fitness Value

For each particle, based on its represented parameter settings, the performance of the corresponding Critic network on the training set is calculated. The Critic network undergoes a training process that involves gradually adjusting its parameters by minimizing the absolute TD error, ultimately enhancing the accuracy of Q-value estimation. Consequently, the magnitude of the TD error serves as a critical fitness metric, guiding the particles toward optimal parameter configurations. Among them, TD error is used to measure the difference between the Q-value estimated by the agent in a given state based on the current strategy and the target Q-value calculated based on experience.

(3) Determine Individual and Global Optimal Positions

Individual and global optimal position refer to the fitness values and corresponding positional parameters of individual particle and whole particle populations in their history, respectively.

(4) Update Velocity and Position

The formulas for position and velocity update are as shown in Equations (22) and (23):

$$V_{gd}(t+1) = \varphi \times \left[\omega V_{gd}(t) + c_1 r_1 \left(p_{\text{best},gd} - x_{id}(t)\right) + c_2 r_2 \left(g_{\text{best},d} - x_{gd}(t)\right)\right] \tag{29}$$

$$X_{gd}(t+1) = X_{gd}(t) + V_{gd}(t+1) \tag{30}$$

here, $g = \{1, 2, \ldots, G\}$, $d = \{1, 2, \ldots, D\}$, r_1 and r_2 are uniformly distributed on [0, 1] and are mutually independent, and φ is the constriction factor.

(5) Determine termination condition

If the termination condition is not satisfied, go back to step 2; otherwise, end. The particle swarm adjusts its velocity and position according to the fitness values, gradually converging towards the global optimal solution.

Each time an operation is performed, all particles execute simultaneously, and the state of each particle is continuously updated, avoiding any waiting issues. During the process of adjusting states, particles adapt their individual states at any moment until the global optimal solution is found. The implementation steps are illustrated in Figure 3.

In the updating process of PSO, the particle swarm searches the entire parameter space, gradually converging to the global optimum. The absolute value of TD-error is employed as the fitness value, the Critic network can learn the value function more accurately, and then guide the Actor network to update the policy, thus improving the performance of the algorithm.

In addition, in order to further enhance the convergence speed of the MATD3 algorithm, the dual experience pools technique is introduced. However, the target region is relatively vast and the capacity of the experience pool is limited. This leads to the pos-

sibility that, under limited exploration, the experience pool may store a large number of suboptimal actions, affecting training efficiency. Therefore, two experience pools, B_1 and B_2, are considered. B_1 indiscriminately stores samples, while B_2 is used to store experiences with single-step rewards $r_t > v$, and its size is much smaller than B_1. During the sampling process, a subset of samples is randomly selected from B_1, and an additional 10% of the total samples are randomly chosen from B_2.

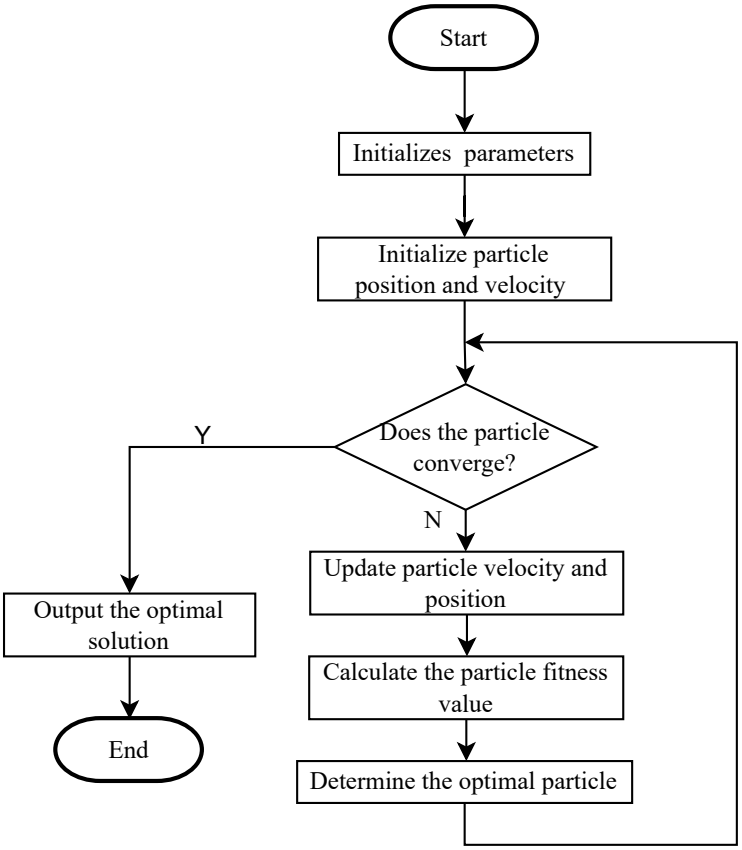

Figure 3. Particle swarm algorithm implementation process.

The DP-MATD3 algorithm combines the PSO algorithm to optimize the parameters of the Critic network while utilizing dual-experience pools to overcome the shortcomings of the MATD3 algorithm. The DP-MATD3 algorithm is illustrated in Figure 4 and the key procedures are generalized in Algorithm 1.

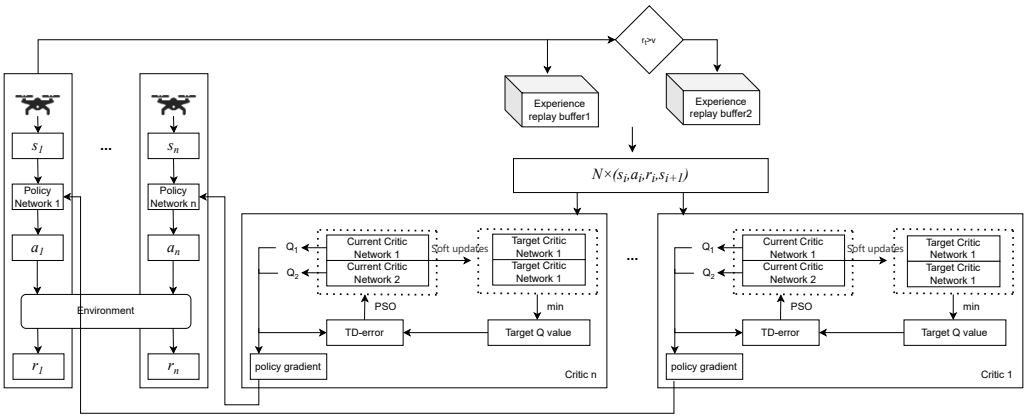

Figure 4. DP-MATD3 algorithm block diagram.

Algorithm 1 DP-MATD3 algorithm

Initialization: The network parameters of each UAV are randomly initialized. Initialize the replay experience pools B_1 and B_2, and set the mini-batch sample size N. Initialize the proportions p_1 and p_2 for sampling from experience pools B_1 and B_2, respectively, with $p_1 = (1 - p_2)$. Initialize the number and positions of IoT devices, as well as the number and positions of UAVs.

for $episode = 0, 1, \ldots, L$ **do**

Obtain the initial environment observation values s for each UAV.

for $t = 1, \ldots, T$ **do**

mth UAV selects an action a_m;
Perform action a_m to obtain reward r_m and new environmental observation s_m;

if the state s_m encounters an obstacle or goes beyond the boundary **do**

break

end if

Store the state transition tuple (s_m, a_m, r_m, s'_m) in B_1;

if $r_m > v$ **do**

Store the state transition tuple (s_m, a_m, r_m, s'_m) in B_2;

end if

Randomly sample $p_1 * N$ tuples $(s_t, a_t, r_t, s'_{t+1})$ from B_1 and $p_2 * N$ tuples (s, a, r, s') from B_2 to learn;
Add exploration noise to the action's output by the Actor target network $a'_m = \mu_{\theta'_m}(s'_m) + \tilde{\varepsilon}, \tilde{\varepsilon} \sim \mathrm{clip}(N(0, \delta), -c, c)$
Setting $\mathrm{y}_m = \mathrm{r}_m + \gamma \min_{\mathrm{j}=1,2} \mathrm{Q}^{\pi}_{\mathrm{m},\theta_{\mathrm{j}}}(s'_m, \mathrm{a}'_1, \ldots, \mathrm{a}'_{\mathrm{M}})$;
Calculate the absolute value of the TD-error $\left| y_m - \min_{j=1,2} \mathrm{Q}^{\pi}_{\mathrm{m},\theta_{\mathrm{j}}}(s_m, \mathrm{a}_1, \ldots, \mathrm{a}_M) \right|$
Based on the optimal particle's location, the Critic network's parameters are updated;

if t mod c_1 then

Update Actor params using the gradient of the sampled data $\nabla_{\theta^{\mu_m}} J(\mu_m) \approx \frac{1}{N} \sum_{i=1}^{N} [\nabla_{a_m} Q(s_m, a_1, \ldots, a_M) \nabla_{\theta_m} \mu_m(s_m \mid \theta^{\mu_m})]$

if t mod c_2 then

Update target net params.

end for

end for

In DP-MATD3, the absolute value of TD-error is used as the fitness value of the PSO algorithm to minimize TD-error faster, so that the Critic network can learn the value function more accurately and guide the Actor network to choose better actions. The structure of dual-experience pools is designed to store more optimal experience information

by adding an additional experience pool so that the network can find the optimal strategy faster and speed up the training of the DP-MATD3 algorithm.

4. Simulation Analysis

4.1. Environmental Configuration

This section simulates the designed the multi-UAV trajectory planning strategy. The simulation scenario consists of a square area measuring 1.5 km × 1.5 km. As shown in Figure 5, the orange circles represent IoT devices, while three UAVs (green circles) depart from locations (0.1 km, 0.1 km), (1.3 km, 0.1 km), and (1.3 km, 1.3 km). The parameters for the UAVs are presented in Table 1.

Table 1. Parameters of the aerial vehicle.

Parameter	Value
channel gain h_0	−50 dB
communication bandwidth B	2 MHz
upload power P_{up}	0.1 W
UAV fixed altitude H	50 m
noise power σ^2	−100 dBm

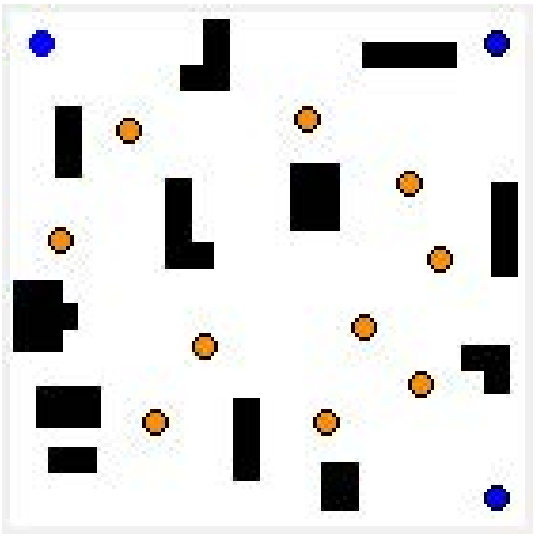

Figure 5. The map of the environment.

The networks in the DP-MATD3 algorithm are constructed with two hidden layers of artificial neural networks, both being fully connected layers. The experimental platform is established using Python 3.7 and PyTorch 1.5.1. The remaining parameters are illustrated in Table 2.

Table 2. Simulation Parameter Setting.

Parameter	Value
discount factor	0.95
Critic learning rate	0.001
Actor learning rate	0.002
experience pool B_1	20,000
experience pool B_2	1000
batch size	256
Maximum training steps	10,000

4.2. Experimental Results and Analysis

The DP-MATD3 algorithm optimizes the MATD3 algorithm's Critic network by introducing the PSO and uses the absolute value of the TD-error as the fitness value for the PSO to achieve faster minimization of TD-error. To assess the algorithm's convergence, the trend of TD-error absolute value over training steps was used as an evaluation criterion. Figures 6 and 7 demonstrate that, under the same number of training steps, the DP-MATD3 algorithm reaches a stable state faster than the MATD3 algorithm. This is attributed to the PSO algorithm's ability to rapidly explore the optimal solution space, optimizing Critic network parameters and quickly reducing TD-error absolute value, thus expediting the overall convergence process.

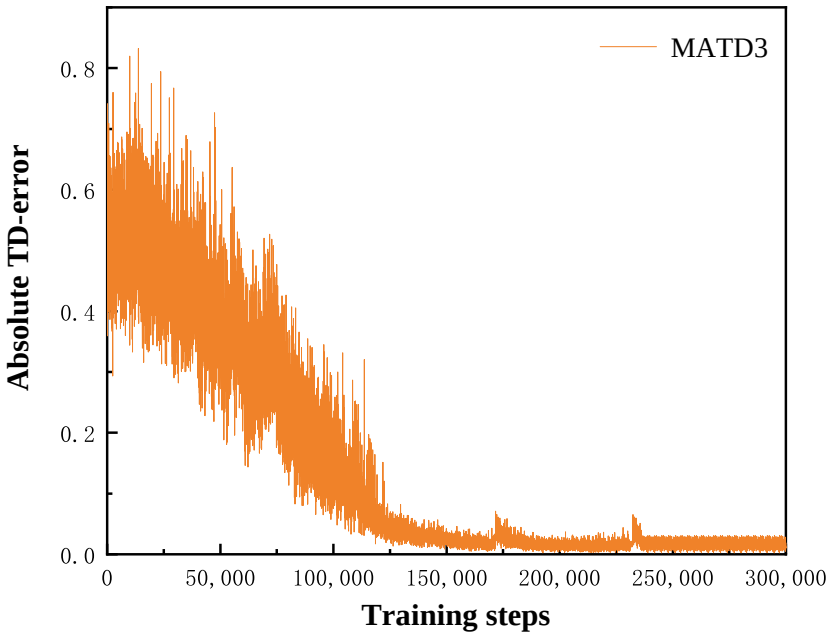

Figure 6. The convergence of the absolute value of TD-error in the MATD3 algorithm.

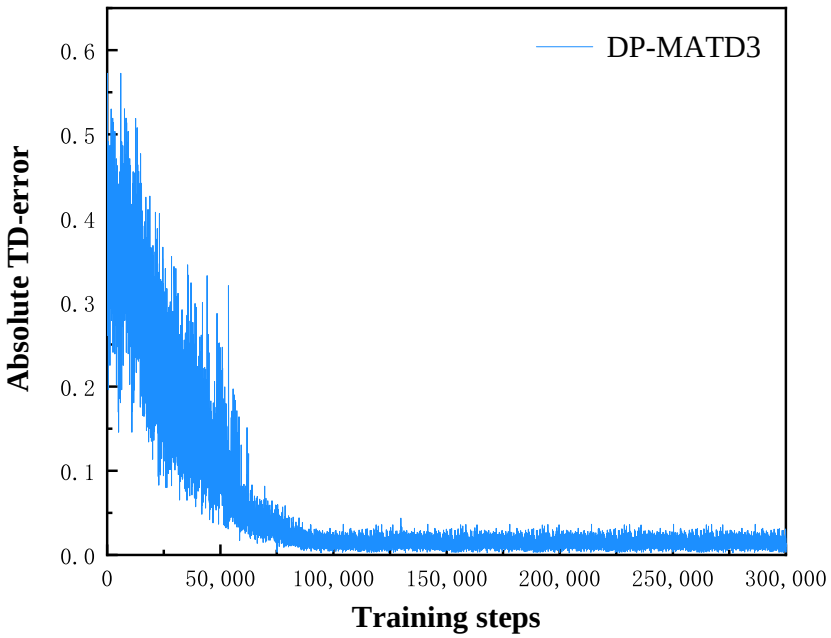

Figure 7. The convergence of the absolute value of TD-error in the DP-MATD3 algorithm.

Figure 8 shows the trajectory diagram under the DP-MATD3 algorithm designed in this chapter. At t = 0, all IoT devices perform a data update. Subsequently, IoT devices

N1 and N6 conduct another data update at t = 100 s, devices N7 and N8 at t = 200 s, and device N3 at t = 300 s. The remaining IoT devices do not perform a second update within the mission cycle. Observing the trajectory in the figure, due to the flight time required for the UAVs to reach the IoT devices, the DP-MATD3 algorithm fully considers the tradeoff between flight time and AoI in trajectory planning. The algorithm achieves the minimization of AoI by jointly planning the collaborative working trajectory of the UAVs and the connection sequence of the UAVs. For example, UAV1 chooses to gather data from N2 first, not N1 nearby. This is because N1 performs a data update at t = 100 s, and by then, UAV1 will have access to the updated data. Therefore, UAV1 chooses to collect data from N2 first, thus reducing the overall AoI. Similarly, for N3, N7, and N8, which have longer update cycles, the UAVs fully consider their update frequencies and locations in the planning, choosing appropriate time points for data collection. These results effectively demonstrate the superiority of the DP-MATD3 algorithm in trajectory planning.

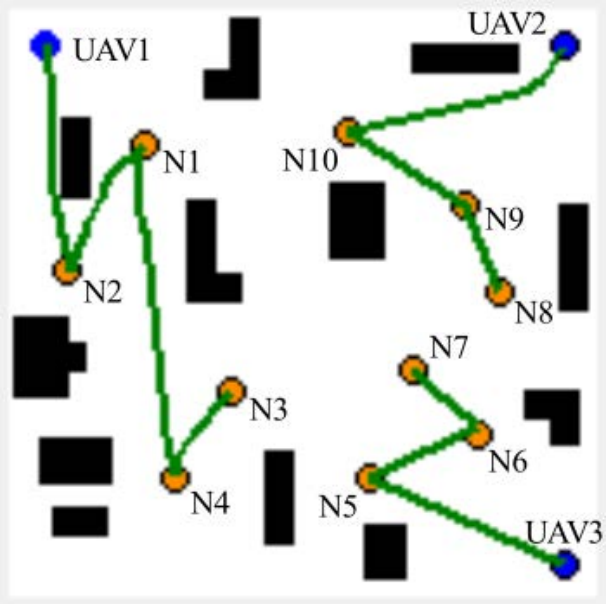

Figure 8. UAV flight trajectory.

The weighted average AoI between the DP-MATD3 and the MATD3 at the UAV speed of 5 m/s and 10 m/s are depicted in Figures 9 and 10, respectively. From the graphs, it is evident that the convergence and stability of the DP-MATD3 algorithm are better at different speeds. For instance, when v = 10 m/s, the weighted average AoI achieved by the DP-MATD3 algorithm is approximately 121.59 s, while the MATD3 algorithm's weighted average AoI is around 155.03 s. It can be seen that DP-MATD3 achieves the reduction in the weighted average AoI and improves the system performance. This is because the DP-MATD3 algorithm can generate more reasonable trajectories based on the actual information update patterns. For example, concerning N1, since it will undergo a second data update at t = 100 s, UAV1 collects its data after collecting N2's data, resulting in a smaller AoI. In contrast, the MATD3 algorithm completes information collection before its second update, leading to a larger AoI, falling into a local optimum.

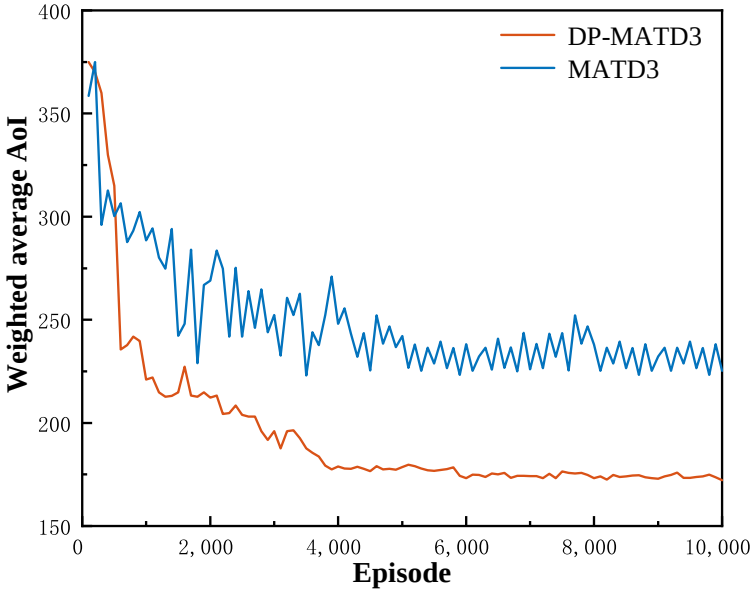

Figure 9. Weighted average AoI at v = 5m/s.

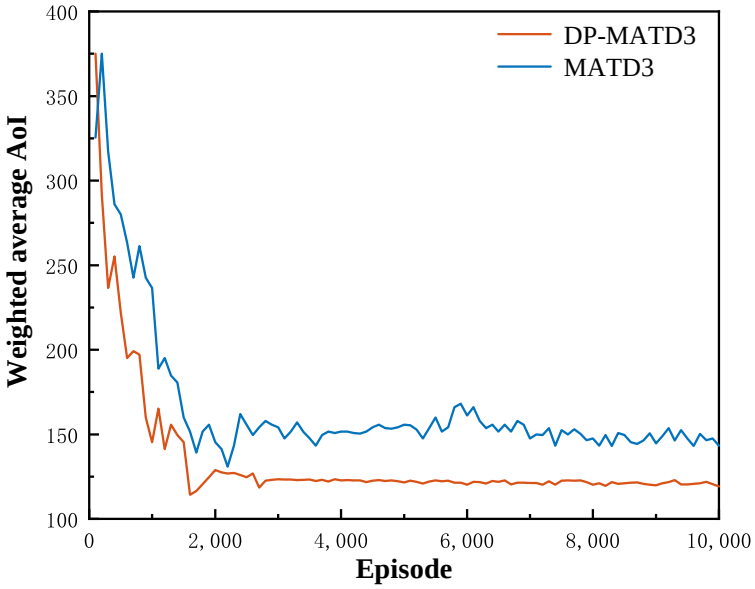

Figure 10. Weighted average AoI at v = 10 m/s.

5. Conclusions

This paper investigated the path planning problem for multi-UAV collaboration in completing data collection tasks and proposed a DP-MATD3 path planning algorithm to minimize the weighted average AoI. The algorithm optimizes the Critic network by introducing the PSO algorithm, overcoming the issue of the MATD3 algorithm being prone to local optima. Additionally, the algorithm is extended and optimized through the introduction of a dual experience pool structure to enhance training efficiency. The simulation experiments, demonstrating the path planning results for multiple UAVs, effectively validated the feasibility of the DP-MATD3 algorithm in path planning. A comparative analysis with baseline strategies also clearly indicated a significant improvement in algorithm performance.

Author Contributions: H.H.: Revised the manuscript and approved the final version. Y.L.: Conceived, conducted experiments, acquired data, and drafted and revised the manuscript. G.S.: Revised the manuscript and supervision. W.G.: Funding acquisition and writing guidance. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Shandong Province (ZR2023MF112).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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.

References

1. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
2. Cui, Q.; Zhang, J.; Zhang, X.; Chen, K.; Tao, X.; Zhang, P. Online anticipatory proactive network association in mobile edge computing for IoT. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 4519–4534. [CrossRef]
3. Li, Z.; Xu, Y.; Wang, P.; Xiao, G. Restoration of multi energy distribution systems with joint district network recon figura-tion by a distributed stochastic programming approach. *IEEE Trans. Smart Grid* **2023**, *15*, 3317780.
4. Samir, M.; Sharafeddine, S.; Assi, C.M.; Nguyen, T.M.; Ghrayeb, A. UAV trajectory planning for data collection from time-constrained IoT devices. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 34–46. [CrossRef]
5. Li, G.; He, J.; Peng, S.; Jia, W.; Wang, C.; Niu, J.; Yu, S. Energy efficient data collection in large-scale internet of things via computation offloading. *IEEE Internet Things J.* **2019**, *6*, 4176–4187. [CrossRef]
6. Samir, M.; Assi, C.; Sharafeddine, S.; Ebrahimi, D.; Ghrayeb, A. Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach. *IEEE Trans. Veh.* **2020**, *69*, 12382–12395. [CrossRef]
7. Huang, H.; Li, Z.; ; Gooi, H.B.; Qiu, H.; Zhang, X.; Lv, C.; Liang, R.; Gong, D. Distributionally robust energy-transportation coordination in coal mine integrated energy systems. *Appl. Energy* **2023**, *333*, 120577. [CrossRef]
8. Kosta, A.; Pappas, N.; Angelakis, V. Age of information: A new concept, metric, and tool. *Found. Trends Netw.* **2017**, *12*, 162–259. [CrossRef]
9. Zheng, H.; Xiong, K.; Fan, P.; Zhong, Z.; Letaief, K.B. Age of information-based wireless powered communication networks with selfish charging nodes. *IEEE J. Sel.* **2021**, *39*, 1393–1411. [CrossRef]
10. Chan, T.; Pan, H.; Liang, J. Age of information with joint packet coding in industrial IoT. *IEEE Wireless Commun. Lett.* **2021**, *10*, 2499–2503. [CrossRef]
11. Chen, Q.; Guo, S.; Xu, W.; Cai, Z.; Cheng, L.; Gao, H. AoI minimization charging at wireless-powered network edge. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems, Bologna, Italy, 10–13 July 2022.
12. Pu, C.; Yang, H.; Wang, P.; Dong, C. AoI-Bounded Scheduling for Industrial Wireless Sensor Networks. *Electronics* **2023**, *12*, 162–259. [CrossRef]
13. Zhao, F.; Sun, X.; Zhan, W.; Wang, X.; Chen, X. Information Freshness in Random-Access Poisson Network: Average AoI versus Peak AoI. In Proceedings of the 2022 IEEE 96th Vehicular Technology Conference, London, UK, 26–29 September 2022.
14. Zhou, B.; Saad, W. Minimizing age of information in the Internet of Things with non-uniform status packet sizes. In Proceedings of the 2019 IEEE International Conference on Communications, Shanghai, China, 20–24 May 2019.
15. Gu, Y.; Chen, H.; Zhai, C.; Li, Y.; Vucetic, B. Minimizing age of information in cognitive radio-based IoT systems: Underlay or overlay? *IEEE Internet Things J.* **2019**, *6*, 10273–10288. [CrossRef]
16. Motlagh, N.H.; Taleb, T.; Arouk, O. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet Things J.* **2016**, *3*, 10273–10288.
17. Wang, J.; Liu, Y.; Niu, S.; Song, H. Reinforcement learning optimized throughput for 5G enhanced swarm UAS networking. In Proceedings of the IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021.
18. Hu, H.; Xiong, K.; Qu, G.; Ni, Q.; Fan, P.; Letaief, K.B. AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks. *IEEE Internet Things J.* **2021**, *8*, 1211–1223. [CrossRef]
19. Gao, X.; Zhu, X.; Zhai, L. AoI-sensitive data collection in multi-uav-assisted wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5185–5197. [CrossRef]
20. Xiong, J.; Li, Z.; Li, H.; Tang, L.; Zhong, S. Energy-Constrained UAV Data Acquisition in Wireless Sensor Networks with the Age of Information. *Electronics* **2023**, *12*, 1739. [CrossRef]
21. Lu, Y.; Hong, Y.; Luo, C.; Li, D.; Chen, Z. Optimization Algorithms for UAV-and-MUV Cooperative Data Collection in Wireless Sensor Networks. *Drones* **2023**, *7*, 408. [CrossRef]
22. Liu, K.; Zheng, J. UAV trajectory optimization for time-constrained data collection in UAV-enabled environmental monitoring systems. *IEEE Internet Things J.* **2022**, *9*, 24300–24314. [CrossRef]

23. Hu, J.; Zhang, H.; Song, L.; Schober, R.; Poor, H.V. Cooperative internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning. *IEEE Trans. Commun.* **2020**, *68*, 6807–6821. [CrossRef]
24. Zhou, C.; He, H.; Yang, P.; Lyu, F.; Wu, W.; Cheng, N.; Shen, X. Deep RL-based trajectory planning for AoI minimization in UAV-assisted IoT. In Proceedings of the 2019 11th International Conference on Wireless Communications and Signal Processing, Xi'an, China, 23–25 October 2019.
25. Abd-Elmagid, M.A.; Ferdowsi, A.; Dhillon, H.S.; Saad, W. Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks. In Proceedings of the 2019 IEEE Global Communications Conference, Waikoloa, HI, USA, 9–13 December 2019.
26. Peng, Y.; Liu, Y.; Li, D.; Zhang, H. Deep reinforcement learning based freshness-aware path planning for UAV-assisted edge computing networks with device mobility. *Remote Sens.* **2022**, *14*, 4016. [CrossRef]
27. Yin, B.; Li, X.; Yan, J.; Zhang, S.; Zhang, X. DQN-based Power Control and Offloading Computing for Information Freshness in multi-DAV-assisted V2X System. In Proceedings of the 2022 IEEE 96th Vehicular Technology Conference, London, UK, 26–29 September 2022.
28. Chen, X.; Wu, C.; Chen, T.; Liu, Z.; Bennis, M.; Ji, Y. Age of information-aware resource management in UAV-assisted mobile-edge computing systems. In Proceedings of the 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020.
29. Lin, X.; Yajnanarayana, V.; Muruganathan, S.D.; Gao, S.; Asplund, H.; Maattanen, H.; Bergstrom, M.; Euler, S.; Wang, Y.-P.E. The sky is not the limit: LTE for unmanned aerial vehicles. *IEEE Commun. Mag.* **2018**, *56*, 204–210. [CrossRef]
30. Bergh, B.V.D.; Chiumento, A.; Pollin, S. LTE in the sky: Trading off propagation benefits with interference costs for aerial nodes. *IEEE Commun. Mag.* **2016**, *54*, 44–50 . [CrossRef]
31. Wang, Y.; Fang, W.; Ding, Y.; Xiong, N. Computation offloading optimization for UAV-assisted mobile edge computing: a deep deterministic policy gradient approach. *Wirel. Netw.* **2021**, *27*, 2991–3006. [CrossRef]