Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT Networks

Lei Yang, Haipeng Yao, *Member, IEEE*, Jingjing Wang, *Member, IEEE*, Chunxiao Jiang, *Senior Member, IEEE*, Abderrahim Benslimane, *Senior Member, IEEE*, and Yunjie Liu

***Abstract*—Unmanned aerial vehicles (UAVs) have been widely used to provide enhanced information coverage as well as relay services for ground Internet-of-Things (IoT) networks. Considering the substantially limited processing capability, the IoT devices may not be able to tackle with heavy computing tasks. In this article, a multi-UAV-aided mobile-edge computing (MEC) system is constructed, where multiple UAVs act as MEC nodes in order to provide computing offloading services for ground IoT nodes which have limited local computing capabilities. For the sake of balancing the load for UAVs, the differential evolution (DE)-based multi-UAV deployment mechanism is proposed, where we model the access problem as a generalized assignment problem (GAP), which is then solved by a near-optimal solution algorithm. Based on this, we are capable of achieving the load balance of these drones while guaranteeing the coverage constraint and satisfying the quality of service (QoS) of IoT nodes. Furthermore, a deep reinforcement learning (DRL) algorithm is conceived for the task scheduling in a certain UAV, which improves the efficiency of the task execution in each UAV. Finally, sufficient simulation results show the feasibility and superiority of our proposed load-balance-oriented UAV deployment scheme as well as the task scheduling algorithm.**

***Index Terms*—Load balance, mobile-edge computing (MEC), multi-UAV deployment, task scheduling, unmanned aerial vehicles (UAVs).**

Manuscript received November 4, 2019; revised January 15, 2020; accepted January 30, 2020. Date of publication February 4, 2020; date of current version August 12, 2020. This work was supported in part by the Director Foundation Project of the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data, in part by the National Key Research and Development Plan under Grant 018YFB1800805, and in part by the Future Intelligent Networking and Intelligent Transportation Joint Laboratory (BUPT-CTTIC). The work of Jingjing Wang was supported by the Shuimu Tsinghua Scholar Program. *(Corresponding author: Haipeng Yao.)*

Lei Yang is with the Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing 100124, China (e-mail: ylkino.xiao5@gmail.com).

Haipeng Yao and Yunjie Liu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yaohaipeng@bupt.edu.cn; liuyj@chinaunicom.cn).

Jingjing Wang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: chinaeephd@gmail.com).

Chunxiao Jiang is with the Tsinghua Space Center, Tsinghua University, Beijing 100084, China (e-mail: jchx@tsinghua.edu.cn).

Abderrahim Benslimane is with LIA/CERI, Avignon University, 84911 Avignon, France (e-mail: abderrahim.benslimane@univ-avignon.fr).

Digital Object Identifier 10.1109/JIOT.2020.2971645

I. Introduction

UNMANNED aerial vehicle (UAV) has witnessed its substantial success in both military and civilian applications due to its easy deployment, low cost, and feasible mobility [1]–[4]. Moreover, given the line-of-sight (LoS) channel, UAVs can be configured in a range of scenarios without ground infrastructures. Considering the large-scale deployment as well as the limited communication capability of Internet-of-Things (IoT) devices, applying UAVs in IoT ecosystems has gained much attention [5]–[7]. In multi-UAV-aided IoT scenarios, drones can both collect data from IoT devices and transmit the data to the ground center [8], [9]. Since IoT devices generally have severely limited power supply and low computing capability, they cannot be able to complete their tasks requiring high-complexity computing [10]. However, the emergence of a variety of mobile applications, such as augmented reality, face recognition, mobile online games, etc., impose great challenges on computing capability as well as requirements on the low latency. Fortunately, the mobile-edge computing (MEC) has been recognized as a promising technology to address these challenges [11]–[13], which provides cloud computing services for the edge networks, where ground IoT nodes can assign certain computing tasks to edge networks. Unlike mobile-cloud computing (MCC), the edge devices, such as access points in the MEC, can be deployed very close to the ground IoT nodes. Hence, MEC is characterized in terms of low-energy consumption, low latency, and relatively high security. Meanwhile, the UAVs can be well equipped with MEC devices to provide mobile-edge services in the large-scale IoT scenarios for the sake of improving users' quality of service (QoS), where the ground IoT nodes can offload heavy computing tasks to the mobile cloudlet via uplink/downlink communication with the UAVs.

In this article, we propose a UAV-aided MEC system as shown in Fig. 1, where multiple UAVs cooperate to provide MEC services for ground IoT nodes. In this case, some tasks in IoT nodes may be offloaded to the UAVs for the sake of satisfying their QoS requirement. Considering the nonuniform distribution of the ground IoT nodes, both the communication performance and the load balance of the tasks from the ground IoT nodes can be substantially affected by the deployment of UAVs [14]. Since computing resources on a UAV are also limited due to its constrained carrying capacity, some of the UAVs may overload, while others are free [15]. Hence, it is necessary for these UAVs to conduct an efficient task scheduling scheme

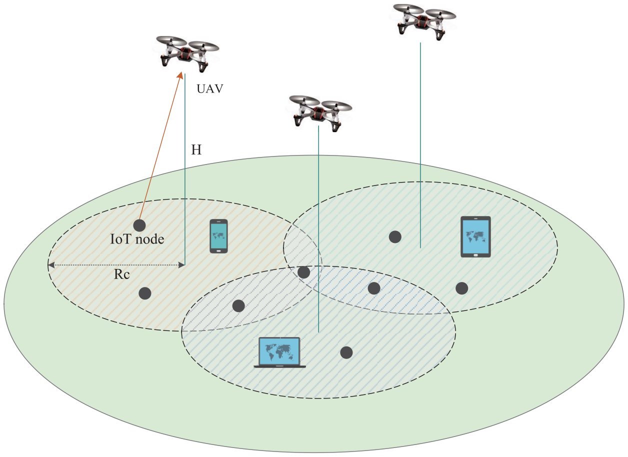

Fig. 1. Multi-UAV-aided IoT scenario.

in order to tackle with a large number of tasks offloaded for the sake of reducing the overall task processing time.

Therefore, both multi-UAV deployment as well as task scheduling impose a severe impact on the efficiency of the MEC network. Specifically, a proper drone's deployment is capable of reducing the transmission delay and of balancing the load of UAVs. In addition, the effective task scheduling strategy among the UAVs is able to reduce the waiting time of tasks and improve the processing efficiency. Hence, this article aims at the load-balancing deployment and the effective task scheduling in the multi-UAV-aided MEC system. The main contributions of this article are summarized as follows.

1) We propose a multi-UAV-aided MEC architecture for IoT networks, where the deployment and the task scheduling of UAVs are concerned considering both the load balance as well as the task processing efficiency.
2) In order to guarantee the load balance of multiple UAVs, we propose a differential evolution (DE)-based UAV deployment algorithm. Moreover, a generalized assignment problem (GAP) is exploited to find the near-optimal IoT node assignment solution. Furthermore, a deep reinforcement learning (DRL)-aided task scheduling method is proposed for the sake of improving the task execution efficiency of each UAV.
3) Extensive simulations evaluate the effectiveness and superiority of our proposed DE-based deployment strategy and the DRL-aided task scheduling scheme. The simulations results show that our proposed multi-UAV-aided MEC network can achieve load-balance task offload and improve the QoS of ground IoT nodes. Furthermore, comparison with other benchmark schemes indicates the better performance of our proposed scheme.

The remainder of this article is organized as follows. The related work is presented in Section II. In Section III, the channel model is defined. Moreover, the load-balance multi-UAV deployment and the latency-aware task scheduling optimization problem are formulated. In Section IV, the DE-based multi-UAV deployment algorithm and the DRL-based task schedule algorithm are elaborated. Section V shows our simulation results, followed by conclusions in Section VI.

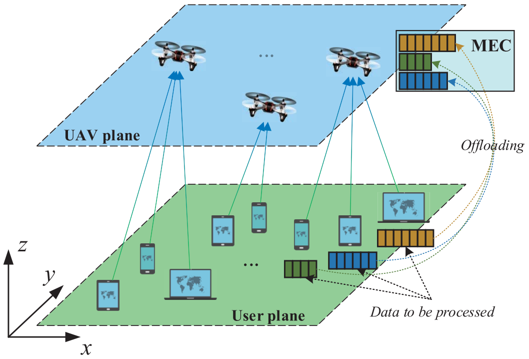

Fig. 2. Network architecture.

II. Related Work

Currently, a range of literature concerns the deployment of UAVs with the objective of improving the performance of the communication system, where UAVs play the role as airborne relays [16]–[18] or as flying base stations [19]–[21]. To elaborate a little further, in [16], by optimizing the source and relay transmission power along with the relay trajectory in mobile relaying systems, the throughput maximization problem was studied with relay nodes mounted on UAVs. Furthermore, Johansen *et al.* [17] used a UAV to operate as a wireless communication relay for the sake of downloading date from an underwater vehicle to a ground station. Kim *et al.* [18] proposed a trajectory planning algorithm and used small UAVs as communication relay nodes. In this way, the communication links were expanded and communication quality for a fleet of naval vessels was improved. Moreover, Mozaffari *et al.* [19] investigated the downlink coverage performance of drone small cells and the optimal drone small cell altitude. Hence, both maximum ground coverage and minimum required transmission power were guaranteed. In [20], through UAV's trajectory optimization, an energy-efficient UAV communication architecture with a ground terminal was built, where the communication throughput and the energy consumption of UAV were considered. Wu *et al.* [21] proposed a multi-UAV-enabled wireless-communication system, where fair performance among users was achieved. In this case, a group of ground users was served by multiple UAV-mounted aerial base. Meanwhile, the authors optimized the multiuser communication scheduling and association, as well as the minimum throughput over all ground users in downlink communication.

With the development of MEC, some researchers have combined UAV with MEC [22]–[27]. Loke [22] put forward an idea of fog calculation on UAVs and outlined some scenarios with using such airborne fog computing architecture. Moreover, Jeong *et al.* [23] used a moving UAV to offer computing offloading services to mobile users, which aimed at minimizing the total mobile energy consumption and satisfying the QoS requirements. In [24], UAV-enabled wireless powered MEC system was studied, with computing bits as well as harvesting causality requirement satisfied. Furthermore, Zhou *et al.* [25] studied a UAV-enabled wireless-powered system, where the energy-harvesting causal as well as the UAV speed constraint were considered. Then, the objective of maximizing computing rate under both partial and binary computing offloading modes was achieved. In [26], for the sake of improving coverage and increasing rate, Zhao *et al.* used UAVs to help small-cell base stations offload traffic via wireless backhaul. In spite of the limited capacity of wireless backhaul, UAVs and small-cell base stations can both store video caches at off-peak time. In [27], a UAV-aided MEC system with a three-layer integrated architecture was established over the social Internet of vehicles. By jointly optimizing the transmit power of the vehicle and the UAV, an optimization framework for total utility maximization was proposed.

III. System Model

A. Network Model

An MEC system model is illustrated in Fig. 2, which consists of K IoT nodes and N UAVs. Meanwhile, IoT nodes on the ground can offload their tasks to the UAVs. Let $\psi = \{m_1, m_2, \ldots, m_K\}$ denote the set of IoT nodes and $\eta = \{u_1, u_2, \ldots, u_N\}$ be the set of the UAVs. In our system, the IoT nodes are constituted by heterogeneous types of devices with different offloading levels. We assume that there are Z types of IoT nodes represented by $e = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_Z\}$, where the corresponding offloading level is denoted by $\tilde{e} = \{\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \ldots, \tilde{\varepsilon}_Z\}$. Without loss of generality, we assume all IoT nodes are fixed on the ground and all UAVs fly at a fixed altitude H $(H > 0)$. The position of the kth IoT node is denoted by $m_k = [x_k^m, y_k^m, 0]$, and the position of the nth UAV is denoted by $u_n = [x_n^u, y_n^u, H]$. Let $\alpha_{n,k}$ indicate the connection between UAV and IoT node, where $\alpha_{n,k} = 1$ means the kth IoT node is connected to the nth UAV. Considering that each IoT node can only connect to one UAV simultaneously, we have

$$\sum_{n=1}^{N} \alpha_{n,k} = 1 \quad \forall k = 1, 2 \ldots, K. \tag{1}$$

Moreover, the distance between u_n and m_k can be calculated by $d(u_n, m_k) = [H^2 + (x_n^u - x_k^m)^2 + (y_n^u - y_k^m)^2]^{(1/2)}$. For an offloaded task $F_z(u_n, m_k)$ $\forall z = 1, 2, \ldots, Z$ between m_k and u_n, given the required computing resource $c_z(u_n, m_k)$, the required execution time $t_z(u_n, m_k)$ and the required communication traffic $h_z(u_n, m_k)$, we define $F_z(u_n, m_k) = [c_z(u_n, m_k), t_z(u_n, m_k), h_z(u_n, m_k)]$.

B. Communication Model

Fig. 1 shows the multi-UAV-aided IoT scenario. Every UAV has a fixed coverage with radius R_c. Moreover, ground IoT nodes can access the UAV via orthogonal frequency-division multiple access, hence the interference among the UAV-node links is neglected. We assume that the communication channels from UAVs to IoT nodes are dominated by the LOS channel, and thus the channel gain between m_k and u_n can be derived as

$$\begin{aligned} g(u_n, m_k) &= \beta_0 d(u_n, m_k)^{-2} \\ &= \frac{\beta_0}{H^2 + \left(x_n^u - x_k^m\right)^2 + \left(y_n^u - y_k^m\right)^2} \end{aligned} \tag{2}$$

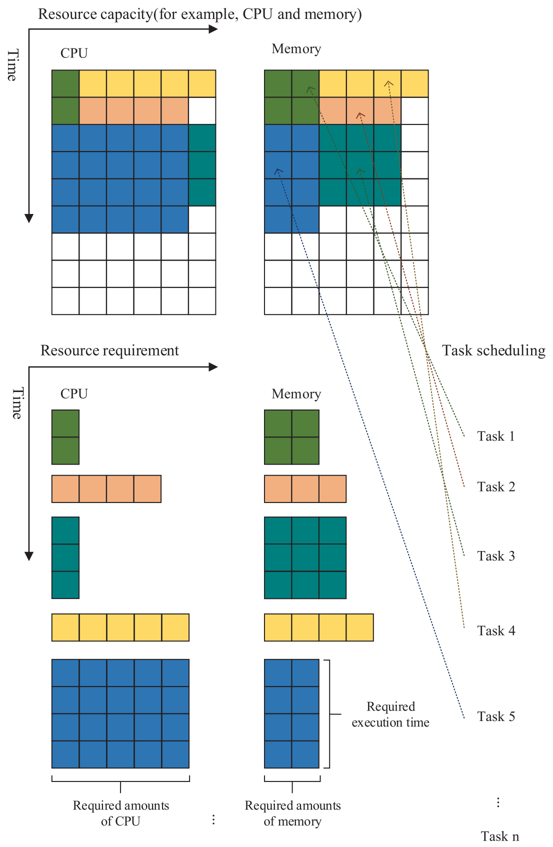

Fig. 3. Toy example for explaining the task scheduling scheme.

where β_0 denotes the channel gain at the reference distance of 1 m. Let $\tilde{p}(u_n)$ represent the transmission power of u_n. Then, the uplink data rate between u_n and m_k can be formulated by

$$\begin{aligned} \tilde{v}(u_n, m_k) &= B \log_2\left(1 + \frac{\tilde{p}(u_n) g(u_n, m_k)}{\sigma^2}\right) \\ &= B \log_2\left(1 + \frac{\beta_0 \tilde{p}(u_n)}{\sigma^2\left(H^2 + \left(x_n^u - x_k^m\right)^2 + \left(y_n^u - y_k^m\right)^2\right)}\right) \end{aligned} \tag{3}$$

where σ^2 denotes the white Gaussian noise variance, and B denotes the channel bandwidth.

The transmission delay between m_k and u_n for task $F_z(u_n, m_k)$ can be shown as

$$t_z^T(u_n, m_k) = \frac{h_z(u_n, m_k)}{\tilde{v}(u_n, m_k)}. \tag{4}$$

Let $\phi(u_n)$ represent the set of IoT nodes connected to u_n, so it follows:

$$\bigcup_{n=1}^{N} \phi(u_n) = \psi. \tag{5}$$

Then, the total offloading level of u_n can be denoted by

$$\Omega_n = \sum_{k=1}^{K} \tilde{\varepsilon}(m_k) \alpha_{n,k} \quad \forall n = 1, 2, \ldots, N \tag{6}$$

where $\tilde{\varepsilon}(m_k)$ represents the offloading level of m_k and $\tilde{\varepsilon}(m_k) \in \tilde{e}$.

C. Computation Model

In our system, P types of resources in UAV are concerned, such as CPU, memory, hard disk, etc., Fig. 3 shows an example of task scheduling. For the task $F_z(u_n, m_k)$, its ideal completion time is $t_z(u_n, m_k)$, so we calculate the total execution time of task $F_z(u_n, m_k)$ according to

$$t_z^A(u_n, m_k) = t_z(u_n, m_k) + t_z^{\text{delay}}(u_n, m_k) \tag{7}$$

where $t_z^{\text{delay}}(u_n, m_k)$ denotes the delay time of task $F_z(u_n, m_k)$. Hence, $t_z^{\text{delay}}(u_n, m_k)$ is 0 when task $F_z(u_n, m_k)$ is executed immediately, or more than 0 when the current processing queue of UAV is full and task needs to wait.

D. Problem Formulation

In this article, we aim at optimizing the average slowdown for offloaded tasks, where the load balance for UAVs is guaranteed. Since the distribution of IoT nodes is generally nonuniform, the deployment strategy of UAVs will have a substantial impact on the performance of the MEC system. Generally, UAVs need to be as close to the IoT nodes as possible to reduce transmission cost. Furthermore, excessive load in the UAV will increase the number of tasks in the processing queue, which will increase the total processing time of the tasks and further affect the system performance. Therefore, for the multi-UAV MEC system, it is necessary to achieve load balance, in order to avoid the overall system performance degradation when some of the UAVs take too many tasks and others are free of tasks.

Specifically, our problem concerns two aspects, namely, the load-balancing multi-UAV deployment and the latency-aware task scheduling on the UAVs. As for the load-balancing multi-UAV deployment, our objective is to achieve the load balance for all UAVs, where the degree of load balance can be expressed as

$$S_O(\Omega) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\Omega_i - \overline{\Omega}\right)^2} \tag{8}$$

where

$$\overline{\Omega} = \frac{1}{N}\sum_{i=1}^{N}(\Omega_i). \tag{9}$$

Note that $S_O(\Omega) \geq 0$. Hence, the larger $S_O(\Omega)$ is, the more nonuniform the distribution of tasks among UAVs is. Meanwhile, the average transmission cost between UAVs and IoT nodes is considered, and we aim at minimizing the average transmission cost according to (4), which can be calculated by

$$t_c(u_n, m_k) = \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} t_z^T(u_n, m_k)\alpha_{n,k}. \tag{10}$$

As for the latency-aware task scheduling, according to (7), we can get the task slowdown, which can be calculated by

$$t_d(u_n, m_k) = \frac{t_z^A(u_n, m_k)}{t_z(u_n, m_k)} \tag{11}$$

where $t_d(u_n, m_k) \geq 1$.

Then, according to (8), (10), and (11), the load-balancing multi-UAV deployment as well as latency-aware task scheduling optimization problem can be formulated as

$$\begin{aligned}
\text{P1}: \min_{(x_n^u, y_n^u)} \quad & \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} t_d(u_n, m_k)\alpha_{n,k} \\
& + \ell_1[S_O(\Omega)] + \ell_2[t_c(u_n, m_k)] \\
\text{s.t.} \quad & d(u_n, m_k) \leq R_c \\
& m_k \in \psi \quad \forall k = 1, 2, \dots, K \\
& u_n \in \eta \quad \forall n = 1, 2, \dots, N \\
& \alpha_{n,k} \in \{0, 1\} \quad \forall k = 1, 2, \dots, K, n = 1, 2, \dots, N \\
& \sum_{n=1}^{N}\alpha_{n,k} = 1 \quad \forall k = 1, 2, \dots, K
\end{aligned} \tag{12}$$

where ℓ_1 and ℓ_2 denote the weight factors for load balancing requirement and average transmission cost, respectively. Specifically, $\ell_1 = 0$ means that the influence of load balance for UAVs is not considered, and $\ell_2 = 0$ means that the influence of average transmission cost is not considered.

IV. Joint Multi-UAV Deployment and Task Scheduling Optimization

In this section, we will elaborate on the proposed multi-UAV deployment algorithm and task scheduling schema. The multi-UAV deployment problem is an NP-hard problem [7]. In order to find a satisfactory solution, we propose a DE-aided multi-UAV deployment algorithm to solve this problem. Initially, all UAVs are randomly positioned, and then we assign the maximum load for each UAV based on the location of UAVs and IoT nodes. In this way, the access problem between UAVs and IoT nodes can be modeled into a GAP. After that, we use an approximation algorithm to determine all the connections between UAVs and IoT nodes. Meanwhile, a deep reinforcement learning algorithm is used to schedule tasks for incoming tasks. Finally, the near-optimal positions of UAVs are obtained through iterations of the DE algorithm.

A. Load Balance for UAVs

Since there are total Z types of IoT nodes represented by $e = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_Z\}$ and the corresponding offloading level is $\tilde{e} = \{\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \dots, \tilde{\varepsilon}_Z\}$, the total offloading level E_t can be formulated as

$$E_t = \sum_{k=1}^{K}\tilde{\varepsilon}(m_k) \tag{13}$$

where $\tilde{\varepsilon}(m_k)$ represents the offloading level of m_k.

Ideally, each UAV carries a task load of $E_{\text{avg}} = E_t/N$ to achieve load balance. However, since the distribution of IoT nodes is generally nonuniform, balancing load for all UAVs will lead to excessive transmission consumption of these IoT nodes when some nodes are far from connected UAV, thus failing to achieve the optimization goal of (12). Hence, according to the location of all UAVs and IoT nodes, let $e_{lb}(u_n)$ be the lower bound of the set of IoT nodes, in which the IoT nodes can only be connected to u_n due to the distance. The total offloading level of $e_{lb}(u_n)$ can be formulated as

$$\tilde{e}_{lb}(u_n) = \sum_{m_k \in e_{lb}(u_n)} \tilde{\varepsilon}(m_k) \quad \forall k = 1, 2, \ldots, K. \tag{14}$$

Similarly, let $e_{ub}(u_n)$ be the upper bound of the set of IoT nodes which u_n can cover. The total offloading level of $e_{ub}(u_n)$ can be formulated as

$$\tilde{e}_{ub}(u_n) = \sum_{m_k \in e_{ub}(u_n)} \tilde{\varepsilon}(m_k) \quad \forall k = 1, 2, \ldots, K. \tag{15}$$

Then, let the reference load of UAV u_n be $\tilde{E}(u_n)$, we can achieve

$$\tilde{e}_{lb}(u_n) \leq \tilde{E}(u_n) \leq \tilde{e}_{ub}(u_n). \tag{16}$$

Considering E_{avg}, $\tilde{e}_{lb}(u_n)$, and $\tilde{e}_{ub}(u_n)$, if $E_{\text{avg}} \geq \tilde{e}_{ub}(u_n)$, the best setting for $\tilde{E}(u_n)$ is $\tilde{E}(u_n) = \tilde{e}_{ub}(u_n)$ for satisfying the total load requirement of u_n. Similarly if $E_{\text{avg}} \leq \tilde{e}_{lb}(u_n)$, the best setting for $\tilde{E}(u_n)$ is $\tilde{E}(u_n) = \tilde{e}_{lb}(u_n)$. In the case of $\tilde{e}_{lb}(u_n) \leq E_{\text{avg}} \leq \tilde{e}_{ub}(u_n)$, u_n has more exclusive IoT nodes when E_{avg} is close to $\tilde{e}_{lb}$, where $\tilde{E}(u_n)$ needs to be appropriately increased. When E_{avg} is close to $\tilde{e}_{ub}$, more IoT nodes covered with u_n will also be covered by other UAVs, where $\tilde{E}(u_n)$ needs to be appropriately reduced. Hence, we can get

$$\begin{aligned}\tilde{E}(u_n) &= E_{\text{avg}} + \lambda_1\big(\tilde{e}_{ub}(u_n) - E_{\text{avg}}\big) \\ &\quad - \lambda_2\big(E_{\text{avg}} - \tilde{e}_{lb}(u_n)\big) \\ &= (1 - \lambda_1 - \lambda_2)E_{\text{avg}} + \lambda_1\tilde{e}_{ub}(u_n) + \lambda_2\tilde{e}_{lb}(u_n)\end{aligned} \tag{17}$$

where λ_1 and λ_2 indicate the influence factors of $\tilde{e}_{ub}(u_n)$ and $\tilde{e}_{lb}(u_n)$, respectively. Then, the reference load $\tilde{E}(u_n)$ can be expressed by

$$\tilde{E}(u_n) = \begin{cases} \tilde{e}_{lb}, & E_{\text{avg}} \leq \tilde{e}_{lb}(u_n) \\ \tilde{e}_{ub}, & E_{\text{avg}} \geq \tilde{e}_{ub}(u_n) \\ (1 - \lambda_1 - \lambda_2)E_{\text{avg}} + \lambda_1\tilde{e}_{ub}(u_n) & \\ \qquad + \lambda_2\tilde{e}_{lb}(u_n), & \text{otherwise.} \end{cases} \tag{18}$$

Then, the sum of all reference load can be expressed as $\tilde{E}_t = \sum_{i=1}^{N} \tilde{E}(u_n)$. Sometimes, it can be aware of that a gap exists between $\tilde{E}_t$ and E_t in the case of $\tilde{E}_t < E_t$, which needs to be filled. Hence, in the case of $\tilde{E}_{lb}(u_n) \leq E_{\text{avg}} \leq \tilde{E}_{ub}(u_n)$, we fill the gap by adding $\tilde{E}_{\text{gap}}$ to $\tilde{E}(u_n)$, and the reference load remains unchanged otherwise. Then, we have

$$\tilde{E}_{\text{gap}} = \frac{E_t - \tilde{E}_t}{n^*} \tag{19}$$

where n^* is the number of UAVs that under the case of $\tilde{E}_{lb}(u_n) < E_{\text{avg}} < \tilde{E}_{ub}(u_n)$. Finally, we rewrite (18), and get the final reference load of all UAVs

$$\tilde{E}(u_n) = \begin{cases} \tilde{e}_{lb}, & E_{\text{avg}} \leq \tilde{e}_{lb}(u_n) \\ \tilde{e}_{ub}, & E_{\text{avg}} \geq \tilde{e}_{ub}(u_n) \\ (1 - \lambda_1 - \lambda_2)E_{\text{avg}} + \lambda_1\tilde{e}_{ub}(u_n) & \\ \qquad + \lambda_2\tilde{e}_{lb}(u_n) + \tilde{E}_{\text{gap}}, & \text{otherwise.} \end{cases} \tag{20}$$

B. GAP-Based Node Assignment

In this section, we aim at optimizing the node assignment among UAVs according to the reference load in the previous section. Given the known position of all UAVs, the node assignment problem can be formulated as a kind of GAP. In the GAP, both tasks and agents have sizes, where the size of each task is different with different agent [28].

In this article, each UAV can be seen as an agent, and each IoT node has a kind of task. Define $w_{n,k}$ as the cost of the task offloaded from the kth IoT node to the nth UAV, where $w_{n,k}$ is determined only by the type of the task. As a result, $w_{n,k} = \tilde{\varepsilon}(m_k)$. Furthermore, let $f_{n,k}$ be the profit of the task, which is primarily affected by the distance between u_n and m_k and the required communication traffic $h_z(u_n, m_k)$. According to (4), we can get

$$f_{n,k} = \begin{cases} \varphi t_z^T(u_n, m_k), d(u_n, m_k) \leq R_c \\ -\infty, d(u_n, m_k) > R_c \end{cases} \tag{21}$$

where $\varphi > 0$ denotes the influence factor. In general, the closer the distance between u_n and m_k, the bigger the $f_{n,k}$ is. When the distance between u_n and m_k exceeds the service radius R_c, $f_{n,k}$ will close to $-\infty$.

In order to maximize the total profit, we formulated this assignment problem as a integer-programming (IP) problem, which can be written as

$$\begin{aligned} \text{P2}: \min_{x_{n,k}} \quad & \sum_{n=1}^{N}\sum_{k=1}^{K} -f_{n,k}x_{n,k} \\ \text{s.t.} \quad & \sum_{k=1}^{K} w_{n,k}x_{n,k} \leq \tilde{E}(u_n) \quad \forall n = 1, 2, \ldots, N \\ & \sum_{n=1}^{N} x_{n,k} = 1 \quad \forall k = 1, 2, \ldots, K \\ & x_{n,k} \in \{0, 1\} \quad \forall n = 1, 2, \ldots, N,\ k = 1, 2, \ldots, K. \end{aligned} \tag{22}$$

Since the problem $P2$ is a GAP, problem $P2$ is NP-hard. Here, we propose an approximate algorithm to find its near-optimal solution, which is described as three steps.

Step 1) *Linear Programming (LP) Relaxation:* Convert the IP problem into the following LP relaxation:

$$\begin{aligned} \text{P3}: \min_{x_{n,k}} \quad & \sum_{n=1}^{N}\sum_{k=1}^{K} -f_{n,k}x_{n,k} \\ \text{s.t.} \quad & \sum_{k=1}^{K} w_{n,k}x_{n,k} \leq \tilde{E}(u_n) \quad \forall n = 1, 2, \ldots, N \\ & \sum_{n=1}^{N} x_{n,k} = 1 \quad \forall k = 1, 2, \ldots, K \\ & x_{n,k} \geq 0 \quad \forall n = 1, 2, \ldots, N,\ k = 1, 2, \ldots, K. \end{aligned} \tag{23}$$

Then, solving this LP gives us a fractional solution.

Step 2) *Bipartite Graph Construction:* Convert the fractional solution to a bipartite graph with fractional edge weights where one side represents IoT nodes and the other side represents slots in UAVs. Define a bipartite graph $C = (\hat{K}, \hat{B})$, where $\hat{K} = [K]$ and $\hat{B} = \{(n', s') | n' \in [N], s' \in [s'_n]\}$. We illustrate the edges in the bipartite graph by only looking at the case $n' = 1$. For $n' = 1$, consider only the IoT node m_k such that $x_{1,k} > 0$ and assume this is $[K]$ without loss of generality. Assume that $f_{1,1} \geq f_{1,2} \geq \cdots f_{1,K}$ and sort them if necessary. For each $k' \in [K]$, we track the partial sum $\sum_{k=1}^{k'} x_{1,k}$. If $\sum_{k=1}^{k'} x_{1,k} \leq s'$, and $\sum_{k=1}^{k'-1} x_{1,k} \geq s' - 1$, then add an edge $(k', (1, s'))$ with weight $x_{1,k'}$. Otherwise, if $\sum_{k=1}^{k'} x_{1,k} > s'$, and $\sum_{k=1}^{k'-1} x_{1,k} < s'$, then add an edge $(k', (1, s' - 1))$ with weight $s' - \sum_{k=1}^{k'-1} x_{1,k}$ and an edge $(k', (1, s'))$ with weight $\sum_{k=1}^{k'} x_{1,k} - s'$.

Step 3) *Deterministic Rounding:* Note that the bipartite graph is a fractional min-cost perfect matching of the underlying cost function $C(k', (n', s')) = -f_{n,k}$. It can be shown that all extreme points of the polytope are integral. Through finding a cycle or augmenting path in the previous bipartite graph, we can convert the fractional solution into an integral one within polynomial time, without increasing the objective value. Call all the edges with weight in the range $(0, 1)$ *unsaturated* and $\{0, 1\}$ *saturated*. We represent an iterative method such that after one loop, the number of *unsaturated* edges decreases by at least 1, and the sum of weights attached to a left vertex is preserved as 1. Since at the end there is no *unsaturated* edge left and each left vertex is attached to edges with weights summing up to 1, each left vertex is mapped to a right vertex via the unique edge with weight 1. We will see shortly that this map is also injective, proving that it is a perfect matching on the left side. Consider the following two cases.

Case 1: If there is a cycle of *unsaturated* edges, color them by blue and red alternately. By increasing the weights of all red edges by a small amount δ and decreasing the weights of blue ones by δ, no LP constraints are violated provided that δ is sufficiently small. We can thus move δ toward that direction until one of the red edges comes to 1 or one of the blue edges comes to 0. Either way, we decreased the number of *unsaturated* edges by at least 1.

Case 2: Call a vertex *single* if it is only attached to a *unsaturated* edge, then the weight on the vertex is easily seen as equal to the weight of the edge. Note that no left vertex can be *single* since they all have weight exactly 1. Suppose now there is no cycle of *unsaturated* edge left, then each maximum path of *unsaturated* edge must start and end at *single* vertices. Choose a maximum path and color the edges by blue and red alternatively. Similar to case 1, perturbing the weight of all blue and red edges by a small amount δ does not violate any of the LP constraints, and we can freely move δ until one red edge reaches 1 or one blue edge reaches 0. It should be noted that, after deterministic rounding, the actual offloading level of a few UAVs may be slightly higher than the reference load, but it will not cause negative effects because it will not exceed the maximum processing capacity of the UAV.

C. Deep-Reinforcement-Learning-Aided Task Scheduling

In this section, we study the task scheduling scheme for incoming tasks relying on the DRL algorithm. The DRL combines deep learning and reinforcement learning in order to achieve end-to-end learning from perception to action. The agent, first of all, senses the state from the environment and then performs an action. The environment may move to another state with time. At the same time, the agent can obtain feedback according to the reward function. It can be seen that DRL is a model of human–environment interaction [29]–[31]. It can be seen that DRL is a model of human–environment interaction. The Q-learning algorithm is a classical reinforcement learning algorithm, and it can find the near-optimal action by interacting with the environment. Moreover, for the step t, the states can be represented as S_t, and the the policy space can be defined as π, then the accumulated rewards $V^{\pi}(S_t)$ can be calculated as

$$V^{\pi}(S_t) = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \tag{24}$$

where R_t is the instant reward in step t, and $\gamma (0 < \gamma < 1)$ represents the discount factor, which reflects the influence of the future reward. In Q-learning algorithm, the Q-value is the function of states S_t and action A_t, which can be formulated as

$$Q_t(S_t, A_t) = R_t + \gamma V^{\pi}(S_{t+1}). \tag{25}$$

Q-learning is a kind of algorithm with the iteration of value. Since we can only get a limited series of samples and only limited samples can be used for operation, we update the Q-value according to

$$\begin{aligned} Q_{t+1}(S_t, A_t) = {} & (1 - \vartheta) Q_t(S_t, A_t) \\ & + \vartheta \big[R_t + \gamma \max Q_t(S_{t+1}, A_{t+1})\big] \end{aligned} \tag{26}$$

where the learning rate ϑ $(0 < \vartheta < 1)$ controls the learning speed.

Deep Q-network (DQN) combines convolutional neural network (CNN) with Q-learning, which uses the target function of Q-learning to construct deep learning tags [32]. In addition, DQN uses an experience replay mechanism to solve data correlation problems. At the same time, a CNN (MainNet) is used to generate the current Q-value, and another CNN (Target) is used to generate the Target Q-value. Furthermore, the loss function of DQN can be given by

$$L(\theta) = E\Big[\big(\mathrm{Q}^T - Q(S_t, A_t; \theta)\big)^2\Big] \tag{27}$$

where the target function Q^T can be formulated as

$$Q^T = R_t + \gamma \max Q(S_{t+1}, A_{t+1}; \theta) \tag{28}$$

where θ is the parameter of the neural network.

In this article, we exploit the DRL algorithm shown as Algorithm 1 to solve the task scheduling problem, for the sake of minimizing the average slowdown of tasks in UAVs. Considering that a set of $\phi(u_n)$ IoT nodes have been accessed

```
Algorithm 1 DRL-Based Task Scheduling
Input: states, $S$; maximum number of iteration, $t_{max}$;
Output: action-value function, $Q(S, A)$;
initialize action-value function Q with random weights;
for $t = 1; t < t_{max}; t++$ do:
    divide step $t$ into $b_t$ sub-steps;
    select a random sub-action in each sub-step, and all sub-actions make up action $a_t$;
    execute action $a_t$, then obtain the reward $r_t$, and arrive at the next state $s_{t+1}$;
    calculate target Q-value according to Eq. (28);
    update the deep Q-network by minimizing the loss $L_\theta$ according to Eq. (27);
end for
```

to the nth UAV according to (5), three key factors of the DQN are given as follows.

1) *State Space:* The resources states of UAV include the current allocated computing resources and the required computing resources of the possible scheduled tasks as shown in Fig. 3. We assume that the resource demand of task $F_z(u_n, m_k) = [c_z(u_n, m_k), t_z(u_n, m_k), h_z(u_n, m_k)]$ is given, where $c_z(u_n, m_k) = [c_{z,1}(u_n, m_k), c_{z,2}(u_n, m_k), \dots, c_{z,p}(u_n, m_k)]$ and the $c_{z,p}(u_n, m_k), p = 1, 2, \dots, P$ indicates the pth type of resources. Furthermore, the current allocated computing resources of the nth UAV are $G_{n,z}$. Hence, the system states with total N UAVs can be given by $\Gamma = [S_1^S, S_2^S, \dots, S_n^S, n = 1, 2, \dots, N]$, where $S_n^S = \{\bigcup_{k=1}^{K} F_z(u_n, m_k)\alpha_{n,k}, G_{n,z}, z = 1, 2, \dots, Z\}$.
2) *Action Space:* Assuming in the tth step, a number of $M_{n,t}$ tasks are queued in the nth UAV. Then, the size of the action space in the nth UAV can be represented by $2^{M_{n,t}}$, which make challenges on the DRL. In order to simplify the action space, we redefine the action space as $A_{n,t} = \{\emptyset, 1, 2, \dots, M_{n,t}\}$, and divide each step into $b_{n,t}$ substeps, where $b_{n,t}$ denotes the total number of subactions that will be scheduled in the tth step. Then, in each substep, only one subaction will be scheduled, and the size of the action space is reduced to $M+1$. Since a task is scheduled in the cluster as shown in Fig. 3, this task will be removed from the waiting queue. The agent then observes a state transition, where the scheduled task is moved to the appropriate position. Once the agent picks $\emptyset$ or an invalid subaction, it indicates that the agent does not wish to schedule further tasks in the current step, and the time actually proceeds, in other words, go to the next step $t + 1$.
3) *Reward:* The reward function is designed for directing the agent to minimize the average slowdown in the UAVs. Specifically, we set the reward in each timestep as $\sum_{M_n} [-1/(t_z(u_n, m_k))]$, where M_n is the set of tasks currently in the nth UAV.

As for the complexity of our proposed DRL-aided task scheduling algorithm, in the training process, the time complexity is $O(E * D/B * T)$, which mainly depends on the CNN training complexity of DQN, where E represents the epochs, D denotes the data set size, B denotes the batch size, and T represents the time complexity of a single iteration. Moreover, the execution time complexity of the algorithm is $O(n)$ relying on it is mainly matrix operation.

D. Differential-Evolution-Based Multi-UAV Deployment

DE is a random model that simulates biological evolution. Through repeated iterations, the populations who are adapted to the environment are preserved. In this article, the dimension of each population is $2N$, which is twice as much as the number of UAVs. Let $X_d(i), d = 1, 2, \dots, D$ indicate the deployment location of the ith generation of UAVs, where D is the total number of populations. At the beginning of the algorithm, we randomly determine the deployment position of UAVs in all populations as $X_d(1), d = 1, 2, \dots, D$. In the gth iteration, each population $X_d(g)$ mutates and generates a mutation vector $H_d(g)$ by randomly selecting three other populations from the generation, for example, $X_{da}(g)$, $X_{db}(g)$, and $X_{dc}(g)$ while $da \neq db \neq dc \neq d$. Then, we have

$$H_d(g) = X_{da}(g) + \xi(X_{db}(g) - X_{dc}(g)) \tag{29}$$

where ξ indicates the factor used to control the influence of difference vectors. Then, the crossover vector $V_d(g)$ can be caculated by

$$I_d(g) = \begin{cases} H_d(g), & \text{rand}(0, 1) < P_{cr} \\ X_d(g), & \text{else} \end{cases} \tag{30}$$

where P_{cr} is the crossover probability. Finally, according to the value of fitness function, we select the best population between crossover vector $V_d(g)$ and the original vector $X_d(g)$ as the next generation, which can be formulated as

$$X_d(g + 1) = \begin{cases} I_d(g), f(I_d(g)) > f(X_d(g)) \\ X_d(g), \text{ else,} \end{cases} \tag{31}$$

where function $f(*)$ is the objective function of (12).

According to the characteristics of DE, after a certain number of iterations, we can get the converged populations, from which we can get the near-optimal deployment of all UAVs. As for the complexity of our proposed DE-based multi-UAV deployment algorithm, in each decision round, we have to update the generation of each population according to (31). Hence, the complexity of each decision round is on the order of $O(n)$. The flow of our proposed DE-based multi-UAV deployment algorithm is shown in Algorithm 2.

V. Simulation Results

In our simlulations, a 400 × 400-m rectangular area is considered, where the ground IoT nodes are randomly distributed. The channel bandwidth is $B = 1$ MHz and the transmission power is $\tilde{p} = 0.5$ W. The variance of the white Gaussian noise is $\sigma^2 = 5 \times 10^{-15}$. All UAVs fly at a fixed altitude $H = 100$ m with a maximum coverage $R_c = 100$ m. The channel gain is set as $\beta_0 = \eta_{los}(4\pi f/c)^2$, where $\eta_{\text{los}} = 1$ is the attenuation factor corresponding of LOS [33], the carrier frequency is $f = 2.4$ GHz, and c is the velocity of light. There are total 100 IoT nodes and five UAVs.

Algorithm 2 DE-Based Multi-UAV Deployment

```
Input: the set of (IoT nodes), $\phi$; objective function; the number of UAVs, $N$; maximum number of iteration, $l_{max}$;
Output: deployment of UAVs, $[x_n^u, y_n^u], n = 1, 2, ..., N$; best IoT node assignment, $\phi(u(n))$;
1:  randomly initialize populations $X_d(1)$;
2:  for $i = 1; i < l_{max}; i++$ do:
3:      obtain mutation populations through differential strategy according to Eq. (29);
4:      obtain crossover populations according to Eq. (30);
5:      determine reference load for multiple UAVs according to Eq. (20);
6:      get the fractional solution of node assignment problem in Eq. (23);
7:      construct the bipartite graph based on the fractional solution;
8:      deterministic round the fractional solution;
9:      find the optimal task scheduling scheme according to Algorithm 1;
10:     calculate the fitness for all populations according to the objective function of Eq. (12);
11:     get the next generation of DE according to Eq. (31);
12: end for
```

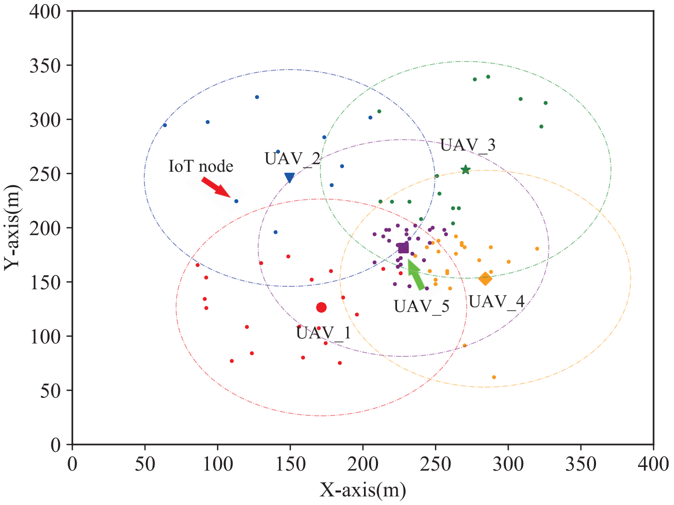

Fig. 4. Simulation result of the multi-UAV deployment in the context of $N = 5$ and $K = 100$.

Fig. 4 shows an example of near-optimal result for the multi-UAV deployment. There are some different colors of UAVs, and the smaller pointers indicate the IoT nodes while each IoT node is connected to the UAV of the same color. Explicitly, given a total of $N = 5$ UAVs and $K = 100$ IoT nodes, we arrive at a near-optimal multi-UAV deployment strategy associated with the coordinates of [171.41, 126.49], [149.50, 245.87], [270.68, 253.43], [284.26, 152.94], and [227.91, 181.27] for each UAV.

As for the task scheduling, we use a neural network to construct our deep Q-learning network with a learning rate of 0.0025. The task generation of IoT nodes follows the Poisson distribution while the arrival rate is λ. There is a total of three kinds of tasks and two kinds of resources required for each task in the system. Meanwhile, the ideal completion time and resource requirement of each task are randomly generated from [1, 15] and [1, 10]. Figs. 5 and 6 plot the reward and the average slowdown, respectively, at each iteration. As expected, both values increase with iterations improve. It clearly shows in Fig. 5 that as the iterations increase, the total reward of DRL will gradually get bigger and converge to an ideal value. Fig. 6 shows that as the iterations increase, UAVs have learned to better schedule incoming tasks, so the average slowdown will gradually approach the minimum value. Figs. 7 and 8 indicate the influence of the quantity of tasks on the scheduling performance of UAVs. Since the task generation of IoT nodes follows the Poisson distribution, we can obtain that with the increase of the task arrival rate, the average slowdown increases and reward decreases owing to the limited processing capability of each UAV.

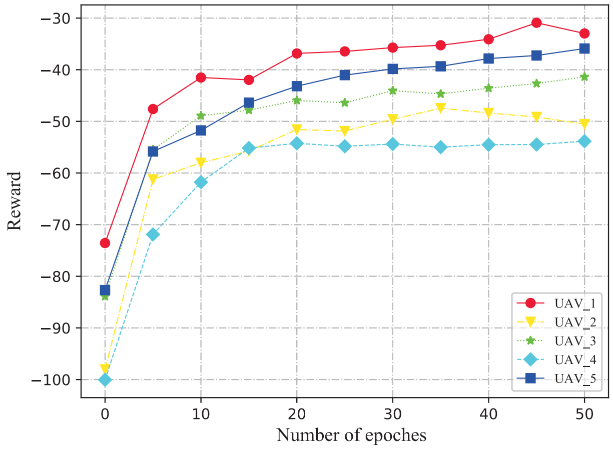

Fig. 5. Reward for task scheduling the UAV network.

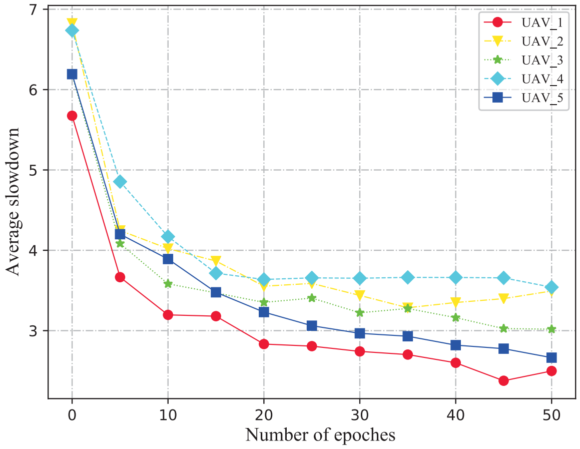

Fig. 6. Average slowdown of offloaded tasks of the UAV network.

We compare our proposed DRL-aided task scheduling algorithm against three classic task scheduling algorithms, i.e., first-come-first-serve (FCFS) algorithm, shortest job first (SJF) algorithm, and round robin (RR) algorithm, as shown in Fig. 9. Specifically, SJF schedules the smaller tasks first in comparison to both FCFS and RR. The performance of the RR algorithm largely depends on the choice of the length of the time quantum. If the time slot is longer, it tends to exhibit the

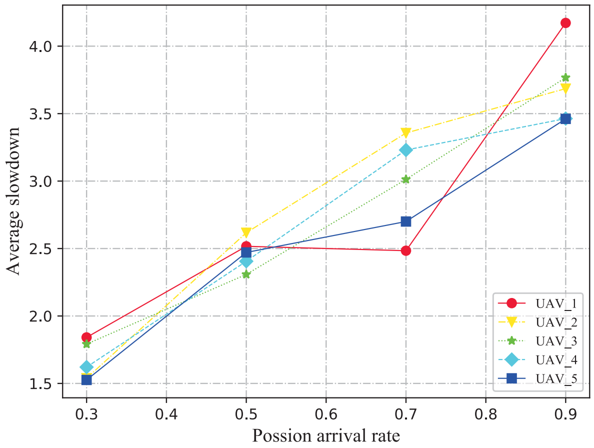

Fig. 7. Average slowdown versus Possion arrival rates in the context of $\lambda = (0.3, 0.5, 0.7, 0.9)$.

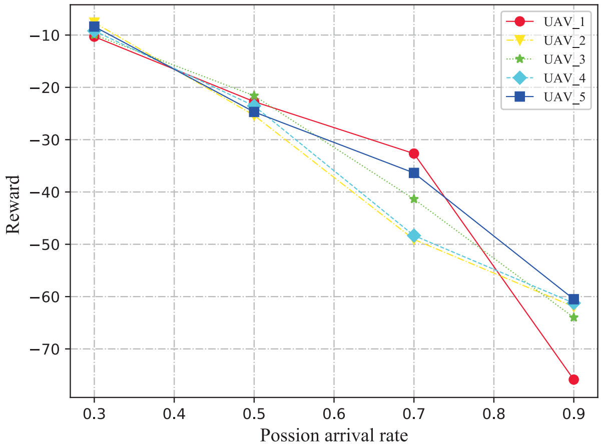

Fig. 8. Reward versus Possion arrival rates in the context of $\lambda = (0.3, 0.5, 0.7, 0.9)$.

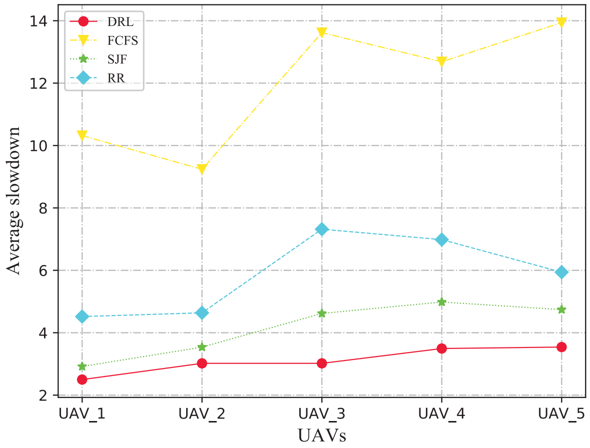

Fig. 9. Average slowdown comparison among DRL, FCFS, SJF, and RR.

same behavior as FCFS. However, if the time slot is shorter than it needed, more task switching can reduce the efficiency of task scheduling. In contrast, the DRL-aided task scheduling algorithm can learn from experience and can reserve the network resources. Hence, the recent arriving small tasks can be dealt with at once, and the average slowdown is minimized.

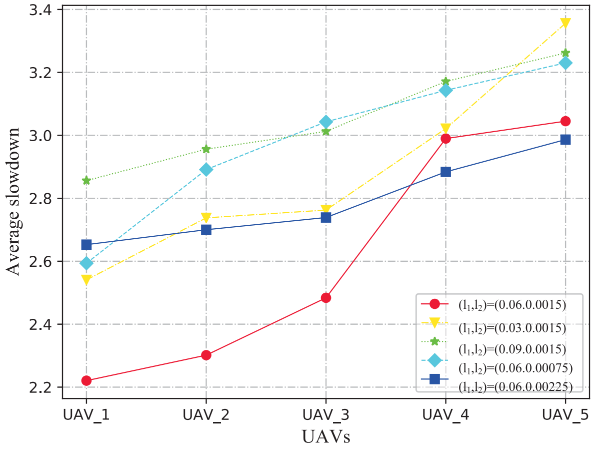

Fig. 10. Average slowdown versus weight factors (ℓ_1, ℓ_2) for the DE algorithm.

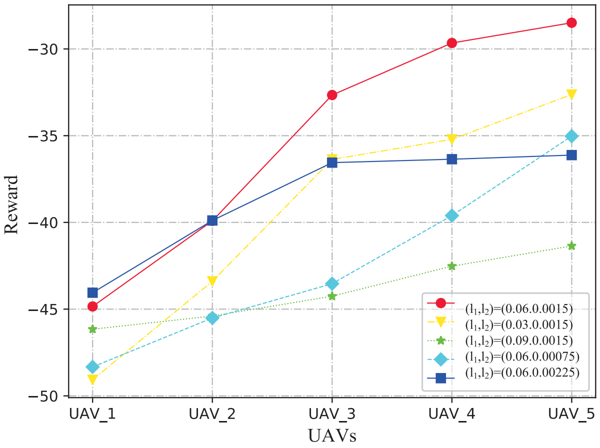

Fig. 11. Reward versus weight factors (ℓ_1, ℓ_2) for the DE algorithm.

Figs. 10 and 11 show the influence of different weight factors ℓ_1 and ℓ_2 according to (12). Since ℓ_1 and ℓ_2 denote the weight factors for load-balancing requirement and average transmission cost, respectively, appropriate ℓ_1 and ℓ_2 will give the MEC system better performance. It can be seen in Figs. 10 and 11 that, when $\ell_1 = 0.06$ and $\ell_2 = 0.0015$, it will get the best overall performance in our system. Furthermore, whether ℓ_1 and ℓ_2 are too small or too large, the average slowdown will increase and the reward will decrease. This is because, underestimating ℓ_1 will lead to an unbalanced load among UAVs while sometimes some MEC nodes cannot meet the QoS requirements, as well as overestimating ℓ_1 will lead to an increase in transmission cost of UAVs. In the same way, underestimating ℓ_2 will result in high transmission cost, and overestimating ℓ_2 will result in an unbalanced load.

Fig. 12 shows the comparison to the classic genetic algorithm (GA) and particle swarm optimization (PSO) in Algorithm 2. In the DE, the population size is 10, and the crossover probability as well as the mutation probability of DE are set as 0.7 and 0.2, respectively. In the PSO, the particle number is 10, which is the same as DE. Moreover, the weight factors of velocity, best position, and globally optimal position are 0.2, 0.5, and 0.5, respectively. In the GA, the population size is 64, and one population is eliminated per generation,

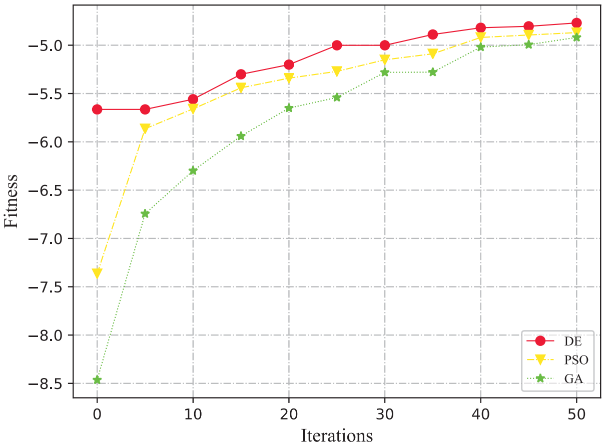

Fig. 12. Fitness comparison among DE, PSO, and GA.

where the crossover probability as well as the mutation probability are the same as DE. Obviously, as iterations increase, the DE algorithm has the best fitness in comparison to the other two algorithms.

VI. Conclusion

The development of the UAV technology has provided a new solution for the task offloading of IoT devices. Compared with fixed ground base stations, UAVs can approach ground IoT devices efficiently due to its easy deployment, low cost, and feasible mobility. In this article, we aim for designing a multiple UAVs-aided MEC system with the goal of global load balance and slowdown minimization. We introduced the DE algorithm to search for near-optimal locations of UAVs. Besides, the GAP-aided approximation algorithm is conceived to determine each connection between UAV and IoT nodes. Moreover, a DRL-based task scheduling scheme is used to schedule offloaded tasks effectively. Finally, simulation results verify the effectiveness and superiority of our proposed algorithm.

References

[1] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in UAV communication networks,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2015.

[2] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[3] J. Zhang *et al.*, “Aeronautical ad hoc networking for the Internet-above-the-clouds,” *Proc. IEEE*, vol. 107, no. 5, pp. 868–911, May 2019.

[4] X. Li, H. Yao, J. Wang, S. Wu, C. Jiang, and Y. Qian, “Rechargeable multi-UAV aided seamless coverage for QoS-guaranteed IoT networks,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10902–10914, Dec. 2019.

[5] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV-based IoT platform: A crowd surveillance use case,” *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[6] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, “UAV-aided MIMO communications for 5G Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1731–1740, Apr. 2019.

[7] N. H. Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[8] J. Wang, C. Jiang, Z. Wei, C. Pan, H. Zhang, and Y. Ren, “Joint UAV hovering altitude and power control for space-air-ground IoT networks,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1741–1753, Apr. 2019.

[9] N. Zhao *et al.*, “UAV-assisted emergency networks in disasters,” *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.

[10] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[11] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[13] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[14] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, and L. Hanzo, “A near-optimal UAV-aided radio coverage strategy for dense urban areas,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9098–9109, Sep. 2019.

[15] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, “Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones,” *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[16] 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.

[17] T. A. Johansen, A. Zolich, T. Hansen, and A. J. Sørensen, “Unmanned aerial vehicle as communication relay for autonomous underwater vehicle—Field tests,” in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*. Austin, TX, USA, Dec. 2014, pp. 1469–1474.

[18] S. Kim, H. Oh, J. Suk, and A. Tsourdos, “Coordinated trajectory planning for efficient communication relay using multiple UAVs,” *Control Eng. Pract.*, vol. 29, pp. 42–49, Aug. 2014.

[19] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Drone small cells in the clouds: Design, deployment and performance analysis,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. San Diego, CA, USA, Dec. 2015, pp. 1–6.

[20] Y. Zeng and R. Zhang, “Energy-efficient UAV communication with trajectory optimization,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[21] 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.

[22] S. W. Loke, “The Internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds,” 2015. [Online]. Available: arXiv:1507.04492.

[23] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[24] F. Zhou, Y. Wu, H. Sun, and Z. Chu, “UAV-enabled mobile edge computing: Offloading optimization and trajectory design,” in *Proc. IEEE Int. Conf. Commun. (ICC)*. Kansas City, MO, USA, May 2018, pp. 1–6.

[25] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, “Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[26] N. Zhao *et al.*, “Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment,” *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.

[27] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, “Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social Internet of vehicles,” *IEEE Access*, vol. 6, pp. 56700–56715, 2018.

[28] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations* (Wiley-Interscience Series in Discrete Mathematics and Optimization). Hoboken, NJ, USA, 1990.

[29] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, “Thirty years of machine learning: The road to Pareto-optimal wireless networks,” *IEEE Commun. Surveys Tuts.*, early access, doi: 10.1109/COMST.2020.2965856.

[30] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, “Distributed *Q*-learning aided heterogeneous network association for energy-efficient IIoT,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2756–2764, Apr. 2020.

[31] H. Yao, T. Mai, C. Jiang, L. Kuang, and S. Guo, “AI routers & network mind: A hybrid machine learning paradigm for packet routing,” *IEEE Comput. Intell. Mag.*, vol. 14, no. 4, pp. 21–30, Nov. 2019.

[32] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” 2013. [Online]. Available: arXiv:1312.5602.

[33] J. Lu, S. Wan, X. Chen, Z. Chen, P. Fan, and K. B. Letaief, “Beyond empirical models: Pattern formation driven placement of UAV base stations,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3641–3655, Jun. 2018.

Lei Yang received the B.S. degree in information management and information system from the Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2014, and the M.S. degree in information and communication engineering from the Beijing University of Technology, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the Beijing Advanced Innovation Center for Future Internet Technology.

His main research interests include the UAV communications and mobile-edge computing.

Haipeng Yao (Member, IEEE) received the Ph.D. degree from the Department of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

He is an Associate Professor with the Beijing University of Posts and Telecommunications. He has published more than 90 papers in prestigious peer-reviewed journals and conferences. He has been engaged in research on future Internet architecture, network AI, big data, cognitive radio networks, and optimization of protocols and architectures for broadband wireless networks.

Jingjing Wang (Member, IEEE) received the B.S. degree (Highest Hons.) in electronic information engineering from Dalian University of Technology, Dalian, China, in 2014, and the Ph.D. degree (Highest Hons.) in information and communication engineering from Tsinghua University, Beijing, China, in 2019.

From 2017 to 2018, he visited the Next Generation Wireless Group chaired by Prof. L. Hanzo, University of Southampton, Southampton, U.K. He is currently a Postdoctoral Researcher with the Department of Electronic Engineering, Tsinghua University. His research interests include resource allocation and network association, learning theory-aided modeling, analysis and signal processing, as well as information diffusion theory for mobile wireless networks.

Dr. Wang was a recipient of the Best Journal Paper Award of the IEEE ComSoc Technical Committee on Green Communications & Computing in 2018 and the Best Paper Award from the IEEE ICC and IWCMC in 2019.

Chunxiao Jiang (Senior Member, IEEE) received the B.S. degree (Hons.) in information engineering from Beihang University, Beijing, China, in 2008, and the Ph.D. degree (Hons.) in electronic engineering from Tsinghua University, Beijing, in 2013.

He was a Postdoctoral Researcher with the Department of Electronic Engineering, Tsinghua University from 2013 to 2016, during which he visited the University of Maryland at College Park, College Park, MD, USA, and the University of Southampton, Southampton, U.K.

Dr. Jiang was a recipient of the IEEE GLOBECOM Best Paper Award in 2013, the IEEE GlobalSIP Best Student Paper Award in 2015, the IEEE IWCMC Best Paper Award in 2017, and the IEEE Communications Society Young Author Best Paper Award in 2017.

Abderrahim Benslimane (Senior Member, IEEE) received the B.S. degree in computer science from the University of Nancy, Nancy, France, in 1987, and the DEA (M.S.) and Ph.D. degrees in computer science from the University of Franche-Comté, Besançon, France, in 1989 and 1993, respectively.

He has been an Associate Professor with the University of Technology of Belfort-Montbéliard since 1994, and a Professor of computer science with Avignon University, Avignon, France, since 2001. He was a Technical International Expert with the French Ministry of Foreign and European Affairs from 2012 to 2016. He served as a Coordinator with the Faculty of Engineering, French University, Cairo, Egypt. His research interests are in the development of communication protocols with the use of graph theory for mobile and wireless networks.

Prof. Benslimane was a recipient of the French Award of Doctoral Supervisions from 2017 to 2021 and attributed the French Award of Scientific Excellency from 2011 to 2014. He was a recipient of the title to supervise researches (HDR 2000) from the University of Cergy-Pontoise, Cergy, France. He served as a Symposium Co-Chair/Leader in many IEEE international conferences, such as ICC, GLOBECOM, AINA, and VTC. He is currently the Chair of the ComSoc TC of Communication and Information Security.

Yunjie Liu received the B.S. degree in technical physics from Peking University, Beijing, China, in 1968.

He is currently the Academician of the China Academy of Engineering, Beijing, the Chief of the Science and Technology Committee of China Unicom, and the Dean of the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing. His research interests include next generation networks, and network architecture and management.