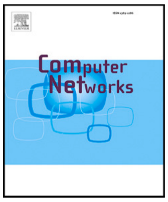
Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



TD3-based trajectory optimization for energy consumption minimization in UAV-assisted MEC system

Fanfan Shen [a], Bofan Yang [a,*], Jun Zhang [b], Chao Xu [a], Yong Chen [a], Yanxiang He [c]

[a] School of Computer Science, Nanjing Audit University, Nanjing 211815, China
[b] College of Software, East China University of Science and Technology, Nanchang 330013, China
[c] School of Computer Science, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Keywords:
PDPSO-TD3
Trajectory optimization
Energy consumption
UAV

ABSTRACT

Unmanned Aerial Vehicle (UAV) assisted Mobile Edge Computing (MEC) systems provide substantial benefits for task offloading and communication services, especially in situations where traditional communication infrastructure is unavailable. Current research emphasizes maintaining communication quality while minimizing total energy consumption and optimizing UAV flight trajectories. However, several issues remain: First, the energy consumption objective function lacks comprehensiveness, neglecting the impact of UAV flight energy consumption; second, an effective Deep Reinforcement Learning (DRL) algorithm has not been employed to address the non-convexity of the objective function; third, there is insufficient discussion regarding the practical significance of the proposed approach. To address these issues, this paper formulates an objective function aimed at minimizing MEC energy consumption by considering task offloading decisions, communication delays, computational energy consumption, and UAV flight energy consumption. We propose a Population Diversity-based Particle Swarm Optimization-Double Delay Deep Deterministic Policy Gradient (PDPSO-TD3) algorithm to find the optimal solution, enhance UAV flight trajectories through optimized offloading decisions, ensure efficient communication, and minimize the total energy consumption of the MEC system. Furthermore, we discuss the practical applicability of PDPSO-TD3 in detail and present the proposed scheme. Experimental results demonstrate that compared to the Deep Deterministic Policy Gradient (DDPG) algorithm, for transmission delay, MEC energy consumption, UAV flight energy consumption, and User Equipments (UEs) access rate metrics. The proposed PDPSO-TD3 algorithm can improvement the performance by about 14.3%, 10.1%, 6.1%, and 3.3%, respectively.

1. Introduction

When processing latency-sensitive tasks such as road traffic monitoring, intelligent vehicle navigation and virtual reality [1], mobile devices often struggle to maintain low power consumption and low latency due to limitations in computational resources and size [2]. For this reason, Mobile Edge Computing (MEC) has been proposed as a solution. Specifically, MEC achieves the goal of reducing transmission latency by offloading tasks from User Equipments (UEs) to edge servers. In addition, MEC effectively reduces the proportion of discarded tasks in time-sensitive operations, thereby improving the Quality of Service (QoS) for users [3].

However, in emergency situations such as natural disasters, the infrastructure communication facilities may collapse, which poses a challenge to the stability of MEC services [4,5]. Fortunately, Unmanned Aerial Vehicle (UAV) has achieved significant advancements in diverse applications, including post-disaster emergency rescue, search operations, oil and gas field mapping, and critical data acquisition [6,7]. The authors in [8] proposed the utilization of UAV as a “mobile aerial base station” to provide emergency communication support when conventional communication facilities fail. Compared with MEC that rely on ground Base Stations (BSs), UAV can extend wireless coverage, such as Line of Sight (LoS) angles, quickly collect sensitive data, and support the efficient offloading of computation-intensive tasks in MEC [9–11].

Based on the above research, we are considering using UAV as mobile BSs to assist with MEC communication when the communication system is obstructed. Our goal is to minimize the energy consumption and transmission delay, and to ensure the reliability of the communication quality between UAV and UEs. Therefore, adaptive UAV selection based on dynamic path planning is an important issue.

[*] Corresponding author.
E-mail addresses: ffshen@whu.edu.cn (F. Shen), MP2209118@stu.nau.edu.cn (B. Yang), zhangjun_whu@whu.edu.cn (J. Zhang), xuchao@nau.edu.cn (C. Xu), chenyong@nau.edu.cn (Y. Chen), yxhe@whu.edu.cn (Y. He).

https://doi.org/10.1016/j.comnet.2024.110882
Received 18 July 2024; Received in revised form 21 October 2024; Accepted 27 October 2024
Available online 1 November 2024

When a UAV acts as relay BSs for communication services, it must fly to each terminal to provide coverage. However, in practical situations, the UAV is unable to ascertain the actual location of the UEs in advance because the location of the UEs are random and not fixed. In addition, the communication rate is also affected by the distance between the UAV and UEs. Therefore, autonomous real-time path planning and dynamic adaptation to random UEs are major challenges.

Deep Reinforcement Learning (DRL) combines the complex feature extraction capabilities of Deep Learning (DL) with the adaptive learning mechanisms of Reinforcement Learning (RL), which can control the agent running in a complex and dynamic environment, and shows excellent performance in multi-dimensional continuous action space [12]. Therefore, DRL has been applied to solve the autonomous path planning problem of UAV [13,14].

In [15], the authors proposed a TD3-BC-PPO algorithm to address the problem of dynamic optimal design under adversarial real-time conditions, but it did not cover the MEC environment. Some works in [16–19] indicated that in a system with multiple UAV supporting MEC, algorithms such as Double Q-Learning (DQL), Multi-Agent Deep Deterministic Policy Gradient (MADDPG), and Multi-Agent Double Delay Deep Deterministic Policy Gradient (MATD3) can do well in collaboratively optimizing UAV trajectories and providing computational services. In [20], the authors used two Deep Q-Network (DQN) networks to solve the UAV trajectory constraint optimization problem. In [21], the authors put forth a joint algorithmic approach that employs the Population Diversity-based Particle Swarm Optimization (PDPSO) algorithm to optimize the task offloading strategy and the Deep Deterministic Policy Gradient (DDPG) algorithm to identify the optimal UAV trajectory. It is important to note that the aforementioned studies are based on the expansion of DQL, DDPG and DQN algorithms. In comparison to Double Delay Deep Deterministic Policy Gradient (TD3), the algorithms themselves present certain issues, including insufficient stability and overestimation, which may result in convergence to local optima and impair the learning ability and exploration performance.

In this paper, we innovatively propose the PDPSO-TD3 algorithm. The trajectory of the UAV is dynamically optimized based on the optimal task offloading strategy, aiming to minimize both energy consumption and transmission latency. To the best of our knowledge, this is the first instance where the PDPSO algorithm and the TD3 algorithm are combined, resulting in excellent performance. Furthermore, we specifically take into account the impact of UAV flight power consumption on the total energy consumption of the MEC system, which has been largely overlooked in previous studies. The main contributions of this paper are summarized as follows:

1. **Task offloading on the MEC system:** We address the cooperative task offloading strategy for UAV-assisted MEC systems, which integrates on-device computation by UEs and UAV-supported partial task processing coupled with offloading. The strategy aims to minimize the anticipated long-term task execution cost, factoring in the temporal disparity between the UAV partial computation and the UEs local task processing and service offloading. Through this optimization, we ascertain the optimal task offloading ratio, thereby formulating the most efficient task offloading strategy.
2. **UAV flight energy consumption model is proposed:** We investigate the effect of UAV flight energy consumption on the total energy consumption of MEC systems, incorporating this parameter into our objective function. Utilizing DRL, we solve the ensuing non-convex optimization problem, thereby reducing MEC energy consumption and concurrently optimizing UAV flight paths.
3. **The PDPSO-TD3 algorithm is innovatively proposed:** The trajectory of the UAV is dynamically updated in real time by our algorithm, based on the optimal task offloading strategy, ensuring effective coverage of UEs within the UAV's communication range. Our algorithm possesses two distinct features, firstly, it utilizes the PDPSO algorithm to determine the optimal offloading rate for decision-making regarding task offloading. Secondly, we enhance the TD3 algorithm to achieve optimal trajectory design for the UAV in a continuous action space, thereby enhancing real-time communication quality.
4. **Real-world applicability:** We specifically address the real-world applicability of the PDPSO-TD3 algorithm, presenting a proposed scheme. This includes identifying relevant real-world scenarios, detailing the application methodologies, and discusses the expected practical implications. These aspects have been largely neglected in the existing literature.

The rest of this paper is structured as follows: Section 2 reviews the related work. Section 3 describes the system model and problem formulation. Section 4 details the PDPSO-TD3 algorithm. Section 5 discusses the algorithm's real-world applications. Section 6 provides experimental results and analysis. Section 7 concludes and suggests future research directions.

2. Related works

The integration of UAV, MEC, and DRL has been the subject of extensive study in a variety of application scenarios, including data collection [22], collision avoidance [23], target tracking [24], resource management [25], and task offloading [26]. In this study, we focus on the trajectory optimization method, task offloading scheme, energy loss and transmission delay of UAV assisted MEC, and divide the research work in related fields into two categories: resource allocation and trajectory optimization.

2.1. Resource allocation

The first category is focused on the implementation of offloading strategies and system architectures designed to reduce energy consumption and transmission latency across the system. In [27], the authors conducted an exhaustive analysis of the computational time and energy expenditure associated with task offloading and proposed a pragmatic strategy to minimize these costs. Subsequent works, as detailed in [28,29], introduced an alternative iterative scheme that employs block descent to refine task offloading decisions on MEC servers, thereby optimizing a cost function that accounts for both energy consumption and transmission delay. In [30], the authors focused on a MISO UAV-assisted MEC network and presented a three-stage iterative algorithm designed to reduce the overall energy usage of the MEC system. Further advancements were made in [31], where the authors addressed a heterogeneous MEC system, with the objective of enhancing energy utilization efficiency through the joint optimization of UAV trajectories and computational resource distribution. In [32], a two-stage UAV operational mode was introduced to effectively manage queued tasks, rendering resource allocation and trajectory planning more efficient while simultaneously reducing energy usage. In [33], the authors proposed an iterative algorithm to streamline UAV trajectories and to adeptly manage the scheduling of computational resources. In [34], the authors implemented dynamic orchestration of the ground network, designing a more flexible framework for terrestrial networks that achieves high-quality network communication while reducing system energy consumption and optimizing UAV flight trajectories. In [35], the authors proposed a dynamic grouping and re-orchestration strategy for vehicular networks, where the latency model for ground networks can process data in cloud or edge servers. In future research on UAV-assisted MEC, this model can be utilized to more effectively evaluate and optimize communication latency.

Despite the contributions of previous studies, they have not yet explored the potential of DRL in addressing non-convex optimization challenges. It is widely recognized that DRL offers significant advantages in the realm of non-convex optimization, particularly when the

Table 1
Comparison between our work and the existing literature.

References	Offload decision	Communication latency	Computational energy consumption	UAV flight energy consumption	UAV trajectory optimization	DRL algorithm update	Real-world applications
[15]	✓					✓	✓
[16]	✓	✓	✓	✓	✓	✓	
[17]	✓	✓	✓	✓	✓	✓	
[18]	✓	✓	✓	✓	✓	✓	
[19]	✓	✓	✓		✓	✓	
[20]	✓		✓		✓	✓	
[21]	✓	✓	✓		✓	✓	
[27]	✓	✓	✓				
[28]	✓	✓	✓		✓		
[29]	✓	✓	✓		✓		
[30]	✓	✓	✓		✓		
[31]	✓	✓	✓	✓	✓		
[32]	✓	✓	✓		✓		
[33]	✓	✓	✓		✓		
[34]	✓	✓	✓	✓	✓		✓
[36]	✓	✓	✓			✓	
[37]	✓	✓	✓	✓	✓	✓	
[38]	✓	✓	✓			✓	
[39]	✓				✓	✓	✓
[40]	✓	✓	✓		✓	✓	
[41]	✓		✓		✓	✓	
[42]	✓		✓		✓	✓	
[43]	✓		✓		✓	✓	
[44]	✓		✓	✓	✓	✓	
Our work	✓	✓	✓	✓	✓	✓	✓

objective is to achieve global optimal, such as in scenarios involving MEC resource allocation and the trajectory optimization of UAV. In contrast to traditional approaches that may be satisfied with local optima, DRL can learn through interaction with the environment and find policies that are either globally optimal or near-optimal. In [36], the authors proposed an online task offloading framework based on DRL, which significantly reduced the computational complexity while making optimal offloading decisions. In [37], the authors studied a DRL-based UAV task offloading network, which enabled priority-based task allocation, UAV power preservation, and trajectory optimization. The authors of [38] proposed an ICRA method for the UANETs in a UAV-assisted maritime monitoring system. By employing a DRL algorithm to determine the optimal clustering strategy, they effectively reduced the transmission latency and enhanced both energy efficiency and service quality.

2.2. UAV trajectory optimization

The second category of investigation focuses on optimizing UAV trajectories to reduce task computation latency and MEC system energy consumption, while also enhancing communication quality. In [39], the authors enhanced the DQN algorithm for flexible deployment on UAV and MEC platforms, enabling the planning of UAV flight trajectories that avoided collisions and optimized energy conservation. The researchers in [40] proposed a two-tiered algorithmic framework. The first tier employed the Lagrange multiplier method to address the task offloading allocation, while the second tier utilized DRL to address the UAV trajectory optimization, thereby reducing the total energy consumption of the MEC system. In [41], the authors explored the UAV-assisted MEC wireless charging scenario, with the goal of maximizing UAV energy utilization efficiency through the strategic optimization of UAV trajectories. Additionally, [42] investigated the design of UAV trajectories within complex 3D environments, aiming to reduce the costs associated with mission decision-making. It is important to note that the aforementioned studies overlooked the impact of UAV flight power consumption on the MEC system. This oversight could lead to an uneven distribution of computational resources and a potential decline in the quality of communication services.

In [43], the authors proposed a trajectory control algorithm based on DRL with the objective of reducing UAV flight power consumption in dynamic environments and facilitating real-time decision-making, thereby optimizing the utilization of the UAV's limited computational resources. In [44], the authors considered the impact of UAV flight power consumption on multi-UAV-assisted MEC systems and proposed a JDPB algorithm to address offloading decisions, resource allocation, and UAV trajectory planning. However, when formulating task offloading strategies, the impact of computation energy consumption and communication delay should be comprehensively considered. This includes evaluating the flight power consumption of the UAV, optimizing its flight trajectory, and solving the non-convex optimization problem using DRL. Currently, the studies in these areas has not yet formed a comprehensive framework.

2.3. Literature summary

In this section, we explore the main differences between this study and the existing literature. Firstly, it is noted that the studies cited in Refs. [19–21,28–30,32,33], and [39–43] have not incorporated UAV flight energy consumption within their energy consumption models. In contrast, the present study integrates task offloading decisions, communication delays, computational energy consumption, and UAV flight energy consumption into a more comprehensive objective function, which is designed to minimize the energy consumption associated with MEC. Secondly, with respect to the challenge posed by non-convex objective functions, the extant literature [27–34] has not employed a more appropriate DRL algorithm, potentially leading to a propensity for the algorithm to converge to local optima. To mitigate this limitation, we introduce an algorithm, denoted as PDPSO-TD3, which is presented herein for the first time. In comparison with existing algorithms, PDPSO-TD3 achieves a better balance between the convergence speed of the algorithm and the quality of the solution, thereby enabling the identification of the global optimal solution more effectively. Finally, this study places particular emphasis on the practical applicability of the PDPSO-TD3 algorithm and proposes a corresponding scheme. This aspect, which is frequently overlooked in the extant research, is deemed to be of paramount importance for enhancing the practical applicability and expediting the industrialization process of pertinent technologies. Table 1 summarizes the differences between our work and the existing literature.

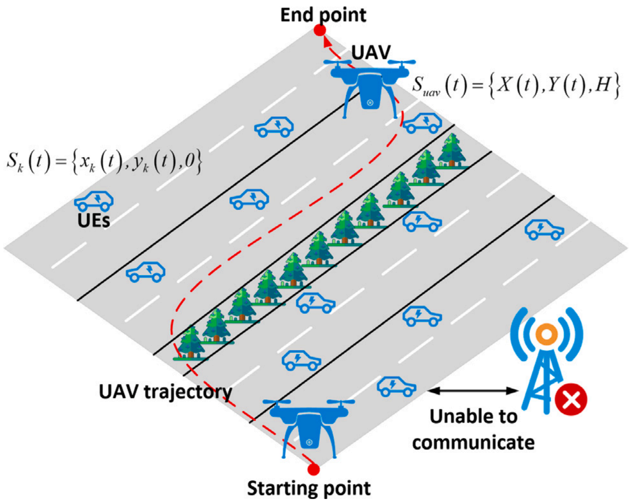


Fig. 1. MEC system model.

3. System model and problem formulation

It is well known that new energy vehicles depend on real-time communication and data exchange with the network for tasks including autonomous route planning, vehicle monitoring, and application operation. These services are highly sensitive to network delays. In the event of a communication base station failure, UAV serve as mobile BSs to provide communication services. As shown in Fig. 1, in a scenario where communication is disrupted along a road section, a UAV flies from the starting point to the end point to offer emergency communication services to K UEs. The positions of both the UAV and the UEs are specified in 3D coordinate space, with the UEs' locations and the data volumes they process being subject to randomness and uncertainty. Due to the real-time requirements of communication, some tasks necessitate immediate local response computation. Furthermore, some data may involve user privacy or security issues, necessitating local computation rather than offloading to the UAV for assisted computation. This indicates that the task offloading decision will be divided into two types: either UEs compute all tasks locally, or they offload some tasks to the UAV for assisted computation, ensuring task timeliness, safety, and stability.

The total system time slot is set to T, each time slot is χ in length, and there are $\frac{T}{\chi}$ time slots of equal length. In tth time slot, we define the position of the K UEs as $S_k(t) = \{x_k(t), y_k(t), 0\} \in \mathbb{R}^{1\times 3}, k = 1, 2, \ldots\ldots, K$. The UAV position is determined as $S_{uav}(t) = \{X(t), Y(t), H\}$. Given the relatively flat road surface and the lack of obstacles or structures that could obstruct wireless communication, the UAV is configured to maintain a constant altitude H. The UAV must determine its flight trajectory, including direction o_t and velocity V_t, at each time slot. The UAV's maximum speed is represented by $V_{\max}$, and its positional coordinates are consequently updated as follows:

$$S_{uav}(t+1) = \begin{cases} X(t+1) = X(t) + \sin\left(o_t \times \pi\right) \times V_t \\ Y(t+1) = Y(t) + \cos\left(o_t \times \pi\right) \times V_t \end{cases} \tag{1}$$

The UAV communicates with only one UE in a single time slot, which is to better observe the UAV flight decision in each time slot. Considering the realistic scenario, the UAV and UEs are moving and bounded by a fixed area $S_{uav}(t), S_k(t) \leqslant \{X_{size}, Y_{size}, H\}$. The intensive task that requires computational processing for K UEs is denoted as $L_k(t), k = 1, 2, \ldots\ldots, K$.

In MEC systems, the provisioning of computing and offloading services for resource-intensive tasks mainly involves two fundamental components: computation and communication. Furthermore, we specifically examine the impact of UAV flight power consumption on the overall energy consumption of the MEC system. In the subsequent sections, we will detail these components in subsections, define the problem, and propose our solution. For clarity, Table 2 provides a summary of the main notations used throughout this paper.

Table 2
Main notations.

Notation	Definition
K	The count of UEs
T	Total duration of the MEC system
χ	The length of each time slot
$S_k(t)$	Coordinate position of UEs in time slot
$S_{uav}(t)$	Coordinate position of UAV in time slot
V_t, $V_{\max}$	UAV flight speed and maximum speed constraint
o_t	The flight direction of the UAV
$L_k(t)$	Task data size generated by the UEs
$h_k(t)$	The channel gain between the UAV and UEs
B_u	Uplink bandwidth
β_0	The channel gain at a distance of 1 m
δ_0^2	The noise power generated by UAV communication
$\varphi_k(t)$	Task offloading strategy
P_{user}	The communication transmission power of UEs
$I_k(t)$	Communication link validity
$T_k^{Tra}(t)$	Transmission delay caused by partial task offloading
$\alpha_k(t)$	Task offloading ratio
$E_k^{Tra}(t)$	Energy consumption generated by communication
P_{uav}	Transmit power of the UAV.
$T_{k,loc}^{Com}(t)$	The delay for UEs local computation
f_{user}	The local computing resources of UEs
C_{user}	The CPU computation cycles for UEs when performing local
$T_{k,par}^{Com}(t)$	The delay caused by the UEs executing partial computations.
$T_{k,uav}^{Com}(t)$	The delay caused by UAV assisted computing
C_{uav}	The CPU computation cycles required for the UAV
f_{uav}	The computing resources of UAV
$E_{k,loc}^{Com}(t)$	Energy consumption from local computation by the UEs.
K_{user}	The CPU capacitance index of UEs
$E_{k,par}^{Com}(t)$	Energy consumption incurred during UEs partial computation
K_{uav}	The CPU capacitance index of UAV
$E_{k,uav}^{Com}(t)$	Energy consumption incurred during UAV task computation.
$T_{par}(t)$	Partial tasks compute the total delay incurred.
$E_{par}(t)$	Total energy consumption from partial task computation.
$T(t)$	Total transmission delay of MEC.
$E(t)$	Total energy consumption of MEC.
$E_{uav}(t)$	The flight energy consumption of the UAV in tth time
ω	Calculate the weights used
ϑ	UAV flight energy consumption weight

3.1. Communication model

When the BSs fail to provide reliable communication service, the UAV is required to function as the mobile BSs for communicating with UEs, and the channel gain $h_k(t)$ between them is determined based on real-time positioning. We do not consider the downlink transmission here, that is [29]:

$$h_k(t) = \frac{\beta_0}{H^2 + \left\|S_{uav}(t) - S_k(t)\right\|^2} \tag{2}$$

$\left\|S_{uav}(t) - S_k(t)\right\|$ denotes the spatial separation between the UAV and UEs within a 3D continuous action space, while β_0 represents the channel gain based on the initial distance, which we have set as 1 m in this context.

Uplink bandwidth B_u required by UEs is uniformly allocated when the UAV, serving as an aerial base station, engages in communication with UEs to facilitate offloading of computationally intensive tasks. Consequently, the data transmission rate $r_k(t)$ can be defined as follows [24]:

$$\mathrm{r}_k(t) = \frac{B_u}{K}\log_2\left(1 + \frac{P_{\mathrm{user}}h_k(t)}{\delta_0^2}\right) \tag{3}$$

The UAV communication is affected by noise interference, with a noise power is δ_0^2. UEs' communication transmission power is P_{user}.

To establish efficient communication links $I_k(t)$ with UEs and ensure timely offloading transmission of all computing tasks $L_{\mathrm{k}}(t)$ generated by UEs within a single time slot, we define the minimum uplink rate κ as follows:

$$\kappa = \frac{L_{\mathrm{k}}(t)}{\chi} \tag{4}$$

Effective communication between UAV and UEs can only be achieved if the minimum communication requirements are met. Therefore, $I_k(t) \in \{0,1\}$, where 1 indicates a successful establishment of communication link, while the opposite indicates communication failure.

$$I_k(t) = \begin{cases} 1, r_k(t) > \kappa \\ 0, otherwise \end{cases} \tag{5}$$

The binary representation is utilized to express our task offloading policy $\varphi_k(t)$. When $\varphi_k(t) = 1$, UEs offload a portion of the data to UAV for assisted computing. It is important to note that when $\varphi_k(t) = 0$, the task data is computed entirely locally on UEs without any interaction with UAV.

$$\varphi_k(t) = \begin{cases} 1, & partial\ offload \\ 0, & total\ local \end{cases} \tag{6}$$

When $\varphi_k(t) = 1$, UEs will offload a portion of their computing tasks to UAV, resulting in the transmission delay between them as stated below:

$$T_{\mathrm{k}}^{Tra}(t) = \frac{\alpha_k(t)L_k(t)\varphi_k(t)}{r_k(t)} \tag{7}$$

In the tth time slot, UEs locally generate tasks $L_k(t)$. To facilitate auxiliary computation, $L_k(t)$ needs to allocate a portion of these tasks to UAV at an allocation ratio of $\alpha_k(t)$.

The total energy consumption resulting from the communication between UAV and UEs can be inferred as follows:

$$E_{\mathrm{k}}^{Tra}(t) = P_{uav}T_{\mathrm{k}}^{Tra}(t) = \frac{P_{uav}\alpha_k(t)L_k(t)\varphi_k(t)}{r_k(t)} \tag{8}$$

where P_{uav} represents transmit power of the UAV.

3.2. Computation model

Task computation consists of two components: the local computation performed by UEs and the involvement of UAV in executing a portion of the computational tasks to support UEs. Local computation refers to UEs independently handling all the computation tasks, i.e., when the offloading strategy is 0. Correspondingly, when the offloading strategy is 1, it means UAV assumes a portion of the computational tasks from UEs, the delay for UEs entirely local computation as [18]:

$$T_{k,loc}^{Com}(t) = \frac{\left(1-\varphi_k(t)\right)L_k(t)C_{user}}{f_{user}} \tag{9}$$

We define the limited local computing resources of UEs as f_{user}. The number of CPU cycles expended by UEs to process 1 bit of data is defined as C_{user}. The latency generated by UEs undertaking partial computational tasks is:

$$T_{k,par}^{Com}(t) = \frac{\left(1-\alpha_k(t)\right)L_k(t)\varphi_k(t)C_{user}}{f_{user}} \tag{10}$$

Similarly, UEs offload some computing tasks to UAV, and the delay caused by UAV assisted computing is as follows:

$$T_{k,uav}^{Com}(t) = \frac{\varphi_k(t)\alpha_k(t)L_k(t)C_{uav}}{f_{uav}} \tag{11}$$

where C_{uav} denotes the CPU cycles needed by UAV to process 1 bit of data, while f_{uav} refers to the local computational capabilities of the UAV. As such, we can deduce the computational energy consumption of UEs and UAV as [18]:

$$\begin{aligned} E_{k,loc}^{Com}(t) &= K_{user}\left(f_{user}\right)^3 T_{k,loc}^{Com}(t) \\ &= K_{user}\left(1-\varphi_k(t)\right)L_k(t)C_{user}\left(f_{user}\right)^2 \end{aligned} \tag{12}$$

$$\begin{aligned} E_{\mathrm{k,par}}^{Com}(t) &= K_{user}\left(f_{user}\right)^3 T_{k,par}^{Com}(t) \\ &= K_{user}\left(1-\alpha_k(t)\right)L_k(t)\varphi_k(t)C_{user}\left(f_{user}\right)^2 \end{aligned} \tag{13}$$

$$\begin{aligned} E_{k,uav}^{Com}(t) &= K_{uav}\left(f_{uav}\right)^3 T_{k,uav}^{Com}(t) \\ &= K_{uav}\varphi_k(t)\alpha_k(t)L_k(t)C_{uav}\left(f_{uav}\right)^2 \end{aligned} \tag{14}$$

where K_{user} and K_{uav} respectively represent the CPU capacitance indexes of the UEs and the UAV. It is important to acknowledge that the energy consumption of task offloading calculations increases with higher transmission delays, as these factors are intricately interconnected within the system model.

Therefore, the total energy consumption and transmission delay can be inferred when UAV assist UEs in task offloading and computation.

$$T_{par}(t) = \max\left(T_{\mathrm{k,par}}^{Com}(t), T_{\mathrm{k}}^{Tra}(t) + T_{k,uav}^{Com}(t)\right) \tag{15}$$

$$E_{par}(t) = E_{\mathrm{k}}^{Tra}(t) + E_{\mathrm{k,par}}^{Com}(t) + E_{k,uav}^{Com}(t) \tag{16}$$

Finally, we can derive the total energy consumption and transmission delay after all UEs complete their computational tasks.

$$E(t) = \sum_{k=1}^{K}\left[\left(1-\varphi_k(t)\right)E_{k,loc}^{Com}(t) + \varphi_k(t)E_{par}(t)\right] \tag{17}$$

$$T(t) = \sum_{k=1}^{K}\left[\left(1-\varphi_k(t)\right)T_{k,loc}^{Com}(t) + \varphi_k(t)T_{par}(t)\right] \tag{18}$$

3.3. UAV flight energy consumption model

Rotorcraft UAV primarily expending energy during flight, including hovering. When employed as a mobile MEC platform for UEs communication, the UAV hovers and transitions to a flying state when navigating towards the UEs. However, due to the limited onboard battery, the UAV flight speed is constrained, preventing it from maintaining peak velocity. Thus, precise modeling of its flight energy consumption is pivotal for effective trajectory planning. In [45], the author derived the flight power of a rotorcraft UAV at a speed of V, disregarding the impact of acceleration on flight energy consumption.

$$\begin{aligned} P_e(V) = {} & P_0\left(1+\frac{3V^2}{U_{tip}^2}\right) + P_i\left(\sqrt{1+\frac{V^4}{4V_0{}^2}} - \frac{V^2}{2V_0{}^2}\right)^{\frac{1}{2}} \\ & + \frac{1}{2}d_0\rho s A V^3 \end{aligned} \tag{19}$$

where P_0 represents the blade profile power of the UAV in hover, P_i represents the induced power, and V_0 represents the average rotor induced velocity, U_{tip}, d_0, ρ, s, and A represent the tip speed of the UAV rotor blade, fuselage drag ratio, air density, rotor solidity, and propeller disk area. Therefore, assuming that the UAV speed v_t remains constant during each time slot, we can derive the energy consumption within that time slot.

$$\begin{aligned} E_{uav}(t) = {} & \left(P_0\left(1+\frac{3V_t^2}{U_{tip}^2}\right) + P_i\left(\sqrt{1+\frac{V_t{}^4}{4V_0{}^2}} - \frac{V_t{}^2}{2V_0{}^2}\right)^{\frac{1}{2}}\right. \\ & \left. + \frac{1}{2}d_0\rho s A V_t{}^3\right) \times \chi \end{aligned} \tag{20}$$

The cumulative flight energy consumption of UAV in the whole time T is expressed as follows:

$$E_{total} = \sum_{t=1}^{T} E_{uav}(t) \tag{21}$$

3.4. Problem formulation

The total energy consumption of the MEC system includes communication energy consumption, task computation energy consumption, and UAV flight energy consumption. Our goal is to minimize the total energy consumption of the MEC system while planning the optimal path for the UAV.

$$\begin{aligned} P:\ & \min_{S_{uav}(t)} \omega\left[\sum_{t=1}^{T} E(t)\right] + (1-\omega)\left[\sum_{t=1}^{T} T(t)\right] + \vartheta E_{total} \\ s.t\quad & C1: S_{uav}(t), S_k(t) \in \left\{X_{size}, Y_{size}, H\right\}, k = 1, 2......, K \\ & C2: \varphi_k(t) \in \{0,1\} \\ & C3: \min\left[T_{par}(t)\right] \\ & C4: S_{uav}^{start} \to S_{uav}^{end} \\ & C5: I_k(t) = 1 \\ & C6: V_t \leqslant V_{\max} \end{aligned} \tag{22}$$

ω represents the weight of energy consumption for communication and task calculation, while ϑ represents the weight of energy consumption for UAV flight. C1 is the location constraint for UAV and UEs, which must be within the specified range. C2 is the task offloading strategy, C3 is the constraint for formulating the task offloading ratio, C4 represents the motion trajectory constraint of the UAV, C5 represents the communication constraint between the UAV and UEs, and C6 represents the flight speed constraint of the UAV.

Obviously, in C3, the total number of tasks that can be delegated by a user within a fixed time frame remains constant. Therefore, the larger the amount of tasks offloaded to the local computation, the smaller the amount of tasks that the corresponding UAV needs to compute. We can derive that $T_{k,par}^{Com}(t)$ is inversely proportional to $T_k^{Tra}(t)+T_{k,uav}^{Com}(t)$. If and only if $T_{k,par}^{Com}(t) = T_k^{Tra}(t) + T_{k,uav}^{Com}(t)$, $T_{par}(t)$ as a whole is minimized. Based on this, we can determine the optimal task offloading ratio $\alpha_k(t)$:

$$\begin{aligned} & \min\left[T_{par}(t)\right] \\ & = \min\left[\max\left(T_{\mathrm{k,par}}^{Com}(t), T_{\mathrm{k}}^{Tra}(t) + T_{k,uav}^{Com}(t)\right)\right] \end{aligned} \tag{23}$$

$$\Rightarrow T_{\mathrm{k,par}}^{Com}(t) = T_{\mathrm{k}}^{Tra}(t) + T_{k,uav}^{Com}(t) \tag{24}$$

$$\Rightarrow \frac{\left(1-\alpha_k(t)\right) L_k(t) C_{user}}{f_{user}} = \frac{\alpha_k(t) L_k(t)}{r_k(t)} + \frac{\alpha_k(t) L_k(t) C_{uav}}{f_{uav}} \tag{25}$$

$$\Rightarrow \alpha_k(t) = \frac{C_{user} f_{uav} r_k(t)}{\left(C_{uav} f_{user} + C_{user} f_{uav}\right) r_k(t) + f_{user} f_{uav}} \tag{26}$$

The above optimization objective is a non-convex optimization problem. When DRL algorithm solves such problems, it can solve the optimal policy through the environment interaction. Therefore, we propose the PDPSO-TD3 algorithm.

4. The proposed algorithm

In this section, we first introduce the PDPSO algorithm separately from the TD3 algorithm. The PDPSO algorithm is responsible for determining the optimal unloading strategy, while the TD3 algorithm is responsible for implementing the optimal path planning of UAV. Finally, we propose the PDPSO-TD3 algorithm.

4.1. PDPSO algorithm

We know that traditional Particle Swarm Optimization (PSO) algorithms are prone to falling into local optima and have low convergence accuracy. To address these issues, we use the PDPSO algorithm, which can effectively utilize the population diversity of particles to continuously adjust the inertia weight ϖ, effectively balancing the global exploration ability and local development ability of particle flight process, and avoiding falling into local optima.

The particle velocity v_i is iteratively updated after weighting ϖ, and the update process is based on the difference between the current particle's optimal solution $pbest$, the swarm's optimal solution $gbest$, and the particle's position x_i. This update incorporates learning factors (c_1, c_2) and random numbers ($rand_1$, $rand_2$) to enhance its effectiveness. Similarly, the particle position x_i is updated according to its velocity v_i.

$$\begin{aligned} v_i(t+1) = \varpi(t) v_i(t) + c_1 rand_1 & \left[pbest - x_i(t)\right] \\ + c_2 rand_2 & \left[gbest - x_i(t)\right] \end{aligned} \tag{27}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{28}$$

After the algorithm converges, the particles will be concentrated near the optimal solution and are basically in a static state. Therefore, we can determine the offloading strategy as follows:

$$Sig\left(v_i(t)\right) = \frac{1}{1+e^{-v_i(t)}} \tag{29}$$

$$\varphi_i(t) = \begin{cases} 1, rand_3 < Sig\left(v_i(t)\right) \\ 0, otherwise \end{cases} \tag{30}$$

$rand_3$ is a random number within the $[0,1]$ range, $\varphi_i(t) = 1$ represents the partial task offloading strategy selected at time slot t, which requires UAV assistance for calculation. Similarly, $\varphi_i(t) = 0$ represents the local calculation of all tasks selected at time slot t.

The update iteration of particle velocity v_i and position x_i is intricately linked to the inertia weight ϖ. In order to determine the appropriate inertia weight ϖ, it is necessary to first optimize the diversity within the population D.

$$D(t+1) = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}\left(d_i^{Ave}(t) - d_i^{Min}(t)\right)^2} \tag{31}$$

We represent the number of UEs by the quantity of particles, denoted as K. The inter-particle distance reflects the distance between UEs. As the distance between UEs is stochastic and variable, we compute both the average distances $d_i^{Ave}(t)$ and minimum inter-particle distances $d_i^{Min}(t)$ for ease of computation.

The improved inertia weight $\varpi(t)$ formula is presented as follows in conclusion:

$$\varpi_i(t+1) = \begin{cases} \varpi_i(t)\left(e^{\frac{1}{D(t+1)+1}-1} + 1\right), D(t+1) \geq D(t) \\ \varpi_i(t)\left(e^{\frac{1}{D(t+1)+1}-1}\right), D(t+1) < D(t) \end{cases} \tag{32}$$

4.2. TD3 algorithm

We know that the DDPG algorithm can solve the continuous action space problem well, and many studies apply it to UAV trajectory control. But it has a problem of overestimation, which leads to slow convergence speed, low reward value and other problems. Therefore, we use the TD3 algorithm here. TD3 mainly has the following improved parts:

(1) To address the overestimation issue in DDPG, the TD3 algorithm utilizes two sets of critic networks, θ_n^1 and θ_n^2, which represent different Q values, and compare and select the minimum Q value as the update target.

$$\begin{aligned} & \mathrm{y}_i\left(r_i, s_{i+1}, \gamma\right) \\ & = r_i + \gamma \min_{i=1,2} Q_n^{\theta_i{}'}\left(s_{i+1}, \mu'\left(s_{i+1} | \theta^{\mu'}\right)\right), i = 1, 2 \end{aligned} \tag{33}$$

where r_i represents $R\left(s_i, a_i\right)$, which is used to evaluate the value of s_i and a_i. s_{i+1} represents the state at time $i+1$, γ represents the discount factor of reward.

(2) Fixing the policy network μ function and only training the Q function can make the network converge and get the best results.

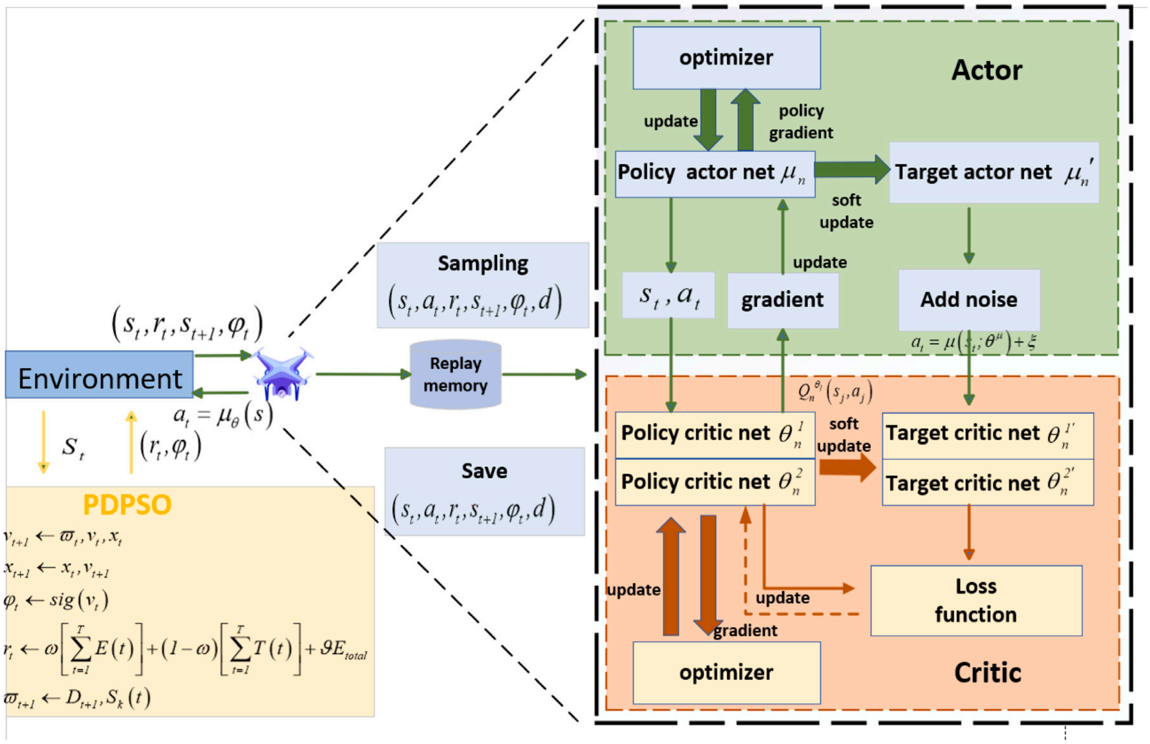

optimizer
Actor
update
policy gradient
Policy actor net μ_n
Target actor net μ_n'
soft update
update
s_t, a_t
gradient
Add noise
$a_t = \mu(s_t; \theta^\mu) + \xi$
$(s_t, r_t, s_{t+1}, \varphi_t)$
Sampling
$(s_t, a_t, r_t, s_{t+1}, \varphi_t, d)$
Environment
Replay memory
$a_t = \mu_\theta(s)$
S_t
(r_t, φ_t)
Save
$(s_t, a_t, r_t, s_{t+1}, \varphi_t, d)$
PDPSO
$v_{t+1} \leftarrow \varpi_t, v_t, x_t$
$x_{t+1} \leftarrow x_t, v_{t+1}$
$\varphi_t \leftarrow sig(v_t)$
$r_t \leftarrow \omega\left[\sum_{t=1}^{T} E(t)\right] + (1-\omega)\left[\sum_{t=1}^{T} T(t)\right] + \vartheta E_{total}$
$\varpi_{t+1} \leftarrow D_{t+1}, S_k(t)$
$Q_n^{\theta_i}(s_j, a_j)$
Policy critic net θ_n^1
Policy critic net θ_n^2
soft update
Target critic net $\theta_n^{1'}$
Target critic net $\theta_n^{2'}$
update
update
gradient
Loss function
optimizer
Critic

Fig. 2. The internal structure of the PDPSO-TD3 algorithm.

Therefore, TD3 algorithm only starts to train the action network when the round number reaches the policy cycle.

(3) The accuracy of policy value evaluation is enhanced in TD3 by introducing random noise to each action generated by the policy output.

$$a'\left(s_{i+1}\right) = clip\left(\mu'\left(s_{i+1}|\theta^{\mu'}\right) + clip\left(\varepsilon, -c, c\right), a_{low}, a_{high}\right) \\ , \varepsilon \sim N\left(0, \varepsilon\right) \tag{34}$$

Next, we will present it in four components: agent, state, action and reward value.

Agent: The UAV acts as an agent that learns its trajectory position and the optimal ratio of local task offloading continuously. It minimizes energy consumption and transmission delay to attain the lowest possible total system cost.

State: We make the assumption that the UAV travels from a predetermined starting point to a designated endpoint, moving in a left-to-right direction. The UEs location interacting with the UAV is randomly generated along the path, and the data transmission amount varies with different offloading ratios. Therefore, the state records both the UAV and UEs locations, and the data amount transmitted during the interaction.

$$State = \left\{S_k\left(t\right), S_{uav}\left(t\right), L_k\left(t\right), \forall S \in \left\{X_{size}, Y_{size}, H\right\}\right\} \tag{35}$$

Action: The action of the UAV in time slot t is:

$$Action = \left\{o_t, v_t\right\} \tag{36}$$

o_t chooses the direction for the UAV action, which ranges from $0 \sim 360°$.

Reward: Our objective is to minimize the total energy consumption and transmission delay of the MEC system, and optimize the UAV trajectory accordingly. To achieve this, we define the reward of each step R_{step} as the product of the UAV movement steps and the weighted sum of energy consumption and transmission delay produced by the system under the optimal offloading policy. Thus, the total reward value is the sum of rewards for each step, and we can derive the final average reward value $R_{average}$ as a metric for evaluating the algorithm. Moreover, in order to ensure reliable communication quality, we impose an area constraint $\left\{X_{size}, Y_{size}, H\right\}$. We will penalize UAV or UEs that go out of range severely.

$$R_{step} = \min_{S_{uav}(t)} \omega \left[\sum_{t=1}^{T} E\left(t\right)\right] + (1-\omega)\left[\sum_{t=1}^{T} T\left(t\right)\right] + \vartheta E_{total} \tag{37}$$

$$R_{average} = -\frac{R_{step}\chi}{T} \tag{38}$$

4.3. The proposed PDPSO-TD3 algorithm

From the above part, PDPSO algorithm is highly exploratory and can effectively formulate task offloading strategy. The TD3 algorithm exhibits high utilization and is capable of planning UAV paths in a high-dimensional continuous action space. The development of the joint algorithm represents a promising endeavor. Consequently, we have innovatively proposed the PDPSO-TD3 algorithm, which combines the high exploration capability of the PDPSO algorithm with the high utilization efficiency of the TD3 algorithm (maximizing regression of current policy using known experience), as shown in Algorithm 1. This ensures that the policy parameters optimized by the PDPSO algorithm can be effectively integrated into the training process of TD3 algorithm. Additionally, effective information exchange is conducted to address overestimation problems. To the best of our knowledge, this is a pioneering approach that successfully combines two exceptional algorithms and demonstrates remarkable performance.

The overall algorithm framework is illustrated in Fig. 2, which consists of three parts: PDPSO environment interaction module, UAV intelligent agent module and TD3 action module. First, the PDPSO algorithm interacts with the environment to acquire the optimal reward r$_t$ and the task offloading strategy φ_t. Next, the UAV infers the state of the next time slot s_{t+1} by leveraging its current state s_t, the offloading strategy φ_t and the reward value r$_t$ from the environment. Then, the UAV stores these information into the experience pool and performs sampling training. This helps the UAV to explore better in the subsequent formal training. Finally, The training process is mainly carried out by the TD3 network, which comprises a network of actor and a network of critic. The actor network has a policy network with weight μ_n and a target network with weight μ'_n. The critic network has two policy networks with weights θ_n^1 and θ_n^2, which can address the overestimation of Q value, the reason for choosing the TD3 algorithm as the foundation. Moreover, the critic network also has two target networks with weights $\theta_n^{1'}$ and $\theta_n^{2'}$, which can enhance the learning and training stability.

The policy critic network θ_n^1 and θ_n^2 evaluates the state s_t and the action a_t to produce the current Q-function $Q_{\mathrm{n}}^{\theta_i}\left(s_j, a_j\right)$. Meanwhile, the target critic network $\theta_n^{1'}$ and $\theta_n^{2'}$ generates a contrast Q-function $Q_{\mathrm{n}}^{\theta'_i}\left(s_j{}', \tilde{a}_j\right)$ to avoid overestimating the Q-value. It is important to note that before generating the contrast Q-function $Q_{\mathrm{n}}^{\theta'_i}\left(s_j{}', \tilde{a}_j\right)$, the target actor network μ'_n from the actor network part adds noise $\tilde{a}_j$ to the target critic network $\theta_n^{1'}$ and $\theta_n^{2'}$ that produces the contrast Q-function, which enables the TD3 policy to explore better.

The weights of the six neural networks are initialized first, followed by the opening of replay buffer B (Lines 1 to 3). In each training round, we limit the action exploration range of the UAV to $[0,1]$ to ensure random exploration but not out of bounds. Initialize the UAV state s_t(Lines 4 to 6). Next, for each step of exploration, the UAV chooses the flight Angle and transmission energy consumption based on the action a_t with the added random noise ξ(Lines 7 to 9).

$$\mathrm{a}_t = \mu\left(s_t; \theta^\mu\right) + \xi \tag{39}$$

We use the PDPSO algorithm to determine the task offloading ratio. We initialize the velocity v_1, initial position x_1, inertia weight ϖ and the local and global optimal solutions *pbest* and *gbest* of each particle. The velocity v_{i+1} and position x_{i+1} of the particle in the subsequent time slot should be timely updated. It is advantageous to compute the offloading decision φ_{i} and determine the optimal task offloading ratio during the current iteration cycle.

$$r_t = \omega\left[\sum_{t=1}^{T} E(t)\right] + (1-\omega)\left[\sum_{t=1}^{T} T(t)\right] + \vartheta E_{total} \tag{40}$$

After every particle iteration, we update the local and global optimal solutions *pbest* and *gbest*, the population diversity D and the inertia weight ϖ (Lines 10 to 19). Following the action selection by the UAV, it obtains an evaluation in the form of a reward value r_t. Furthermore, the UAV receives feedback on the next state value s_{t+1}. We store the current experience $\left(s_t, a_t, r_t, s_{t+1}, \varphi_t, done\right)$ in B in replay buffer B to improve the stability of the training process. We randomly select a small batch of values $\left(s_t, a_t, r_t, s_{t+1}, \varphi_t, done\right)$ in B from the replay buffer for each training and generate the corresponding policy using the policy actor network μ_n. It is updated utilizing a gradient-based policy:

$$\begin{aligned}&\nabla_{\mu_n} J\left(\mu_n\right)\\&= \frac{1}{N}\sum_i \left[\nabla_{\mu_n}\mu\left(s_i; \theta^\mu\right)\nabla_a Q^{\theta^1}(s, a)\,|s = s_i, a = \mu\left(s_i; \theta^\mu\right)\right]\end{aligned} \tag{41}$$

We use the current policy $\mu\left(s_i; \theta^\mu\right)$ to get two Q values $Q^{\theta^1}\left(s_i, \mu\left(s_i; \theta^\mu\right)\right)$ and $Q^{\theta^2}\left(s_i, \mu\left(s_i; \theta^\mu\right)\right)$ from two policy critic networks θ_n^1 and θ_n^2, and update the Q values by minimizing the loss function $L\left(\theta_n^j\right)$:

$$L\left(\theta_n^j\right) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - Q_n^{\theta_j}\left(s_i, a_i\right)\right]^2, j = 1, 2 \tag{42}$$

To stabilize the training process, the update of the three target networks is based on the time step d. Where lr represents the learning rate, τ represents the update rate.

$$\begin{aligned}&\mu_n \leftarrow \mu_n - lr\nabla_{\mu_n} J\left(\mu_n\right)\\&\theta_n^i \leftarrow \theta_n^i - lr\nabla_{\theta_n^i} L\left(\theta_n^i\right), i = 1, 2\end{aligned} \tag{43}$$

$$\begin{aligned}&\mu_n{}' = \tau\mu_n + (1-\tau)\,\mu_n{}'\\&\theta_n^{i'} = \tau\theta_{\mathrm{n}}^i + (1-\tau)\,\theta_n^{i'}, i = 1, 2\end{aligned} \tag{44}$$

In addition, the overall reward value is evaluated after each training session (Lines 20 to 29).

$$reward = R_{average} \tag{45}$$

Algorithm 1 PDPSO-TD3 Algorithm.

```
1:  Initialize policy actor network $\mu_n$ and target actor network $\mu'_n$;
2:  Initialize policy critic network $\theta_n^1$, $\theta_n^2$ and target critic network $\theta_n^{1'}$, $\theta_n^{2'}$;
3:  Initialize replay memory B;
4:  for $episode\ ep = 1, ..., M$ do
5:    Define a finite action exploration space $\eta$, which is fixed to $[0,1]$;
6:    Obtain the current initial state value $s_t$;
7:    for $step\ t = 1, ..., T$ do
8:      Each action $a_t = \mu\left(s_t; \theta^\mu\right) + \xi$ at every step needs to be modified with
        random noise $\xi$;
9:      The UAV sets its own mobile and transmission power according to
        the action $a_t$;
10:     Initialize the particle's optimal solution $pbest$, the swarm's optimal
        solution $gbest$;
11:     particle position $x_1$, velocity $v_1$, weight $\varpi$;
12:     for $iteration\ i = 1, ..., I$ do
13:       for $eachparticle\ n = 1, ..., K$ do
14:         Update velocity $v_{i+1}$ and position $x_{i+1}$;
15:         The selection of partial or complete offloading is based on the
            offloading strategy:
            $\varphi_i(t) = \begin{cases} 1, rand_3 < Sig\left(v_i(t)\right) \\ 0, otherwise \end{cases}$
16:       end for
17:       Get in (40);
18:       Update the particle's optimal solution $pbest$, the swarm's optimal
          solution $gbest$, diversity $D$ and weight $\varpi$;
19:     end for
20:     obtains the reward value $r_t$, and the next state $s_{t+1}$;
21:     Store $\left(s_t, a_t, r_t, s_{t+1}, \varphi_t, done\right)$ in B;
22:     Sample a random mini-batch of
        $\left(s_t, a_t, r_t, s_{t+1}, \varphi_t, done\right)$ from B;
23:     Set $y_i = r_i + \gamma \min_{i=1,2} Q_n^{\theta_i'}\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^{\mu'}\right)\right), i = 1, 2$;
24:     Update policy critic network $\theta_n^1$, $\theta_n^2$ by minimizing loss function in
        (42);
25:     Update policy actor network $\mu_n$ with (41);
26:     Update the target networks with (44);
27:   end for
28:   The final reward: $reward = R_{average}$;
29: end for
```

4.4. Complexity analysis

We consider the complexity of individual algorithms and studied how they interact when combined into a unified framework. Specifically, the PDPSO algorithm needs to solve the problem of formulating a specific offloading scheme for intensive tasks, that is, deciding whether to perform the task locally or with UAV assistance. The time complexity of this problem is affected by the number of particles K in the swarm and the number of iterations I, which is $O(I * K)$. The TD3 algorithm plans the UAV's path based on the task offloading strategy, and the problem is divided into T time slots, with a time complexity of $O(T * I * K)$. The number of training rounds required to stabilize at the optimal solution is M. Therefore, ignoring constant factors, the approximate total time complexity of PDPSO-TD3 algorithm is $O(M * T * I * K)$.

It is worth noting that the population particle number represents the number of UEs K. In other words, the number of UEs K, the algorithm's learning rate lr, the added action noise ξ, the reward discount factor γ, and the sample number of training batches M_{b}, all affect the convergence speed of the algorithm, and then affect the

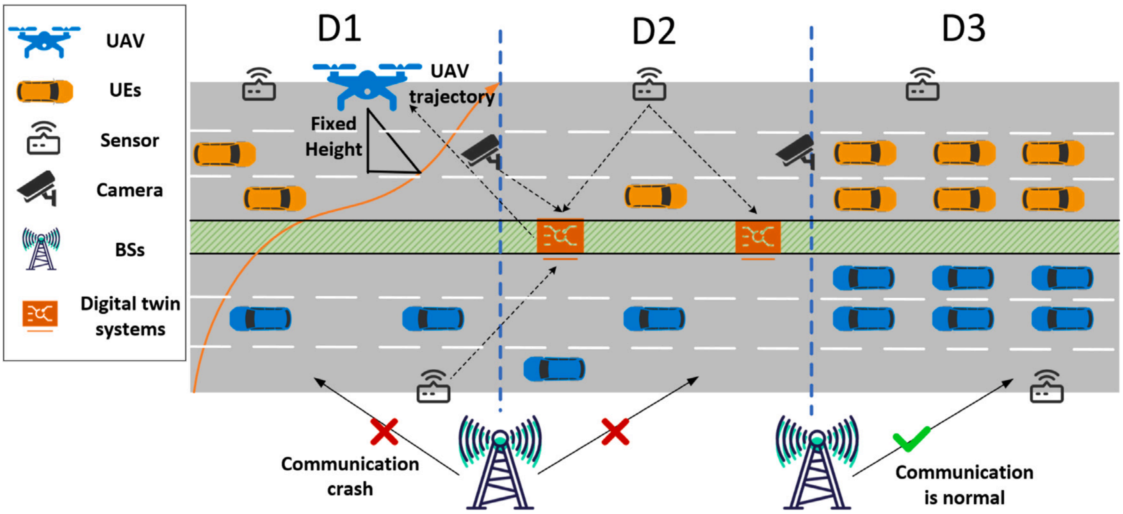


Fig. 3. Analysis of real-world applications.

overall time complexity. Therefore, it is necessary to conduct parameter experiments continuously to achieve a balance, and minimize the system's total energy consumption to achieve efficient utilization of UAV computing resources. In the Section 6.4, we compare various parameters and select a set of optimal ones to reduce time complexity.

5. Analysis of real-world applications

In this section, we apply the proposed PDPSO-TD3 algorithm to practical scenarios and outline our scheme. As illustrated in Fig. 3, our research targets urban road areas but can also be extended to disaster emergency response and agricultural monitoring. We segment a traffic area into multiple square regions, labeled as $D1$, $D2$, and $D3$. Two scenarios are considered. In the first scenario, communication BSs have failed, and UEs require urgent communication services provided by UAV in the $D1$ and $D2$. In the second scenario, the communication BSs is operational, but the signal degradation will lead to traffic congestion in the $D3$, then we need UAV support for UEs. These scenarios are common in practice, and we will detail the proposed solutions next.

UAV face operational constraints due to limited power, necessitating regional division for effective coverage. Furthermore, the training of the PDPSO-TD3 algorithm in realistic environments presents a significant challenge. This is due to the necessity for extensive UAV exploration to collect unknown environmental data, which hinders the swift convergence of the algorithm and makes it less able to handle unforeseen circumstances. Drawing inspiration from [46], digital twin systems offer a promising solution, with studies affirming their feasibility and benefits. By integrating a digital twin system to create a duplicate of the UAV's operational environment, we can conduct tests and refine the PDPSO-TD3 algorithm within this context, while simultaneously optimizing the UAV's path planning, resource allocation, and energy management. This approach is designed to enhance the coverage efficiency and performance of the MEC system. Consequently, our detailed recommendation is as follows:

5.0.1. Data collection and modeling

The digital twin system is utilized to generate a virtual representation of roadways that are prone to accidents. This representation integrates parameters such as meteorological conditions and vehicular traffic patterns. This can be accomplished using existing GIS, sensors, and surveillance cameras. Additionally, UAV is deployed in specific areas to update the digital twin system in real time, ensuring the accuracy and relevance of environmental information.

5.0.2. Simulation and optimization

The PDPSO-TD3 algorithm, integrated within a digital twin system, leverages real-time environmental data for continuous pre-training and optimization, enabling swift convergence and provision of essential communication services in the event of unforeseen accidents. Concurrently, algorithmic parameters are dynamically adjusted based on training outcomes, thereby minimizing actual training costs and enhancing algorithmic efficiency.

5.0.3. Real-time monitoring and analysis

The digital twin system, equipped with an embedded algorithm and real-time environmental data, enables real-time analysis of traffic conditions, including vehicle congestion and adverse weather. This facilitates timely dissemination of early warning messages, allowing the command center to strategically allocate rescue resources, including UAV.

It is important to note that in our model, the altitude of the UAV is fixed. This is due to the fact that the communication services require a higher altitude for the UAV, which correlates positively with coverage for UEs and negatively with energy consumption and transmission delay. Balancing UAV altitude with energy loss poses challenges. Consequently, we have set the optimal altitude for the UAV to 250 m, as this altitude ensures high-quality communication services while minimizing excessive energy consumption in road environments without obstruction.

6. Experimental results and analysis

In this section, we detail the experimental setup, baselines, performance of different algorithms, sensitivity for parameters, and evaluation results.

6.1. Experimental setup

We conduct numerical experiments based on Intel i5-12500H, NVIDIA GTX 3050Ti, Python 3.7.12, and Pytorch 1.9.1. Next, we assume that there is a road area where communication is cut off, with an area of 1000×1000 m^2. The UAV assumes the role of a temporary mobile BSs, providing communication services. As shown in Fig. 4, there are a total of 10 UEs randomly distributed in this area, with each UEs data size ranging from $[1, 10]$ Mbits. Similarly, the positions of UAV and UEs are also restricted within a range of $[1000\text{ m}, 1000\text{ m}, 250\text{ m}]$. The CPU-cycle requirements for local and UAV-assisted task execution are 800 cycles/bit and 1000 cycles/bit, respectively. Additionally, our

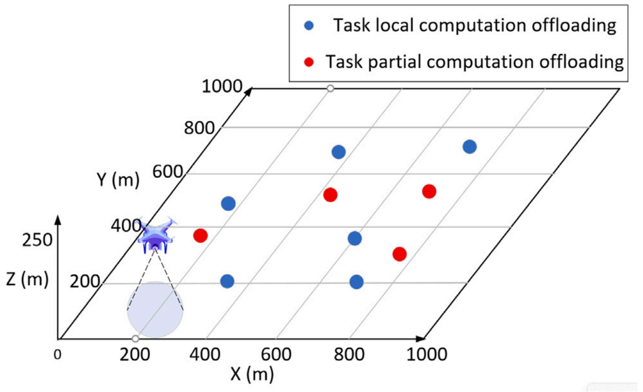


Fig. 4. Initial random positions of UEs.

Table 3
Parameter setting.

Parameters	Values
The count of UEs K	10
Task data size generated by the UEs $L_k(t)$	$[1, 10]$ Mbits
The predetermined altitude of the UAV H	250 m
The range of randomly generated UEs $Xsize, Ysize$	$[1000$ m, 1000 m$]$
The channel gain at a distance of 1 m β_0	−50 dB
The aggregate uplink bandwidth demanded by the UEs B_u	100 MHz
The communication transmission power of UEs P_{user}	0.5 W
The receiving power of UAV P_{uav}	0.5 W
The noise power generated by UAV communication δ_0^2	−70 dB m/Hz
The CPU capacitance index of UEs K_{user}	10^{-27}
The CPU capacitance index of UAV K_{uav}	10^{-28}
The CPU computation cycles for UEs when performing local C_{user}	800 cycles/bit
The CPU computation cycles required for the UAV C_{uav}	1000 cycles/bit
The local computing resources of UEs f_{user}	1 GHz
The computing resources of UAV f_{uav}	3 GHz
Calculate the weights used ω	0.75
UAV flight energy consumption weight ϑ	0.65
UAV flight maximum speed constraint $V_{\max}$	15 m/s
The blade profile power of the UAV in hover P_0	25 W
The induced power P_i	25 W
Mean rotor induced velocity in hover V_0	4.8 m/s
The tip speed of the UAV rotor blade U_{tip}	180 m/s
Fuselage drag ratio d_0	0.3
Air density ρ	1.225 kg/m^3
Rotor solidity s	0.05
Propeller disk area A	0.75 m^2

Table 4
The parameter configuration of the PDPSO-TD3 algorithm.

Parameter	Value
The overall count of training iterations M	6000
The number of iterations I	200
Learning rate lr	0.0003
Action noise variance ξ	0.1
Reward discount factor γ	0.99
Mini-batch size M_b	128

simulation environment is designed to validate the algorithm's effectiveness, which can be theoretically extended to broader operational ranges based on the specific performance metrics and real-world scenarios of various UAV models. Table 3 summarizes the main simulation parameters, while Table 4 lists the parameter settings of our algorithm.

6.2. Baselines

DQN, DDPG, and Proximal Policy Optimization (PPO) are pivotal algorithms in DRL, with numerous studies building upon them for innovation and improvement. Consequently, this paper primarily focuses on comparing these algorithms with the PDPSO-TD3 algorithm.

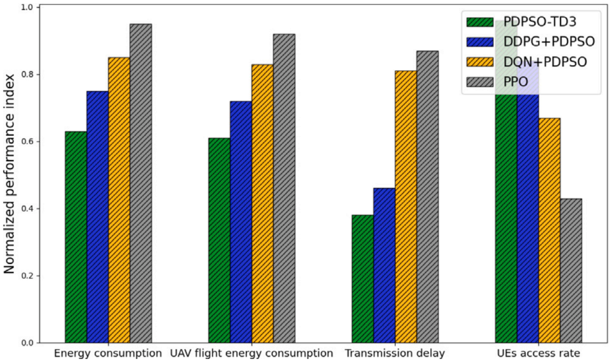


Fig. 5. Performance of different algorithms.

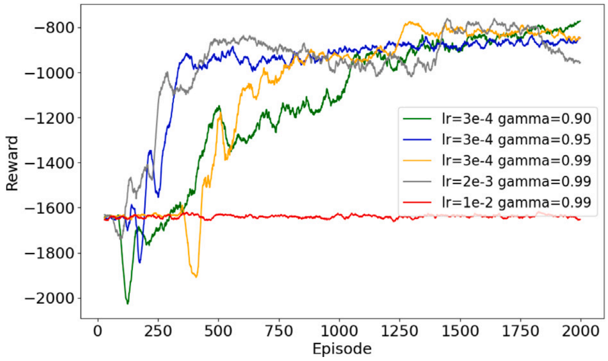


Fig. 6. The impact of learning rate and reward discount factor on the value of rewards.

Additionally, we developed the DQN+PDPSO algorithm to facilitate more accurate comparisons in a consistent environment.

- **DQN+PDPSO:** The DQN algorithm optimized the UAV flight trajectory, while the PDPSO algorithm addressed the offloading strategy. Notably, DQN is favored for high-dimensional, continuous action spaces owing to its convergence and scalability. However, its slow convergence and tendency for overestimation can adversely affect the learning process.
- **DDPG+PDPSO** [21]: The DDPG algorithm optimized the UAV flight trajectory, and the PDPSO algorithm refined its offloading strategy. As with DQN, DDPG confronts challenges, including prolonged training episodes and potential overestimation issues.
- **PPO:** PPO is a policy optimization algorithm that constrains policy updates within a specified range by controlling the step size, ensuring stability and broad applicability in continuous action space reinforcement learning tasks. However, it has high computational demands and is prone to local optima in complex environments, potentially impacting policy performance.

6.3. Performance of different algorithms

In Fig. 5, the "normalized performance index" is employed to assess algorithmic efficiency. It is notable that PDPSO-TD3 demonstrates superior performance with minimal energy expenditure, effectively minimizing UAV flight power and transmission latency. Furthermore, PDPSO-TD3's UEs access rate approaching 1 signifies extensive coverage of the MEC system, ensuring reliable and robust communication quality. We analyze the average transmission delay, energy consumption, UAV flight energy, and UEs access rate for various algorithms, as detailed in Table 5.

Table 5
Performance comparison of PDPSO-TD3 algorithm(UEs = 80).

Performance comparison	DQN+PDPSO	DDPG+PDPSO	PPO	PDPSO-TD3 (Ours)
AVG transmission delay	1.23 (s)	0.91 (s)	1.31 (s)	**0.78 (s)** **Decrease: 36.6%, 14.3%, 40.5%**
AVG energy consumption	5061 (W)	4342 (W)	5954 (W)	**3906 (W)** **Decrease: 22.9%, 10.1%, 34.4%**
AVG UAV flight energy	5261 (W)	4210 (W)	5754 (W)	**3955 (W)** **Decrease: 24.7%, 6.1%, 31.2%**
UEs access rate	0.74	0.92	0.69	**0.95** **Increase: 28.4%, 3.3%, 37.7%**

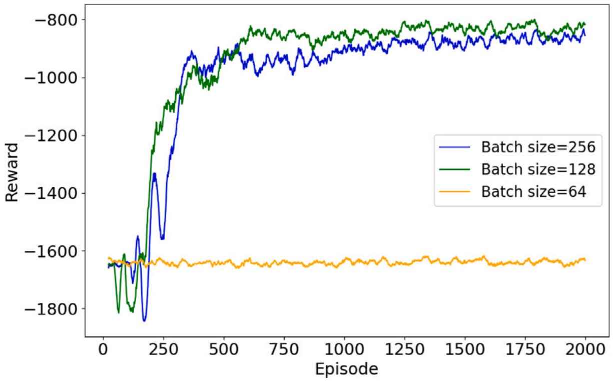


Fig. 7. The impact of varying batch sizes on reward values.

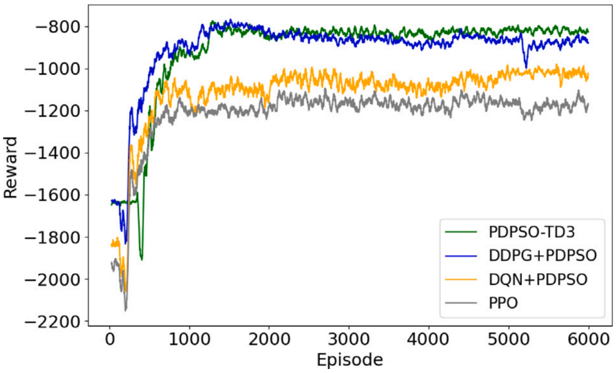


Fig. 8. Reward curves.

6.4. *Sensitivity for parameters*

6.4.1. *Sensitivity of learning rate*

The UAV's ability to explore and the rate at which the value function updates are closely tied to the size of the learning rate. An elevated learning rate prompts the UAV to focus on immediate environmental feedback, which can lead to oscillations in policy and value functions, impeding effective convergence. On the other hand, a reduced learning rate causes the UAV to rely more on historical experiences, resulting in a decreased learning pace and potential stagnation. Furthermore, the discount factor of rewards, a critical hyperparameter, strictly regulates the valuation of future rewards. An increased discount factor allows the UAV to adopt a long-term perspective and thus accumulate higher rewards. In contrast, a lower discount factor biases the UAV towards immediate rewards. Essentially, the discount factor dictates the temporal scope of the training task and the importance of cumulative rewards. Considering these factors, our primary objective is to determine an appropriate learning rate and then identify a corresponding discount factor, ensuring that this combination enhances both the velocity and stability of the model's learning process. Consequently, this enhances the effectiveness of decision-making.

In Fig. 6, we experimented with different parameter combinations, evaluating their impact on the training reward. Consequently, it can be observed that, in the case of a learning rate of $3e-4$, γ variations primarily affect the speed of reward convergence, while having a minimal impact on the reward value. When the learning rate is relatively high ($lr = 1e-2$), the model tends to become trapped in a local optimum with reduced fluctuations in the convergence value.

6.4.2. *Sensitivity of batch size*

The batch size denotes the number of samples fed into the model per iteration. To achieve greater refinement and precision, we have examined the algorithm's performance across a range of batch sizes, as illustrated in Fig. 7. When the batch size is set to 64, the algorithm tends to converge to a suboptimal solution with a relatively low convergence value. However, as the batch size is increased to 128 and 256, there is a significant enhancement in both the convergence speed and the quality of the algorithm's results. Notably, at a batch size of 128, the convergence value is significantly higher compared to that at 256, although the convergence speeds are nearly identical. Thus, a batch size of 128 represents a reasonable compromise, as it does not significantly prolong the training time nor does it substantially compromise the algorithm's performance.

6.5. *Evaluation results*

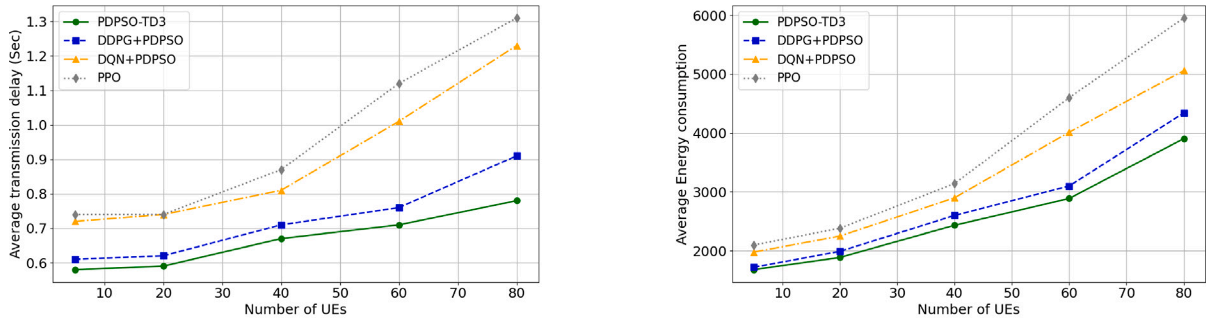
This subsection provides a thorough assessment and discussion of the experimental outcomes, concentrating on the algorithm's convergence, transmission delay, energy consumption, UAV flight energy, UEs access rate, UAV path optimization, and system throughput.

6.5.1. *Convergence performance*

In this section, we conduct a comparison of DQN+PDPSO, DDPG+PDPSO, and PPO over 6000 training episodes using consistent parameters to facilitate a thorough evaluation of the PDPSO-TD3 algorithm. Each reward component in this study incorporates a weighted value that balances energy consumption and transmission delay. As illustrated in Fig. 8, all four algorithms demonstrate the ability to converge. However, PDPSO-TD3 shows faster convergence, enhanced stability, and achieves higher convergence values compared to other algorithms. This advantage can be attributed to TD3's use of twin critic networks and delayed target network updates, which effectively mitigate the Q-value overestimation issues encountered by DQN and DDPG. While PPO primarily relies on policy optimization methods, it may still be prone to value function overestimation under certain conditions.

6.5.2. *Comparison of transmission delay and energy consumption*

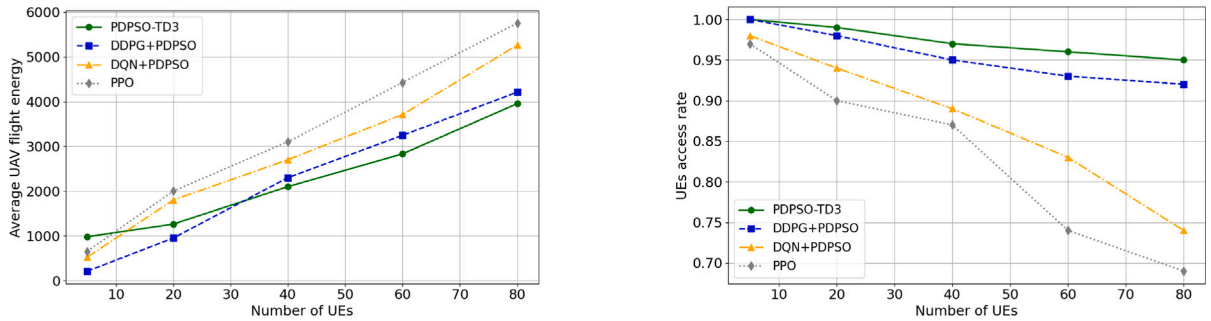
Due to the limited operational range of the UAV and UEs, we consider incrementally increasing the number of UEs within this range to observe the impact on system energy consumption and transmission latency, as shown in Fig. 9. For comparative analysis, we used the



(a) Average transmission delay

(b) Average energy consumption

Fig. 9. The transmission delay and energy consumption under different numbers of UEs.



(a) Average UAV flight energy consumption

(b) UEs access rate

Fig. 10. The flight energy consumption and UEs access rate under different numbers of UEs.

DQN+PDPSO, DDPG+PDPSO and PPO algorithms. To ensure fairness, the UAV start point was kept consistent across all experiments. In particular, we observed a positive correlation between the number of UEs and both system energy consumption and transmission delay. This correlation was attributed to the limited computational capacity, which was unable to cope with an increasing number of computation and offloading tasks. Considering the constraints of current communication scenarios and channel congestion, the maximum number of UEs was limited to 80.

Our results showed that the PDPSO-TD3 algorithm performed best in all scenarios, effectively reducing energy consumption and average transmission delay, especially when the number of UEs was large. This superior performance is due to the algorithm's ability to rationally allocate computational resources and provide an efficient task offload ratio, thus benefiting a larger number of UEs. In addition, the performance of the DQN, DDPG, and PPO algorithms is found to be inferior to that of the PDPSO-TD3 algorithm, suggesting their suitability primarily for scenarios with a limited number of UEs.

6.5.3. Comparison of UAV flight energy consumption and UEs access rate

Fig. 10 presents an investigation into the impact of varying UEs counts on UAV flight energy consumption and UEs access rate for diverse algorithms. Fig. 10(a) demonstrates that PDPSO-TD3 results in higher energy consumption at lower UEs counts (5 to 20). This is attributed to the introduction of action noise for initial exploration, which requires the UAV to continuously navigate in order to gather environmental data. Once the UEs count exceeds 30, PDPSO-TD3 efficiency becomes apparent, with substantially lower energy consumption than alternative algorithms. This indicates its superior capability to reduce UAV energy expenditure.

The UEs access rate is a critical performance indicator for UAV-assisted MEC systems, signifying the extent to which UEs can successfully establish a connection with the UAV. Furthermore, it serves as an indicator of the effectiveness of the UAV flight trajectory in covering UEs, which is essential for evaluating the coverage of the MEC system.

As shown in Fig. 10(b), PDPSO-TD3 exhibits the highest UEs access rate, stabilizing at approximately 95%, which suggests extensive communication coverage and a high coverage rate for the MEC system. Notably, the access rate for UEs decreases as the number of UEs increases, and the performance divergence among algorithms becomes more pronounced. At 80 UEs, the access rate of PDPSO-TD3 is nearly 25% higher than that of PPO, highlighting a significant performance advantage.

6.5.4. UAV trajectory optimization verification

In this study, we have validated the effectiveness of PDPSO-TD3 in UAV trajectory planning through simulation experiments, in particular its ability to maximize the communication coverage of UEs. The experimental setup involves the random distribution of 10 UEs within the simulation environment, emulating the uncertainty and randomness of the real world. The UAV takes off from a coordinate position of $(0\text{ m}, 0\text{ m}, 250\text{ m})$ and plans to fly to a destination at $(1000\text{ m}, 1000\text{ m}, 250\text{ m})$, maintaining a constant altitude throughout the journey. Red dots represent UEs using a partial offload strategy, uploading part of their computational tasks to the UAV for processing. Blue dots represent UEs using a local offloading strategy, where all computational tasks are performed locally. The real-time trajectory of the UAV is represented by a green curve. Fig. 11 demonstrates that, regardless of the offloading strategy employed, the UAV flight trajectory is designed to align with a diagonal path, simultaneously striving to maintain proximity to the UEs to facilitate communication and computational services.

6.5.5. Average throughput analysis

Fig. 12 examines the average throughput changes of algorithms across training rounds. At 1000 episodes, PDPSO-TD3's throughput is lower than DDPG+PDPSO due to its strategy of action noise to

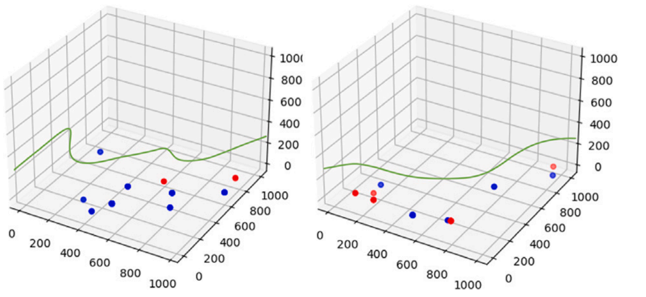


Fig. 11. Optimal trajectory of UAV. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

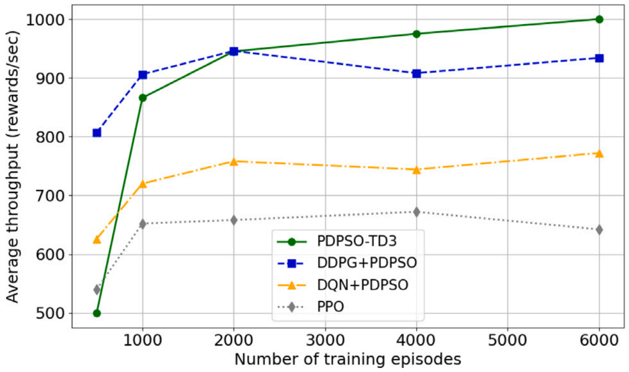


Fig. 12. Comparison of the average throughput.

diversify early exploration, which results in initial lower performance. By approximately 2000 episodes, PDPSO-TD3 has converged, surpassing other algorithms in average throughput and showing a stable increase, indicating effective strategy formulation and improved resource utilization.

7. Conclusion

The paper investigates the utilization of UAV as mobile BSs for providing communication services in scenarios where the underlying communication infrastructure is damaged. Additionally, the UAV is also responsible for assisting in offloading task calculations from UEs. Specifically, we propose an innovative algorithm called PDPSO-TD3. Through theoretical analysis and mathematical derivation, we calculate the optimal task offloading strategy, which encompasses local offloading of computing tasks and UAV-assisted offloading of certain computing tasks. Furthermore, we incorporate the TD3 algorithm to optimize the UAV trajectory, taking into account factors such as random distribution of UEs, data transmission size, and UAV altitude selection. The experimental results show that the PDPSO-TD3 algorithm is superior to the traditional methods in terms of convergence speed and convergence results. Moreover, the UAV is capable of autonomously devising real-time paths to maximize communication coverage, ensure reliable communication quality, and minimize overall energy consumption and transmission delay within the system.

In future work, we aim to explore more complex application scenarios. This includes studying the joint communication of multiple units and UAV in dynamic environments, as well as designing optimization strategies for multiple trajectory intersections. Additionally, building an extended and simplified model of the wireless network poses great challenges, taking into account reduction of transmission errors and increased anti-interference between UEs.

CRediT authorship contribution statement

Fanfan Shen: Writing – review & editing. **Bofan Yang:** Writing – original draft. **Jun Zhang:** Data curation. **Chao Xu:** Formal analysis. **Yong Chen:** Investigation. **Yanxiang He:** Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Basic Science (Natural Science) Research Project of Colleges and Universities in Jiangsu Province (22KJA520004), the National Natural Science Foundation of China (61902189, 71972102, 62472227, 61972293, 62162002), the Natural Science Foundation of Jiangxi Province (20212BAB202002), Postgraduate Research & Practice Innovation Program of Jiangsu Province ($SJCX$24_1166).

Data availability

Data will be made available on request.

References

[1] Y. Liu, Y. Tang, J. Zhao, O. Sun, M. Lv, L. Yang, 5G+ VR industrial technology application, in: 2020 International Conference on Virtual Reality and Visualization, ICVRV, IEEE, 2020, pp. 336–337.

[2] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2322–2358.

[3] M. Tang, V.W. Wong, Deep reinforcement learning for task offloading in mobile edge computing systems, IEEE Trans. Mob. Comput. 21 (6) (2020) 1985–1997.

[4] Z. Yang, M. Chen, X. Liu, Y. Liu, Y. Chen, S. Cui, H.V. Poor, AI-driven UAV-NOMA-MEC in next generation wireless networks, IEEE Wirel. Commun. 28 (5) (2021) 66–73.

[5] P.A. Apostolopoulos, G. Fragkos, E.E. Tsiropoulou, S. Papavassiliou, Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty, IEEE Trans. Mob. Comput. 22 (1) (2021) 175–190.

[6] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, K.B. Letaief, AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks, IEEE Internet Things J. 8 (2) (2020) 1211–1223.

[7] M. Dai, T.H. Luan, Z. Su, N. Zhang, Q. Xu, R. Li, Joint channel allocation and data delivery for uav-assisted cooperative transportation communications in post-disaster networks, IEEE Trans. Intell. Transp. Syst. 23 (9) (2022) 16676–16689.

[8] Y. Zeng, R. Zhang, T.J. Lim, Wireless communications with unmanned aerial vehicles: Opportunities and challenges, IEEE Commun. Mag. 54 (5) (2016) 36–42.

[9] Y. Yu, X. Bu, K. Yang, H. Yang, X. Gao, Z. Han, UAV-aided low latency multi-access edge computing, IEEE Trans. Veh. Technol. 70 (5) (2021) 4955–4967.

[10] W. Mao, K. Xiong, Y. Lu, P. Fan, Z. Ding, Energy consumption minimization in secure multi-antenna UAV-assisted MEC networks with channel uncertainty, IEEE Trans. Wireless Commun. 22 (11) (2023) 7185–7200.

[11] X. Xu, H. Zhao, H. Yao, S. Wang, A blockchain-enabled energy-efficient data collection system for UAV-assisted IoT, IEEE Internet Things J. 8 (4) (2020) 2431–2443.

[12] J. Xu, X. Yan, C. Peng, X. Wu, L. Gu, Y. Niu, UAV local path planning based on improved proximal policy optimization algorithm, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[13] B. Xin, C. He, DRL-based improvement for autonomous UAV motion path planning in unknown environments, in: 2022 7th International Conference on Control and Robotics Engineering, ICCRE, IEEE, 2022, pp. 102–105.

[14] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, X. Liang, UAV autonomous target search based on deep reinforcement learning in complex disaster scene, IEEE Access 7, 117227–117245.

[15] X.-Y. Xu, Y.-Y. Chen, T.-R. Liu, TD3-BC-PPO: Twin delayed DDPG-based and behavior cloning-enhanced proximal policy optimization for dynamic optimization affine formation, J. Franklin Inst. 361 (12) (2024) 107018.

[16] B. Shi, Z. Chen, Z. Xu, A deep reinforcement learning based approach for optimizing trajectory and frequency in energy constrained multi-UAV assisted MEC system, IEEE Trans. Netw. Serv. Manag. (2024) 1, http://dx.doi.org/10.1109/TNSM.2024.3362949.

[17] M. Ejaz, J. Gui, M. Asim, M.A. El-Affendi, C. Fung, A.A. Abd El-Latif, RL-planner: Reinforcement learning-enabled efficient path planning in multi-UAV MEC systems, IEEE Trans. Netw. Serv. Manag. 21 (3) (2024) 3317–3329.

[18] Y. He, Y. Gan, H. Cui, M. Guizani, Fairness-based 3-D multi-UAV trajectory optimization in multi-UAV-assisted MEC system, IEEE Internet Things J. 10 (13) (2023) 11383–11395.

[19] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, D. Niyato, Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing, IEEE Trans. Wireless Commun. 21 (9) (2022) 6949–6960.

[20] A. Nehra, P. Consul, I. Budhiraja, G. Kaur, N. Nasser, M. Imran, Federated learning based trajectory optimization for UAV enabled MEC, in: ICC 2023-IEEE International Conference on Communications, IEEE, 2023, pp. 1640–1645.

[21] Y. Gan, Y. He, Trajectory optimization and computing offloading strategy in UAV-assisted MEC system, in: 2021 Computing, Communications and IoT Applications, ComComAp, IEEE, 2021, pp. 132–137.

[22] Y. Zeng, J. Tang, MEC-assisted real-time data acquisition and processing for UAV with general missions, IEEE Trans. Veh. Technol. 72 (1) (2022) 1058–1072.

[23] E. Figetakis, A. Refaey, UAV path planning using on-board ultrasound transducer arrays and edge support, in: 2021 IEEE International Conference on Communications Workshops, ICC Workshops, IEEE, 2021, pp. 1–6.

[24] B. Yang, X. Cao, C. Yuen, L. Qian, Offloading optimization in edge computing for deep-learning-enabled target tracking by internet of UAVs, IEEE Internet Things J. 8 (12) (2020) 9878–9893.

[25] Y. Ding, Y. Feng, W. Lu, S. Zheng, N. Zhao, L. Meng, A. Nallanathan, X. Yang, Online edge learning offloading and resource management for UAV-assisted MEC secure communications, IEEE J. Sel. Top. Sign. Proces. 17 (1) (2022) 54–65.

[26] K. Xiang, Y. He, UAV-assisted MEC system considering UAV trajectory and task offloading strategy, in: ICC 2023-IEEE International Conference on Communications, IEEE, 2023, pp. 4677–4682.

[27] X. Deng, J. Li, P. Guan, L. Zhang, Energy-efficient UAV-aided target tracking systems based on edge computing, IEEE Internet Things J. 9 (3) (2021) 2207–2214.

[28] F. Pervez, A. Sultana, C. Yang, L. Zhao, Energy and latency efficient joint communication and computation optimization in a multi-UAV-assisted MEC network, IEEE Trans. Wireless Commun. 23 (3) (2024) 1728–1741.

[29] J. Ji, K. Zhu, C. Yi, D. Niyato, Energy consumption minimization in UAV-assisted mobile-edge computing systems: Joint resource allocation and trajectory design, IEEE Internet Things J. 8 (10) (2020) 8570–8584.

[30] B. Liu, Y. Wan, F. Zhou, Q. Wu, R.Q. Hu, Resource allocation and trajectory design for MISO UAV-assisted MEC networks, IEEE Trans. Veh. Technol. 71 (5) (2022) 4933–4948.

[31] W. Liu, H. Wang, X. Zhang, H. Xing, J. Ren, Y. Shen, S. Cui, Joint trajectory design and resource allocation in UAV-enabled heterogeneous MEC systems, IEEE Internet Things J. 11 (19) (2024) 30817–30832.

[32] Y. Zeng, S. Chen, Y. Cui, J. Du, Efficient trajectory planning and dynamic resource allocation for UAV-enabled MEC system, IEEE Commun. Lett. 28 (3) (2024) 597–601.

[33] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F.R. Yu, V.C. Leung, UAV trajectory optimization for data offloading at the edge of multiple cells, IEEE Trans. Veh. Technol. 67 (7) (2018) 6732–6736.

[34] P.A. Karegar, A. Al-Anbuky, UAV-assisted data gathering from a sparse wireless sensor adaptive networks, Wirel. Netw. 29 (3) (2023) 1367–1384.

[35] D.Z. Al-Hamid, A. Al-Anbuky, Vehicular networks dynamic grouping and re-orchestration scenarios, Information 14 (1) (2023) 32.

[36] L. Huang, S. Bi, Y.-J.A. Zhang, Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks, IEEE Trans. Mob. Comput. 19 (11) (2019) 2581–2593.

[37] X. Zheng, Y. Wu, L. Zhang, M. Tang, F. Zhu, Priority-aware path planning and user scheduling for UAV-mounted MEC networks: A deep reinforcement learning approach, Phys. Commun. 62 (2024) 102234.

[38] J. Guo, H. Gao, Z. Liu, F. Huang, J. Zhang, X. Li, J. Ma, ICRA: An intelligent clustering routing approach for UAV ad hoc networks, IEEE Trans. Intell. Transp. Syst. 24 (2) (2023) 2447–2460.

[39] S. Ouahouah, M. Bagaa, J. Prados-Garzon, T. Taleb, Deep-reinforcement-learning-based collision avoidance in uav environment, IEEE Internet Things J. 9 (6) (2021) 4015–4030.

[40] L.P. Qian, H. Zhang, Q. Wang, Y. Wu, B. Lin, Joint multi-domain resource allocation and trajectory optimization in UAV-assisted maritime IoT networks, IEEE Internet Things J. 10 (1) (2022) 539–552.

[41] F. Song, M. Deng, H. Xing, Y. Liu, F. Ye, Z. Xiao, Energy-efficient trajectory optimization with wireless charging in UAV-assisted MEC based on multi-objective reinforcement learning, IEEE Trans. Mob. Comput. (2024) 1–18, http://dx.doi.org/10.1109/TMC.2024.3384405.

[42] Y. Gao, X. Yuan, D. Yang, Y. Hu, Y. Cao, A. Schmeink, UAV-assisted MEC system with mobile ground terminals: DRL-based joint terminal scheduling and UAV 3D trajectory design, IEEE Trans. Veh. Technol. 73 (7) (2024) 10164–10180.

[43] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, A. Nallanathan, Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing, IEEE Trans. Mob. Comput. 21 (10) (2021) 3536–3550.

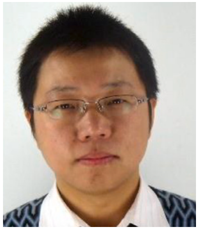
[44] Y. Zhang, Z. Kuang, Y. Feng, F. Hou, Task offloading and trajectory optimization for secure communications in dynamic user multi-UAV MEC systems, IEEE Trans. Mob. Comput. (2024) 1–15, http://dx.doi.org/10.1109/TMC.2024.3442909.

[45] Y. Zeng, J. Xu, R. Zhang, Energy minimization for wireless communication with rotary-wing UAV, IEEE Trans. Wirel. Commun. 18 (4) (2019) 2329–2345.

[46] H. He, W. Yuan, S. Chen, X. Jiang, F. Yang, J. Yang, Deep reinforcement learning-based distributed 3D UAV trajectory design, IEEE Trans. Commun. 72 (6) (2024) 3736–3751.

Fanfan Shen received the PhD degree from Wuhan University, Hubei, China. He is now a fulltime associate professor at the Nanjing Audit University, Jiangsu, China. His main research interests include emerging non-volatile memory, embedded system and artificial intelligence (ffshen@whu.edu.cn).

Bofan Yang graduated from Henan University of Technology with a bachelor's degree in IOT engineering in 2021. He is currently pursuing a Master's degree at Nanjing Audit University, where his research interests are UAV path planning and MEC networks.



Jun Zhang received the PhD degree from Wuhan University, Hubei, China. He is now a professor at East China University Of Technology. His main research interests include high performance computing and computer architecture.

Chao Xu received the BS and PhD degrees from the Computer School at the Wuhan University, Hubei, China. He is now a professor at Nanjing Auditing University, China. His main research interests include trust computing and bigdata audit.

Yong Chen received the Ph.D. degree from Wuhan University, in 2013. He is currently a senior engineer with the School of Information Science, Nanjing Auditing University, China. His main research interests include compilation optimization, software reliability and software defect prediction.

Yanxiang He received the PhD degree from Wuhan University, Hubei, China. He is a Professor in the Computer School at the Wuhan University. His research interests include trusted software, distributed parallel processing and high performance computing.