electronics

MDPI

Article

Joint AP Selection and Task Offloading Based on Deep Reinforcement Learning for Urban-Micro Cell-Free UAV Network

Chunyu Pan [1,2], Jincheng Wang [1,2,*], Xinwei Yue [1,2], Linyan Guo [3] and Zhaohui Yang [4]

[1] Key Laboratory of Information and Communication Systems, Ministry of Information Industry, Beijing Information Science and Technology University, Beijing 100101, China; chunyupan@bistu.edu.cn (C.P.); xinwei.yue@bistu.edu.cn (X.Y.)
[2] Key Laboratory of Modern Measurement & Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing 100101, China
[3] School of Geophysics and Information Technology, Beijing Campus, China University of Geosciences, Beijing 100083, China; guoly@cugb.edu.cn
[4] School of Information Science and Electronic Engineering, Department of Information and Communication Engineering, Yuquan Campus, Zhejiang University, Hangzhou 310027, China; yang_zhaohui@zju.edu.cn
* Correspondence: jincheng.wang@bistu.edu.cn

Abstract: The flexible mobility feature of unmanned aerial vehicles (UAVs) leads to frequent handovers and serious inter-cell interference problems in UAV-assisted cellular networks. Establishing a cell-free UAV (CF-UAV) network without cell boundaries effectively alleviates frequent handovers and interference problems and has been an important topic of 6G research. However, in existing CF-UAV networks, a large amount of backhaul data increases the computational pressure on the central processing unit (CPU), which also increases system delay. Meanwhile, the mobility of UAVs also leads to time-varying channel conditions. Therefore, designing dynamic resource allocation schemes with the help of edge computing can effectively alleviate this problem. Thus, aiming at partial network breakdown in an urban-micro (UMi) environment, an urban-micro CF-UAV (UMCF-UAV) network architecture is proposed in this paper. A delay minimization problem and a dynamic task offloading (DTO) strategy that jointly optimizes access point (AP) selection and task offloading is proposed to reduce system delay in this paper. Considering the coupling of various resources and the non-convex feature of the proposed problem, a dynamic resource cooperative allocation (DRCA) algorithm based on deep reinforcement learning (DRL) to flexibly deploy AP selection and task offloading of UAVs between the edge and locally is proposed to solve the problem. Simulation results show fast convergence behavior of the proposed algorithm compared with classical reinforcement learning. Decreased system delay is obtained by the proposed algorithm compared with other baseline resource allocation schemes, with the maximize improvement being 53%.

Keywords: unmanned aerial vehicle (UAV); 6G communication; cell-free network; deep reinforcement learning; edge computing; resource allocation

Citation: Pan, C.; Wang, J.; Yue, X.; Guo, L.; Yang, Z. Joint AP Selection and Task Offloading Based on Deep Reinforcement Learning for Urban-Micro Cell-Free UAV Network. *Electronics* **2023**, *12*, 4777. https://doi.org/10.3390/electronics12234777

Academic Editor: Carlos Tavares Calafate

Received: 24 October 2023
Revised: 14 November 2023
Accepted: 24 November 2023
Published: 25 November 2023

1. Introduction

In recent years, there have been frequent natural disasters around the world, such as the earthquake that affected central Mexico in 2017 and the wildfires in Washington State in 2020, which can have a very serious impact if they occur in densely populated urban environments. The infrastructure of traditional ground networks is easily damaged by disasters [1], which leads to the failure of mobile phones that can be used to transmit information to the outside world and, thus, increases the difficulty of rescue [2]. One solution is to use satellite networks, but satellite communication costs are extremely high and have a high system delay, while an air base station tethered to a balloon lacks mobility [3]. In order to cope with emergency scenarios, it is an effective response measure to build a

flexible ground-to-air network by using unmanned aerial vehicles (UAVs) [1–3]. However, in traditional cellular networks, the flexible mobility of UAVs leads to frequent handovers and inter-cell interference issues [4]. The emergence of cell-free networks without cells and cell boundaries effectively reduces this problem and has been an important topic of future 6G research [5]. A cell-free network is a distributed multiple-input multiple-output (MIMO) system with a large number of access points (APs). All APs are connected to the central processing unit (CPU) through the backhaul link for data transmission pre-processing and signal detection, and multiple APs simultaneously provide services to a user in the same time–frequency resources block [6]. A cell-free UAV (CF-UAV) network can be constructed to reduce the problem of frequent handovers and inter-cell interference of UAVs in traditional cellular-based ground-to-air networks. However, all APs need to communicate with the CPU, which leads to high backhaul link overhead, high CPU computation pressure, and high system delay. To solve this problem, it is a feasible solution to equip edge servers on the AP side to alleviate the CPU computational pressure in CF-UAV networks through edge computing [7]. Further, for a CF-UAV network, the channel between the UAV and the AP changes with the mobility of the UAV in each time slot. Such channel time variability causes time-varying computing and communication resource allocation demands, which is a challenge in urban emergency scenarios. The urban-micro (UMi) scenario is the most sensitive scenario compared with other urban scenarios, and delay is the most critical performance index in this scenario [8]. Therefore, it is necessary to design an efficient AP selection and task offloading scheme to reduce the system delay of CF-UAV networks in a UMi environment.

1.1. Related Works and Motivations

Existing studies have made some progress on cell-free networks and resource allocation problems. In ref. [9], the authors studied the impact of different CPU deployment methods on the resource allocation of cell-free networks. Ref. [10] proposed a cell-free network architecture based on network slicing and carried out a resource allocation scheme. Ref. [11] proposed a directed acyclic graph-based algorithm to solve the task offloading problem in cell-free networks. In ref. [12], the authors proposed an integrated sensing and communication system to solve the resource allocation problem in cell-free networks. Ref. [13] used cell-free networks to build a federated learning architecture and reduced uplink training time through power control. Refs. [14,15] focused on optimizing AP selection for cell-free networks through machine learning.

Compared with a ground network, a ground-to-air network has higher-dimensional resources due to the introduction of UAVs, and on their own, the resource management measures of the ground network cannot cope with the high-dimensional and dynamic resources brought by UAVs. Therefore, academia and industry have studied many resource allocation problems in UAV-assisted networks. Ref. [16] investigates a UAV-enabled wireless communication system with energy harvesting to optimize UAV path planning and energy. In ref. [17], a cache-enabled UAV network was studied, and the authors jointly optimized user association, spectrum allocation, and content caching. Ref. [18] applied a UAV network to an intelligent reflective surface to improve the energy efficiency of the system by optimizing the UAV trajectories. Ref. [19] adopted a continuous convex approximation algorithm to conduct joint optimization of UAV trajectories, power control, and user associations. Ref. [20] combined mobile edge computing and network function virtualization technology to optimize the system delay in UAV-assisted industrial internet scenarios.

In order to reduce the frequent handovers and inter-cell interference problems caused by the mobile characteristics of UAVs, the CF-UAV network came into being. In refs. [21–25], the authors considered a UAV-assisted cell-free network and optimized the resources in the network. In ref. [21], the authors adopted block optimization and quadratic transformations for resource optimization in the special case of hardware impairments in CF-UAV networks. In ref. [22], a dynamic AP selection strategy based on average power strength was proposed in CF-UAV networks. Ref. [23] optimized network resources through UAV deployment in

CF-UAV networks. Ref. [24] proposed three UAV trajectory design schemes to improve the spectral efficiency. Ref. [25] studied resource allocation in CF-UAV networks through the combination of a gradient-based algorithm and a Gibbs sampling algorithm.

In recent years, there has been a significant increase in the number of mobile devices, which generates a large amount of data [26]. On the one hand, the increase in data volume causes additional computing burden on the CPU of the network, and a large amount of data is transmitted to the CPU through the backhaul link, which also has certain privacy risks [27]. Therefore, edge computing can effectively alleviate this problem, and users can offload data to a nearby edge server for calculation, reducing the communication overhead and security risks generated by sending data back to the CPU. Ref. [28] provides an overview of practical distributed edge learning techniques and their interplay with advanced communication optimization designs Refs. [29,30] combined cell-free networks with edge computing. User tasks were offloaded to edge servers to solve the joint communication and computational resource allocation problem and reduce CPU computational pressure. In ref. [31], a UAV was used for mobile edge computing, which was combined with DRL to optimize task offloading in 5G-supporting software-defined networks. On the other hand, the increasing number of mobile devices brings more diverse and complex resources, which leads to the difficulty of communication and computing resource allocation. In order to solve complex optimization problems with multiple resources, artificial intelligence algorithms based on machine learning are gradually being applied to resource allocation problems [32,33].

Deep reinforcement learning (DRL) is one of the most typical representatives of machine learning algorithms. With the advantage of dealing with continuous states, DRL is very suitable for solving UAV resource allocation problems with time-variance and high-dimensionality [34]. Many authors have used DRL for resource allocation studies in UAV networks. In ref. [35], a heterogeneous UAV communication network was constructed in an emergency scenario, and a resource allocation algorithm based on DRL was adopted to optimize energy efficiency. Refs. [36,37] both applied DRL to the trajectory control of UAVs to achieve joint optimization of computing resources and user associations and to reduce energy consumption. Refs. [38,39] proposed a computational offloading strategy based on DRL to solve the problem of the resource dimensional curse in UAV-assisted communication networks. Ref. [40] applied DRL to CF-UAV networks and proposed a DRL algorithm based on a soft actor–critic to optimize UAV deployment and power allocation. Ref. [41] proposed a UAV-assisted cell-free multi-group broadcast network architecture. In order to solve the coupling between resources, an optimization algorithm based on DRL was used to effectively handle the video transmission from UAVs to virtual reality users. However, there are relatively few studies on resource allocation for CF-UAV networks in urban emergency scenarios. Therefore, it is necessary to study a resource allocation strategy to cope with emergency scenarios in a UMi environment and use the advantages of advanced DRL to solve the coupling problem between optimized variables.

Therefore, aiming at the background of local network breakdown caused by natural disasters or data congestion in an urban environment, an urban-micro CF-UAV network (UMCF-UAV) is proposed in this paper. We consider the importance of reducing communication delay in UMi emergency scenarios. Therefore, an optimization problem to minimize the system delay is formulated, and a dynamic task offloading (DTO) strategy with joint AP selection is proposed to reduce the system delay in the UMCF-UAV network. Since the variables in the optimization problem are tightly coupled in the unit time slot, an advanced dynamic resource cooperative allocation (DRCA) algorithm based on DRL is proposed to solve the optimization problem. The experimental results show that the proposed algorithm achieves the best performance compared with the classical Q-learning, random allocation, and equal allocation strategies.

1.2. Contributions and Organization

The main contributions of this paper can be summarized as follows:

- A UMCF-UAV network adapted to a UMi disaster emergency environment is proposed in this paper. The different channel states of LoS and NLoS in the ground-to-air communication model are fully considered in this paper. The minimum mean square error estimation (MMSE) is used to derive the closed-form expression of the uplink transmission rate that can be achieved by UAVs in the proposed network.
- In this paper, edge computing is used to reduce the computing pressure on the CPU in the UMCF-UAV network. The optimization problem of delay minimization is proposed. In order to reduce the system delay, a DTO strategy based on a UMi emergency scenario to jointly optimize AP selection is proposed in this paper. The proposed strategy comprehensively considers the impacts of various network factors on task offloading.
- A DRL-based DRCA algorithm is proposed in this paper to solve this non-convex problem with tightly coupled variables. Compared with the baseline algorithm, the algorithm proposed in this paper approximates the global optimal solution obtained by traversal search, which effectively reduces the delay.

The structure of this paper is as follows: Section 2 introduces the system model and channel modeling of the network. Section 3 introduces the channel estimation and the transmission process of the uplink and downlink. Section 4 proposes the optimization problem of this paper. Section 5 introduces the DRL-based DRCA algorithm. Section 6 gives the parameter settings and simulation results. Section 7 concludes the paper.

Symbol description: in this paper, vectors and matrices are represented by bold italic lowercase letters $\boldsymbol{a}$ and bold regular uppercase letters $\mathbf{A}$, respectively, superscript a^* represents a conjugate, $\boldsymbol{a}^H$ and $\mathbf{A}^H$ represent a conjugate transpose, $\|\|$ represents a Euclidean norm, $\mathbb{E}\{\}$ represents an expectation operator, $\triangleq$ is used as a definition expression, and finally, $a \sim \mathcal{CN}\left(0, \sigma^2\right)$ denotes a circularly symmetric complex Gaussian random variable a with zero mean and variance σ^2.

2. System Model

In this section, we focus on the system model and channel modeling method of the network, including the realistic environment applicable to the system model and the specific mathematical definition of each part of the channel gain.

2.1. System Description

As shown in Figure 1, the proposed network system contains a set of APs $\mathcal{M} = \{1, 2, \ldots, M\}$ and UAVs $\mathcal{N} = \{1, 2, \ldots, N\}$, where M>N, and all APs and UAVs are equipped with a single antenna and are randomly distributed in the service area. Each AP is equipped with an edge server, and all APs can be connected to the CPU through the backhaul link in the same time–frequency resource blocks. All APs can serve all users at the same time, and users in the network include ground users (GUs) and UAVs. We assume that the UAV has a certain computing capacity and can perform simple calculations. The GUs in the waiting area need to transmit tasks to UAVs through wireless data links, and UAVs need to offload task data to the available APs in the service area. For ease of presentation, we define the waiting area as follows.

Definition 1. *The "waiting area" is the area of local network congestion or paralysis due to emergencies or natural disasters in the UMi scenario. APs in the waiting area cannot provide network services for GUs.*

Transmission between an AP and the user adopts the form of TDD, so there is reciprocity between the uplink channel and the downlink channel: the uplink channel and the downlink channel can adopt the same channel gain. We assume that the UAV has enough energy supply to ensure all the work needs in the cycle. Thus, the energy consumption of the UAV is ignored in this paper [20].

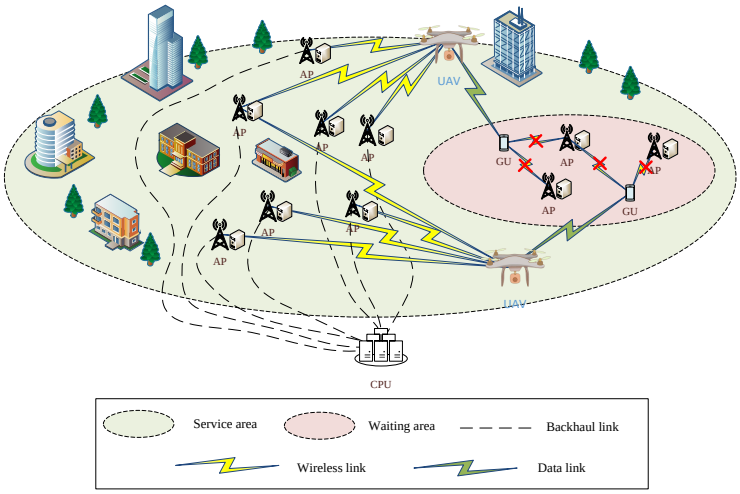

Figure 1. The proposed framework of UMCF-UAV network.

2.2. Channel Model

In the channel model, we use g_{mn} to represent the channel gain between the m-th AP and the n-th UAV, where β_{mn} represents the large-scale fading between the m-th AP and the n-th UAV, and h_{mn} represents the small-scale fading between the m-th AP and the n-th UAV. As a result, g_{mn} is given by

$$g_{mn} = \sqrt{\beta_{mn}} h_{mn}. \tag{1}$$

We assume that $h_{mn} \sim \mathcal{CN}(0,1)$, and in each channel, h_{mn} is independently and identically distributed (i.i.d.); large-scale fading β_{mn} can be defined as follows

$$\beta_{mn} = PL_{mn} 10^{\frac{\sigma_{sh} z_{mn}}{10}}, \tag{2}$$

where $10^{\frac{\sigma_{sh} z_{mn}}{10}}$ represents shadow fading, σ_{sh} represents the shadow fading standard deviation, $z_{mn} \sim N(0,1)$, and PL_{mn} represents path loss, which is related to the distance from the AP to the UAV. Since the UMi network contains obstacles such as tall buildings, and considering the line-of-sight (LoS) link characteristics of the UAV itself [42,43], path loss is defined as 3GPP TR.36.777 [21] according to link probability

$$PL_{mn} = \begin{cases} max\left\{PL'_{mn}, 30.9 + (22.25 - 0.5\log_{10}(H))\log_{10}(d^{3D}_{mn}) + 20\log_{10}(f_c)\right\}, & LoS, \\ max\left\{PL^{LOS}_{mn}, 32.4 + (43.2 - 7.6\log_{10}(H))\log_{10}(d^{3D}_{mn}) + 20\log_{10}(f_c)\right\}, & NLoS, \end{cases} \tag{3}$$

where f_c represents the central frequency, PL^{LOS}_{mn} represents the LoS link pass loss, and PL'_{mn} represents the free space path loss, defined as $PL'_{mn} = \frac{d_0}{H^2 + \left\| q^{UAV}_n - q^{AP}_m \right\|^2}$, where q^{UAV}_n represents the two-dimensional coordinates of the n-th UAV on the plane (X_n, Y_n). The variable q^{AP}_m represents the two-dimensional coordinates of the m-th AP in the plane (X_m, Y_m), H represents the height of the UAV, d_0 represents the reference distance, and d^{3D}_{mn} represents the three-dimensional space distance from the m-th AP to the n-th UAV. Further, $d^{3D}_{mn} = \sqrt{(X_n - X_m)^2 + (Y_n - Y_m)^2 + H^2}$. We define the standard deviation of shadow fading as the following expression

$$\sigma_{sh} = \begin{cases} max\{5exp(-0.01H), 2\}, & LoS, \\ 8 \qquad\qquad\qquad\quad\;\; , & NLoS. \end{cases} \tag{4}$$

LoS link probability is expressed as

$$P_{LOS} = \begin{cases} 1 & , \quad d_{mn}^{2D} \le d, \\ \frac{d}{d_{mn}^{2D}} + \left(1 - \frac{d}{d_{mn}^{2D}}\right) \exp\left(\frac{-d_{mn}^{2D}}{\varpi}\right), & d_{mn}^{2D} > d, \end{cases} \tag{5}$$

where d_{mn}^{2D} represents the two-dimensional distance between the m-th AP and the n-th UAV, $d_{mn}^{2D} = \sqrt{(X_n - X_m)^2 + (Y_n - Y_m)^2}$, and ϖ and d are variables determined by the height of the UAV. These are defined as $\varpi = 233.98\log_{10}(H) - 0.95$ and $d = max\{294.05\log_{10}(H) - 432.94, 18\}$, and the probability of a NLoS link is $P_{NLOS} = 1 - P_{LOS}$. It can be seen that P_{LOS} in Equation (5) gradually approaches 1 as the height H of the UAV increases. By substituting Equations (3) and (4) into Equation (2), we can derive two components $\beta_{mn}^{LOS} = PL_{mn}^{LOS} 10^{\frac{\sigma_{sh}^{LOS} z_{mn}}{10}}$ and $\beta_{mn}^{NLOS} = PL_{mn}^{NLOS} 10^{\frac{\sigma_{sh}^{NLOS} z_{mn}}{10}}$ under the probability distributions of LoS and NLoS, respectively. Equation (2) for large-scale fading is further rewritten as a weighted sum of two components

$$\beta_{mn} = \beta_{mn}^{LOS} P_{LOS} + \beta_{mn}^{NLOS} P_{NLOS}; \tag{6}$$

large-scale fading β_{mn} changes more slowly than small-scale fading h_{mn}, which belongs to slow fading. Therefore, it is considered that in unit timeslot t, the small-scale fading of each channel is an independent and uniformly distributed variable as mentioned above, while large-scale fading can be regarded as a constant.

3. Data Transmission Process

In this section, we focus on the data transmission process and derive the closed-form expression for the transmission rate. The transmission process in the unit timeslot t includes: uplink channel estimation, downlink data transmission, and uplink data transmission. In the uplink channel estimation stage, all UAVs send pilot sequences to the AP, and the AP estimates the channel between each UAV. In the downlink data transmission stage, the downlink data are pre-coded, and transmitted power is allocated according to the uplink estimated channel; channel estimation is not carried out on the downlink alone because the downlink does not send pilot frequencies but relies on channel hardening, which makes the channel gain close to its expected value and a definite constant [6]. All APs do not share instantaneous channel state information (CSI) during data transmission, so conjugate beamforming technology is used for downlink transmission, while matching filtering technology is used for uplink reception. Note that we fully introduce the whole process of data transmission in this section, but the problem studied in this paper is the computing offloading delay of UAVs. Therefore, this paper focus on data transmission from UAVs to APs: that is, the uplink in the proposed network. We introduce downlink transmission only for formula derivation and focus on uplink data transmission optimization in the later computations.

3.1. Channel Estimation

Let τ_c be the length of the coherence interval (in symbol), which is equal to the product of the coherence time and the coherence bandwidth. Let τ_p be the duration of uplink pilot training, $\tau_p < \tau_c$, and define $\sqrt{\tau_p}\boldsymbol{\varphi}_n \in \mathbb{C}^{\tau_p \times 1}$ as the pilot sequence sent by each UAV, satisfying $\|\boldsymbol{\varphi}_n\|^2 = 1$. Where $n = 1, \ldots, N$, the pilot signal $\mathbf{y}_{pm} \in \mathbb{C}^{1 \times \tau_p}$ received at the m-th AP of the channel estimation stage can be expressed as

$$\mathbf{y}_{pm} = \sqrt{\tau_p \rho_p} \sum_{n=1}^{N} g_{mn} \boldsymbol{\varphi}_n^H + \mathbf{w}_{pm}, \tag{7}$$

where ρ_p represents the transmit power of each pilot symbol, and $\mathbf{w}_{pm} \in \mathbb{C}^{1 \times \tau_p}$ is an additive white Gaussian noise vector, the elements of which are random variables that follow the independent and identically distributed $\mathcal{CN}(0, \sigma^2)$.

Based on the received pilot signal, the ideal estimate of channel g_{mn} by the m-th AP can be expressed as a projection along the direction of the pilot vector $\boldsymbol{\varphi}_n$,

$$\check{y}_{mn} = \mathbf{y}_{pm}\boldsymbol{\varphi}_n = \sqrt{\tau_p\rho_p}g_{mn} + \sqrt{\tau_p\rho_p}\sum_{i\neq n}^{N} g_{mi}\boldsymbol{\varphi}_i^H\boldsymbol{\varphi}_n + \mathbf{w}_{pm}\boldsymbol{\varphi}_n; \tag{8}$$

the second term in Equation (8) represents the influence caused by pilot contamination. We assume the pilot to be orthogonal to itself: namely, $\|\boldsymbol{\varphi}_n\|^2 = 1$. In this paper, we also assume the influence caused by pilot contamination: namely, $\boldsymbol{\varphi}_i^H\boldsymbol{\varphi}_n \neq 0 (i \neq n)$, so the second term cannot be ignored. Equation (8) only shows the ideal estimation method, but due to the existence of noise, the ideal estimation is not accurate. MMSE can be used to calibrate the estimation results, and the MMSE estimated value of the channel $\hat{g}_{mn}$ is obtained as follows

$$\hat{g}_{mn} = \frac{\mathbb{E}\{g_{mn}\check{y}_{mn}^*\}}{\mathbb{E}\{\check{y}_{mn}\check{y}_{mn}^*\}}\check{y}_{mn} = \frac{\sqrt{\tau_p\rho_p}\beta_{mn}}{\tau_p\rho_p\sum\limits_{i=1}^{N}\beta_{mi}\left|\boldsymbol{\varphi}_i^H\boldsymbol{\varphi}_n\right|^2 + \sigma^2}\check{y}_{mn}; \tag{9}$$

each AP independently estimates the channel of each UAV, and the APs do not share the channel estimation information with each other.

3.2. Downlink Data Transmission

Since channel estimation is performed at each AP, all APs take the channel estimate $\hat{g}_{mn}$ of the uplink channel as the true value of the channel. The information of the network is encoded as symbol q_{dn} and adopts conjugate beamforming to transmit information. The signal transmitted from the m-th AP can be expressed as follows

$$x_m = \sqrt{\rho_d}\sum_{n=1}^{N}\sqrt{\eta_{mn}^{DL}}\hat{g}_{mn}^* q_{dn}, \tag{10}$$

where q_{dn} is the symbol that is sent to the n-th UAV and satisfies $\mathbb{E}\left\{\left|q_{dn}\right|^2\right\} = 1$, ρ_d indicates the downlink average transmitting power of each AP, η_{mn}^{DL} indicates the power coefficient sent from the m-th AP to the n-th UAV. For all APs, the power coefficient should meet the following constraints

$$\mathbb{E}\left\{\left|x_m\right|^2\right\} < \rho_d, \tag{11}$$

Equation (12) can be derived from Equation (11)

$$\sum_{n=1}^{N}\eta_{mn}^{DL}\mathbb{E}\left\{\left|\hat{g}_{mn}\right|^2\right\} < 1, \tag{12}$$

where

$$\gamma_{mn} \triangleq \mathbb{E}\left\{\left|\hat{g}_{mn}\right|^2\right\} = \frac{\tau_p\rho_p\beta_{mn}^2}{\tau_p\rho_p\sum\limits_{i=1}^{N}\beta_{mi}\left|\boldsymbol{\varphi}_i^H\boldsymbol{\varphi}_n\right|^2 + \sigma^2}. \tag{13}$$

After the m-th AP sends the downlink signal Equation (10) and passes through the real channel g_{mn}, the received signal at the n-th UAV can be expressed as

$$r_{dn} = \sum_{m=1}^{M} g_{mn}x_m + w_{dn} = \sqrt{\rho_d}\sum_{m=1}^{M}\sum_{i=1}^{N}\sqrt{\eta_{mi}^{DL}}g_{mn}\hat{g}_{mi}^* q_{di} + w_{dn}, \tag{14}$$

where w_{dn} is the additive Gaussian white noise obeying $\mathcal{CN}\left(0, \sigma^2\right)$ at the n-th UAV.

3.3. Uplink Data Transmission

In uplink data transmission, each UAV encodes the task data into a symbol q_{un}, satisfying $\mathbb{E}\left\{|q_{un}|^2\right\} = 1$. The symbol is then allocated a transmit amplitude value $\sqrt{\eta_n^{UL}}$ to generate a baseband signal for wireless transmissions. The signal sent by the n-th UAV can be defined as

$$\phi_n = \sqrt{\eta_n^{UL}} q_{un}, \tag{15}$$

where η_n^{UL} is the emission power coefficient of the n-th UAV and satisfies $0 < \eta_n^{UL} < 1$. The UAV transmission signal is transmitted to the m-th AP through the real channel g_{mn}, and the received signal at the m-th AP is as follows

$$y_m = \sqrt{\rho_u} \sum_{n=1}^{N} g_{mn} \phi_n + w_{um}, \tag{16}$$

where ρ_u is the average uplink transmit power of each UAV, and w_{um} is the additive white Gaussian noise at the m-th AP subject to $\mathcal{CN}\left(0, \sigma^2\right)$. Since the uplink signal transmission adopts the receiving mode of matching the filter at the m-th AP, the signal of Equation (16) is multiplied by the conjugate estimated channel at the AP to obtain the maximum output signal-to-noise ratio (SNR). Then, the result of matching the filter at each AP, $\hat{g}_{mn}^* y_m$, is transmitted to the CPU for decoding to obtain the sent signal. The CPU receives

$$r_{um} = \sum_{m=1}^{M} \hat{g}_{mn}^* y_m = \sqrt{\rho_u} \sum_{m=1}^{M} \sum_{i=1}^{N} \hat{g}_{mn}^* g_{mi} \phi_i + \sum_{m=1}^{M} \hat{g}_{mn}^* w_{um}; \tag{17}$$

when the CPU receives Equation (17), it decodes q_{un}.

In a time slot t, each AP sends downlink data to the served UAV, indicating its remaining computing capacity, and then the UAV needs to assign the task data offloaded to each AP after confirming the association information locally. The data offloading rate is decided by the uplink's achievable rate. Then, Equation (17) can be rewritten as

$$\begin{aligned} r_{um} = & \sqrt{\rho_u} \mathbb{E}\left\{ \sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} \right\} q_{un} \\ & + \sqrt{\rho_u} \left(\sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} - \mathbb{E}\left\{ \sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} \right\} \right) q_{un} \\ & + \sqrt{\rho_u} \sum_{i \neq n}^{N} \left(\sum_{m=1}^{M} \sqrt{\eta_i^{UL}} \hat{g}_{mn}^* g_{mi} \right) q_{ui} + \sum_{m=1}^{M} \hat{g}_{mn}^* w_{um}; \end{aligned} \tag{18}$$

we split and rename each item in Equation (18) as follows

$$DS_n = \sqrt{\rho_u} \mathbb{E}\left\{ \sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} \right\}, \tag{19}$$

$$BU_n = \sqrt{\rho_u} \left(\sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} - \mathbb{E}\left\{ \sum_{m=1}^{M} \sqrt{\eta_n^{UL}} \hat{g}_{mn}^* g_{mn} \right\} \right), \tag{20}$$

$$UI_{ni} = \sqrt{\rho_u} \sum_{m=1}^{M} \sqrt{\eta_i^{UL}} \hat{g}_{mn}^* g_{mi}, \tag{21}$$

$$N_n = \sum_{m=1}^{M} \hat{g}_{mn}^* w_{um}, \tag{22}$$

the above Equation (19) represents the strength of the expected signal, Equation (20) is the uncertainty of beamforming, Equation (21) is the inter-user interference caused by the i-th

UAV, and Equation (22) is the additive noise interference at the m-th AP. We can take the addition and sum of the three interference terms Equations (20)–(22) as "effective noise", so it can be deduced that the achievable uplink transmission rate of the n-th UAV in the worst case considering all interference effects can be expressed by the Shannon formula as follows

$$R_n^{UL} = \frac{\tau_u}{\tau_c} B \log_2 \left(1 + \frac{|DS_n|^2}{\mathbb{E}\left\{|BU_n|^2\right\} + \sum\limits_{i \neq n}^{N} \mathbb{E}\left\{|UI_{ni}|^2\right\} + \mathbb{E}\left\{|N_n|^2\right\}} \right), \tag{23}$$

where B represents the channel bandwidth. Then, we further derive the uplink's closed-form achievable rate under the proposed network according to Equation (23).

Theorem 1. *The achievable uplink transmission rate of the n-th UAV in the UMCF-UAV network is given by*

$$R_n^{UL} = \frac{\tau_u}{\tau_c} B \log_2 \left(1 + \frac{\rho_u \eta_n^{UL} \left(\sum\limits_{m=1}^{M} \gamma_{mn} \right)^2}{\rho_u \sum\limits_{i \neq n}^{N} \eta_i^{UL} \left(\sum\limits_{m=1}^{M} \gamma_{mn} \frac{\beta_{mi}}{\beta_{mn}} \right)^2 \left| \boldsymbol{\varphi}_i^H \boldsymbol{\varphi}_n \right|^2 + \rho_u \sum\limits_{i=1}^{N} \eta_i^{UL} \sum\limits_{m=1}^{M} \gamma_{mn} \beta_{mi} + \sigma^2 \sum\limits_{m=1}^{M} \gamma_{mn}} \right). \tag{24}$$

Proof of Theorem 1. See Appendix A. □

4. Problem Description

In this section, we focus on the proposed optimization problem. Firstly, we model the needed data and propose a computational model. Then, we analyze and propose the optimization problem. Finally, we give the complete process of the proposed DTO strategy in the UMCF-UAV network.

4.1. Computational Model

As shown in Figure 2, the whole communication cycle L is divided into T communication time slots t. The system delay considered in this paper is only for the task offloading process during uplink transmission and ignores the channel estimation and downlink transmission delay. Further, we consider that all APs are connected to the CPU by backhaul links with infinite capacity, and each AP is equipped with an edge server locally. Therefore, the backhaul link delay between AP and CPU and the transmission delay between AP and edge server are not considered in this paper. The delay of the system can be divided into three parts: offloading delay, edge computing delay, and local computing delay. In one time slot t, the local computing of the UAV is synchronized with the task transfer, and the servers perform edge computing when the task is offloaded to the servers. Each server transmits task data through its own wireless link with the UAV, and the time delay in each link is different. Finally, the system delay to be optimized is equal to the link corresponding to the server with the highest delay in the time slot t, and our goal is to gradually reduce the delay of the entire system by reducing the maximum link delay for each time slot t.

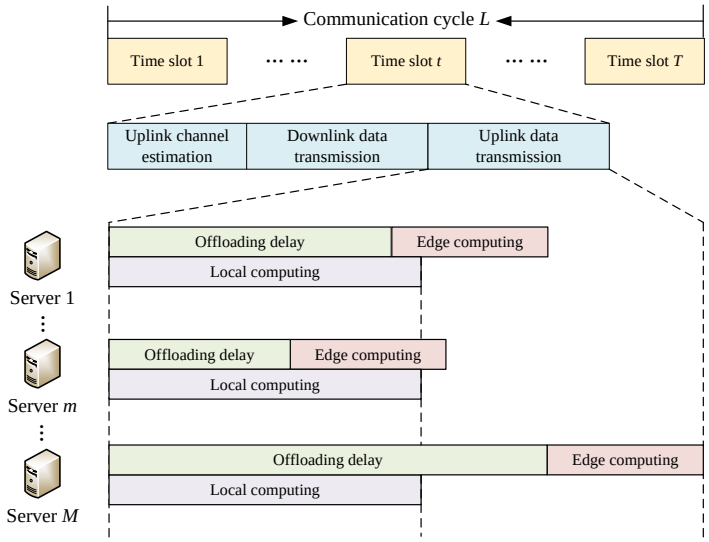

Figure 2. Time slot structure for UMCF-UAV network.

4.1.1. Offloading Delay

The offloading delay in the system is defined as the time taken to transfer the task from the n-th UAV to the m-th AP in unit time slot t, which can be expressed as follows

$$l_{offload} = \mu_{mn} \frac{c_m K_n D_n}{R_n^{UL}}, \tag{25}$$

where D_n is the sum of the GUs' tasks carried by the UAV in time slot t, K_n is the ratio of tasks that the n-th UAV decides to offload to the edge APs and satisfies $K_n \in [0,1]$, c_m is the ratio of tasks that the m-th AP gets from the n-th UAV and satisfies $c_m \in [0,1]$. The term $\mu_{mn} \forall m \in M$ is a binary variable that represents AP selection and is used to decide whether to transmit the task to be offloaded to the target AP and is defined as

$$\mu_{mn} = \begin{cases} 1, & \text{select this AP as the server and compute the offloading task,} \\ 0, & \text{this AP is not selected as the server.} \end{cases}$$

4.1.2. Edge Computing Delay

The edge computing delay in the system is defined as the time required for the edge server on the m-th AP side to compute the task offloaded from the UAV in a unit time slot t. It can be expressed as

$$l_{edge} = \frac{c_m K_n D_n \omega_{AP}}{f_m^{AP}}, \tag{26}$$

where ω_{AP} indicates the number of CPU cycles required by the AP to process each unit byte and is measured in cycles/bit, and f_m^{AP} indicates the computing capacity of the edge server on the m-th AP side in cycles/s.

4.1.3. Local Computing Delay

The local computation delay in the system is defined as the time required for the n-th UAV to compute the remaining tasks locally in unit time slot t and can be expressed as follows

$$l_{local} = \frac{(1 - K_n) D_n \omega_{UAV}}{f_n^{UAV}}, \tag{27}$$

where $(1 - K_n) D_n$ indicates the number of tasks remaining at the UAV that need to be computed locally, The term ω_{UAV} indicates the number of CPU cycles required by the

UAV to process each unit byte and is measured in cycles per bit, and f_n^{UAV} indicates the computing capacity of the n-th UAV in cycles/s.

4.2. Problem Formulation

This paper jointly optimizes AP selection and task offloading in order to reduce the system delay of a UMCF-UAV network, in which the impact of multi-dimensional resources such as computing capacity, UAV coordinates, and channel environment on task offloading are considered in the resource allocation computation. By optimizing the system delay within each time slot t, the delay of the whole communication cycle is thus reduced. Therefore, Equations (25)–(27) are rewritten as an expression related to time slot t. The optimization problem is specified as follows

$$\min_{\mu_{mn}(t),c_m(t),K_n(t),q_n^{UAV}(t)} \sum_{t=1}^{T}\sum_{m=1}^{M} \mu_{mn}(t)\max\Big\{l_{local}(t), l_{edge}(t)+l_{offload}(t)\Big\} \tag{28a}$$

$$s.t. \quad \mu_{mn}(t) \in \{0,1\}, \forall m \in \{1,\ldots,M\}, \forall t \in \{1,\ldots,T\}, \tag{28b}$$

$$0 \le \sum_{m=1}^{M} \mu_{mn}(t) \le M, \forall n \in \{1,\ldots,N\}, \tag{28c}$$

$$0 \le c_m(t) \le 1, \forall m \in \{1,\ldots,M\}, \tag{28d}$$

$$0 \le K_n(t) \le 1, \forall n \in \{1,\ldots,N\}, \tag{28e}$$

$$\sum_{m=1}^{M} c_m(t) \in \{0,1\}, \forall m \in \{1,\ldots,M\}, \tag{28f}$$

$$q_n^{UAV}(t) \in \big\{(X_n(t), Y_n(t))\big| x_n(t) \in [0, X_{upper}], y_n(t) \in [0, Y_{upper}]\big\} \tag{28g}$$

where constraints (28b) and (28c) guarantee that all UAVs can select any number of APs within slot t. Constraint (28d) represents the size of the tasks received by each AP. Constraint (28e) represents the size of task offloading by each UAV. Constraint (28f) guarantees the right that the UAV can perform local offloading. The constraint (28g) represents the coordinate range of the UAV, where X_{upper} and Y_{upper} are the upper bounds of the horizontal and vertical coordinates of the UAV. The UAV can remain unchanged for several consecutive time slots or can change its deployment position.

4.3. DTO Strategy Model

Now we express the task offloading strategy for jointly optimizing AP selection. We consider that the UAV formulates an offloading strategy at each time slot. We define the set $\mathcal{L} = \{\mathcal{L}_1, \ldots \mathcal{L}_t, \ldots \mathcal{L}_T\}$ to represent the strategy set, where $\mathcal{L}_t = \{K_n(t), c_1(t), \ldots c_m(t), \ldots c_M(t)\}$ represent the DTO strategy at time slot t. Specifically, the UAV does not perform task offloading and all tasks are completed locally when $K_n(t) = 0$. The UAV performs task offloading and the size of offloaded tasks is $K_n(t)$ when $K_n(t) \neq 0$. The UAV does not select the m-th AP when $K_n(t) \neq 0$ and $c_m(t) = 0$. The UAV selects the m-th AP and offloads the task to the m-th AP by the value of $c_m(t)$ when $K_n(t) \neq 0$ and $c_m(t) \neq 0$. The elements in $\mathcal{L}_t$ satisfy $K_n(t) \in [0,1], \forall n \in \mathcal{N}$ and $c_m(t) \in [0,1], \forall m \in \mathcal{M}$. We define the UMCF-UAV network information $\lambda = \{\lambda_u, \lambda_d\}$ and the transmission rate set $\mathcal{R} = \{R_1^{UL}(t), \ldots R_n^{UL}(t), \ldots R_N^{UL}(N)\}$, where λ_u and λ_d represent the uplink and downlink network information, respectively. The specific value of $\mathcal{L}_t$ depends on the combination of λ and $\mathcal{R}$ and satisfies the optimization Equation (28a).

As shown in Figure 3, a complete process of the proposed DTO strategy in the UMCF-UAV network can be divided into the following steps.

- Step 1. Downlink network information acquisition: GUs in the waiting area generate tasks and transfer all tasks to the UAV. Meanwhile, the UAV sends pilot signals to APs for channel estimation. The AP encodes λ_d as symbol q_{dn}, where $\lambda_d = \{f_m^{AP}, \hat{g}_{mn}, \omega_{AP}, q_m^{AP}\}$, $\forall m \in \mathcal{M}$, $\forall n \in \mathcal{N}$. The AP sends the network information λ_d to the UAV through the downlink wireless link.
- Step 2. AP selection and task offloading: The UAV obtains λ_u locally and calculates $\mathcal{R}$ based on the information, where $\lambda_u = \{f_n^{UAV}, D_n, \omega_{UAV}, q_n^{UAV}, \eta_n^{UL}\}$, $\forall m \in \mathcal{M}$, $\forall n \in \mathcal{N}$. The DTO strategy $\mathcal{L}_t$ in the current time slot t is generated locally by combining λ_d for downlink transmission. The $\mathcal{L}_t$ and task are encoded as q_{un} and sent to the selected AP in the strategy.
- Step 3. Task computing and uplink transmission: The AP transfers the received tasks to the edge server for task computing and uploads the computed available resource allocation data to the CPU to detect the symbols.
- Step 4. Update network information: The CPU receives the uplink data and updates λ. The AP sends the data of the available resource allocation result and the downlink network information $\lambda_d^{'}$ at the next time slot $t+1$ to the UAV over the wireless link. Then, the UAV sends the data of task processing results back to the GUs in the waiting area and starts a new round of task offloading.

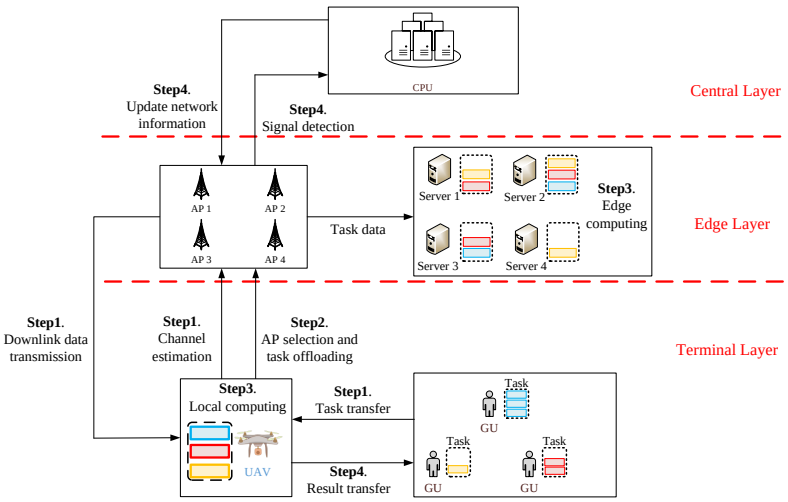

Figure 3. The process of the proposed DTO strategy in UMCF-UAV network.

This paper assumes that all UAVs have the same f_n^{UAV}. As can be seen from Equations (25)–(27), l_{local} is only determined by the DTO offload strategy $\mathcal{L}_t$. Besides the offload strategy, R_n^{UL} and f_m^{AP} are also the factors that determine $l_{offload}$ and l_{edge}. Therefore, reducing delay is not a simple choice of maximizing R_n^{UL} or f_m^{AP}. For example, the UAV may offload an uncertain size of tasks at the highest rate to the edge server on the AP side with very low computing capacity. Such task assignment does not reduce the system delay because there is no mathematical relationship between R_n^{UL} and f_m^{AP}. However, R_n^{UL}, f_m^{AP} and the offloading strategy affect the system delay together. As a result, both the numerator and the denominator of the formula in Equation (28a) affect the solution of the optimization problem.

In order to solve the non-convex optimization problem, we design a DTO offloading strategy to reduce the system delay. We consider that the optimization variables involved in the strategy are tightly coupled, which makes the optimization problem impossible to be solved with traditional linear programming. In order to implement the proposed strategy, we propose a DRL-based DRCA algorithm.

5. Algorithm Description

In order to solve the coupled and non-convex problem, this paper first introduces the theoretical framework of the DRCA algorithm; then, it defines the state space, action space, and reward function of the DRCA algorithm; and finally, it introduces the process of training the DRCA algorithm.

For the optimization problem presented in this paper, the UAV provides services to the GUs and performs task offloading with the AP. Throughout the resource allocation process, UAVs are interacting with the external environment as agents. In each time slot t, the UAV receives a state $s_t \in S$ from the external environment. For this state s_t, the UAV makes an action $a_t \in A$ by making decision π, and then, the action causes the environment to change. The environment changes to the next state s_{t+1}, and a reward $r(s_t, a_t)$ is fed back to the UAV to inform the UAV of the quality of the action selection. This process can be seen as a Markov decision process (MDP). The DRCA algorithm is developed to solve the MDP problem in the proposed algorithm. By introducing a neural network as a Q network to replace the Q-table in Q-learning, the proposed algorithm can process high-dimensional data such as pictures or videos. We define the Q value as $Q(s_t, a_t, w)$, where w represents the network weight in the Q network. In order to solve the bootstrapping phenomenon of updating the current time s_t with the next time s_{t+1} data in traditional reinforcement learning, two neural networks are introduced in DRCA to compute Q_{eval} and Q_{target}, which are called the main network and the target network, respectively. The structure of the two networks is exactly the same, and the difference is only in the weight w. The Q value update formula in DRCA is as follows

$$Q_{eval} = Q(s_t, a_t, w), \tag{29}$$

$$Q_{target} = r_t + \gamma \max_{a \in A} Q\left(s_{t+1}, a, w^{-}\right), \tag{30}$$

$$Q(s_t, a_t, w) = Q(s_t, a_t, w) + \alpha \left[r_t + \gamma \max_{a \in A} Q\left(s_{t+1}, a, w^{-}\right) - Q(s_t, a_t, w) \right], \tag{31}$$

where w^{-} represents the network weight in the target network.

The complete DRCA is shown in Figure 4. First, the agent obtains the state s_t by interacting with the environment and chooses the action a_t to execute according to the ε-based greedy policy; this affects the environment, making the state change to the next moment state s_{t+1}, and provides the reward r_t at the same time. The environment forms a four-element array of the four items of data obtained from this interaction and stores them in an experience pool. The agent then repeats this process to fill the experience pool. Secondly, the current quaternal array is input into the main network each time to predict Q_{eval}. When the accumulated experience in the experience pool reaches a certain amount, part of the array is taken out to form a batch, which is input into the target network, and Q_{target} is computed according to Equation (30). Then, the difference between Q_{eval} and Q_{target} is computed, and the network weight of the main network is updated by gradient descent. Finally, the network weights of the main network are directly assigned to the target network every once in a while so as to update the weights of the target network.

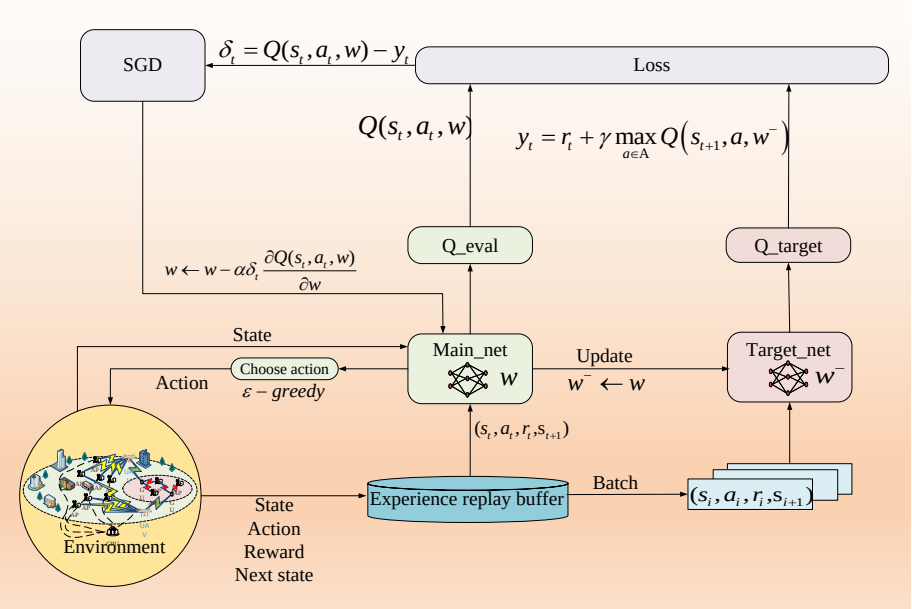

Figure 4. The proposed DRCA structure.

In the proposed DRCA algorithm, the environment, state space, action space, and reward function are defined as follows

Environment: The role of the environment in the algorithm is to interact with the agent and provide the state information that the algorithm needs. For the proposed optimization problem, we define the environment as the proposed UMCF-UAV network, and the agent is the UAV.

State Space: The state space design starts from the description of the environment. In the UMCF-UAV network proposed in this paper, the state space is defined as

$$S = \begin{pmatrix} D_1(t), \ldots D_n(t), \ldots D_N(t), \\ g_{11}(t), \ldots g_{mn}(t), \ldots g_{MN}(t), \\ \eta_1^{UL}(t), \ldots \eta_n^{UL}(t), \ldots \eta_N^{UL}(t), \\ f_1^{AP}(t), \ldots f_m^{AP}(t), \ldots f_M^{AP}(t) \end{pmatrix}, \tag{32}$$

where $D_n(t)$ represents the total size of pending tasks carried by the n-th UAV from GUs in time slot t; $g_{mn}(t)$ represents the channel gain between the m-th AP and the n-th UAV in slot t. The term $\eta_n^{UL}(t)$ represents the power coefficient of the n-th UAV at time slot t, and $f_m^{AP}(t)$ represents the remaining computing capacity of the edge server on the m-th AP side at time slot t. The composition of the state space mainly depends on the network information parameters λ in the DTO strategy. The agent acquires state information before the beginning of each time slot and updates it at the end.

Action Space: The design of the action space is related to the behavior of the agent, which is the behavior that the UAV can make in the network. In the UMCF-UAV network assisted by the UAV proposed in this paper, the action space is defined as

$$A = \begin{pmatrix} \mu_{11}(t), \ldots \mu_{mn}(t), \ldots \mu_{MN}(t), \\ c_1(t), \ldots, c_m(t), \ldots c_M(t), \\ K_1(t), \ldots K_n(t), \ldots K_N(t), \\ q_1^{UAV}(t), \ldots q_n^{UAV}(t), \ldots q_N^{UAV}(t) \end{pmatrix}, \tag{33}$$

where $\mu_{mn}(t)$ indicates whether the n-th UAV chooses the m-th AP as the AP-selected variable for offloading the target in time slot t, $K_n(t)$ represents the ratio of tasks that the

n-th UAV decides to offload to the edge APs at time slot t, $c_m(t)$ represents the ratio of tasks that the m-th AP gets from the n-th UAV at time slot t, and $q_n^{UAV}(t)$ represents the coordinates of the n-th UAV at time slot t. The action space depends on the strategy set $\mathcal{L}$ in the DTO strategy, and the agent makes a selection based on the state information at each time slot.

Reward Function: The choice of reward function depends on the goal of optimization. For the optimization problem proposed in this paper, its reward function can be defined as follows

$$Reward = \begin{cases} -1, & \text{if } l_{delay}(t) \geq slot\ t, \\ e^{-10 l_{delay}(t)}, & \text{if } l_{delay}(t) < slot\ t. \end{cases} \tag{34}$$

In the command, $l_{delay}(t) = \mu_{mn}(t)\max\left\{l_{local}(t), l_{edge}(t) + l_{offload}(t)\right\}$ indicates the network delay of the system at time t. If the delay exceeds the current time slot t, computing and offloading cannot be completed in the current time slot. This will affect the next time slot t+1, which will cause an additional data burden for task offloading at the next moment. The increase in network delay will also reduce the experience of users in the waiting area, and it is not conducive to life safety of people in emergency situations. When the time delay is less than the current time slot t, we prefer the time delay to be as low as possible so as to obtain a higher reward. Therefore, an exponential function with e as the base is designed as the reward function. It can be seen that the reward at this time is inversely proportional to the increase of the delay, with the highest reward being 1 and the lowest infinite approaching 0.

The training procedure of the proposed DRCA algorithm can be seen in Algorithm 1. Firstly, in the initialization stage, the network parameters, including channel gain, power coefficient, and AP residual computing capacity, are initialized to build the UMCF-UAV network and obtain the initial network state s_1. Then, at the beginning of each training, the current state s_t is updated to the initial state s_1. For each time slot t, the UAV relies on the greedy policy to randomly select the action, execute the selected action in the constructed UMCF-UAV network, and calculate the network delay l_{delay} in the system after the action is executed. The reward value r_t of the UMCF-UAV network feedback is computed according to Equation (34), and the current network state is changed to the next instant state s_{t+1}. The UAV encapsulates the results of this round of interaction with the environment into a transition (s_t, a_t, r_t, s_{t+1}), which is used to compute Q_{eval} in each time and also stores this information in the experience pool. The agent selects a part of the transition from which to form a batch of capacity $\mathcal{B}$ and enters the target network to compute Q_{target} when the experience pool is full. The loss function $\delta_t = Q(s_t, a_t, w) - r_t + \gamma \max_{a \in A} Q(s_{t+1}, a, w^-)$ is computed based on the difference between Q_{target} and Q_{eval}. The Stochastic gradient descent (SGD) method is adopted to update the weight of the main network in each time slot, which is as follows

$$w = w - \alpha\delta_t \frac{\partial Q(s_t, a_t, w)}{\partial w}. \tag{35}$$

The state s_{t+1} is updated at the next moment to the initial state $s_1 = s_{t+1}$ in the next round of interaction, ending the current round of interaction, and after several interactions, the UAV updates the weight of its target network through the following Equation (36)

$$w^- = w. \tag{36}$$

The next action choice of UAVs after each iteration will be closer to the high reward choice, and the UAVs can select the best AP selection and task offloading scheme after the model converges.

Algorithm 1 DRCA Algorithm

Randomly initialize all AP positions q_m^{AP}
Initialize action space **A**, number of actions $\mathcal{A}$, number of states $\mathcal{S}$
Initialize replay memory $\mathcal{D}$ with memory capacity $\mathcal{J}$, batch size $\mathcal{B}$, learning rate α, decay factor γ, epsilon ε
Initialize main network $\mathbb{Q}_{eval}$ with weight w, target network $\mathbb{Q}_{target}$ with weight w^-
for each episode **do**
    Reset simulation parameters of the UMCF-UAV system and obtain initial observation s_1
    for each time slot t **do**
        Let $s_t = s_1$
        With ε-*greedy* policy to choose action a_t
        Normalization each AP's $f_m^{AP}(t)$
        for each UAV **do**
            Choose a task offloading ratio and AP selection based on g_{mn}, $f_m^{AP}(t)$, D_n from a_t
            Compute R_n^{UL} with η_n^{UL} and q_n^{UAV} based on Equation (24)
            Compute system delay l_{delay} based on Equations (25)–(27) and r_t based on Equation (34)
            if the m-AP is selected **then**
                Reduce $f_m^{AP}(t)$ to $f_m^{AP}(t+1)$ with probability
            else
                Keep or increase $f_m^{AP}(t)$ to $f_m^{AP}(t+1)$ with probability
            end if
            Update $f_m^{AP}(t) = f_m^{AP}(t+1)$ and reset η_n^{UL},q_n^{UAV},D_n
        end for
        Observe system delay l_{delay}, reward r_t and next state s_{t+1}
        if the capacity of replay memory has reached $\mathcal{J}$ **then**
            Randomly sample a batch size $\mathcal{B}$ of transition (s_t, a_t, r_t, s_{t+1}) from replay memory $\mathcal{D}$
            Eliminate the first transition (s_t, a_t, r_t, s_{t+1}) from replay memory
        else
            Store transition (s_t, a_t, r_t, s_{t+1}) into replay memory $\mathcal{D}$
        end if
        Send batch to target network $\mathbb{Q}_{target}$, compute Q_{target} based on Equation (30)
        Compute loss function with $\delta_t = Q(s_t, a_t, w) - r_t + \gamma \max_{a \in A} Q(s_{t+1}, a, w^-)$
        Update weight w of main network $\mathbb{Q}_{eval}$ by SGD based on Equation (35)
        Update observation $s_1 = s_{t+1}$
    end for
end for

In this paper, the GUs of the waiting area transmit task data to the UAV, and the task is computed and offloaded by the UAV and the AP in the service area. According to the DRCA algorithm proposed in this paper, the UAV determines its selected APs, the size of tasks that need to be offloaded to the AP, and the size of tasks that need to be left for local computing. The goal of the proposed algorithm is to reduce the delay of the proposed UMCF-UAV network. We define a network with N UAVs and M APs where each UAV is required to perform its own randomly selected action for a single episode. The selection of the AP and task offloading does not require traversing all $2^M - 1$ possibilities, so the complexity to execute the offloading strategy in each episode is $O(N)$ and is executed only once. In Algorithm 1, the number of episodes is defined by $\mathcal{P}$ and the number of time slot iterations in each episode is defined as $\mathcal{T}$. Therefore, the algorithm complexity of DRCA can be expressed as $O(N\mathcal{P}\mathcal{T})$.

6. Numerical Results

In this section, we evaluate and discuss the simulation results of the DRCA algorithm based on DRL. First, parameters used in the simulation are given and explained in Table 1. Second, the delay performance of the proposed UMCF-UAV network is compared with the traditional UAV-assisted cellular network, and the DTO strategy adopted in the DRCA algorithm is compared with other allocation methods. Finally, DRCA simulation results are compared with classical Q-learning and traversal system method (TSM) algorithms.

Table 1. Simulation parameters.

Parameter	Value
f_c	1.9 GHz
d_0	1 m
B	20 MHz
ρ_p	100 mW
ρ_d	200 mW
ρ_u	100 mW
H	100 m
σ^2	1 dB
t	0.1 ms
ω_{AP}	800 cycles/bit
ω_{UAV}	800 cycles/bit
α	0.001
γ	0.9
ε	0.995
$\mathcal{J}$	1000
$\mathcal{B}$	64

6.1. Parameter Setting

A circular area with a radius of 1000 m is considered in the proposed system model, a total of $M = 4$ APs and $N = 2$ UAVs are configured for different comparisons, and all APs are randomly distributed in this area. The total length of the coherence interval in the system is set as $\tau_c = 200$ symbols, which includes the uplink pilot estimation length $\tau_p = 25$ symbols, the downlink data transmission length $\tau_d = 75$ symbols, and the uplink data transmission length $\tau_u = \tau_c - \tau_p - \tau_d$. We assume that an emergency situation has occurred at this time and the UAV has received the offloaded tasks from the GUs in the waiting area. Therefore, what the UAV needs to do is to select the appropriate offloading target from all APs in the network and reduce the system delay in this process. The parameters used in the rest of the simulation are listed in Table 1.

6.2. Simulation Results and Analysis

Figure 5 shows the changes in rewards of the system at different learning rates. It can be seen that the algorithm can converge at $\alpha = 0.01$ and $\alpha = 0.001$. However, the convergence speed of the algorithm with $\alpha = 0.001$ is significantly faster than that with $\alpha = 0.01$. And a higher learning rate of $\alpha = 0.1$ has not yet reached convergence. This is because the high learning rate produces a large learning step, which leads to the inability to find the optimal solution quickly. When $\alpha = 0.00001$, the algorithm fails to converge because the learning rate is too low, making the algorithm remain in the exploratory phase. Therefore, according to the simulation results, $\alpha = 0.001$ has a good convergence speed, and we use it in the following simulation.

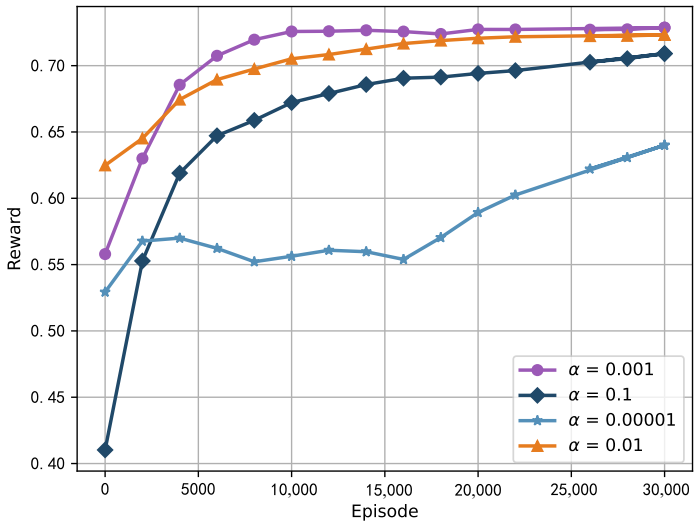

Figure 5. Comparison of DRCA algorithms under different learning rates.

As shown in Figure 6, we apply the same optimization problem and the proposed algorithm in both the traditional UAV-assisted cellular network and the UMCF-UAV network. It can be seen from the Figure 6 that the UMCF-UAV network has a significant gap in system delay compared with the traditional UAV-assisted cellular network; this is mainly reflected in the initial stage of training, where the performance of the UMCF-UAV network is already 83% higher than that of the traditional UAV-assisted cellular network. And the performance advantage of the UMCF-UAV network is maintained from the beginning of training to the end of training. However, under the compensation of the DRCA algorithm proposed in this paper, it is obvious that the traditional UAV-assisted cellular network also quickly converges to a lower level. It can be seen from Figure 6 that the UMCF-UAV network converges to 0.1 ms after 5000 episodes, while the traditional UAV-assisted cellular network converges only to 0.19 ms after 20,000 episodes, which indicates the performance advantage of the UMCF-UAV network and proves the feasibility of our proposed network.

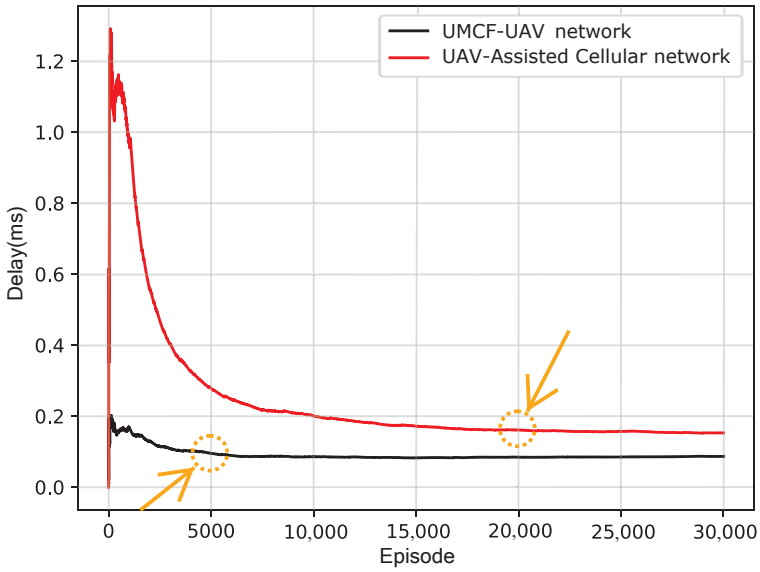

Figure 6. System delay comparison between UMCF-UAV network and UAV-assisted cellular network under DRCA algorithm.

As shown in Figure 7, in order to verify the performance of the edge server, AP selection and task offloading of 50 episodes of single training were randomly selected. The simulation results show that based on the proposed offloading strategy, the edge server added at the AP side can offload part of the tasks and reduce the computational pressure on the CPU.

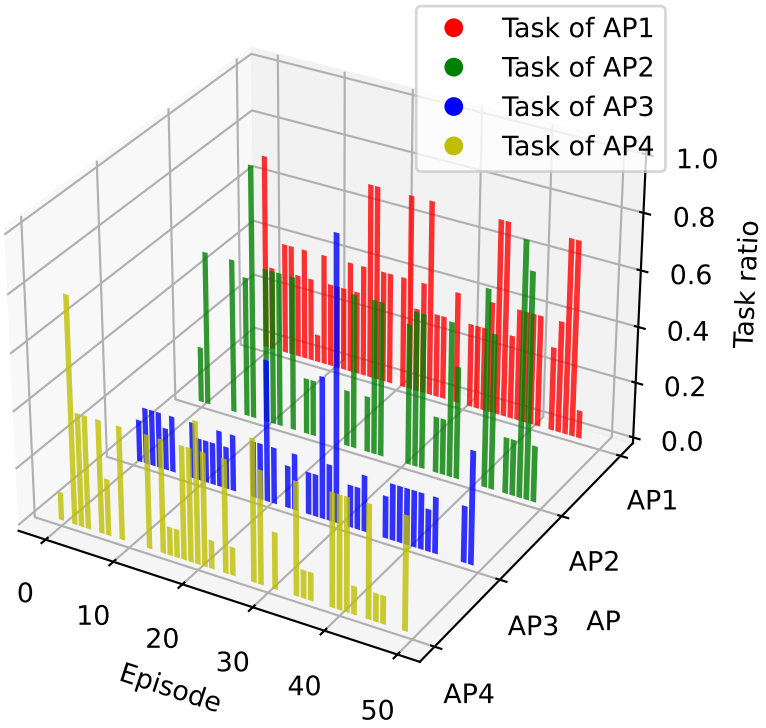

Figure 7. Task offloading of the AP side edge server in 50 random episodes.

In Figure 8, the proposed DTO scheme is compared with random allocation (RA), equal allocation (EA), and ideal local (IL) computing. Among them, RA indicates that tasks are offloaded to all APs in an area with random probability, EA indicates that tasks are equally offloaded to all APs in an area, and IL indicates that there is no edge offloading and all tasks are finished locally with strong computing power, which eliminates transmission delays. Therefore, IL is an ideal task offload assignment mode that cannot be implemented. As can be seen from Figure 8, the EA scheme is the worst case in the proposed network because the UAV equally allocates the same size of tasks to all APs with different remaining computing capacities, which may cause some APs to overload and waste the computing resources of other APs that still have computing capacity, and the overload of some APs affects resource allocation in the next moment. With the extension of time, the computing capacity of APs in the network will be polarized. As a result, the APs with the lowest computing capacity always determine the system delay of the network. However, due to the RA scheme adopting random assignment of tasks, it will also lead to the similar results of EA. In some situations, such as in the initial state, the computing capacity matches the task size; thus, the network delay under the RA scheme is better than EA in the initial training. But as the training cycles increase, the delay also starts to increase, and the gap between RA and EA schemes gradually narrows. The IL scheme is an ideal scenario: we assume that the UAV computing task is not limited by power and the flight life cycle; at this time, the UAV becomes a mobile base station, eliminating the network delay of data transmission. In fact, due to the energy consumption and the flight life cycle of UAVs, the ideal situation is difficult to achieve, so we take IL as a lower bound of the delay and as a reference scheme. As shown in Figure 8, the proposed scheme improves the delay optimization by 53% compared with the EA scheme and 47% compared with the RA scheme, and it is closest to the ideal IL scheme among all schemes.

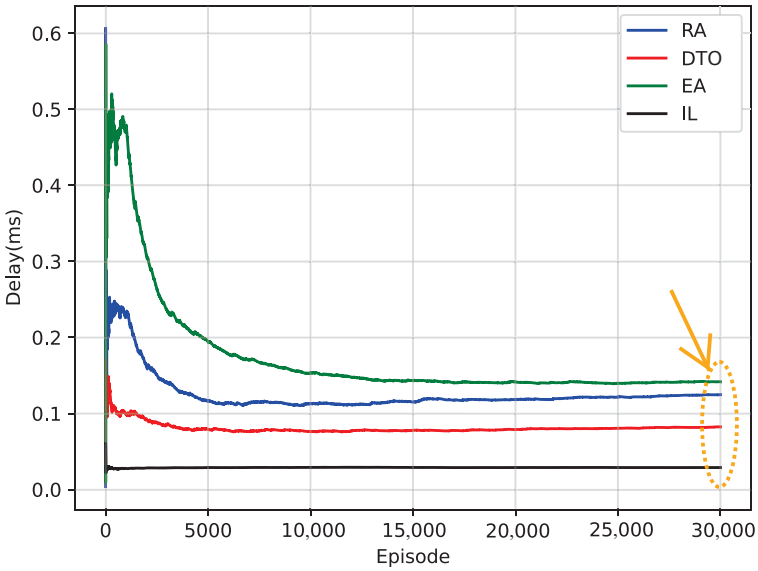

Figure 8. Comparison of system delay generated by different task offloading schemes under DRCA algorithm.

As shown in Figure 9, we compared the proposed DTO strategy with RA and EA at several training times, and the results show that the proposed DTO strategy can obtain lower system delay than RA and EA after each training, which indicates that the stability of the proposed offloading strategy is better than that of the two baseline strategies EA and RA. This is because the proposed strategy takes into account AP selection and multiple dynamic resource information within the network in each training, while the EA and RA strategies only consider the task data size and the number of APs.

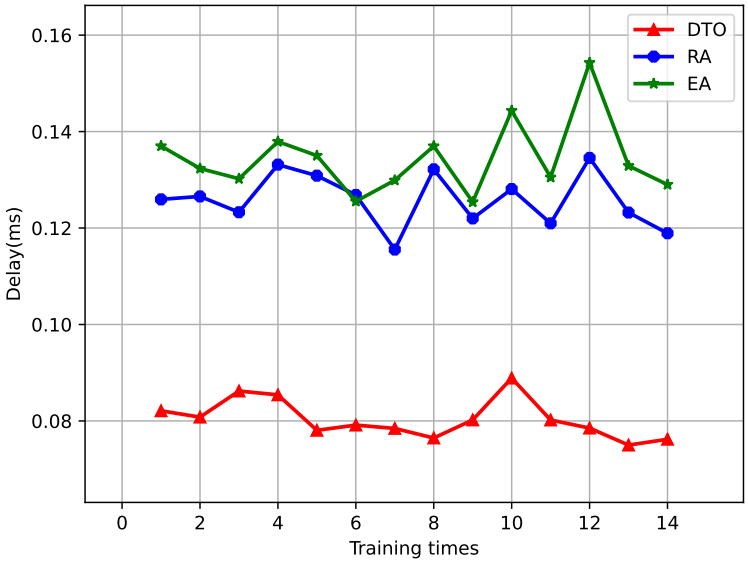

Figure 9. Comparison of the proposed DTO strategy with other RA and EA offloading strategies under multiple training.

As shown in Figure 10, in order to measure the efficiency of the DTO strategy in processing edge tasks, we compare the delay changes of the three strategies with different task sizes. It can be seen from Figure 10 that the system delay of all three strategies improves as the size of offloading tasks increases. However, compared with the RA and EA strategies, the proposed DTO strategy increases slowly and is always lower than the other two baseline

strategies under the same task size. This is because the proposed strategy jointly optimizes AP selection and improves the adaptability of the strategy to environmental changes.

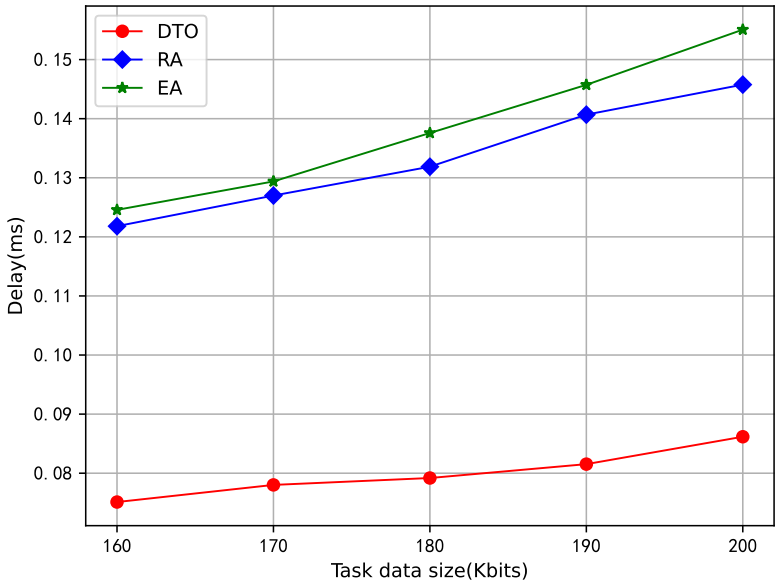

Figure 10. Comparison of the proposed DTO strategy with RA and EA offloading strategies under different task sizes.

Finally, we compare the performance of the proposed DRCA algorithm with classical Q-learning and TSM in Figure 11. It can be clearly seen from Figure 11 that the proposed scheme is superior to the traditional reinforcement learning Q-learning. The reason is that the input involved in the resource allocation of the network proposed in this paper is a high-dimensional variable. Q-learning uses a Q-table, which has more difficultly meeting the needs of high-dimensional variables. The proposed DRCA algorithm is based on the DRL algorithm, which uses neural network to deal with such problems and achieves better performance. However, when using the algorithm based on DRL, only the local optimal solution can be obtained, and the global optimal solution cannot be guaranteed. In order to further prove the performance of the proposed algorithm, we include a number of small-scale UAVs and use the TSM to verify the performance from the perspective of experiments. At each training moment, TSM traverses all possible decisions made by UAVs at that moment, computes the delay in turn, and compares and selects the minimum value. Therefore, TSM is a method to directly search for the global optimal solution. By taking the optimal TSM results as reference, the lowest delay gap between Q-learning and TSM in Figure 11 is 0.16 ms after 30,000 episodes, while the delay gap between the DRCA algorithm and TSM at the same time is only 0.015 ms, which reduces the delay by 90.63% compared with Q-learning. It can be seen that compared with Q-learning, the local optimal solution computed by the DRCA algorithm in this paper can approximate the global optimal result of TSM.

Figure 12 shows the comparison of the convergence performance of the DRCA and Q-learning algorithms. It can be obviously seen that DRCA has faster convergence speed and higher reward values than Q-learning: the reason is that the introduction of a neural network in the DRCA algorithm shortens the time required for convergence compared with Q-learning's table lookup.

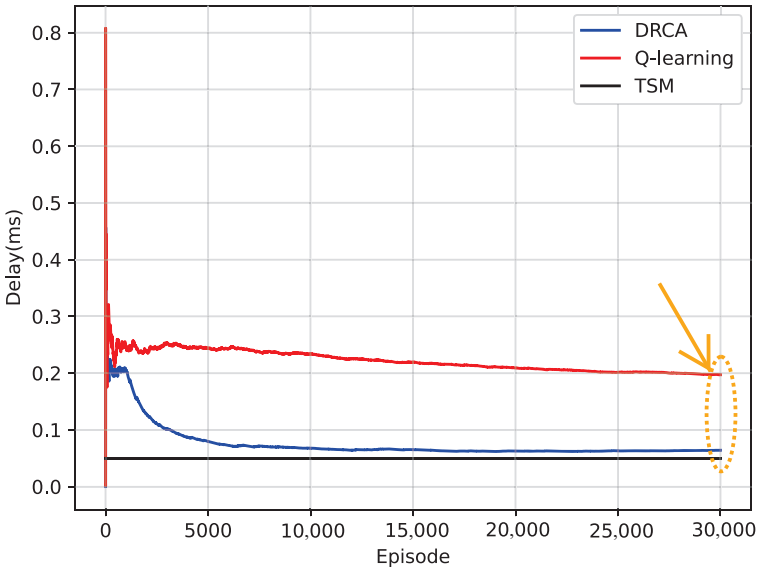

Figure 11. System delay comparison between DRCA algorithm, Q-learning, and TSM.

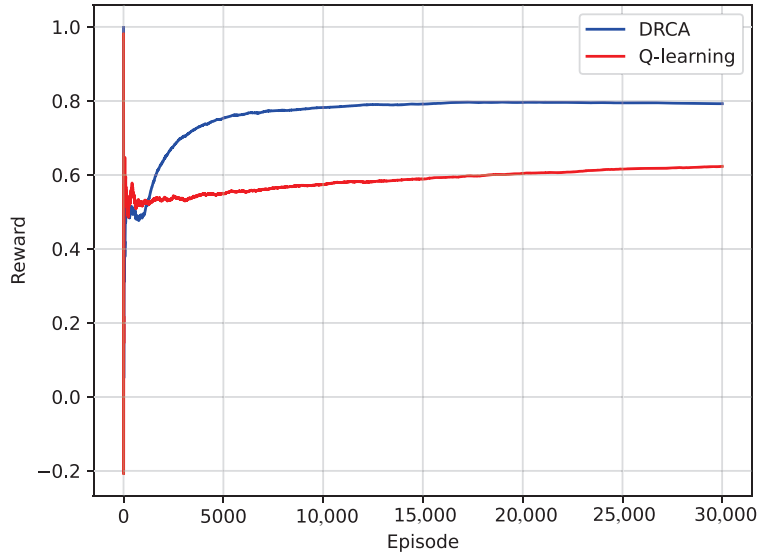

Figure 12. Comparison of convergence of DRCA algorithm to Q-learning.

As shown in Figure 13, in order to further explain the performance advantages of the proposed DRCA algorithm compared with Q-learning, we compare the delay variation of the two algorithms with different numbers of UAVs and APs. It can be seen that as the number of APs in the network increases, the performance of both algorithms is improved, and the delay is further reduced. This is because the increase in APs brings more computing resources to the network, which reduces the computational pressure on the edge servers at each AP side. Meanwhile, the time delay of the two algorithms gradually improves with the increasing number of UAVs. This is because UAVs can be regarded as aerial users in the system, and the increasing number of UAVs brings more computational burden to the edge. As can be seen from Figure 13, the DRCA algorithm is always superior to the Q-learning algorithm and obtains lower system delay when the number of APs and UAVs is fixed.

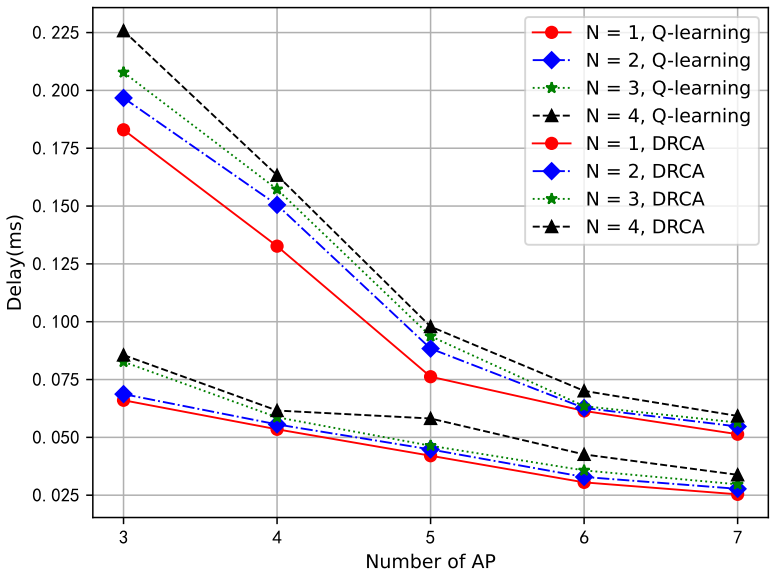

Figure 13. Comparison of the DRCA algorithm and Q-learning with different numbers of APs and UAVs.

7. Conclusions

Aiming at the emergency scenario of network interruption in a UMi environment, a UMCF-UAV network architecture is proposed in this paper. In the UMCF-UAV network, edge servers are introduced to reduce CPU computing pressure and system delay, and a dynamic resource allocation scheme of AP selection and task offloading is proposed to minimize the transmission delay of the system. In order to solve the proposed non-convex problem with tightly coupled optimization variables, we converted the problem into an MDP problem and proposed a dynamic resource cooperative allocation algorithm. The proposed algorithm has been compared with traditional Q-learning: the comparison results show that the proposed algorithm has lower delay and faster convergence than the Q-learning algorithm. Further, the resource allocation scheme proposed in this paper is compared with different resource allocation schemes such as RA, EA, and IL, and the results show that the proposed algorithm has the best performance. In order to further verify the performance of the algorithm, a small number of UAVs are used to compare with the optimal performance obtained based on the TSM algorithm in this paper. Simulation results show that the proposed algorithm is close to the optimal bounds of TSM.

Author Contributions: Conceptualization, C.P.; methodology, C.P. and J.W.; software, J.W.; validation, J.W., C.P. and Z.Y.; formal analysis, X.Y.; investigation, J.W.; resources, C.P. and L.G.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, J.W., C.P. and Z.Y.; visualization, X.Y.; supervision, C.P., X.Y. and L.G.; project administration, X.Y. and L.G.; funding acquisition, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Natural Science Foundation Haidian Original Innovation Joint Fund (No. L212026), R&D Program of Beijing Municipal Education Commission (KM202211232011).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Theorem 1

In order to derive the closed-form expression of the uplink rate, we need to compute $|DS_n|^2$, $\mathbb{E}\left\{|BU_n|^2\right\}$, $\mathbb{E}\left\{|UI_{ni}|^2\right\}$, and $\mathbb{E}\left\{|N_n|^2\right\}$.

Appendix A.1. Compute $|DS_n|^2$

The error of the channel estimate is defined as $\varepsilon_{mn} \triangleq g_{mn} - \hat{g}_{mn}$, which is the difference between the actual channel gain and the estimated channel value. Substituting the error ε_{mn} into Equation (19) yields

$$\begin{aligned} DS_n &= \sqrt{\rho_u}\mathbb{E}\left\{\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\hat{g}_{mn}^*(\varepsilon_{mn}+\hat{g}_{mn})\right\} \\ &= \sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\mathbb{E}\{\hat{g}_{mn}^*\varepsilon_{mn}\} + \sqrt{\eta_n^{UL}}\mathbb{E}\{\hat{g}_{mn}^*\hat{g}_{mn}\}. \end{aligned} \tag{A1}$$

According to the nature of MMSE estimation, we know that the error cannot be reduced by improving the estimation when the best estimate point is reached. Therefore, ε_{mn} and $\hat{g}_{mn}$ are independent of each other, and $\mathbb{E}\{\hat{g}_{mn}^*\varepsilon_{mn}\} = 0$, so Equation (19) can be further simplified by

$$DS_n = \sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\mathbb{E}\{\hat{g}_{mn}^*\hat{g}_{mn}\} = \sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\mathbb{E}\left\{|\hat{g}_{mn}|^2\right\} = \sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\gamma_{mn}, \tag{A2}$$

$$|DS_n|^2 = \rho_u\eta_n^{UL}\left(\sum_{m=1}^{M}\gamma_{mn}\right)^2. \tag{A3}$$

Appendix A.2. Compute $\mathbb{E}\left\{|BU_n|^2\right\}$

$$\begin{aligned} \mathbb{E}\left\{|BU_n|^2\right\} &= \mathbb{E}\left\{\left|\sqrt{\rho_u}\left(\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\hat{g}_{mn}^*g_{mn} - \mathbb{E}\left\{\sum_{m=1}^{M}\sqrt{\eta_n^{UL}}\hat{g}_{mn}^*g_{mn}\right\}\right)\right|^2\right\} \\ &\overset{(a)}{\Rightarrow} \rho_u\sum_{m=1}^{M}\eta_n^{UL}\left(\mathbb{E}\left\{|\hat{g}_{mn}^*g_{mn}|^2\right\} - |\mathbb{E}\{\hat{g}_{mn}^*g_{mn}\}|^2\right) \\ &= \rho_u\sum_{m=1}^{M}\eta_n^{UL}\left(\mathbb{E}\left\{\left|\hat{g}_{mn}^*\varepsilon_{mn} + |\hat{g}_{mn}|^2\right|^2\right\} - \left|\mathbb{E}\left\{\hat{g}_{mn}^*\varepsilon_{mn} + |\hat{g}_{mn}|^2\right\}\right|^2\right) \\ &= \rho_u\sum_{m=1}^{M}\eta_n^{UL}\left(\mathbb{E}\left\{|\hat{g}_{mn}^*\varepsilon_{mn}|^2\right\} + \mathbb{E}\left\{|\hat{g}_{mn}|^4\right\} - \gamma_{mn}^2\right) \\ &\overset{(b)}{\Rightarrow} \rho_u\sum_{m=1}^{M}\eta_n^{UL}\left(\gamma_{mn}(\beta_{mn}-\gamma_{mn}) + 2\gamma_{mn}^2 - \gamma_{mn}^2\right) \\ &= \rho_u\eta_n^{UL}\sum_{m=1}^{M}\gamma_{mn}\beta_{mn}, \end{aligned} \tag{A4}$$

where (a) is derived from $\mathbb{E}\left\{(X-\mathbb{E}X)^2\right\} = \mathbb{E}\left\{X^2 - 2X*\mathbb{E}X + (\mathbb{E}X)^2\right\} = \mathbb{E}X^2 - 2\mathbb{E}X*\mathbb{E}X + \mathbb{E}\left\{(\mathbb{E}X)^2\right\} = \mathbb{E}X^2 - (\mathbb{E}X)^2$, and (b) is based on $\mathbb{E}\left\{|\hat{g}_{mn}|^4\right\} = 2\gamma_{mn}^2$ and $\mathbb{E}\left\{|\varepsilon_{mn}|^2\right\} = \beta_{mn} - \gamma_{mn}$.

Appendix A.3. Compute $\mathbb{E}\left\{|UI_{ni}|^2\right\}$

$$\mathbb{E}\left\{|UI_{ni}|^2\right\} = \mathbb{E}\left\{\left|\sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_i^{UL}}\hat{g}_{mn}^*g_{mi}\right|^2\right\}. \tag{A5}$$

We simplify Equation (9) as follows

$$\hat{g}_{mn} = \frac{\sqrt{\tau_p \rho_p}\beta_{mn}}{\tau_p \rho_p \sum_{i=1}^{N} \beta_{mi} \left|\boldsymbol{\varphi}_i^H \boldsymbol{\varphi}_n\right|^2 + \sigma^2} \check{y}_{mn} = c_{mn} \check{y}_{mn}; \tag{A6}$$

substituting Equations (8) and (A6) into Equation (A5), we have

$$\begin{aligned}
\mathbb{E}\left\{|UI_{ni}|^2\right\} &= \mathbb{E}\left\{\left|\sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_i^{UL}} c_{mn} \check{y}_{mn}^* g_{mi}\right|^2\right\} \\
&= \mathbb{E}\left\{\left|\sqrt{\rho_u}\sum_{m=1}^{M}\sqrt{\eta_i^{UL}} c_{mn}\left(\sqrt{\tau_p \rho_p}\sum_{n'=1}^{N} g_{mn'} \boldsymbol{\varphi}_{n'}^H \boldsymbol{\varphi}_n + \mathbf{w}_{pm}\boldsymbol{\varphi}_n\right)^* g_{mi}\right|^2\right\} \\
&\overset{(c)}{\Rightarrow} \rho_u \tau_p \rho_p \eta_i^{UL} \mathbb{E}\left\{\left|\sum_{m=1}^{M} g_{mi}\sum_{n'=1}^{N} c_{mn} g_{mn'}^* \boldsymbol{\varphi}_{n'}^T \boldsymbol{\varphi}_n^*\right|^2\right\} + \rho_u \eta_i^{UL}\mathbb{E}\left\{\left|\sum_{m=1}^{M} c_{mn} g_{mi} \mathbf{w}_{pm}^* \boldsymbol{\varphi}_n^*\right|^2\right\} \\
&\overset{(d)}{\Rightarrow} \rho_u \tau_p \rho_p (\xi_1 + \xi_2) + \rho_u \eta_i^{UL} \sum_{m=1}^{M} c_{mn}^2 \beta_{mi}.
\end{aligned} \tag{A7}$$

In the above derivation (c), based on the derivation of the mathematical expectation formula, when X and Y are independent random variables, $\mathbb{E}\left\{|X+Y|^2\right\} = \mathbb{E}\{|X|^2\} + \mathbb{E}\{|Y|^2\}$, and (d) relies on us to define the channel gain $g_{mi} = \sqrt{\beta_{mi}} h_{mi}$ and $\widetilde{\mathbf{w}}_{pm} = \mathbf{w}_{pm}^* \boldsymbol{\varphi}_n^* \sim \mathcal{CN}(0, \sigma^2)$, where ξ_1 and ξ_2 are given by

$$\xi_1 \triangleq \eta_i^{UL}\mathbb{E}\left\{\left|\sum_{m=1}^{M} g_{mi} c_{mn} g_{mi}^* \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2\right\}, \tag{A8}$$

$$\xi_2 \triangleq \eta_i^{UL}\mathbb{E}\left\{\left|\sum_{m=1}^{M} g_{mi}\sum_{n'\neq i}^{N} c_{mn} g_{mn'}^* \boldsymbol{\varphi}_{n'}^T \boldsymbol{\varphi}_n^*\right|^2\right\}. \tag{A9}$$

Compute ξ_1 first,

$$\begin{aligned}
\xi_1 &\triangleq \eta_i^{UL}\mathbb{E}\left\{\left|\sum_{m=1}^{M} g_{mi} c_{mn} g_{mi}^* \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2\right\} = \eta_i^{UL}\mathbb{E}\left\{\left|\sum_{m=1}^{M} c_{mn}|g_{mi}|^2 \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2\right\} \\
&= \eta_i^{UL}\left|\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2 \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{m'=1}^{M} c_{mn} c_{m'n} |g_{mi}|^2 |g_{m'i}|^2\right\} \\
&= \eta_i^{UL}\left|\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2 \mathbb{E}\left\{\sum_{m=1}^{M} c_{mn}^2 |g_{mi}|^4\right\} + \eta_i^{UL}\left|\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2 \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{m'\neq m}^{M} c_{mn} c_{m'n} |g_{mi}|^2 |g_{m'i}|^2\right\} \\
&= 2\eta_i^{UL}\left|\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2 \mathbb{E}\left\{\sum_{m=1}^{M} c_{mn}^2 \beta_{mi}^2\right\} + \eta_i^{UL}\left|\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_n^*\right|^2 \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{m'\neq m}^{M} c_{mn} c_{m'n} \beta_{mi} \beta_{m'i}\right\}.
\end{aligned} \tag{A10}$$

In the same way, the complete square expansion of ξ_2 can be obtained

$$\xi_2 = \eta_i^{UL}\sum_{m=1}^{M}\sum_{n'\neq i}^{N} c_{mn}^2 \beta_{mi} \beta_{mn'} \left|\boldsymbol{\varphi}_{n'}^T \boldsymbol{\varphi}_n^*\right|^2; \tag{A11}$$

substituting Equations (A10) and (A11) into Equation (A7), we have

$$\mathbb{E}\left\{|UI_{ni}|^2\right\} = \rho_u \eta_i^{UL} \left(\sum_{m=1}^{M} \gamma_{mn} \frac{\beta_{mi}}{\beta_{mn}} \right)^2 \left| \boldsymbol{\varphi}_i^H \boldsymbol{\varphi}_n \right|^2 + \rho_u \eta_i^{UL} \sum_{m=1}^{M} \gamma_{mn} \beta_{mi}. \tag{A12}$$

Appendix A.4. Compute $\mathbb{E}\left\{|N_n|^2\right\}$

$$\mathbb{E}\left\{|N_n|^2\right\} = \mathbb{E}\left\{ \left| \sum_{m=1}^{M} \hat{g}_{mn}^* w_{um} \right|^2 \right\} = \sum_{m=1}^{M} \mathbb{E}\left\{ |\hat{g}_{mn}^* w_{um}|^2 \right\} = \sigma^2 \sum_{m=1}^{M} \gamma_{mn}. \tag{A13}$$

Substituting Equations (A3), (A4), (A12) and (A13) into Equation (23), we have Equation (24).

References

1. Chen, C.; Zhang, T.; Xu, W.; Yang, X.; Wang, Y. Multi-UAV Cooperation Based Edge Computing Offloading in Emergency Communication Networks. In Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6. [CrossRef]
2. Yu, H.; Liu, Y.; Han, M.; Zhao, X.; Fu, M.; Wu, Z.; Li, D. Latency Optimization for UAVs Based Emergency Communication. In Proceedings of the 2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA), Ningbo, China, 18–22 August 2023; pp. 1280–1285.
3. Yang, J.; Kim, J.; Kang, S.; Ok, H.; Yoo, Y.; Koh, H.; Kim, P.; Head, A.L.; Smith, A. Establishing Wireless Intranet Network Using UAVs and Web Application for Emergency Communications. In Proceedings of the 2023 IEEE Sensors Applications Symposium (SAS), Ottawa, ON, Canada, 18–20 July 2023; pp. 1–6. [CrossRef]
4. Abuzgaia, N.; Younis, A.; Mesleh, R. UAV Communications in 6G Cell-Free Massive MIMO Systems. In Proceedings of the 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Benghazi, Libya, 21–23 May 2023; pp. 634–639. [CrossRef]
5. Zong, J.; Liu, Y.; Liu, H.; Wang, Q.; Chen, P. 6G Cell-Free Network Architecture. In Proceedings of the 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 27–29 May 2022; pp. 421–425. [CrossRef]
6. Ngo, H.Q.; Ashikhmin, A.; Yang, H.; Larsson, E.G.; Marzetta, T.L. Cell-Free Massive MIMO Versus Small Cells. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1834–1850. [CrossRef]
7. Li, S.; Tan, F.; Liu, Q. Mobile Edge Computing Tasking Offloading Strategy in Cell-Free Massive MIMO with Graph Neural Network. In Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 14–16 June 2023; pp. 1–3. [CrossRef]
8. Ozpolat, M.; Al-Rubaye, S.; Williamson, A.; Tsourdos, A. Integration of unmanned aerial vehicles and LTE: A scenario-dependent analysis. In Proceedings of the IEEE 2022 International Conference on Connected Vehicle and Expo (ICCVE), Lakeland, FL, USA, 7–9 March 2022; pp. 1–6.
9. Ammar, H.A.; Adve, R.; Shahbazpanahi, S.; Boudreau, G.; Srinivas, K.V. Distributed Resource Allocation Optimization for User-Centric Cell-Free MIMO Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 3099–3115. [CrossRef]
10. Wu, S.; Liu, L.; Zhang, W.; Meng, W.; Ye, Q.; Ma, Y. Revenue-Maximizing Resource Allocation for Multitenant Cell-Free Massive MIMO Networks. *IEEE Syst. J.* **2022**, *16*, 3410–3421. [CrossRef]
11. Lu, X.; Yu, X.; Wang, C.; Wang, W.; Zhen, J. Collaborative Task Offloading Based on Scalable DAG in Cell-Free HetMEC Networks. In Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 14–16 June 2023; pp. 1–6. [CrossRef]
12. Cao, Y.; Yu, Q.Y. Joint Resource Allocation for User-Centric Cell-Free Integrated Sensing and Communication Systems. *IEEE Commun. Lett.* **2023**, *27*, 2338–2342. [CrossRef]
13. Zhang, J.; Zhang, J.; Ng, D.W.K.; Ai, B. Federated Learning-Based Cell-Free Massive MIMO System for Privacy-Preserving. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 4449–4460. [CrossRef]
14. Freitas, M.; Souza, D.; Borges, G.; Cavalcante, A.M.; da Costa, D.B.; Marquezini, M.; Almeida, I.; Rodrigues, R.; Costa, J.C.W.A. Matched-Decision AP Selection for User-Centric Cell-Free Massive MIMO Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6375–6391. [CrossRef]
15. Ghiasi, N.; Mashhadi, S.; Farahmand, S.; Razavizadeh, S.M.; Lee, I. Energy Efficient AP Selection for Cell-Free Massive MIMO Systems: Deep Reinforcement Learning Approach. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 29–41. [CrossRef]

16. Yang, Z.; Xu, W.; Shikh-Bahaei, M. Energy efficient UAV communication with energy harvesting. *IEEE Trans. Veh. Technol.* **2019**, *69*, 1913–1927. [CrossRef]
17. Chen, M.; Saad, W.; Yin, C. Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 1504–1517. [CrossRef]
18. Yao, Y.; Lv, K.; Huang, S.; Li, X.; Xiang, W. UAV trajectory and energy efficiency optimization in RIS-assisted multi-user air-to-ground communications networks. *Drones* **2023**, *7*, 272. [CrossRef]
19. Wu, Q.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [CrossRef]
20. Li, X.; Fang, Y.; Pan, C.; Cai, Y.; Zhou, M. Resource Scheduling for UAV-Assisted Failure-Prone MEC in Industrial Internet. *Drones* **2023**, *7*, 259. [CrossRef]
21. Tentu, V.; Sharma, E.; Amudala, D.N.; Budhiraja, R. UAV-Enabled Hardware-Impaired Spatially Correlated Cell-Free Massive MIMO Systems: Analysis and Energy Efficiency Optimization. *IEEE Trans. Commun.* **2022**, *70*, 2722–2741. [CrossRef]
22. Shi, K.; Dai, Z.; Zeng, Y. Dynamic AP Association for UAV-Centric Aerial-Ground Communication with Cell-Free Massive MIMO. In Proceedings of the 2023 International Conference on Communications, Computing and Artificial Intelligence (CCCAI), Shanghai, China, 23–25 June 2023; pp. 139–144. [CrossRef]
23. Wang, L.; Zhang, Q. Cell-Free Massive MIMO with UAV Access Points: UAV Location Optimization. In Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC), Foshan, China, 11–13 August 2022; pp. 262–267. [CrossRef]
24. Zheng, J.; Zhang, J.; Ai, B. UAV communications with WPT-aided cell-free massive MIMO systems. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3114–3128. [CrossRef]
25. Diaz-Vilor, C.; Lozano, A.; Jafarkhani, H. Cell-Free UAV Networks: Asymptotic Analysis and Deployment Optimization. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 3055–3070. [CrossRef]
26. Vu, T.T.; Ngo, D.T.; Ngo, H.Q.; Dao, M.N.; Tran, N.H.; Middleton, R.H. Joint resource allocation to minimize execution time of federated learning in cell-free massive MIMO. *IEEE Internet Things J.* **2022**, *9*, 21736–21750. [CrossRef]
27. Zhang, C.; Luo, X.; Liang, J.; Liu, X.; Zhu, L.; Guo, S. POTA: Privacy-Preserving Online Multi-Task Assignment with Path Planning. *IEEE Trans. Mob. Comput.* **2023**, *early access*. [CrossRef]
28. Xu, W.; Yang, Z.; Ng, D.W.K.; Levorato, M.; Eldar, Y.C.; Debbah, M. Edge Learning for B5G Networks with Distributed Signal Processing: Semantic Communication, Edge Computing, and Wireless Sensing. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 9–39. [CrossRef]
29. Tilahun, F.D.; Abebe, A.T.; Kang, C.G. Multi-Agent Reinforcement Learning for Distributed Resource Allocation in Cell-Free Massive MIMO-enabled Mobile Edge Computing Network. *IEEE Trans. Veh. Technol.* **2023**, 1–15. [CrossRef]
30. Femenias, G.; Riera-Palou, F. Mobile Edge Computing Aided Cell-Free Massive MIMO Networks. *IEEE Trans. Mob. Comput.* **2022**, 1–16. [CrossRef]
31. Song, I.; Tam, P.; Kang, S.; Ros, S.; Kim, S. DRL-Based Backbone SDN Control Methods in UAV-Assisted Networks for Computational Resource Efficiency. *Electronics* **2023**, *12*, 2984. [CrossRef]
32. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4245–4257. [CrossRef]
33. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
34. Wang, Y.; Chen, M.; Yang, Z.; Luo, T.; Saad, W. Deep Learning for Optimal Deployment of UAVs with Visible Light Communications. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7049–7063. [CrossRef]
35. Wu, S. Resource Allocation Based on Reinforcement Learning for Heterogeneous Air Network. In Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 14–16 June 2023; pp. 1–5. [CrossRef]
36. Zhan, C.; Zeng, Y. Energy Minimization for Cellular-Connected UAV: From Optimization to Deep Reinforcement Learning. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 5541–5555. [CrossRef]
37. Wang, L.; Wang, K.; Pan, C.; Xu, W.; Aslam, N.; Nallanathan, A. Deep Reinforcement Learning Based Dynamic Trajectory Control for UAV-Assisted Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 3536–3550. [CrossRef]
38. Qin, P.; Wang, S.; Lu, Z.; Xie, Y.; Zhao, X. Deep Reinforcement Learning-Based Energy Minimization Task Offloading and Resource Allocation for Air Ground Integrated Heterogeneous Networks. *IEEE Syst. J.* **2023**, *17*, 4958–4968. [CrossRef]
39. Luo, Q.; Luan, T.H.; Shi, W.; Fan, P. Deep Reinforcement Learning Based Computation Offloading and Trajectory Planning for Multi-UAV Cooperative Target Search. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 504–520. [CrossRef]
40. Xu, F.; Ruan, Y.; Li, Y. Soft Actor-Critic Based 3-D Deployment and Power Allocation in Cell-Free Unmanned Aerial Vehicle Networks. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 1692–1696. [CrossRef]
41. Hu, F.; Deng, Y.; Aghvami, A.H. Cooperative Multigroup Broadcast 360° Video Delivery Network: A Hierarchical Federated Deep Reinforcement Learning Approach. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 4009–4024. [CrossRef]

42. Shen, C.; Chang, T.H.; Gong, J.; Zeng, Y.; Zhang, R. Multi-UAV Interference Coordination via Joint Trajectory and Power Control. *IEEE Trans. Signal Process.* **2020**, *68*, 843–858. [CrossRef]
43. Zhang, J.; Zeng, Y.; Zhang, R. Multi-Antenna UAV Data Harvesting: Joint Trajectory and Communication Optimization. *J. Commun. Inf. Netw.* **2020**, *5*, 86–99. [CrossRef]