IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 12, 2021, accepted January 19, 2021, date of publication January 28, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3055335

Joint Power and QoE Optimization Scheme for Multi-UAV Assisted Offloading in Mobile Computing

QI WANG[1,2], (Student Member, IEEE), ANG GAO[1,2], AND YANSU HU[3]

[1] School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China
[2] Yangtze River Delta Research Institute, Northwestern Polytechnic University, Taicang 215400, China
[3] School of Electronics and Control, Chang'an University, Xi'an 710064, China

Corresponding author: Ang Gao (gaoang@nwpu.edu.cn)

This work was supported in part by the China and Shaanxi Postdoctoral Science Foundation under Grant 2017M623243 and Grant 2018BSHYDZZ26, in part by the Shaanxi and Guangxi Keypoint Research and Invention Program under Grant 2019ZDLGY13-02-02 and Grant AB19110036, and in part by the Taicang Keypoint Science and Technology Plan under Grant TC2018SF03 and Grant TC2019SF03.

ABSTRACT Recent years, unmanned aerial vehicles (UAVs) have attracted much attention for providing intermediate relay to ground mobile user equipments (UEs) for their flexible mobility. UEs can offload computing-intensive task to mobile cloud computing (MCC) or mobile edge computing (MEC) for fast processing. However, with multi-UAV and ground mobile UEs in the system, heterogeneous performance requirement as well as fast-changing communication condition make the system more complicated. Meanwhile, both UEs and UAVs are battery-driven. How to optimize the energy efficiency for UEs' transmission and UAVs' position should be carefully considered. Since this is a non-convex and mixed-integer optimization problem, a heuristic joint power and quality of experience (HJPQ) algorithm is proposed in this article, where the UEs' offloading delay, MIMO channel, transmission power, as well as UAVs' placement are jointly optimized. The numeral simulations not only reveal the effectiveness of HJPQ, but also guarantee the great quality of experience (QoE) performance for UEs with different priorities. Furthermore, the comparison experiments with random assignment and deep deterministic policy gradient (DDPG) show the superiority of HJPQ in lower complexity, faster convergence, shorter offloading delay as well as higher energy efficiency.

INDEX TERMS UAVs, offloading, quality of experience, HJPQ, DDPG.

I. INTRODUCTION

With the rapid development of ground mobile user equipments (UEs), the data traffic has been witnessed an exponential growth in recent years, which can provide a powerful platform to various applications. However, UEs are still limited by their physical size and suffered from unsatisfied real-time to resist to the contradiction between the computation-intensive requirement and the limit computation capability [1], [2]. For the computing-intensive applications such as simultaneous localization, mapping (SLAM) and virtual reality (VR), offloading task to mobile cloud computing (MCC) or mobile edge computing (MEC) server in parallel is a promising solution to provide location awareness, maintain low latency, support heterogeneity, and ameliorate quality of service (QoS) for real-time applications [3], which enables UEs to offload partial or complete computation-intensive tasks to improve offloading performance on limited battery power and reduce the energy consumption for computing [4]–[6]. After the MEC/MCC server performs computation, the computation results can be transmitted back to the UEs.

Unmanned aerial vehicles (UAVs) are especially suitable for unexpected or temporary events for the features of cost-effective and swiftly deployed [7]. As a result, UAVs have attracted significant interest in assisted wireless network for various application such as data collection, network topology building, energy harvesting, etc [8]–[10]. In addition, UAVs can act as intermediate relays for air-ground integrated mobile edge networks [11] (AGMEN), due to their high

The associate editor coordinating the review of this manuscript and approving it for publication was Pietro Savazzi.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

maneuverability and flexibility for on-demand placement. It has been shown that not only short-distance line-of-sight (LoS) communication links between UAVs and ground UEs can be efficiently exploited in multi-UAV assisted wireless networks for performance enhancement by location assignment, but also UAVs can fly close to the edge UEs who are far away from ground base station (GBS) or none-line-of sight caused by terrain to provide offloading relay service.

However, such mechanism for multi-UAV assisted system induces new issues. First, with multi-UAV and ground mobile UEs in the system, heterogeneous performance requirement as well as fast-changing communication condition make the problem more complicated. Moreover, the system is sensitive to energy consumption for endurance, not only for mobile UEs but also for UAVs themselves [11]. The main issues for multi-UAV network in task offloading are summarized as follows:

1) Different with cellular communication that only has one GBS, there generally exist multi-UAV in the range as aerial relays. UEs are differ in processing capacity, and different types of on-board applications may generate heterogeneous user-perceived QoS, which is also known as quality of experience (QoE). For example, SLAM and control signals should be superior to the general sensing data transmission. While lower-changing data such as temperature and moisture measurements is corresponding more delay-tolerant than that of fast-changing data such as real-time video and audio stream. So an effective optimization algorithm related to UAVs' placement has to be developed to maximize the overall throughput and provide satisfying offloading rate to meet UEs' QoE requirement.
2) Energy efficiency related to system endurance should be considered for both UAVs and mobile UEs in the offloading-and-relay scenario [12]. For the former, UAVs are generally battery-driven and tend to move to a "better" position to improve the channel condition and enhance the transmission rate. However, such movement may consume extra propulsion energy and shorten the flying time. For the latter, it is also expected that the mobile UEs in the ground enjoy a higher transmission rate but with lower power consumption.

In summary, how to allocate UAVs' position with limited energy consumption and intricate offloading QoE requirement is a great challenge. However, UAVs as the wireless communication aerial platform can only provide relay to finite UEs, and moving close to one UE will deteriorate other UEs' channel condition. Hence, multi-UE in the range actually compete for limited service with each other. So the position of UAVs in the system should be optimized to meet all UEs offloading requirement. Meanwhile, the mobility as well as the uncertainty of offloading task, make the problem impossible to reserve resource precisely ahead of time.

In this article, we consider a multi-UAV assisted wireless communication system where multi-UE are enabled to offload computing-intensive tasks to MCC/MEC for fast processing. On the basis, a joint optimization algorithm of communication delay and energy efficiency is proposed. As a non-convex problem, a heuristic scheme is adopted to search the optimal solution and proved to be effective by numeral simulations. The major contributions of the paper are shown as follows:

1) A multi-UAV assisted communication model related to offloading delay and energy efficiency is established. Furthermore, MIMO technology is introduced into the channel model by which data can be transmitted independently and in parallel. As the computation capacity is limited, the offloading task is processed on cloud or cloudlet.
2) To tackle such a mix integer non-linear programming (MINLP) problem, the paper proposes a genetic based heuristic joint power and QoE (HJPQ) algorithm to search the optimal solution. Further, the complexity and convergence of the proposed algorithm are analyzed.
3) As an important member of reinforcement learning (RL) algorithm, deep deterministic policy gradient (DDPG) is adopted for experimental comparison in the features of algorithm convergence and system performance. The numeral simulations demonstrate the superiority of our proposed scheme.

The rest of paper is organized as follows. Sec II reviews related works. In Sec III, the model of multi-UAV assisted aerial offloading scheme is detailed. The mathematical description of HJPQ algorithm is in Sec IV. In Sec V, a series of experiments are carried. The results analysis is also presented in this section. Sec VI analyzes the complexity of both HJPQ and DDPG. Finally, we conclude the paper in Sec VII.

II. RELATED WORK

Recent years, many studies have spotted on how to optimize the UAVs placement to enhance the system performance. The common idea is to model the issue as an optimization problem with different constrains, such as coverage [13]–[15], throughput [16], [17], data rate [18], delay [19]–[21], energy efficiency [22], [23] and etc. According to how to solve such multi-constraints optimization, the work can briefly divided into three categories, i.e., convex based approach, learning based approach and heuristic based approach.

Papers in [17], [18], [22] all adopted successive convex optimization technique. The paper [22] jointly optimize the sensor-nodes(SNs)' wake-up schedule and UAV's trajectory to minimize the energy consumption of all SNs. And authors in [18] proposed a joint optimizing scheme related to UEs scheduling, UAVs trajectory as well as power control to maximize the communication throughput. Different from [22] and [18] that only considered downlink or uplink, the UAV acted as a full-duplex base station in [17] to maximize the sum of uplink and downlink throughput by alternately optimizing the UAV trajectory, downlink/uplink user scheduling, and uplink user transmit power. Paper [19] aimed to max-min average throughput via jointly optimizing the UAV trajectory and OFDMA resource allocation. Although convex can get closed-form solution, it can hardly put into practice due to convex requirement, which is not a general case in complex environment.

Heuristic based approach can provide a feasible way to tackle intricate issues such as paper [14], [15]. Paper [14] proposed a novel genetic algorithm-based 2D placement approach to maximum coverage with consideration of data rate distribution, its drawback is only feasible to 2D movement. Paper [15] took usage of particle swarm optimization (PSO) to adjust flight altitude and beam angle-tomaximizing the coverage under the constraint of transmitting power, however, the mobility of UAVs was not taken into consideration.

Deep Reinforcement Learning (DRL) has recently attracted much attention due to its model-free feature and high learning ability for non-linear approximation feature. Paper [24] aimed at maximizing migration throughput for user by DRL-based scheme with limited UAV energy, however which was only feasible to the single UAV scenario. Similarly, the authors in [25] extended Q-learning to multi-UAV enabled system, in which UAVs can effectively reduce users' total consumptions in terms of time and energy. It is worth noting that UAVs were deployed at a particular height in all the researches above, i.e., only horizontal movement. Fair 3-D placement and energy replenishment policy for multi-UAV with a deep reinforcement learning approach were jointly studied in [23], where UAVs moved around to ensure each UE can be covered.

Regrettably, DRL can solve such multi-constrains optimization issue, however, it need extra on-line or off-line training, and high-complexity induces longer convergence time comparing with heuristic algorithm. Despite of the extensive researches and applications in UAV assisted wireless communication networks, few literature refers to the UAVs placement, channel allocation as well as UEs scheduling with the comprehensively consideration of system QoE requirement such as delay, throughput and energy efficiency. Besides, in literatures like [14], [22], the height of UAVs was fixed in the papers above. Actually, in 3D space, UAVs always tend to fly close to UEs for a better communication quality not only in the horizontal distance but also in the height.

III. SYSTEM MODELS AND PROBLEM DEFINITION

Fig. 1 shows the multi-UAV assisted aerial offloading system, where UAVs are dispatched to provide data offloading relay for ground UEs. The offloading task of UEs can be cooperatively relayed by UAVs and then executed at cloud in parallel to achieve a better real-time performance. Thus the system can enforce resource assignment by selecting properly UAVs as access point. In this section, the model of communication, data offloading as well as mobile energy consumption is described. Based on this, a multi-objective minimum optimization problem is formulated related to UAVs placement, transmission power and UEs scheduling.

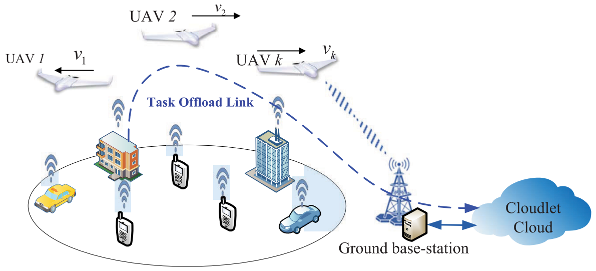


FIGURE 1. Multi-UAV assisted aerial offloading system.

A. COMMUNICATION MODEL

In this model, the placement of UAVs and the transmission power of UEs should be properly controlled and well-considered. As shown in Fig. 1, there are $\mathcal{N} = \{1, 2, \cdots, N\}$ UAVs and $\mathcal{M} = \{1, 2, \cdots, M\}$ ground mobile UEs in the cell. The position of UAVs and UEs are measured by Cartesian coordinate, which are denoted by $C_n = (x_n, y_n, z_n) \in \mathbb{R}^{3\times1}$ and $C'_m = (x_m, y_m) \in \mathbb{R}^{2\times1}$ respectively, where $x_n, y_n, x_m, y_m \in [0, a]$. And z_{max} and z_{min} are the maximum and minimum allowed height of UAVs.

Assuming that the communication channel follows quasi-static fading, i.e., channel's characteristic remains unchanged at each time slot in the offloading period. MIMO technology is used in UAV assisted communication by antenna configuration. In specific, the signal is transmitted and received through multiple antennas at the transmitter and receiver, i.e., UAVs and UEs in this article. The spatial multiplexing and diversity gain are obtained to improve the bit error rate (BER) or data transmission rate of each UE. Meanwhile, MIMO can also multiply the capacity and spectrum utilization of the communication system without increasing the bandwidth. It is reasonable to assume that the antenna spacing is large enough in such a communication model. As a result, the subchannels can be independent and obtain multiplexing gain.

Assuming that UAVs and UEs are equipped with I and J MIMO antennas respectively, and there will be no channel interference during transmission and the data can be transmitted and received simultaneously. Since UAVs act as intermediate relays, it is necessary for UEs to select a proper UAV to offload tasks. Consequently, a decision variable should be used for service matching between UAVs and UEs, to indicate whether to offload and where to offload. The paper takes the flag matrix $\mathbf{U} = [u_{m,n}]$ as decision variable, where $u_{m,n}$ is a binary variable, and $u_{m,n} = 1$ means UAV n provides the offloading relay to UE m. Otherwise, $u_{m,n} = 0$. Due to the power and carriage limitation, the number of antennas at both UAVs and UEs are rare. It is also assumed that each UE only takes one antenna to communication with one UAV once a

TABLE 1. Notation.

$\mathcal{N}$	Number of UAVs
$\mathcal{M}$	Number of UEs
$\mathbf{C} = [C_n]$	Coordinates of UAVs
$\mathbf{U} = [u_{m,n}]$	Decision variable of the offloading task
$\mathbf{P} = [P_{m,n}]$	Transmiting power vector of each channel
p_{mov}	mobility power of each UAV
I	Number of antennas at UAVs
J	Number of antennas at UEs
B_m	Offloading task size in bits of UE m
$\tau_m^{\max}$	Maximum tolerated delay of UE m
$\mathcal{D}_{m,n}$	Distance between UAV n and UE m
$\mathcal{D}_n$	Moving distance of UAVs
$\mathcal{W}_n$	Overall energy consumption of UAV n
$\mathcal{W}_m$	Overall energy consumption of UE m
$L_{m,n}$	Path loss
ξ_{LoS}	Average additional loss
$R_{m,n}$	Offloading rate

time. In summarize, there are the constraints as follows:

$$\sum\nolimits_{m=1}^{M} u_{m,n} \le I, \quad \forall n, \tag{1}$$

$$\sum\nolimits_{n=1}^{N} u_{m,n} \le J, \quad \forall m, \tag{2}$$

$$u_{m,n} \in \{0, 1\}, \quad \forall m, n. \tag{3}$$

B. DATA OFFLOADING DELAY MODEL

The wireless communication between UEs and UAVs consists of front-haul and back-haul links. Due to the fact that the computation result of an intensive task executed at cloud is considered very small, which can be insignificant compared with the size of task itself [26], the paper only considers the offloading front-haul link delay, i.e., task offloading link in Fig.1. Differ from terrestrial channel, air-to-ground channel has large chance of line-of-sight (LoS) connectivity as the altitude of UAVs is much higher than that of ground UEs [27] and the LoS channel of the UAV communication links are much more predominant than other channel impairments, such as small scale fading or shadowing [21]. Moreover, the Doppler frequency shift caused by UAV mobility is assumed to be compensated at the receivers. The altitude of UEs as well as the antenna height are neglected in this model. Based on such a general fading channel model, the channel quality only depends on the UAV-UE distance. So the large-scale channel fading model can be expressed by:

$$L_{m,n}(dB) = 20\log(\frac{4\pi f_c \mathcal{D}_{n,m}}{c}) + \xi_{LoS}, \tag{4}$$

where $L_{m,n}(dB)$ is the average path loss in dB from UE m to UAV n, $\mathcal{D}_{m,n} =\| C_n - C'_m \|$ is the Euclidean distance, i.e., the distance of UAV-UE. c is the speed of light, f_c is the carrier frequency, and ξ_{LoS} is the average additional loss of free space propagation, which is a constant depending on the environment (suburban, urban, dense urban, highrise urban or others) [23]. Let $P_{m,n}$ be the transmission power from UE m to UAV n, so the received power at UAV antennas $\overline{P}_{m,n}$ is:

$$\overline{P}_{m,n} = P_{m,n} - L_{m,n}, \tag{5}$$

$$R_{m,n} = \mathcal{B}\log_2(1 + \frac{\overline{P}_{m,n}}{\sigma^2}), \tag{6}$$

where $R_{m,n}$ is the achievable rate between UE m and UAV n, which can be designed based on the UAV's location, σ^2 is the power of additive white Gaussian noise (AWGN) at the receiver, and $\mathcal{B}$ is the bandwidth of each MIMO channel.

Thus, for UE m, the overall transmission rate denoted by R_m can be expressed as:

$$R_m = \sum_{n=1}^{N} u_{m,n} \cdot \mathcal{B}\log_2(1 + \frac{\overline{P}_{m,n}}{\delta^2}). \tag{7}$$

Generally, computing-intensive applications such as VR, SLAM and video/audio transmission are divided into a series of tasks denoted by $\mathcal{G}_m = \{B_m, \tau_m^{\max}\}$, where B_m is the task size and $\tau_m^{\max}$ is the maximum tolerance delay for UE m. So delay τ_m should satisfy the QoE constraint:

$$\tau_m = \frac{B_m}{R_m} \le \tau_m^{max}. \tag{8}$$

Thus the service matching issue can be treated as a minimization problem:

$$\min_{\mathbf{U},\mathbf{P},\mathbf{C}} \frac{1}{M} \sum_{m=1}^{M} (\tau_m). \tag{9}$$

Eq.(9) is expected that the average delay among all UEs should be minimum.

C. MOBILE ENERGY MODEL

UAVs are battery-driven, which makes the energy consumption be a critical index for acceptable performance, high availability, and economically viable drone small cells (DSCs) coverage [28]. UAVs can not only act as static aerial relays for ground UEs, but also fly close to a specific UE to improve offloading rate.

However, the improvement of one UE's offloading rate will deteriorate the others' because of the propagation pass loss. So UAVs tend to be "lazy" to move and the movement decision should be well balanced according to the different QoE requirement of multi-UE. Moreover, the movement of UAVs in the air needs extra energy consumption, especially the altitude climbing may cost ten times energy comparing with hovering [29].

Considering the problems above, before providing offloading relay to the assigned UEs, UAVs need to move to the most suitable position for energy optimization. So the movement $\mathcal{D}_n(t)$ of UAV n is defined by:

$$\mathcal{D}_n =\| D_n[0] - D_n[F] \|, \tag{10}$$

where ‖ denotes the Euclidean norm. The initial location of UAV n is assumed to be pre-determined and denoted as $D_n[0]$, and the final position of UAV n is $D_n[F]$.

Supposing that the energy consumption by unit movement of UAV n is denoted by w_n, which is given by:

$$w_n = p_{mov}/\vartheta, \tag{11}$$
$$\mathcal{W}_n = w_n \mathcal{D}_n, \tag{12}$$

where p_{mov} denotes the mobility power and ϑ represents the velocity of UAVs. Further, with the movement distance, the mobility energy consumption of UAV n can be known as $\mathcal{W}_n$. Since UAVs are driven by battery, in order to ensure the quality of service and maintain the lifetime, the flight time should not be too long. It is assumed that the maximum flight time is T, then $\mathcal{W}_n$ for each UAV should satisfy the constraint $\mathcal{W}_n \leq p_{mov}T$.

For UE m, the overall energy consumption during the data transmission can be expressed as:

$$\mathcal{W}_m = \tau_m \sum_{n=1}^{N} u_{m,n} \cdot P_{m,n}. \tag{13}$$

As the total energy of each UE consumed for data transmission is quite limited due to equipment specification, the transmit power constraint and the total energy constraint for UE m can be described as:

$$\mathcal{P}_m = \sum_{n=1}^{N} u_{m,n} P_{m,n} \leq \mathcal{P}_m^{\max}, \tag{14}$$
$$\mathcal{W}_m \leq \tau_m^{max} \mathcal{P}_m^{\max}. \tag{15}$$

So the overall energy consumption on the multi-UAV assisted system can be attained as:

$$E_{over} = \alpha \sum_{m=1}^{M} \mathcal{W}_m + (1-\alpha) \sum_{n=1}^{N} \mathcal{W}_n, \tag{16}$$

where $\alpha \in [0, 1]$ is the cost weight. Similarly, it is expected that the average energy consumed by all UAVs and UEs can be minimized, which can be written as:

$$\min_{\mathbf{U},\mathbf{P},\mathbf{C}} \alpha \frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_m + (1-\alpha) \frac{1}{N} \sum_{n=1}^{N} \mathcal{W}_n. \tag{17}$$

D. PROBLEM FORMULATION

According to the system model mentioned above, an optimization problem is formulated in this subsection. Since the data size of computing results from cloud or cloudlet is much smaller compared to the offloading task itself, the delay and energy consumption caused by sending back results to ground UEs can be neglected. Thus the task offloading problem can be formulated into a joint optimization of energy consumption and tolerant delay in the uplink, i.e., the multi-objective minimization problem of Eq.(9) and Eq.(17). The variable parameters to be optimized are the decision variable $\mathbf{U}$, the transmission power $\mathbf{P}$ of UEs and the placement $\mathbf{C}$ of UAVs, which yield the following problem:

$$\text{P1}: \min_{\mathbf{U},\mathbf{P},\mathbf{C}} \begin{cases} \frac{1}{M} \sum_{m=1}^{M} (\tau_m) \\ \alpha \frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_m + (1-\alpha) \frac{1}{N} \sum_{n=1}^{N} \mathcal{W}_n \end{cases} \tag{18}$$
$$\text{s.t. } u_{m,n} \in \{0, 1\}, \quad \forall n, m, \tag{C1}$$
$$\sum_{m=1}^{M} u_{m,n} \leq I, \quad \forall n, \tag{C2}$$
$$\sum_{n=1}^{N} u_{m,n} \leq J, \quad \forall m, \tag{C3}$$
$$\mathcal{P}_m = \sum_{n=1}^{N} u_{m,n} P_{m,n} \leq \mathcal{P}_m^{\max}, \quad \forall m, \tag{C4}$$
$$\tau_m \leq \tau_m^{\max}, \quad \forall m, \tag{C5}$$
$$\mathcal{W}_n \leq p_{mov}T, \quad \forall n, \tag{C6}$$
$$\mathcal{W}_m \leq \tau_m^{max} \mathcal{P}_m^{\max} \quad \forall m, \tag{C7}$$

where C1 is related to the binary decision of offloading channel. C2 indicates that each UAV can only assist at most I UEs for the limitation of antennas. Similarly, C3 addresses that each UE can only offload task to at most J UAVs simultaneously. C4 means that the overall transmission power at UE m is up-bounded by $\mathcal{P}_m^{\max}$. C5 guarantees the maximum delay on UE m. C6 states that UAV n cannot exceed the maximum available mobility energy, and C7 limits the up-bounded of $\mathcal{W}_n$ and $\mathcal{W}_m$ respectively.

Generally, the multiple-objective optimal problem P1 can be converted into a simple objective optimization by linear weighted summation, which can be reformulated as follows:

$$\begin{aligned} \text{P2}: \min_{\mathbf{U},\mathbf{P},\mathbf{C}} \mathbb{C} = \gamma \frac{1}{M} \sum_{m=1}^{M} (\tau_m) + (1-\gamma) \cdot [\alpha \frac{1}{M} \sum_{m=1}^{M} \mathcal{W}_m \\ +(1-\alpha) \frac{1}{N} \sum_{n=1}^{N} \mathcal{W}_n] \\ \text{s.t. } (C1), (C2), (C3), (C4), (C5), (C6), (C7), \end{aligned} \tag{19}$$

where $0 < \gamma < 1$ is used to achieve the maximum throughput while minimizing the energy consumption, it is clear that problem P2 is a mix integer non-linear programming (MINLP) problem,, which is challenging since it contains discrete binary variables $u_{m,n}$ and high coupling constraints.

IV. GA BASED JOINT OPTIMIZATION ALGORITHM

In this section, we focus on solving the problem P2. The objective function is related to the decision variable $\mathbf{U}$, the transmission power $\mathbf{P}$ and the position of UAVs $\mathbf{C}$. However, the joint optimization is difficult to solve due to $\mathbf{P} \in \mathcal{R}^{\mathcal{M}}$ and $\mathbf{C} \in \mathcal{R}^{3\times\mathcal{N}}$ are continuous vectors, while

$\mathbf{U} \in \mathcal{Z}^{\mathcal{M}\times\mathcal{N}}$ is a binary matrix, which makes the problem P2 be non-smooth and non-differential or continuous. There will be $2^{\mathcal{M}\times\mathcal{N}}/IJ$ permutations for $\mathbf{U}$, and for each permutation, the transmission power $\mathbf{P}$ and UAVs' location $\mathbf{C}$ also need to be allocated and deployed.

For such a MINLP problem, i.e., NP-hard problem, reinforcement learning algorithms such as Q-learning and deep Q-network (DQN) [30], deterministic policy gradient (DPG) [31], [32] are proposed for optimization. However, they are lack of QoE consideration and have large complexity in time and space, which is proved by the comparison experiments in Section V and the algorithm complexity is analysed in Section VI. Besides, there are many meta-heuristic algorithms such as simulated annealing (SA), iterated local search (ILS), particle swarm optimization (PSO) [33], [34] and other intelligent swarm algorithm [35] with lower computational complexity to mitigate such issue.

In particular, genetic algorithm (GA) has an ability to enable the searching behaviour to jump out of local extreme points and obtain the global optimization solution. Thus in this article, a genetic based heuristic joint power and QoE (HJPQ) algorithm is proposed, which can obtain the global optimization solution by iteratively searching a better artificial population.

GA is a randomly searching algorithm to tackle troublesome problems by mimicking biological evolution which takes the survival fittest principle. At each step, the genetic algorithm randomly selects several individuals from the current population as parents, and then generates offspring population. After several successive generations, the population is evolved towards an optimal solution.

There are three key elements in genetic algorithm, i.e., "selection", "crossover", and "mutation". GA starts with an initial set of solutions and products optimization iteratively through genetic operations until reaching the maximum iteration number $\mathbb{G}$ or convergence. Especially, the crossover and mutation operations of GA can maintain the population diversity and extend the searching region, so that it is not easy to fall into local optimal points. For this reason, it is impactful in searching the global domain. HJPQ proposed in this article is summarized in Algorithm 1 and the details is described as follows:

1) Chromosome: In HJPQ algorithm, each chromosome corresponds to a solution, i.e., an encoded individual. Binary encoding is used in this article. The continuous variable $\mathbf{P} = \{P_{m,n}, m \in \mathcal{M}, n \in \mathcal{N}\}$, $\mathbf{C} = \{C_n, n \in \mathcal{N}\}$ are rounded into integers and the binary matrix $\mathbf{U} = \{u_{m,n}, m \in \mathcal{M}, n \in \mathcal{N}\}$ is interpreted into binary chromosome by 16bits Gray-code. So that each individual can be represented by a $\mathcal{N}(17\mathcal{M} + 48)$ bits long binary vector as genes in a chromosome.

$$\mathbf{I}^{\text{binary}} = [\mathbf{U}^{\text{binary}}_{\mathcal{M}\times\mathcal{N}\times 1}, \mathbf{P}^{\text{binary}}_{16\times\mathcal{M}\times\mathcal{N}}, \mathbf{C}^{\text{binary}}_{16\times 3\times\mathcal{N}}]. \tag{20}$$

Furthermore, define the coding and decoding function used in Algorithm 1, i.e. $\mathbf{I}^{\text{binary}} = \mathcal{F}_{\text{code}}(\mathbf{U}, \mathbf{C}, \mathbf{P})$ and $[\mathbf{U}, \mathbf{C}, \mathbf{P}] = \mathcal{F}_{\text{decode}}(\mathbf{I}^{\text{binary}})$. Then set the size of population $\mathbb{T}$, crossover probability p_c and mutation probability p_m according to model parameters, where the superscript g denotes the number of generation. Function $\mathbb{R}$ is adopted to evaluate the fitness of each individual, which can be given by:

$$\mathbb{R}^g_t = e^{-\mathbb{C}}, \tag{21}$$

where $\mathbb{C}$ is defined in Eq.(19).

2) Selection: In fact, roulette wheel is widely used for selection operation [36]. In roulette wheel, each individual is allocated to a simulated wheel size, which is in proportion to the fitness value of individual to determine the amount of each individual inherited into the next generation population, which can be expressed as:

$$P(\mathbb{R}^g_t) = \frac{\mathbb{R}^g_t}{\sum_t^{\mathbb{T}} \mathbb{R}^g_t}, \tag{22}$$

so the cumulative probability of each individual q_t can be expressed as:

$$q^g_t = \sum_{t=1}^{t} P(\mathbb{R}^g_t). \tag{23}$$

Then, generate a pseudo-random number r obeyed uniform distribution, which is belonged to [0,1]. And next find the interval of r in the roulette, i.e., the selected individual. Therefore, it is reasonable for an individual with a higher fitness to be selected. At last, repeat this step until the number of individuals in the new population is equal to the size of parent population.

3) Crossover: Crossover is the key operation for generating new individuals in HJPQ algorithm, i.e., selecting genes from the parent chromosome and generating new offspring to improve the fitness. For example, as shown in Fig.2, there are i crossover points, and $k_i \in \{1, 2, \cdots, l-1\}$, where k is the crossover point and l is the length of chromosome. These crossover points are selected by random numbers, which are not repetitive and arranged in ascending order. The genes in parent chromosome are exchanged between two crossover points, resulting in two new offspring. There is no exchange between the individual's first gene and the first crossover point, i.e., odd regions remain unchanged.

4) Mutation: After performing the crossover, mutation is implemented to randomly change the offspring to prevent all solutions in a population from falling into a local optimum. This step is another operation to generate new individuals. Simple mutation is taken in this algorithm, which refers to performing mutation operation on a certain point or a few genes randomly designated by individual chromosome. An example of mutation is shown in Fig.3. If the original gene value is 0, the mutation operation will change the gene value

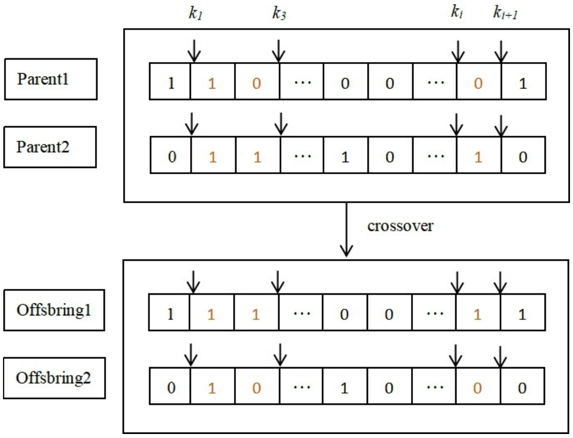


FIGURE 2. Crossover of chromosome.

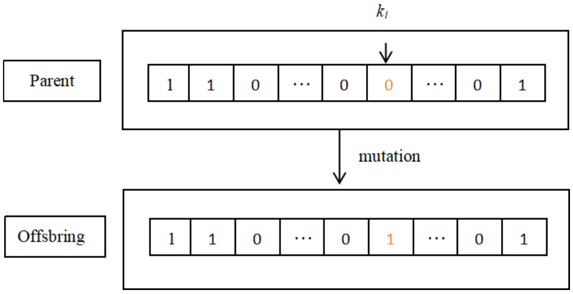


FIGURE 3. Mutation of chromosome.

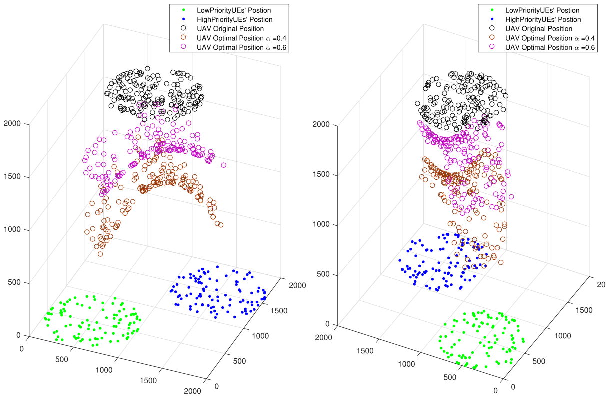
to 1. Otherwise, if the original gene value is 1, the mutation operation will change it to 0, i.e., binary-reverse.

V. SIMULATION RESULTS AND DISCUSSIONS

In this section, we display simulations to analyze the performance of HJPQ, which includes UAVs location distribution and differentiated QoE performance in the dynamic environment. Comparison experiments with other benchmark algorithms are also presented in this section. The Matlab and Python programming tools are used for HJPQ and the DDPG algorithm respectively.

A. UAVS LOCATION DISTRIBUTION

As shown in Fig.4, UEs and UAVs are randomly distributed in a disc of 1km radium. Supposing that there are $\mathcal{M} = 150$ UAVs providing offloading aerial-relay service for $\mathcal{N} = 200$ ground UEs in the range. Considering that both UAVs and UEs are located in a 3D cube area with the size of 2000m × 2000m × 2000m. The corresponding parameters configuration in HJPQ is listed in Tab.2. The minimum and maximum allowed flying height of UAVs are 500m and 2000m respectively, and the distribution centre for UAVs is (1000, 1000, 2000). UEs are equally divided into two types: the first is marked as high priority with a smaller delay tolerance $\tau_1^{\max}$ and the distribution centre is (500, 500, 0). The other type belongs to low priority with a larger delay tolerance $\tau_2^{\max}$, which distribution centre is (1500, 1500, 0).



(a) Perspective View 1 (b) Perspective View 2

FIGURE 4. The optimized UAVs' distribution by HJPQ.

TABLE 2. Parameters of HJPQ.

Parameters Config	Values
Population Size	$\mathbb{T} = 300$
Max Generation	$\mathbb{G} = 1400$
Select Probability	$r = 0.9$
Crossover probability	$P_c = 0.7$
Mutation probability	$P_m = 0.1$

So the delay tolerance proportion is fixed in this article, i.e., $\tau_1^{\max}/\tau_2^{\max} = 0.5$.

First, it can be seen from Fig.6 that the total reward of fitness function in HJPQ gradually increases and converges by almost 320 steps of evolution, which confirms the effectiveness of our algorithm.

More concretely, we show the 3D distributions of UAVs' optimal position from two perspective views in Fig.4(a) and Fig.4(b) respectively:

1) UAVs always tend to move towards UEs based on Eq.(16). But at the same time, due to the movement of UAVs is limited by energy consumption, the position of UAVs is saddle-distribution.
2) The priority difference of UEs will infer the location distribution of UAVs, which means UAVs are bias to the side of UEs with high priority to reduce the offloading delay.
3) It is worth pointing out that as parameter α changing from 0.6 to 0.4, the movement weight caused by energy consumption increases from 0.4 to 0.6, which means that UAVs prefer to minimize the energy consumption caused by movement to balance the overall cost.

B. DIFFERENTIATED QOE PERFORMANCE

Since UEs are divided into two priorities, a set of experiments are producted to verify the system differentiated QoE performance, i.e., offloading delay and energy efficiency on UEs and movement distance on UAVs. The task size of UEs is assigned changing from 1Mbits to 10Mbits. The delay

Algorithm 1 HJPQ Algorithm

```
Input: Size of the population $\mathbb{T}$, Maximum generation
       $\mathbb{G}$, Crossover probility $p_c$, Mutation probility $p_m$
Output: $\arg\max_{\mathbf{U},\mathbf{C},\mathbf{P}} \mathbb{R}$
Initialization:
Randomly initialize $\mathbb{T}$ sets of optimization's variables
  $[\mathbf{U},\mathbf{C},\mathbf{P}]_t$, $\{t = 1, \cdots, \mathbb{T}\}$ as the initial population
  with constrains C1-C5.
Coding the $[\mathbf{U},\mathbf{C},\mathbf{P}]_t$ in to $I_t^{\text{binary}}$ by function $\mathcal{F}_{\text{code}}$.
  $\mathbb{I}^{g=1} = \{I_1^{\text{binary}}, \cdots I_{\mathbb{T}}^{\text{binary}}\}$
while $g \leq \mathbb{G}$ do
    Calculate the fitness value of each individual $\mathbb{R}_t^g$
      according $\mathcal{F}_{\text{decode}}$ and Eq.(21)
    Calculate the selection probability for each
      chromosome by Eq.(22)
    Save the best fitness $R_\star^g$ and the corresponding
      individual $I_\star^{\text{binary}}$
    if $||R_\star^g$ -$R_\star^{g-1}|| \leq \delta$ then
        return $\mathcal{F}_{\text{decode}}(I_\star^{\text{binary}})$
    end
    Selection: Randomly choose $\mathbb{T}$ chromosomes as a
      new population $\hat{\mathbb{I}}^g$ by Roulette Wheel selection
      according to Eq.(23).
    Crossover: For every two pair of individuals in $\hat{\mathbb{I}}^g$,
      take multi-point crossover at every gene position
      with probability $P_c$.
    Mutation: For every individual in $\hat{\mathbb{I}}^g$, take
      binary-reverse at every gene position with the
      probability $P_m$.
    $\mathbb{I}^g \leftarrow \hat{\mathbb{I}}^g \cap I_\star^{\text{binary}}$;
    $g = g + 1$
end
return $\mathcal{F}_{\text{decode}}(I_\star^{\text{binary}})$
```

tolerance proportion is still fixed to $\tau_1^{\max}/\tau_2^{\max} = 0.5$. The results are shown in Fig.5.

1) Fig.5(a) shows that whether low or high priority, the average delay of UEs increases with the data size of offloading task. It is worth noting that although UEs with high priority always enjoy a better offloading delay, the delay ratio maintains constant which is approximately equal to the fixed value 0.5. It implies that HJPQ algorithm can not only guarantee the QoE of high priority, but also avoid the "over-sacrifice" of low priority.
2) Fig.5(b) shows the energy consumption of UEs with different priorities along with the increasing of offloading data size. Opposed to Fig.5(a), UEs with high priority present low energy efficiency. That is because they have to consume more transmission power for a better transmission rate, which leads to a lower energy efficiency. Nevertheless, the trade-off is still meaningful because delay constraint is the most important one which determines whether the task will be success or fail.
3) Fig.5(c) shows that the movement distance of UAVs which serve UEs with high priority is longer than those serving low priority. That is because UAVs have to fly closer to high priority UEs for a better transmission rate. Meanwhile, we can further conclude that as the data size climbs, all UAVs will move more distance to UEs for better transmission rate to meet delay requirement.

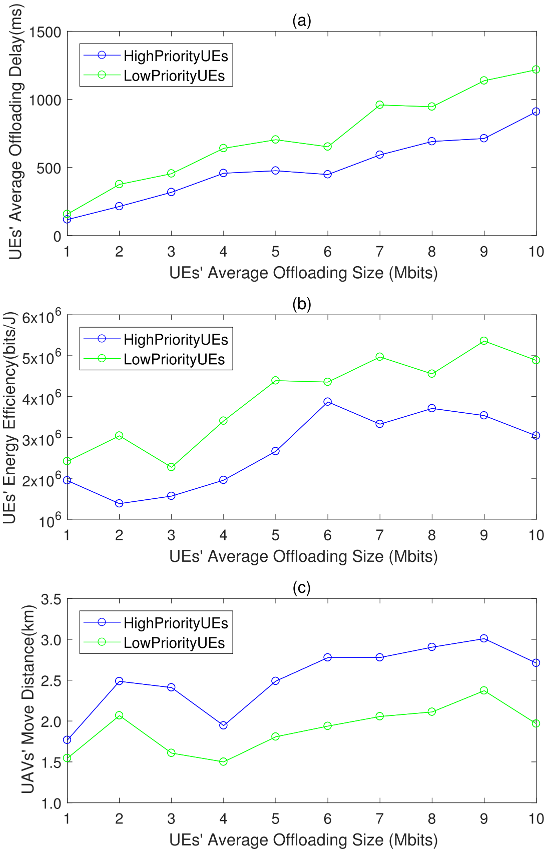


FIGURE 5. QoE performance along with offloading data size.

C. PERFORMANCE COMPARISON WITH LEARNING BASED OPTIMIZATION

DRL is a promising technology to tackle the non-convex issue by training neuron network to interact with the environment by itself and learning from its mistake. Both DRL and heuristic algorithms can provide a feasible way to MINP, so this manuscript take proposed HJPQ and deep deterministic policy gradient (DDPG) [23] for comparison. Different from typical DRL algorithm like DQN that quantizes the time sharing into a finite discrete action space with low dimension, DDPG can solve the optimization issue in a high dimensional and consecutive space.

Assume that T is divided into k equal time slot δ_t, i.e., $T = k\delta_t$, which can be denoted as $T = \{1, 2, \cdots k\}$ for simple. During a time slot, each UAV serves its associated ground

UEs to offload task. Particularly, the UAV's location changing within a time slot can be assumed negligible compared with the link distances from UAV to all ground UEs, i.e., the distribution of $L_{m,n}(dB)[n]$ keeps unchanged within a time slot but varies over different time slots.

DDPG which can directly use the raw observations to learn, and use fewer steps of experience learning than DQN in the Atari domain [37], contains an actor network for generating actions, and a critic network for judging the quality of actions. Each network consist of two sub-networks, i.e., online network and target network. Actor online network $\mu(s|\theta^{\mu})$ is responsible for iterative updating the policy network parameter θ^{μ} and selecting the current action a according to the current state s. While actor target network $\mu'(s|\theta^{\mu'})$ updates the network parameter $\theta^{\mu'}$ and $Q'(s, a|\theta^{Q'})$. Critic online network $Q(s, a|\theta^{Q})$ is responsible for iterative updating the critic network parameter θ^{Q} and calculating the current Q value. While critic target network $Q'(s, a|\theta^{Q'})$ calculates Q', which is used to calculate the target Q value and soft update $\theta^{Q'}$.

The algorithm also adopts experience pool, random sampling and target network, so the real-Q value is actually calculated by two target networks. In the framework of DDPG, state s_k, action a_k and reward r_k are modelled as following:

1) State: s_k is defined as the observation space at time slot k, which includes the flag matrix of offloading task, the position of UAVs and overall power consumption.

$$s_k = \{u_{m,n}[k], P_{m,n}[k], C_n[k]\}.$$

2) Action: a_k is based on the movement of UAVs. Supposing that $\Delta_n[k] = (\psi_n[k], \phi_n[k], d_n[k])$ denotes the polar angle, azimuthal angle and move distance in polar coordinates at time slot k, and $\Delta P_{m,n}[k]$ is the increment of transmission power. So a_k can be expressed as:

$$a_k = \{\Delta_n[k], \Delta P_{m,n}[k]\}.$$

3) Reward: r_k denotes the system reward at the time slot k, which is calculated by the environment after taking the action a_k with state s_k. Similar to HJPQ, the largest reward is expected to be the learning goal according to Eq.(21).

DDPG based optimization algorithm is summarized in Algorithm 2, which adopts a deterministic behaviour strategy a_k in each time slot k, and adds noise $\mathcal{N}_k$ to explore better strategies potentially to the action, which can be depicted as $a_k = \mu(s_k|\theta^{\mu}) + \mathcal{N}_k$. The environment will execute a_k, return r_k and produce new state s_{k+1}. The replay buffer (RB) stores transition data (s_k, a_k, r_k, s_{k+1}) and randomly samples $\mathbb{L}$ transition data from RB as a mini-batch training data to optimize policy. In the critic network, the parameter θ^{Q} is updated by using the temporal difference (TD) error method, and the loss function is formulated as:

$$L = \frac{1}{N}\sum_{i}(y_i - Q(s_i, a_i|\theta^{Q}))^2, \tag{24}$$

where y_i is given as

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}). \tag{25}$$

Algorithm 2 DDPG Offloading Optimization Algorithm

```
Input: Training epoch length $L$, critic learning rate $r_c$,
       actor learning rate $r_a$; discount factor $\gamma$, soft
       update factor $\tau$; replay buffer $RB$, mini-batch
       size $\mathbb{L}$; Gaussian distributed behavior noise $\mathcal{N}_k$
Initialization:
Random Initialize actor network $\mu(s|\theta^{\mu})$ and critic
network $Q(s, a|\theta^{Q})$ with $\theta^{\mu}$ and $\theta^{Q}$;
Initialize target network $Q'$ and $\mu'$ with $\theta^{Q'} \leftarrow \theta^{Q}$,
$\theta^{\mu'} \leftarrow \theta^{\mu}$;
Empty replay buffer $RB$;
for $episode = 1, L$ do
    Initialize a Gaussian noise $\mathcal{N}_k$ with $mean = 0$ and
    variance $var \leftarrow 3$.
    Receive initial oberservision state $s_0$.
    for $T = \{0, 1, \cdots, k\}$ do
        Select action $a_k = \mu(s_k|\theta^{\mu}) + \mathcal{N}_k$;
        Execute action $a_k$ and calculate reward $r_l$ and
        get new state $s_{k+1}$;
        Store transition $(s_k, a_k, r_k, s_{k+1})$ in $RB$;
        Sample a random minibatch of $N$ samples
        $(s_i, a_i, r_i, s_{i+1})$ from $RB$;
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$;
        Update critic network by minimizing the loss
        by Equ.24;
        Update the actor policy by the sampled policy
        gradient by Equ. 27;
        Update the target networks $\theta^{\mu'}$, $\theta^{Q'}$ by
        Equ.28;
        if all UEs' data is offloaded then
            break;
        end
    end
end
```

For the actor network, the estimation of policy gradient $\nabla_{\mu}J$ can be written as:

$$\nabla_{\theta^{\mu}} J_{\beta}(\mu) \approx E_{s\sim\rho^{\beta}}[\nabla_a Q(s, a|\theta^{Q})|_{a=\mu(s)} \cdot \nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})], \tag{26}$$

Further, the minibatch (s_i, a_i, r_i, s_{i+1}) can be obtained by random sampling from RB. As a result, $\nabla_{\mu}J$ can be rewritten as Equ.27:

$$\nabla_{\theta^{\mu}} J_{\beta}(\mu) \approx \frac{1}{\mathbb{L}}(\nabla_a Q(s, a|\theta^{Q})|_{s=s_i, a=\mu(s_i)} \cdot \nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s=s_i}). \tag{27}$$

The running average method is used to soft update the target network parameters as Equ.28, which makes the learning

TABLE 3. Network parameters of DDPG.

Parameters Config	Number	Size	Acti-func	Network
Input Layer	1	$\chi_{a,0} = 2\mathcal{M}\mathcal{N} + 3\mathcal{N}$	Relu	Actor
Hidden Layer	3	$\chi_{a,1:3} = 300,$	Relu6	
Output Layer	1	$\chi_{a,4} = 300, \chi_a{}^*$	sigmoid	
Input Layer	1	$\chi_{c,0} = 2\mathcal{M}\mathcal{N} + 6\mathcal{N} + \mathcal{M}$	Relu	Critic
Hidden Layer	3	$\chi_{c,1:3} = 300$	Relu6	
Output Layer	1	$\chi_{c,4} = 300, \chi_c{}^*$	NA	

[1] $\chi_{a,i}$ and $\chi_{c,j}$ present the input size of each layer in actor network and critic network respectively.
[2] $\chi_a{}^*$ is the output size of output layer in actor network, i.e., $\chi_{a,5} = 3\mathcal{N} + \mathcal{M}$.
[3] $\chi_c{}^*$ is the output size of output layer in critic network, i.e., $\chi_{c,5} = 1$.

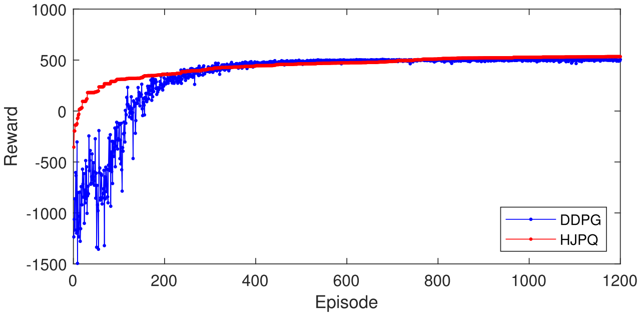


FIGURE 6. The system overall reward converge with DDPG.

process more stable and easier to converge.

$$\begin{cases} \theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}, \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}, \end{cases} \tag{28}$$

where τ is 0.001 generally.

In this article, both actor network and critic network are feed-forward fully connected neural network and the corresponding parameters configuration for DDPG are listed in Tab.3. The training is executed with 1200 episodes by the example of 150 UEs. The episode will terminate when all UEs' data is offloaded or the length of episode is longer than k. Gaussian noise is used with mean 0 and variance *var*. The value of *var* is 3 in the beginning and times 0.9995 after each time slot. *RB* is set to 10000 and the size of minibanch $\mathbb{L}$ is 64.

The reward changing with episode for HJPQ and DDPG is shown in Fig.6. It can be observed that both them can converge to an optimal solution. However, compared with HJPQ (converges in almost 320 steps gently), DDPG takes more time (about 400 episodes) to converge and exits jitter.

Taking random assignment as the benchmark, Fig.7 shows the UAVs' movement distance, UEs' delay, and system energy efficiency of HJPQ and DDPG, where $\tau_1^{\max}$ and $\tau_2^{\max}$ are fixed to be 300ms and 500ms respectively and the energy efficiency equals to data volume divided by energy consumption.

1) Fig.7(a) shows the UAVs moving distance in three different algorithms. It is obvious that UAVs in random assignment have to fly the longest distance for there is no trajectory optimization. HJPQ presents slightly superior to DDPG because it can find the best position in real time instead of iterated operation step by step.
2) Fig.7(b) implies that random flight does not care about the delay constraint for different UEs priorities, i.e., without the ability to provide QoE service. On the contrast, even the number of UEs increasing from 100 to 200, both HJPQ and DDPG can guarantee the delay ratio to the fixed value. However, as the number of UEs increasing, the advantage of HJPQ comes into sight because UAVs can find the best position for the maximum transmission rate more quickly, which is also related to the converge time in Fig.6.
3) Fig.7(c) shows that no matter which algorithms, the energy efficiency always enhances with the increasing number of UEs, which is because the cover probability of UEs will enhance correspondingly. HJPQ is able to minimize the UE's transmission power, the mobile energy consumption of UAVs and optimize UAVs trajectory through effective constraints on offloading delay, which leads to an outperformed energy efficiency compared with other algorithms. It is worth noticed that when the number of UEs up to 200, the energy efficiency of HJPQ is lower than that of DDPG. That is because in order to meet the delay constraint of UEs (as shown in Fig.7(b)), HJPQ will prompt UEs to enhance the transmission power to reach a better transmission rate, which causes more energy consumption and deteriorates the energy efficiency.

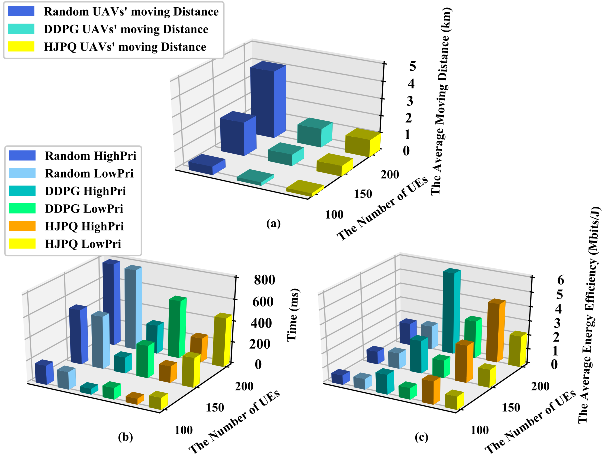


FIGURE 7. The comparison of mobile distance, delay and energy efficiency over different algorithms and UEs' number.

VI. COMPLEXITY ANALYSIS

According to the description above, the algorithm complexity HJPQ and DDPG is analyzed as follows:

A. HJPQ COMPLEXITY ANALYSIS

1) In HJPQ, the maximum generation is $\mathbb{G}$, and $\mathbb{T}$ individuals in the population are updated iteratively with $\mathbb{G}$. First, HJPQ calculates the fitness value of each individual, which loops T times. Then each chromosome updates itself through selection, crossover and

mutation, and the cycle times are $\mathbb{T}$, $\mathbb{T}/2$ and $\mathbb{T}$ respectively. Therefore, the overall time complexity is:

$$O(\mathbb{G}(\mathbb{T}+\mathbb{T}+\mathbb{T}+\mathbb{T}/2))) = O(\mathbb{G}\mathbb{T}). \tag{29}$$

2) The size of population is $\mathbb{T}$. Each chromosome is composed of a binary code, which length is $\mathcal{N}(17\mathcal{M}+48)$ in the population. The space complexity of the HJPQ algorithm is obviously the total length of the population, i.e.:

$$O(\mathbb{T}log(\mathcal{N}(17\mathcal{M}+48))) = O(\mathbb{T}log(\mathcal{N}\mathcal{M})). \tag{30}$$

B. DDPG COMPLEXITY ANALYSIS

DDPG mainly includes a replay buffer and four neural networks. Assuming that the actor network contains $\mathcal{I}$ fully connected layers and the critic network contains $\mathcal{J}$ fully connected layers. Thus the time complexity and space complexity of DDPG can be derived with regard to floating point operations per second(FLOPS) [37].

1) The neural networks for every layer have a vector $\chi_{a,i}$ and a matrix $\chi_{a,i} \times \chi_{a,i+1}$ for a fully connected layer to perform dot product. The FLOPS computation is $(2\chi_{a,i}-1)\times\chi_{a,i+1}$, i.e., multiply $\chi_{a,i}$ times and add $\chi_{a,i}-1$ times. Activation layers also should be taken into consideration, which is calculated without dot product. It is only measured by FLOPS, where addition, subtraction, multiplication, division, exponent, square root, etc, are counted as a FLOPS. So the time complexity can be defined as follows:

$$\begin{aligned}2\sum_{i=0}^{\mathcal{I}-1}&((2\chi_{a,i}-1)\chi_{a,i+1}+\kappa\chi_{a,i+1})\\ &+2\sum_{j=0}^{\mathcal{J}-1}((2\chi_{c,j}-1)\chi_{c,j+1}+\kappa\chi_{c,j+1})\\ &=o(\sum_{i=0}^{\mathcal{I}-1}\chi_{a,i}\chi_{a,i+1}+\sum_{j=0}^{\mathcal{J}-1}\chi_{c,j}\chi_{c,j+1}),\end{aligned} \tag{31}$$

where κ is the corresponding parameter of the activation layer [37].

2) The experience replay buffer *RB* in DDPG occupies some space to store the transition, hence the space complexity is the memory space of RB, which can be defined as $\mathcal{Q}$. For a fully connected layer in both the actor network and critic network, there are a $\chi_{a,i}\times\chi_{a,i+1}$ matrix and a $\chi_{a,i+1}$ bias vector. Hence, the memory of one fully connected layer is $(\chi_{a,i}+1)\chi_{a,i+1}$. So the space complexity of the neural networks can be written as follows:

$$\begin{aligned}\sum_{i=0}^{\mathcal{I}-1}&(\chi_{a,i}+1)\chi_{a,i+1}+\sum_{j=0}^{\mathcal{J}-1}(\chi_{c,j}+1)\chi_{c,j+1}+\mathcal{Q}\\ &=o(\sum_{i=0}^{\mathcal{I}-1}\chi_{a,i}\chi_{a,i+1}+\sum_{j=0}^{\mathcal{J}-1}\chi_{c,j}\chi_{c,j+1})+o(\mathcal{Q}).\end{aligned} \tag{32}$$

Combined with Tab.2 and Tab.3, the above analysis shows that HJPQ has much less complex than DDPG indeed.

VII. CONCLUSION

In multi-UAV assisted system, UAVs can help UEs to offload traffic to cloudlet or cloud for fast execution. The paper focuses on the UEs' offloading and UAVs' intermediate relay scheme with dual constrains of QoE and battery limitation. Which can be formulated into a non-convex and mixed-integer optimized problem. For this issue, the paper proposes a HJPQ algorithm related to UEs' offloading delay, MIMO channel, transition power assignment as well as UAV's placement. The numeral simulations demonstrate that even in the dynamic environment (the number and offloading task size of UEs are changing), HJPQ still can not only minimize the overall system cost, but also meet different QoE requirement.

REFERENCES

[1] T.-Y. Kan, Y. Chiang, and H.-Y. Wei, "Task offloading and resource allocation in mobile-edge computing system," in *Proc. 27th Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2018, pp. 1–4.

[2] X. Zhang, Y. Zhong, P. Liu, F. Zhou, and Y. Wang, "Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization," *IEEE Access*, vol. 7, pp. 113345–113354, 2019.

[3] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017.

[4] J. Shuja, S. Mustafa, R. W. Ahmad, S. A. Madani, A. Gani, and M. Khurram Khan, "Analysis of vector code offloading framework in heterogeneous cloud and edge architectures," *IEEE Access*, vol. 5, pp. 24542–24554, 2017.

[5] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 140–146, Feb. 2020.

[6] F. Zhou and R. Q. Hu, "Computation efficiency maximization in wireless-powered mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3170–3184, May 2020.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[9] L. Jingnan, L. Pengfei, and L. Kai, "Research on UAV communication network topology based on small world network model," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2017, pp. 444–447.

[10] Q. Zhang, Z. Wang, P. Zhang, H. Zhang, X. Wan, and Z. Fan, "Sum energy maximization for UAV-enabled wireless power transfer networks with nonlinear energy harvesting model," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Automat. Control Conf. (ITNEC)*, Jun. 2020, pp. 1417–1420.

[11] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[12] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

[13] M. Alzenad and H. Yanikomeroglu, "Coverage and rate analysis for unmanned aerial vehicle base stations with LoS/NLoS propagation," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.

[14] X. Zhong, Y. Huo, X. Dong, and Z. Liang, "QoS-compliant 3-D deployment optimization strategy for UAV base stations," *IEEE Syst. J.*, early access, Aug. 24, 2020, doi: 10.1109/JSYST.2020.3015428.

[15] B. Li, C. Chen, R. Zhang, H. Jiang, and X. Guo, ''The energy-efficient UAV-based BS coverage in Air-to-Ground communications,'' in *Proc. IEEE 10th Sensor Array Multichannel Signal Process. Workshop (SAM)*, Jul. 2018, pp. 578–581.

[16] M. D. Nguyen, T. M. Ho, L. B. Le, and A. Girard, ''UAV placement and bandwidth allocation for UAV based wireless networks,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[17] M. Hua, L. Yang, C. Pan, and A. Nallanathan, ''Throughput maximization for full-duplex UAV aided small cell wireless systems,'' *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 475–479, Apr. 2020.

[18] Q. Wu, Y. Zeng, and R. Zhang, ''Joint trajectory and communication design for multi-UAV enabled wireless networks,'' *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[19] Q. Wu and R. Zhang, ''Common throughput maximization in UAV-enabled OFDMA systems with delay consideration,'' *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, Dec. 2018.

[20] V. Sharma, R. Sabatini, and S. Ramasamy, ''UAVs assisted delay optimization in heterogeneous wireless networks,'' *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2526–2529, Dec. 2016.

[21] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, ''Joint offloading and trajectory design for UAV-enabled mobile edge computing systems,'' *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[22] C. Zhan, Y. Zeng, and R. Zhang, ''Energy-efficient data collection in UAV enabled wireless sensor network,'' *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[23] H. Qi, Z. Hu, H. Huang, X. Wen, and Z. Lu, ''Energy efficient 3-D UAV control for persistent communication service and fairness: A deep reinforcement learning approach,'' *IEEE Access*, vol. 8, pp. 53172–53184, 2020.

[24] J. Li, Q. Liu, P. Wu, F. Shu, and S. Jin, ''Task offloading for UAV-based mobile edge computing via deep reinforcement learning,'' in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2018, pp. 798–802.

[25] M. Wang, S. Shi, S. Gu, X. Gu, and X. Qin, ''Q-learning based computation offloading for multi-UAV-enabled cloud-edge computing networks,'' *IET Commun.*, vol. 14, no. 15, pp. 2481–2490, Sep. 2020.

[26] X. Chen, ''Decentralized computation offloading game for mobile cloud computing,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[27] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, ''Efficient 3-D placement of an aerial base station in next generation cellular networks,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–5.

[28] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, ''A green strategic activity scheduling for UAV networks: A sub-modular game perspective,'' *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 58–64, May 2016.

[29] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, ''UAV trajectory optimization for data offloading at the edge of multiple cells,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[30] F. Wu, H. Zhang, J. Wu, and L. Song, ''Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning,'' *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4175–4189, Jul. 2020.

[31] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, ''Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning,'' *IEEE Access*, vol. 7, pp. 146264–146272, 2019.

[32] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, ''Intelligent trajectory design in UAV-aided communications with reinforcement learning,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.

[33] K. A. Ghamry, M. A. Kamel, and Y. Zhang, ''Multiple UAVs in forest fire fighting mission using particle swarm optimization,'' in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 1404–1409.

[34] K. Chen, Q. Sun, A. Zhou, and S. Wang, ''Adaptive multiple task assignments for uavs using discrete particle swarm optimization,'' in *Proc. Int. Conf. Internet Vehicles. (IOV)*, 2018, pp. 220–229.

[35] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, ''Mobile edge computing empowered energy efficient task offloading in 5G,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[36] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, ''An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing,'' *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.

[37] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, ''Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications,'' *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.

QI WANG (Student Member, IEEE) is currently a master student under the supervision of Prof. Ang Gao with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. Her research interests include unmanned aerial vehicle, mobile edge computing and deep reinforcement learning in wireless communication networks.

ANG GAO received his Ph.D. degree in control theory and control engineering from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2011. He currently serves as an Associate Professor at the School of Electronics and Information, Northwestern Polytechnical University. His research interests include QoS control, resource management and allocation in wireless communication and cloud.

YANSU HU received her Ph.D. degree in control theory and control engineering from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2012. She currently serves as an Associate Professor at the School of Electronics and Control, Chang'an University. Her research interests include networked control and resource allocation in cloud computing.