

## Article

# UAV-Assisted Mobile Edge Computing: Dynamic Trajectory Design and Resource Allocation

Zhuwei Wang <sup>1</sup>, Wenjing Zhao <sup>1</sup>, Pengyu Hu <sup>2,\*</sup>, Xige Zhang <sup>3</sup>, Lihan Liu <sup>4</sup>, Chao Fang <sup>1,3</sup> and Yanhua Sun <sup>1</sup>

<sup>1</sup> Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; wangzhuwei@bjut.edu.cn (Z.W.); 836118497@emails.bjut.edu.cn (W.Z.); fangchao@bjut.edu.cn (C.F.); sunyanhua@bjut.edu.cn (Y.S.)

<sup>2</sup> Department of Smart Agriculture Engineering, Shanghai Vocational College of Agriculture and Forestry, Shanghai 201699, China

<sup>3</sup> Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China; zhangxg01casc@163.com

<sup>4</sup> School of Statistics and Data Science, Beijing Wuzi University, Beijing 101149, China; liulihan@bwu.edu.cn

\* Correspondence: hupy@shafc.edu.cn; Tel.: +86-185-0214-5050

**Abstract:** The recent advancements of mobile edge computing (MEC) technologies and unmanned aerial vehicles (UAVs) have provided resilient and flexible computation services for ground users beyond the coverage of terrestrial service. In this paper, we focus on a UAV-assisted MEC system in which the UAV equipped with MEC servers is used to assist user devices in computing their tasks. To minimize the weighted average energy consumption and delay in the UAV-assisted MEC system, a LQR-Lagrange-based DDPG (LLDDPG) algorithm, which jointly optimizes the user task offloading and the UAV trajectory design, is proposed. To be specific, the LLDDPG algorithm consists of three subproblems. The DDPG algorithm is used to address the issue of UAV desired trajectory planning, and subsequently, the LQR-based algorithm is employed to achieve the real-time tracking control of UAV desired trajectory. Finally, the Lagrange duality method is proposed to solve the optimization problem of computational resource allocation. Simulation results indicate that the proposed LLDDPG algorithm can effectively improve the system resource management and realize the real-time UAV trajectory design.

**Keywords:** UAV; MEC; trajectory design; task offloading; resource allocation



**Citation:** Wang, Z.; Zhao, W.; Hu, P.; Zhang, X.; Liu, L.; Fang, C.; Sun, Y. UAV-Assisted Mobile Edge Computing: Dynamic Trajectory Design and Resource Allocation. *Sensors* **2024**, *24*, 3948. <https://doi.org/10.3390/s24123948>

Academic Editor: Sergio Toral Marin

Received: 5 May 2024

Revised: 8 June 2024

Accepted: 11 June 2024

Published: 18 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The rapid development of mobile intelligent devices is boosting the growth of the Internet of Things (IoT) and the advent of complex mobile applications with intelligent features, such as face recognition, video processing, and online games [1]. These applications are typically latency- and computation-sensitive. However, IoT devices, due to their relatively low computing and battery capabilities, are unable to maintain superior performance [2]. Although cloud computing can offload terminal computing tasks to cloud servers, thereby alleviating the computational burden on mobile devices, the task offloading will cause excessive latency and link congestion problems [3,4].

Mobile edge computing (MEC) provides a cost-efficient solution for computationally intensive and latency-critical tasks, by allocating computational resources towards the network edge to users [5]. The edge execution of user tasks extends the battery life of devices, reduces the power consumption and latency associated with communication and local computing, and improves the quality of service [6,7]. However, in traditional MEC application scenarios, the communication links are dominated by non-line-of-sight (NLoS), which indicates that the data transmission rate is severely restricted by the poor quality of the communication channel [8]. In addition, it poses a significant challenge in deploying the terrestrial MEC unit in certain situations, such as in remote areas or during emergency events [9].

Fortunately, the technology of unmanned aerial vehicles (UAVs), characterized by flexible mobility, easy deployment, and line-of-sight (LoS) connections, has gradually become an important component of future wireless networks. The UAV-assisted MEC system provides a potential solution to address the aforementioned challenges in terrestrial MEC systems [10,11]. Compared to traditional wireless networks, UAV-assisted MEC networks offer a multitude of advantages in terms of mobility, flexibility, cost, coverage, and reconfiguration. Moreover, UAVs equipped with MEC servers can approach users closely to provide services, which can notably reduce energy consumption and transmission delay.

However, designing a joint optimal scheme for resource management and UAV trajectory planning faces significant challenges due to the UAV's inherent dynamics constraints and limited onboard computation capability and energy resources [12–14]. On the one hand, the actual UAV flight acceleration and velocity cannot be adjusted arbitrarily, so sudden acceleration, deceleration, and turning are impossible. However, they are often completely overlooked in existing UAV trajectory planning, resulting in significant deviations between the actual flight trajectory and the theoretically designed trajectory of the UAV [15]. On the other hand, UAV trajectory planning requires achieving coverage for all users and satisfying the task offloading requirements. However, both the users' task offloading ratio and the communication channel between the user and the UAV are time-varying, which causes performance degradation on offloading efficiency, latency, and energy efficiency. Therefore, given the high dynamic scenarios and the frequent task offloading requirements of users, the resource management of MEC systems and the design of UAV flight trajectories have become crucial research topics.

Motivated by the above-mentioned reasons, this paper focuses on the UAV-assisted MEC system considering UAV flight dynamics constraints. A novel linear quadratic regulator (LQR)-Lagrange-based deep deterministic policy gradient (LLDDPG) algorithm is proposed to minimize the weighted energy consumption and delays of the system through the joint optimization of dynamic computation resources and UAV flight trajectory. In fact, in light of the UAV flight dynamics restriction that the velocity and acceleration of the UAV cannot change arbitrarily, the UAV is required to replan a feasible flight trajectory based on the UAV's current flight state and the task offloading requirements of users, thereby enhancing the system performance. The main contributions of this work are summarized as follows:

- Taking into account the dynamic control of the UAV trajectory, the system architecture for a UAV-assisted MEC is investigated. The communication model, UAV control model, as well as the computing and transmission model are analyzed in detail. Subsequently, the joint optimization problem minimizing the weighted energy consumption and delay is formulated when considering the UAV dynamics constraint.
- Constrained by the system dynamics of the UAV, where the velocity and acceleration are not allowed to change arbitrarily, a LLDDPG algorithm is proposed to address the joint dynamic trajectory and resource allocation problem. Specifically, for a practical solution, the optimization problem is decomposed into three distinct subproblems. Firstly, a DDPG-based UAV trajectory design algorithm is developed to acquire the desired optimal trajectory. Subsequently, the LQR-based tracking control algorithm is introduced to derive the actual UAV flight trajectory subject to the system dynamics. Finally, the resource allocation problem regarding the offloading ratio and computation frequency assignments is solved using the Lagrange duality method.
- Numerical simulation results extensively demonstrate the efficacy of the proposed LLDDPG algorithm in terms of learning rate, loss function, and reward convergence. Additionally, the performance evaluations with different weight parameters and the effectiveness of the LLDDPG algorithm in actual UAV flight control are also investigated and analyzed.

The remainder of the paper is organized as follows. In Section 2, the related works are reviewed. The UAV-assisted MEC system model and the optimization problem formulation

are presented in Section 3. In Section 4, the LLDDPG algorithm is proposed. In Section 5, numerical simulations and results are presented. Finally, conclusions are drawn in Section 6.

## 2. Related Works

This section briefly reviews the works related to UAV-assisted MEC resource allocation and trajectory design, and the existing issues and challenges are also discussed.

### 2.1. UAV-Assisted MEC Resource Allocation

In recent years, the increasing maturity of UAV-assisted wireless communication technology has boosted the further development of MEC systems. How to combine UAV advantages and MEC networks has become a research hotspot [16–25]. Guo et al. [16] introduced a UAV-enabled MEC system, in which the UAV served as a relay between the base station and the offloading user. This work investigated the joint optimization of flight trajectories and computational offloading, considering both user service quality and energy consumption. In [17], it maximized the UAV's transmit power efficiency by jointly optimizing bandwidth assignment, transmission time, UAV placement, and power allocation control. Furthermore, Qin et al. [18] investigated the energy efficiency of a UAV-assisted MEC system by considering energy consumption and the device task requirements.

The above works mainly focused on the energy efficiency of the MEC system, while ignoring the influence of the UAV trajectory design. In order to improve the network lifetime and computation capability associated with the UAV, Wang et al. [19] investigated an optimization problem that aimed to minimize the total energy consumption of the UAV through a combined approach of zone division and UAV trajectory planning. Wang et al. [20] addressed the efficiency maximization problem by jointly optimizing bandwidth management, UAV trajectory, computation offloading, and computation resource assignment. Diao et al. [21] optimized the computational offload strategy and UAV trajectory in the UAV-enabled MEC system. Their objective was to minimize the total energy consumption and delay while enhancing the user's service quality. Hu et al. [22] also focused on the joint optimization problem to maximize the data offloading efficiency while minimizing the UAV energy consumption. Liu et al. [23] put forward a system energy minimization problem subject to constraints such as UAV trajectory, transmit power, and CPU frequency. Zeng et al. [24] investigated the problem of minimizing UAV energy consumption, including propulsion energy and communication-related energy, while satisfying the communication throughput requirements of each ground node. By leveraging the traveling salesman problem (TSP) with neighborhood and convex optimization techniques, a successive convex approximation (SCA)-based algorithm is proposed. Yang et al. [25] considered a UAV-enabled MEC system to jointly optimize UAV energy and trajectory control while satisfying long-term data queue stability, and then a perturbed Lyapunov optimization-based offloading and trajectory (PLOT) control algorithm was proposed.

In the aforementioned research, the authors delved into the joint optimization of variables such as UAV trajectory, offloading strategy, computation frequency, and transmission power. Nevertheless, these existing works predominantly focus on the design of desired UAV flight trajectory, completely disregarding the restrictions imposed by UAV flight dynamics. In reality, the inherent limitations in the UAV's flight capabilities can result in a growing deviation between the desired and actual flight trajectories, which might potentially lead to significant performance degradation.

### 2.2. UAV Trajectory Control

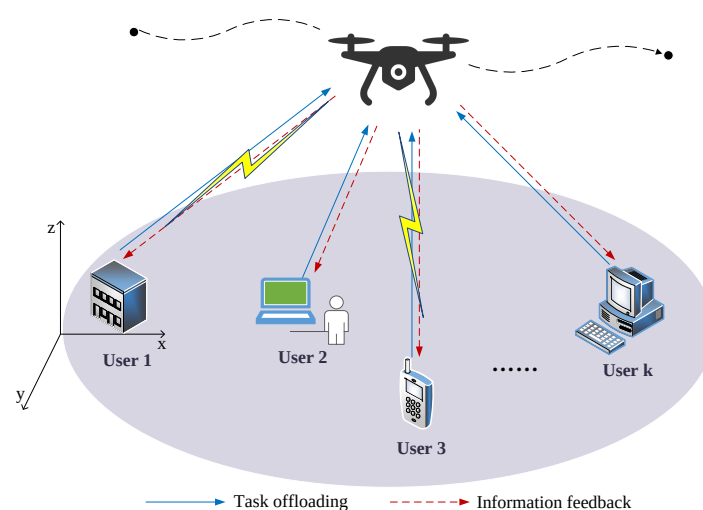
In order to ensure the efficiency of the UAV-assisted MEC system, it is crucial to jointly optimize the system resource allocation and UAV trajectory control. Since the UAV trajectory directly influences the MEC system's energy consumption and user service quality, it is of great importance to track and control the flight trajectory. The trajectory flight control problem continues to attract significant attention due to its potential to enhance the system's adaptability and its capability to handle dynamics and uncertainties [26–30]. Addressing

the UAV trajectory tracking control problem, Yan et al. [26] proposed a dynamic tracking method for UAV landing trajectories based on chaos genetic algorithms. Lee et al. [27] proposed a trajectory tracking control methodology utilizing backward stepping and LQR control. Furthermore, Li et al. [28] presented a control-oriented UAV trajectory design approach that incorporates both the kinematics and dynamics equations of the UAV. However, these works primarily focused on the UAV's trajectory tracking control and neglected the effect of tracking deviation on the overall performance of the UAV-assisted MEC system. Zhang et al. [29] tried to investigate this limitation by considering a network control system with delays, and an adaptive dynamic programming-based tracking control algorithm was proposed to generate real-time control actions. Liu et al. [30] focused on the UAV trajectory planning problem for an environmental monitoring system. The formulated optimization problem was divided into two subproblems: the UAV velocity optimization and trajectory optimization. To address these subproblems, the solving algorithms, based on SCA and general algorithm (GA), respectively, were proposed.

Regrettably, there have been few studies that take into account the joint design of system resource allocation and real-time UAV trajectory control in the UAV-assisted MEC system. Most of the existing articles have studied the desired trajectory planning, assuming that the UAV has the perfect flight capability and operates in a static transmission environment. In this paper, we aim to address this gap to investigate the inherent constraints of UAV flight dynamics, and focus on the joint optimization problem of dynamic trajectory design and resource allocation for a UAV-assisted MEC system.

### 3. System Modeling and Optimization Problem Formulation

A UAV-assisted MEC system is depicted in Figure 1, which consists of the UAV and multiple users. The UAV, equipped with an MEC server, is capable of simultaneously transmitting information and providing edge computing service. Considering the limited computation capacities, the users are required to offload a portion of their computing tasks to the MEC server through the shared wireless network. Generally, the user's computing tasks can be divided into two parts; one is computed locally, and the other is offloaded to the UAV for processing. The UAV aggregates the received information to form a new global model, and then feeds back the global information to the users. In order to enhance the energy efficiency and address the dynamic nature of entire system environment, the UAV's flight needs to be frequently adjusted and controlled.



**Figure 1.** A UAV-assisted MEC system.

### 3.1. Communication Model

Let  $q_k = \{x_k, y_k\}$  denote the location of the  $k$ -th ground user, which is assumed to be known by the UAV. The position of the UAV in the  $n$ -th time slot can be represented as  $q[n] = \{x_u[n], y_u[n], H\}$ , where  $H$  is the fixed flight altitude of the UAV. Typically, it is assumed that the wireless channel between the user and the UAV is mainly dominated by the LoS. Thus, the channel gain  $g_k[n]$  between the user  $k$  and the UAV in the  $n$ -th time slot can be expressed as [31]

$$g_k[n] = \beta_0 d_k^{-2}[n] \quad (1)$$

where  $\beta_0$  is the channel coefficient and  $d_k[n]$  is the distance between the user  $k$  and the UAV for which  $d_k[n] = \sqrt{(x_u[n] - x_k)^2 + (y_u[n] - y_k)^2 + H^2}$ .

Subsequently, the transmission data rate from the user  $k$  to the UAV can be derived as

$$R_k[n] = B \log_2(1 + \frac{g_k[n]p}{N_0}) \quad (2)$$

where  $N_0$  is the noise power, and  $B$  and  $p$  represent the assigned bandwidth and transmit power, respectively.

Similarly, the transmission data rate from UAV to the user  $k$  is given by  $R_u[n] = B \log_2(1 + g_k[n]p_u/N_0)$ , where  $p_u$  represents the transmit power of the UAV.

### 3.2. Computing and Transmission Model

Considering the partial offloading computation scenario, the computation tasks can be divided into two parts. One is handled locally, while the other is offloaded to the MEC server for processing.

(1) Local Computation: Each user has a restricted computation capability for performing local computing, and the CPU frequency  $f_k[n]$  serves as the key factor. The delay  $T_k^L[n]$  and the energy consumption  $E_k^L[n]$  for local computing can be, respectively, deduced as follows [32].

$$T_k^L[n] = \frac{\beta_k[n]L_k[n]C_k}{f_k[n]} \quad (3a)$$

$$E_k^L[n] = \eta T_k^L[n]f_k^3[n] = \eta\beta_k[n]C_kL_k[n]f_k^2[n] \quad (3b)$$

where  $C_k$  is the number of CPU cycles required for computing,  $\eta$  denotes the effective capacitance coefficient for which  $\eta f_k^3[n]$  is the CPU power consumption,  $L_k[n]$  is the total task sizes, and  $\beta_k[n]L_k[n]$  represents the task processed at the local level.

(2) Task Offloading: The offloading delay is determined by the offloading task size, which is given by

$$T_k^O[n] = \frac{(1 - \beta_k[n])L_k[n]}{R_k[n]} \quad (4)$$

Similarly, the relevant transmission energy consumption of user task offloading is given by

$$E_k^O[n] = p_u T_k^O[n] = \frac{(1 - \beta_k[n])L_k[n]p_u}{R_k[n]} \quad (5)$$

(3) UAV Computation: Once the user offloads the task to the MEC server, the UAV processes the task, which causes the processing delay as

$$T_k^c[n] = \frac{(1 - \beta_k[n])L_k[n]C_k}{f_{u,k}[n]} \quad (6)$$

where  $f_{u,k}[n]$  denotes the CPU computing frequency of the UAV allocated to user  $k$ .

Similar to (3b), the energy consumption for offloaded task processing can be obtained as

$$E_k^c[n] = \psi f_{u,k}^3[n] T_k^c[n] = \psi f_{u,k}^2[n] (1 - \beta_k[n]) L_k[n] C_k \quad (7)$$

where  $\psi$  is the UAV effective capacitance coefficient.

(4) Result Feedback: Once the UAV task computation is completed, the results will be fed back to the relevant user, and the transmission-induced delay  $T_k^u[n]$  is given by

$$T_k^u[n] = \frac{L_u[n]}{R_u[n]} \quad (8)$$

where  $L_u[n]$  represents the transmission data size back to the user.

Then, the energy consumption for the information feedback is

$$E_k^u[n] = p_u T_k^u[n] = \frac{L_u[n] p_u}{R_u[n]} \quad (9)$$

From (3) to (9), the total delay and energy consumption in each time slot are given by

$$T[n] = \sum_{k=1}^K T_k^L[n] + T_k^o[n] + T_k^c[n] + T_k^u[n] \quad (10a)$$

$$E[n] = \sum_{k=1}^K E_k^L[n] + E_k^o[n] + E_k^c[n] + E_k^u[n] \quad (10b)$$

### 3.3. UAV Control Model

The existing solutions for UAV trajectory planning are typically carried out under the assumption of perfect UAV flight capability, in which case the velocity and acceleration of UAV can change arbitrarily. However, it is impractical in actual UAV flight, considering the constraints on acceleration and velocity as well as the underlying dynamics principles. In addition, the time-varying task requirements of users and the dynamic transmission environment contribute to the frequent adjustments in the UAV trajectory. Therefore, real-time UAV trajectory control is raised to reduce the performance degradation induced by the state deviations.

Typically, the dynamics of the UAV can be expressed as

$$\dot{q}(t) = v(t) \quad (11a)$$

$$\dot{v}(t) = a(t - \Delta) \quad (11b)$$

where  $q(t)$  and  $v(t)$ , respectively, denote the UAV's location and velocity,  $a(t)$  is the UAV acceleration, and  $\Delta$  is the time delay.

Define a new state vector as

$$w(t) = [q(t), \dot{q}(t)] = [q(t), v(t)] \quad (12)$$

Based on (11) and (12), the dynamics model can be rewritten as

$$\dot{w}(t) = Aw(t) + Ba(t - \Delta) \quad (13)$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, the relevant discrete-time dynamics is given by [33]

$$w[n+1] = A_0 w[n] + B_1 a[n] + B_2 a[n-1] \quad (14)$$

where

$$A_0 = e^{A\Delta T}, B_1 = \int_0^{\Delta T-\Delta} e^{A\Delta T} dt B, w[n] = w[n\Delta T], B_2 = \int_{\Delta T-\Delta}^{\Delta T} e^{A\Delta T} dt B, \quad (15)$$

where  $a[n]$  is the control strategy (i.e., acceleration) of UAV, and  $\Delta T$  is the duration of one time slot.

The propulsion energy is the significant flight energy consumption for the UAV, which is typically given by [34]

$$E_u^{fly}[n] = (N_1 \|v[n]\|^3 + \frac{N_2}{\|v[n]\|} (1 + \frac{\|a[n]\|^2}{a_0^2})) \Delta T + \frac{1}{2} M (\|v[n]\|^2 - \|v[n-1]\|^2) \quad (16)$$

where  $N_1$  and  $N_2$  are system-determined parameters,  $M$  is the mass of the UAV, and  $a_0$  is the gravitational acceleration.

### 3.4. Optimization Problem Formulation

The objective of UAV-assisted MEC is to minimize the weighted energy consumption and delay through the optimization design of UAV trajectory, task offloading strategy, and the allocation of computation resources for both UAV and users, which can be formulated as

$$P1 : \min_{F, \beta, Q} \sum_{n=1}^N \varepsilon_1 E_u^{fly} + \varepsilon_2 E[n] + \varepsilon_3 T[n] \quad (17a)$$

$$s.t. w[n+1] = A_0 w[n] + B_1 a[n] + B_2 a[n-1] \quad (17a)$$

$$\sum_{k=1}^K f_{u,k} \leq F_{u,max}, \forall k \quad (17b)$$

$$0 \leq f_{u,k}, \forall k \quad (17c)$$

$$0 \leq f_k \leq F_k, \forall k \quad (17d)$$

$$0 \leq \beta_k \leq 1, \forall k \quad (17e)$$

where  $F \triangleq \{f_k[n], f_{u,k}[n]\}$  represents the assigned CPU frequencies,  $\beta = \{\beta_k[n]\}$  is the offloading ratio,  $Q = \{q_u[n]\}$  is the UAV flight trajectory, and  $\varepsilon_1, \varepsilon_2, \varepsilon_3$  are the weight coefficients. In the optimization problem, (17a) is the corresponding discrete-time dynamics of the UAV, (17b)–(17d) are the maximum CPU computation frequency constraints, and (17e) is the offloading ratio constraint of the user.

## 4. LLDDPG Algorithm Design

As mentioned above,  $P1$  is a mixed optimization problem due to the system dynamics restriction (17a), which has been commonly ignored in existing studies. For a practical solution, the optimization problem  $P1$  can be decomposed into three distinct subproblems. The detailed analysis will be presented below.

### 4.1. UAV Trajectory Design

Given the offloading ratio and the CPU computation frequency assignments, the optimization problem  $P1$  can be simplified as a UAV trajectory design problem.

#### 4.1.1. DDPG-Based Desired Trajectory Design

Assuming the perfect UAV flight capacity, the optimization problem  $P1$  can be simplified to a desired trajectory design challenge such that

$$P1.1 : \min_Q \sum_{n=1}^N (\varepsilon E_u^{fly} + \sum_{k=1}^K (E_k^o[n] + E_k^u[n])) + \sum_{k=1}^K (T_k^o[n] + T_k^u[n]) \quad (18)$$

To address the subproblem of minimizing the system cost in terms of energy consumption and delay as in  $P1.1$ , deep reinforcement learning (DRL), which has shown remarkable ability in solving intricate network optimization challenges, is employed to achieve the desired trajectory of the UAV. Typically, DRL can be formulated as a Markov decision process (MDP), where the next system state depends solely on the current state and the action determined by the agent. To be specific, the MDP is defined by the tuple  $(S, A, R, P)$ , where  $S$ ,  $A$ , and  $R$ , respectively, represent the set of states, actions, and rewards, and  $P$  is the transition probability from state  $s_n$  to state  $s_{n+1}$  [35]. The following are the definitions for the state, action, and reward functions.

(1) State: Based on the MEC system model and task offloading model formulated in Section 3, the state  $s_n$  consists of the locations of the UAV and users, as well as the user task requirements, which can be defined as

$$s_n = \{x_1[n], y_1[n], x_2[n], y_2[n], \dots, x_K[n], y_K[n], L_1[n], L_2[n], \dots, L_K[n], x_u[n], y_u[n]\} \quad (19)$$

(2) Action: The UAV is required to determine its movements, including flight velocity  $v[n]$  and direction  $\theta[n]$ , which is given by

$$a_n = \{v[n] \cos(\theta[n]), v[n] \sin(\theta[n])\} \quad (20)$$

(3) Reward: The reward function is highly associated with the optimization objective. The objective in  $P1.1$  can be directly served as the reward function:

$$r_n = -(\varepsilon_1 E_u^{fly} + \varepsilon_2 (E_k^o + E_k^u) + \varepsilon_3 (T_k^o + T_k^u)) \quad (21)$$

In the proposed MDP framework, the UAV acts as the agent, interacting with the environment by observing a state  $s_n$ . Then, it executes an action  $a_n$  based on its policy  $\pi$ . Following the execution of action  $a_n$ , the agent receives a reward  $r_n$  and transitions to the next state  $s_{n+1}$ . DDPG, as one of the classic DRL algorithms, stands out for its ability to leverage low-dimensional observations to learn effective strategies in continuous action spaces. In dynamic environments, this method has been demonstrated to be highly effective in making decisions and achieving the desired trajectory of the UAV. Therefore, a DDPG-based algorithm for the design of desired UAV trajectory is proposed to minimize both the energy consumption and time delays induced by the UAV and users.

The architecture of the DDPG-based algorithm is depicted in Figure 2. The DDPG network comprises actor and critic neural networks. Specifically, the actor network, serving as the policy network, takes the current environmental state as input and generates relevant actions through the analysis of the neural network. To enhance the efficacy of the iterative update strategy, the critic network, which utilizes a value-based learning approach, can be updated at each step.

The evaluation value  $Q(s_n, a_n | \theta^Q)$  is acquired by executing action  $a_n$  in the state  $s_n$ . The action  $a_n = \mu(s_n | \theta^\mu)$  is taken in each state, reaching a specific value through a deterministic behavioral strategy. DDPG draws upon the dual network structure of DQN and experience replay to dissociate the behavior strategy network from the evaluation strategy network. The actor and critic have two networks with a similar structure but have asynchronous parameter renewals. In this manner, the convergence speed is quicker when training the network, and the soft update formula of the target network is set as follows:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (22a)$$

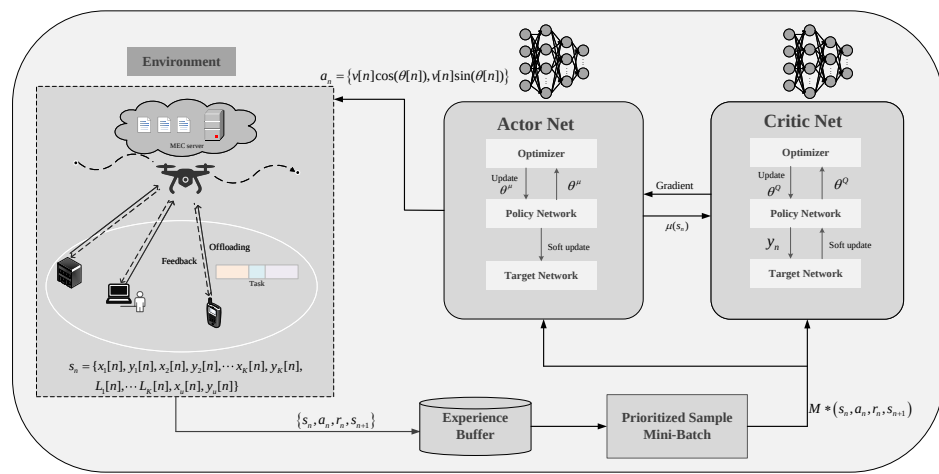
$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'} \quad (22b)$$

where  $\tau$  represents the update rate,  $\theta^Q$  and  $\theta^\mu$  are the parameters of the critic network, while  $\theta^{Q'}$  and  $\theta^{\mu'}$  are the parameters of the target network.

The critic network parameter is updated as

$$y_i = R + \gamma Q'(s_{n+1}, \mu'(s_{n+1} | \theta^{\mu'})) | \theta^{Q'} \quad (23)$$

where  $y_i$  represents the actual evaluation value calculated by the target network, and  $\gamma$  is the reward decay rate.



**Figure 2.** Schematic of DDPG algorithm.

Then, the loss function can be expressed as

$$L = \frac{1}{N} \sum_n (y_n - Q(s_n, a_n | \theta^Q))^2 \quad (24)$$

The actor network parameter is updated by

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s_n, a | \theta^Q) \quad (25)$$

where  $a = \mu(s_n) \nabla_{\theta^\mu} \mu(s_n | \theta^Q)$ ,  $\nabla \mu$  denotes the modification trend of the actor parameters, and  $\nabla Q$  indicates the actor network update direction calculated by the critic.

After completing the training process, the optimized network parameters of the actor, denoted as  $\theta^{\mu*}$ , are obtained. Subsequently, the desired UAV trajectory is given by

$$q^*[n] = \mu(s_n | \theta^{\mu*}) \quad (26)$$

Consequently, the DDPG-based algorithm for the design of desired UAV trajectory can be summarized in Algorithm 1.

**Algorithm 1** DDPG-based algorithm for desired UAV trajectory design**Input:** The positions of the UAV  $q(t)$ **Output:** UAV movement policy

```

1: Initialize the main actor network and critic network.
2: Initialize the target actor network and critic network.
3: Initialize the replay memory B and initialize  $\sigma^2 = 2.0$ ,  $\varepsilon = 0.9$  for action exploration.
4: for episode  $:= 1, \dots, M$  do
5:   for step  $n := 1, \dots, N$  do
6:     Update the environment status, observe the current environment state  $s_n$ .
7:     Set the current action  $a_n \sim N(\mu(s_n|\theta^\mu), \sigma^2)$ ;
8:     Execute the action  $a_n$ , obtain the reward  $r_n$ , and transit to the next state  $s_{n+1}$ .
9:     Store the experience tuple  $(s_n, a_n, r_n, s_{n+1})$  into replay memory B.
10:    if Update then
11:      Randomly sampling the mini-batch transitions from B.
12:      Renew the critic network through minimizing the critic loss.
13:      Renew the actor network through maximizing the actor loss.
14:      Renew the target networks based on (29).
15:      Decay the action  $\sigma^2 \leftarrow \sigma^2 \varepsilon$ 
16:    end if
17:  end for
18: end for

```

## 4.1.2. LQR-Based Trajectory Tracking Control

In practical applications, the UAV faces a dynamic transmission environment, time-varying user task requirements, and diverse flight disturbances. To enhance system performance and control stability, the UAV must dynamically adjust its flight trajectory. In addition, due to the inherent limitations in the UAV's velocity and acceleration, the actual flight state cannot be adjusted arbitrarily to match the desired trajectory  $q^*$  obtained from subproblem P1.1. Therefore, there exist inevitable deviations between the desired and actual flight trajectories, potentially degrading the system performance and even significantly diminishing the effectiveness of the optimization design. To mitigate this issue, it is imperative to minimize the desired UAV trajectory deviation through real-time flight control. In this regard, an LQR-based UAV trajectory tracking control algorithm is proposed to align the actual flight trajectory with the desired trajectory as closely as possible, thereby improving the overall system performance.

Given the desired UAV trajectory design  $q^*$  obtained from subproblem P1.1, the trajectory tracking control problem is given by

$$\begin{aligned}
 P1.2 : q &\rightarrow q^* \\
 s.t. \quad w[n+1] &= A_0 w[n] + B_1 a[n] + B_2 a[n-1]
 \end{aligned} \tag{27}$$

The UAV location and velocity deviations can be, respectively, obtained as

$$\tilde{q} = q - q^* \tag{28a}$$

$$\tilde{v} = v - v^* \tag{28b}$$

Define a new vector  $\tilde{w} = (\tilde{q}, \tilde{v})$ , and then the UAV deviation dynamics can be derived based on (12), (13), and (14) as

$$\tilde{w}[n+1] = A_0 \tilde{w}[n] + B_1 a[n] + B_2 a[n-1] \tag{29}$$

Then, by using the typical quadratic cost function, the UAV trajectory tracking control problem P1.2 can be equivalent to be the following optimization problem [36]:

$$\begin{aligned} \min_{a[n]} & \tilde{w}^T[n]Q\tilde{w}[n] + \sum_{n=0}^{N-1} \tilde{w}^T[n]Q\tilde{w}[n] + a^T[n]Ra[n] \\ \text{s.t. } & \tilde{w}[n+1] = A_0\tilde{w}[n] + B_1a[n] + B_2a[n-1] \end{aligned} \quad (30)$$

where  $N$  denotes the finite time horizon, and  $Q$  and  $R$  are system-determined parameters.

It can be seen that the objective of (30) is to minimize the trajectory tracking deviation through the optimal design of UAV flight  $a[n]$ . Then, an LQR-based trajectory tracking control algorithm is proposed to solve the optimization problem (30).

Define a new state vector as

$$z[n] = [\tilde{w}[n], a[n-1]] \quad (31)$$

The optimization problem (30) can be rewritten as

$$\begin{aligned} \min_{a[n]} & z^T[N]\dot{Q}z[N] + \sum_{n=0}^{N-1} z^T[n]\dot{Q}z[n] + a^T[n]Ra[n] \\ \text{s.t. } & z[n+1] = C_nz[n] + D_na[n] \end{aligned} \quad (32)$$

where

$$\dot{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}, C[n] = \begin{bmatrix} A[n] & B_2[n] \\ 0 & 0 \end{bmatrix}, D[n] = \begin{bmatrix} B_1[n] \\ 0 \end{bmatrix} \quad (33)$$

The optimization problem (32) is a classic LQR-based control problem, and the optimal control strategy can be derived as [33]

$$a[n] = -l[n]z[n] \quad (34)$$

where

$$\begin{aligned} l[n] &= [D^T[n]S[n+1]D[n] + R]^{-1}D^T[n]S[n+1]C[n], \\ S[n] &= C^T[n]S[n+1]C[n] + \dot{Q} - l^T[n]D^T[n]S[n+1]C[n], \\ S[N] &= \dot{Q} \end{aligned} \quad (35)$$

Based on (34) and (35), the actual UAV flight trajectory  $q[n]$  can be obtained based on the UAV acceleration strategy  $a[n]$ .

#### 4.2. Computation Resource Allocation Optimization

After the actual flight trajectory  $q[n]$  is determined from Sections 4.1 and 4.2, the optimization problem in P1 can be equivalent to be a computation resource allocation problem, which is given by

$$\text{P1.3 : } \min_{F, \beta} \sum_{n=1}^N E_k^L[n] + E_k^c[n] + T_k^L[n] + T_k^c[n] \quad (36a)$$

$$\text{s.t. } \sum_{k \in K} f_k \leq F_k, \forall k \quad (36b)$$

$$f_k, f_u \geq 0, \forall k \quad (36c)$$

$$0 \leq \beta_k \leq 1, \forall k \quad (36d)$$

Subproblem P1.3, as a convex problem, can be typically solved using the Lagrange duality method as follows [37].

**Theorem 1.** The UAV trajectory  $q[n]$ , the CPU frequencies of the users and UAV, as well as the optimal offloading ratios, respectively, denoted by  $\beta_k^*[n]$ ,  $f_u^*[n]$ , and  $f_k^*[n]$ , can be expressed as follows.

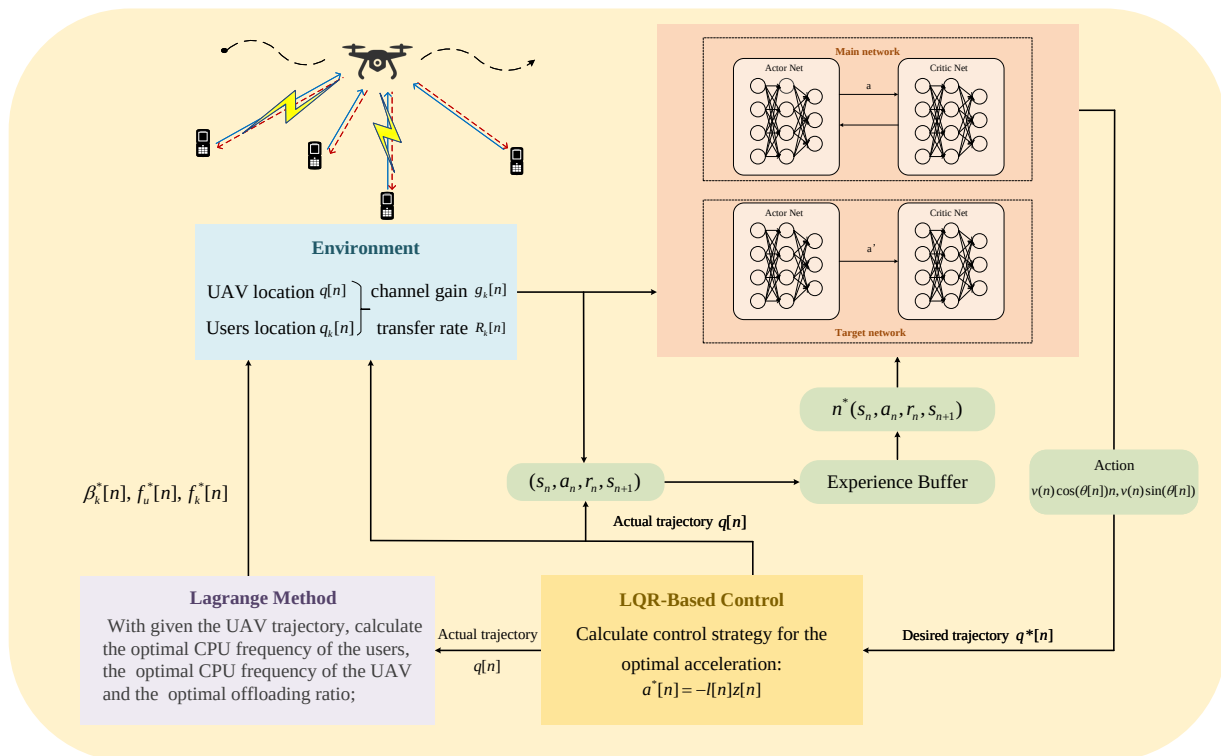
$$f_k^*[n] = \sqrt{\frac{\gamma_k}{3\sigma_c MK \sum_{i=n}^N \kappa_{k,i}}}, \quad k \in K, n \in N \quad (37a)$$

$$f_u^*[n] = \begin{cases} 0, & n = 1 \\ \sqrt{\frac{\lambda_N - \sum_{i=n}^{N-1} \lambda_i}{3\sigma_c M}}, & n = 2, \dots, N-1 \\ \sqrt{\frac{\lambda_N}{3\sigma_c M}}, & n = N \end{cases} \quad (37b)$$

$$\beta_k^*[n] = B\psi \log_2 \left\{ \frac{Bg_k[n] \left[ \sum_{i=n+1}^{N-1} \lambda_i + \xi_k - \lambda_N \right]}{\sum_{i=n}^N \kappa_{k,i} \sigma^2 \ln 2} \right\} \quad (37c)$$

where  $\xi_k \geq 0$ ,  $\kappa_{k,n} \geq 0$ ,  $\lambda_n \geq 0$ .

Finally, based on the algorithm analysis presented in Sections 4.1 and 4.2, the schematic of the proposed LLDDPG algorithm for the joint optimization of the UAV dynamic trajectory and resource allocation can be illustrated as shown in Figure 3, and its algorithmic procedure can be summarized in Algorithm 2.



**Figure 3.** Schematic of the proposed LLDDPG algorithm.

**Algorithm 2** LLDDPG Algorithm

- 1: Initialization:  $q_k, q_u[0], q_u[N], L_k[0], \beta_k[0], f_{u,k}[0], f_k[0], p_u, p, B, C_k, N_0, \eta, \psi$ .
- 2: Solve the subproblem P1.1, and obtain the desired trajectory of UAV  $q^*[n]$  based DDPG algorithm as Algorithm 1.
- 3: Solve the subproblem P1.2 by using the LQR method:
- 4: Derive the control coefficient  $l[n]$  offline based on (35).
- 5: Obtain the acceleration control strategy  $a[n]$  as in (34) based on UAV state deviations and previous control strategies.
- 6: Then, the actual UAV flight trajectory  $q[n]$  can be obtained based on acceleration control strategy  $a[n]$ .
- 7: Solve the subproblem P1.3 by using the Lagrange duality method:
- 8: Obtain the optimal offloading ratios  $\beta_k^*[n]$  and the CPU frequency of the users  $f_u^*[n]$  and UAV  $f_k^*[n]$  based on (37).
- 9:  $q[n], \beta_k^*[n], f_u^*[n]$ , and  $f_k^*[n]$  are fed back for parameter update of DDPG algorithm.

**5. Simulations**

In this section, the performances of the proposed LLDDPG algorithm are comprehensively evaluated through simulations and numerical results. Specifically, the convergence performance of the algorithm is analyzed, and the performance comparisons with other existing works are given.

**5.1. Simulation Settings**

In simulations, the number of user devices is set to 100 and the task duration time is set to 10 min. The ground users are distributed in a 50 m  $\times$  50 m area. The flight height of the UAV is 10 m and the maximum flight velocity  $V_{\max} = 10$  m/s. At the beginning, the UAV starts the task at a random location. The vertical and horizontal coverage radius of the UAV are set, respectively, to  $X_d = 25$  m and  $X_h = 10$  m. The user's data cache is updated each slot time with a Poisson process. The data buffer capacity  $U_{\max}$  is set to 5000 packets, and the relevant data transfer size is  $Q = 10$  Mbits. The UAV and user's transmit power are, respectively, set to  $P_u = 30$  dBm and  $P_k = -20$  dBm. The other system corresponding parameters are shown in Table 1, where parameter settings refer to [38]. The structure and parameters of the DDPG network are shown in Table 2. During the implementation, the final output layer of the actor network is set to the tanh layer, and all hidden layers are completely connected and activated using ReLU functions.

**Table 1.** Simulation parameter settings.

| Symbol     | Description                                | Setting         |
|------------|--------------------------------------------|-----------------|
| $H$        | UAV flight altitude                        | 10 m            |
| $V_{\max}$ | Maximum flight velocity                    | 10 m/s          |
| $B$        | Bandwidth                                  | 1 Mz            |
| $\beta_0$  | Channel power gain                         | −30 dB          |
| $N_0$      | Noise power                                | −90 dBm         |
| $a, b$     | LoS probability                            | 10, 0.6         |
| $C_k$      | Number of CPU cycles per bit               | 1000 cycles/bit |
| $\psi$     | Effective capacitance coefficient of UAV   | $10^{-28}$      |
| $\eta$     | Effective capacitance coefficient of users | $10^{-28}$      |
| $\tau$     | Time slot                                  | 0.1 s           |

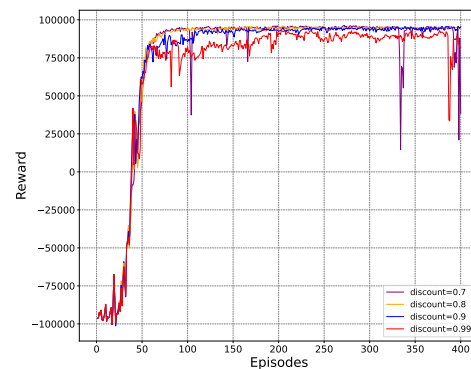
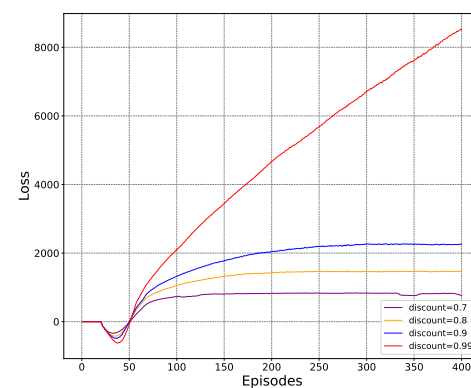
**Table 2.** Parameters of DDPG training.

| Parameter               | Value |
|-------------------------|-------|
| Learning rate of critic | 0.001 |
| Learning rate of actor  | 0.001 |
| Update rate             | 0.001 |
| Discount factor         | 0.9   |
| Batch size              | 64    |

## 5.2. Results and Analysis

Figure 4 shows the convergence of the reward function and the effect of different discount factors on the reward. The results reveal that the model convergence speed is relatively fast, and the final convergence level of the reward function is comparable under the conditions of different discount factors. When the discount factor is 0.99 and 0.7, there are abnormal fluctuations in the subsequent convergence stage, indicating that the exploration of action space is not comprehensive. Since there is no significant change in the final performance when the discount factor is 0.9, it indicates that the agent is able to learn the optimal policy.

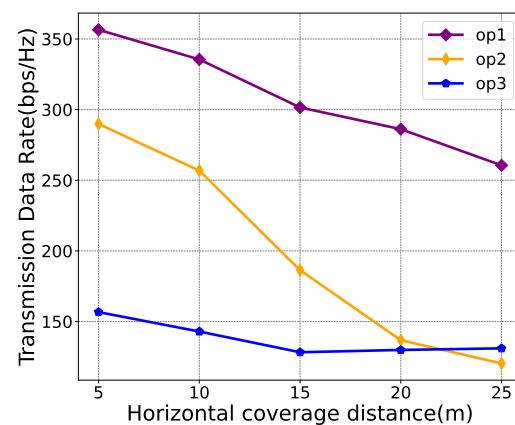
Figure 5 presents the correlation between the learning rate and the loss function. Initially, the learning rate shows a high sensitivity to the loss function of the model. In the case of a low learning rate, the value of the loss function still increases slowly even after multiple training episodes. Conversely, in the case of a higher learning rate, the loss function rises rapidly, but it eventually takes a long time to converge. To ensure more comprehensive exploration of the agent's action space, it is better to keep a smooth increase in the loss function and ultimately reach the optimal value. To sum up, the discount factor 0.9 is selected as a moderate discount factor to achieve the desired result.

**Figure 4.** Reward of different discount factors.**Figure 5.** Loss of different discount factors.

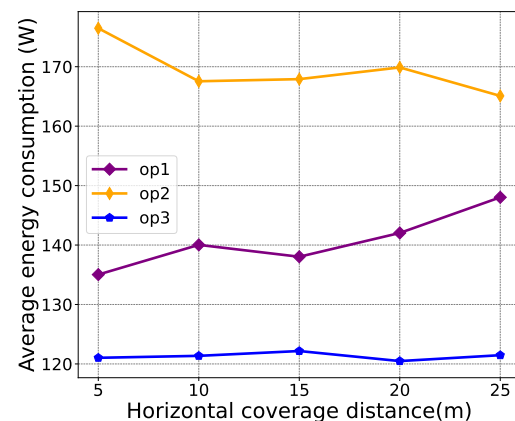
For performance evaluations, Figures 6–8 illustrate the effects of different weight parameters, namely  $\varepsilon_1$ ,  $\varepsilon_2$ , and  $\varepsilon_3$ , on different system performances, including the transmission data rate, average energy consumption, and number of service users. The parameters for the comparison experiments are set as shown in Table 3. The horizontal coverage distance is set to 5 m, 10 m, 15 m, 20 m, and 25 m. It is observed from Figure 6 that, with the increase in the horizontal coverage distance, the transmission data rate decreases under all policies. In Figure 7, the average energy consumption under “op2” is higher than that of “op1” and “op3” because the weight of energy consumption is set to 0 in “op2”. As can be seen from Figure 8, as the horizontal coverage distance increases, the number of service users also increases. Overall, the transmission data rate and the number of service users under the “op1” policy are better than the other two strategies, and the energy consumption performance is slightly worse than that of the “op2” policy. This also validates that the proposed algorithm can successfully learn control strategies that simultaneously optimize multiple optimization objectives.

**Table 3.** Comparison of experiment parameters.

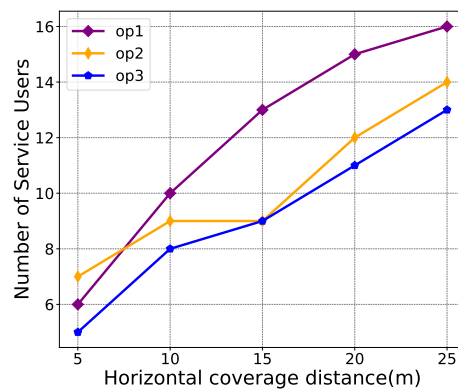
| Name        | Parameter                                                        |
|-------------|------------------------------------------------------------------|
| <i>opt1</i> | $\varepsilon_1 = 0.001, \varepsilon_2 = 0.01, \varepsilon_3 = 1$ |
| <i>op2</i>  | $\varepsilon_1 = 0, \varepsilon_2 = 0, \varepsilon_3 = 1$        |
| <i>op3</i>  | $\varepsilon_1 = 0, \varepsilon_2 = 1, \varepsilon_3 = 0$        |



**Figure 6.** Transmission data rate comparison under different weight parameters.

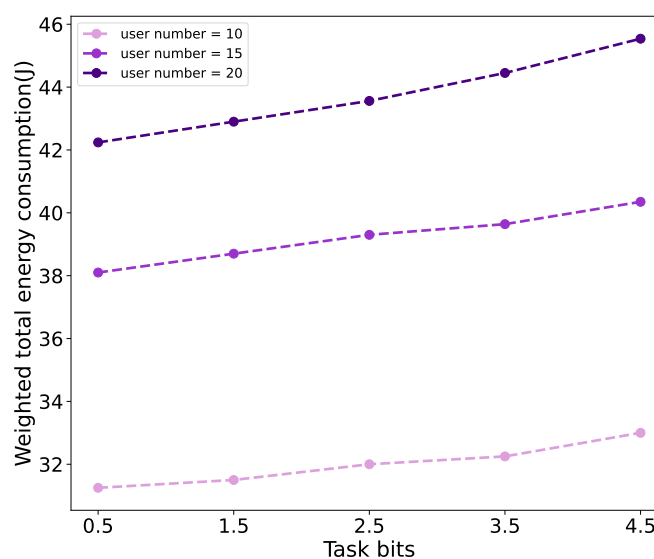


**Figure 7.** Average energy consumption comparison under different weight parameters.



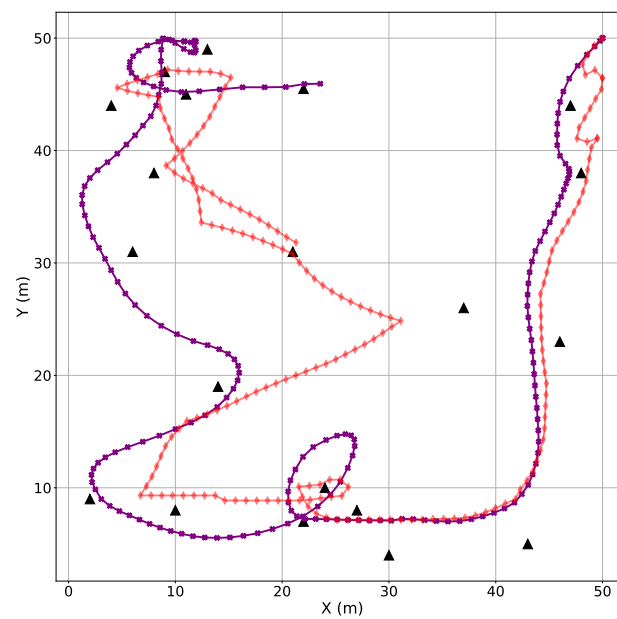
**Figure 8.** Number of service users comparison under different weight parameters.

Figure 9 shows the relationship between the weighted total energy consumption of all users and the quantity of user tasks under scenarios with different numbers of users. It can be seen that the weighted total energy consumption increases when the amount of computing tasks becomes larger, and this can be inferred from Formulas (3) and (5). In addition, the weighted total energy consumption increases with the increase in the number of users, owing to the requirement of consuming more energy between the UAV and the users for computing and transmission purposes.

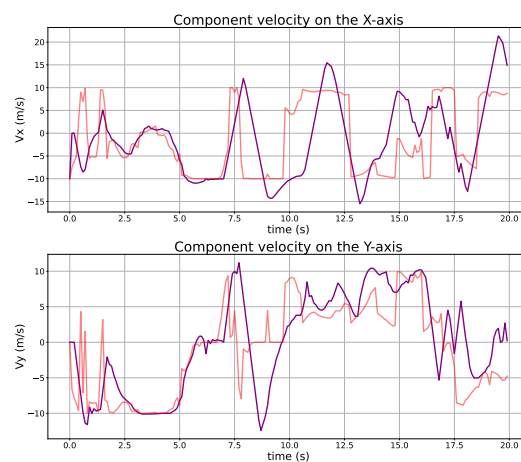


**Figure 9.** The relationship between weighted total energy consumption and tasks under different numbers of users.

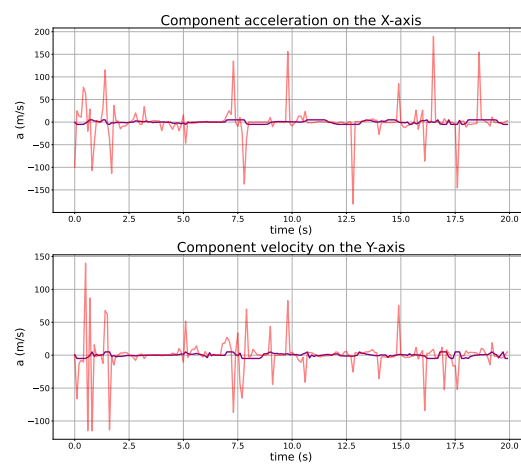
In order to investigate the effectiveness of the LLDDPG algorithm in actual UAV flight control, a numerical simulation of the scenario of UAV-assisted users in task offloading is conducted. The comparisons of the UAV trajectory, velocity, and acceleration are, respectively, presented in Figures 10–12. It can be observed that the size of the offloading task for each user has a significant effect on the UAV trajectory design, and the UAV needs to approach the user with a higher task demand as quickly as possible. Initially, the global planning of the UAV trajectory is carried out based on the DDPG algorithm to ensure the performance of each user and save more energy consumption. Subsequently, the LQR algorithm is used to track the desired trajectory to mitigate the performance degradation introduced by the UAV dynamics constraints.



**Figure 10.** Comparisons between UAV desired and actual trajectories (actual: purple, desired: red).



**Figure 11.** Comparisons of UAV velocity (actual: purple, desired: red).

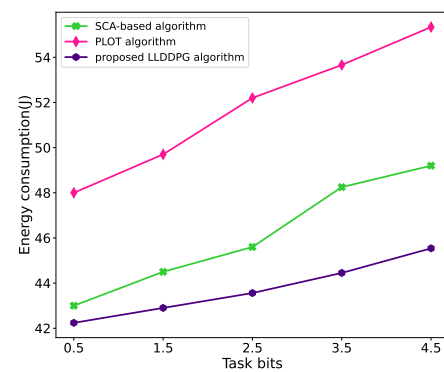


**Figure 12.** Comparisons of UAV acceleration (actual: purple, desired: red).

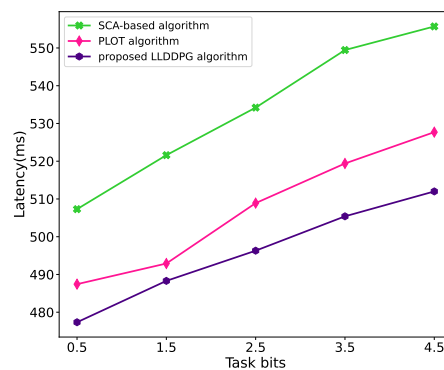
A rigorous trajectory analysis reveals that the initial alignment between the actual and desired trajectories is apparent. However, as the desired trajectory is formulated based

on the user's initial task size, it gradually diverges from the actual trajectory, which can be dynamically replanned in accordance with the evolving user task size and the current UAV flight state. Consequently, the deviation between the UAV's actual and desired trajectories translates into an increasing gap. To address this, the proposed LLDDPG algorithm demonstrates its proficiency in dynamically adjusting the flight trajectory in real-time, taking into account both the user's task requirements and the UAV's current flight status, ultimately enhancing the overall performance.

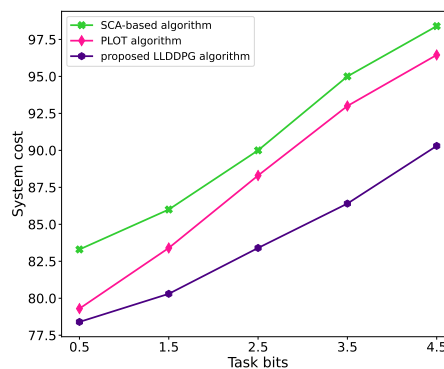
Figure 13 depicts the performance comparisons, including energy consumption, latency, and system cost, against existing optimization algorithms [24,25]. In [24], by leveraging the TSP with neighborhood and convex optimization techniques, an SCA-based algorithm is proposed to address the UAV trajectory planning problem, while the work in [25] proposed a PLOT control algorithm to maximize the aggregate execution of local and offloading tasks.



(a) Energy consumption



(b) Latency



(c) System cost

Figure 13. Comparison with other algorithms.

As depicted in Figure 13, it is revealed that, with the increasing task bit size, the energy consumption of the GCO algorithm escalates significantly greater than the others.

Meanwhile, the TSP algorithm exhibits a notably higher latency compared to the other two algorithms. In contrast, the proposed LLDDPG algorithm achieves the lowest total system cost in the UAV-assisted MEC system, effectively optimizing the total weighted energy consumption and delay of the system.

## 6. Conclusions

This work focus on the intricate issue of resource allocation and real-time trajectory control for a UAV-assisted MEC system operating in a partial offloading mode. Through the joint optimization of CPU frequency, offload ratio, and UAV trajectory, the minimization of the weighted average energy consumption and delay is achieved. In particular, to address the trajectory planning problem, the DDPG and LQR algorithms are employed together to realize the real-time control for actual UAV flight. For the computation resource allocation problem, as a convex problem, a low-complexity Lagrange duality method is proposed to derive the optimal expressions for CPU frequency and offload ratio. Finally, the efficacy of the proposed LLDDPG algorithm is comprehensively evaluated through simulations and numerical results.

**Author Contributions:** Conceptualization, Z.W., W.Z., P.H. and C.F.; methodology, Z.W., W.Z., P.H., X.Z., L.L. and C.F.; software, Z.W., W.Z., X.Z., L.L. and Y.S.; validation, Z.W., W.Z., P.H., X.Z., L.L., C.F. and Y.S.; formal analysis, W.Z., P.H., X.Z. and C.F.; investigation, W.Z., X.Z. and C.F.; data curation, W.Z., X.Z. and Y.S.; writing—original draft preparation, Z.W., W.Z., P.H., L.L., C.F. and Y.S.; writing—review and editing, Z.W., W.Z., P.H., X.Z., L.L., C.F. and Y.S.; project administration, Z.W. and P.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China under Grant 62371014, Beijing Natural Science Foundation 4222002, Urban Carbon Neutral Science and Technology Innovation Fund Project of Beijing University of Technology 040000514122607, and Special Research Program of Academic Cooperation between Taipei University of Technology and Beijing University of Technology NTUT-BJUT-112-02.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor. Syst.* **2015**, *17*, 2347–2376. [\[CrossRef\]](#)
2. Alwarafy, A.; Al-Thelaya, K.A.; Abdallah, M.; Schneider, J.; Hamdi, M. A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Int. Things J.* **2020**, *8*, 4004–4022. [\[CrossRef\]](#)
3. Hu, Z.; Zhong, R.; Fang, C.; Liu, Y. Caching-at-STARS: The next generation edge caching. *IEEE Trans. Wire. Commun.* **2024**, *early access*. [\[CrossRef\]](#)
4. Fang, C.; Hu, Z.; Meng, X.; Tu, S.; Wang, Z.; Zeng, D.; Ni, W.; Guo, S.; Han, Z. Drl-driven joint task offloading and resource allocation for energy-efficient content delivery in cloud-edge cooperation networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 16195–16207. [\[CrossRef\]](#)
5. Hou, Y.; Wang, C.; Zhu, M.; Xu, X.; Tao, X.; Wu, X. Joint allocation of wireless resource and computing capability in MEC-enabled vehicular network. *China Commun.* **2021**, *18*, 64–76. [\[CrossRef\]](#)
6. Baccour, E.; Mhaisen, N.; Abdellatif, A.A.; Erbad, A.; Mohamed, A.; Hamdi, M.; Guizani, M. Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence. *IEEE Commun. Surv. Tutor. Syst.* **2022**, *24*, 2366–418. [\[CrossRef\]](#)
7. Porambage, P.; Okwuibe, J.; Liyanage, M.; Ylianttila, M.; Taleb, T. Survey on multi-access edge computing for internet of things realization. *IEEE Commun. Surv. Tutor. Syst.* **2018**, *20*, 2961–2991. [\[CrossRef\]](#)
8. Yuan, S.; Zhang, Z.; Li, Q.; Li, W.; Zhang, Y. Joint optimization of dnn partition and continuous task scheduling for digital twin-aided mec network with deep reinforcement learning. *IEEE Wire. Commun.* **2023**, *11*, 27099–27110. [\[CrossRef\]](#)
9. Nadeem, L.; Azam, M.A.; Amin, Y.; Al-Ghamdi, M.A.; Chai, K.K.; Khan, M.F.N.; Khan, M.A. Integration of D2D, network slicing, and MEC in 5G cellular networks: Survey and challenges. *IEEE Access* **2021**, *9*, 37590–37612. [\[CrossRef\]](#)

10. Mehrabi, M.; You, D.; Latzko, V.; Salah, H.; Reisslein, M.; Fitzek, F.H. Device-enhanced MEC: Multi-access edge computing (MEC) aided by end device computation and caching: A survey. *IEEE Access* **2019**, *7*, 166079–166108. [\[CrossRef\]](#)
11. Lin, Z.; Lin, M.; De Cola, T.; Wang, J.B.; Zhu, W.P.; Cheng, J. Supporting IoT with rate-splitting multiple access in satellite and aerial-integrated networks. *IEEE Int. Things J.* **2021**, *8*, 11123–11134. [\[CrossRef\]](#)
12. Zhao, C.; Liu, J.; Sheng, M.; Teng, W.; Zheng, Y.; Li, J. Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach. *IEEE J. Sel. Area Commun.* **2021**, *39*, 3193–3207. [\[CrossRef\]](#)
13. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-aided hybrid satellite-terrestrial relay networks: Joint beamforming design and optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [\[CrossRef\]](#)
14. Niu, H.; Chu, Z.; Zhou, F.; Zhu, Z.; Zhen, L.; Wong, K.K. Robust design for intelligent reflecting surface-assisted secrecy SWIPT network. *IEEE Trans. Wire. Commun.* **2021**, *21*, 4133–4149. [\[CrossRef\]](#)
15. Yang, M.; Jeon, S.W.; Kim, D.K. Optimal trajectory for curvature-constrained UAV mobile base stations. *IEEE Wire. Commun. Lett.* **2020**, *9*, 1056–1059. [\[CrossRef\]](#)
16. Guo, F.; Zhang, H.; Ji, H.; Li, X.; Leung, V.C. Joint trajectory and computation offloading optimization for UAV-assisted MEC with NOMA. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 1–6.
17. Pham, Q.V.; Zeng, M.; Ruby, R.; Huynh-The, T.; Hwang, W.J. UAV communications for sustainable federated learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3944–3948. [\[CrossRef\]](#)
18. Qin, X.; Song, Z.; Hou, T.; Yu, W.; Wang, J.; Sun, X. Joint optimization of resource allocation, phase shift and UAV trajectory for energy-efficient RIS-assisted UAV-enabled MEC systems. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 1778–1792. [\[CrossRef\]](#)
19. Wang, D.; Tian, J.; Zhang, H.; Wu, D. Task offloading and trajectory scheduling for UAV-enabled MEC networks: An optimal transport theory perspective. *IEEE Wire. Commun. Lett.* **2021**, *11*, 150–154.
20. Wang, L.; Zhou, Q.; Shen, Y. Computation efficiency maximization for UAV-assisted relaying and MEC networks in urban environment. *IEEE Trans. Green Commun. Netw.* **2021**, *7*, 565–578. [\[CrossRef\]](#)
21. Diao, X.; Guan, X.; Cai, Y. Joint offloading and trajectory optimization for complex status updates in UAV-assisted Internet of Things. *IEEE Int. Things J.* **2022**, *9*, 23881–23896. [\[CrossRef\]](#)
22. Hu, Z.; Zeng, F.; Fu, B.; Jiang, H.; Chen, H. Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1630–1645. [\[CrossRef\]](#)
23. Liu, B.; Wan, Y.; Zhou, F.; Wu, Q.; Hu, R.Q. Resource allocation and trajectory design for MISO UAV-assisted MEC networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4933–4948. [\[CrossRef\]](#)
24. Zeng, Y.; Xu, J.; Zhang, R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wire. Commun.* **2019**, *18*, 2329–2345. [\[CrossRef\]](#)
25. Yang, Z.; Bi, S.; Zhang, Y.J.A. Dynamic offloading and trajectory control for UAV-enabled mobile edge computing system with energy harvesting devices. *IEEE Trans. Wire. Commun.* **2022**, *21*, 10515–10528. [\[CrossRef\]](#)
26. Yan, J.; Liu, Z.; Chen, C.; Zhao, J. Dynamic tracking method for landing trajectory of power line patrol UAV Based on Chaos Genetic Algorithm. In Proceedings of the 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Chongqing, China, 9–11 July 2021; pp. 301–304.
27. Lee, S.H.; Kang, S.H.; Kim, Y. Trajectory tracking control of quadrotor UAV. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017; pp. 281–285.
28. Li, B.; Li, Q.; Zeng, Y.; Rong, Y.; Zhang, R. 3D trajectory optimization for energy-efficient UAV communication: A control design perspective. *IEEE Trans. Wire. Commun.* **2021**, *21*, 4579–4593. [\[CrossRef\]](#)
29. Zhang, M.; Wu, S.; Jiao, J.; Zhang, N.; Zhang, Q. Energy- and Cost-Efficient Transmission Strategy for UAV Trajectory Tracking Control: A Deep Reinforcement Learning Approach. *IEEE Int. Things J.* **2022**, *10*, 8958–8970. [\[CrossRef\]](#)
30. Liu, K.; Zheng, J. UAV trajectory optimization for time-constrained data collection in UAV-enabled environmental monitoring systems. *IEEE Int. Things J.* **2022**, *9*, 24300–24314. [\[CrossRef\]](#)
31. Zhou, F.; Wu, Y.; Hu, R.Q.; Qian, Y. Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. *IEEE J. Sel. Area Commun.* **2018**, *36*, 1927–1941. [\[CrossRef\]](#)
32. Qian, Y.; Wang, F.; Li, J.; Shi, L.; Cai, K.; Shu, F. User association and path planning for UAV-aided mobile edge computing with energy restriction. *IEEE Wire. Commun. Lett.* **2019**, *8*, 1312–1315. [\[CrossRef\]](#)
33. Wang, Z.; Gao, Y.; Fang, C.; Liu, L.; Zhou, H.; Zhang, H. Optimal control design for connected cruise control with stochastic communication delays. *IEEE Trans. Veh. Technol.* **2020**, *69*, 15357–15369. [\[CrossRef\]](#)
34. Zeng, Y.; Zhang, R. Energy-efficient UAV communication with trajectory optimization. *IEEE Trans. Wire. Commun.* **2017**, *16*, 3747–3760. [\[CrossRef\]](#)
35. Zhang, M.; Zhang, Y.; Gao, Z.; He, X. An improved DDPG and its application based on the double-layer BP neural network. *IEEE Access* **2020**, *8*, 177734–177744. [\[CrossRef\]](#)
36. Wang, Z.; Jin, S.; Liu, L.; Fang, C.; Li, M.; Guo, S. Design of intelligent connected cruise control with vehicle-to-vehicle communication delays. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9011–9025. [\[CrossRef\]](#)

37. Zhou, F.; Wu, Y.; Sun, H.; Chu, Z. UAV-enabled mobile edge computing: Offloading optimization and trajectory design. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
38. Yu, Y.; Tang, J.; Huang, J.; Zhang, X.; So, D.K.C.; Wong, K.K. Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm. *IEEE Trans. Commun.* **2021**, *69*, 6361–6374. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.