A Reinforcement Learning Based Stochastic Game for Energy-Efficient UAV Swarm Assisted MEC with Dynamic Clustering and Scheduling

Jialiuyuan Li, Changyan Yi, *Member, IEEE*, Jiayuan Chen, You Shi, Tong Zhang, *Member, IEEE*, Xiaolong Li, Ran Wang, *Member, IEEE*, and Kun Zhu, *Member, IEEE*

***Abstract*—In this paper, we study the energy-efficient unmanned aerial vehicle (UAV) swarm assisted mobile edge computing (MEC) with dynamic clustering and scheduling. In the considered system model, UAVs are divided into multiple swarms, with each swarm consisting of a leader UAV and several follower UAVs. These UAVs serve as mobile edge servers, providing computing services to their covered ground end-users. Unlike existing works, we allow UAVs to dynamically cluster into different swarms, in other words, each follower UAV can change its leader based on the time-varying spatial positions, updated application placement, etc. in a dynamic manner. With the objective of maximizing the long-term energy efficiency of the UAV swarm assisted MEC system, a joint optimization problem of UAV swarm dynamic clustering and scheduling is formulated. Considering the inherent cooperation and competition among intelligent UAVs, we further reformulate this problem as a combination of a series of strongly interconnected multi-agent stochastic games, and theoretically prove the existence of the corresponding Nash Equilibrium (NE). Then, we propose a novel reinforcement learning based UAV swarm dynamic coordination (RLDC) algorithm for obtaining such an equilibrium. Furthermore, the convergence and complexity of the RLDC algorithm are analyzed. Simulations are performed to evaluate the performance of RLDC and illustrate its superiority compared to existing approaches.**

***Index Terms*—UAV swarm, MEC, long-term energy efficiency, stochastic game, reinforcement learning**

I. Introduction

Recently, the concept of unmanned aerial vehicle (UAV) assisted mobile edge computing (MEC) [2]–[4] has attracted significant attention due to its high mobility, flexible coverage and rapid deployment in providing fast-responsive supplementary computing services to end-users (e.g., IoT devices). Specifically, in a UAV swarm, a leader UAV possesses the capability to dynamically guide their follower UAVs to approach end-users. Furthermore, by forming into swarms (each of which consists of a leader and multiple followers [5]), UAV swarm assisted MEC can further improve the collaboration among UAVs for enhancing service quality, and thus has become a popular trend for terrain limited and emergency applications, such as wireless inland ship [6], [7] and maritime ship [8].

Although UAV swarm assisted MEC is envisioned as a lightweight and highly efficient paradigm, it faces several inherent restrictions. For instance, the computing workload among different swarms may be severely unbalanced with fixed clustering. Furthermore, the restricted energy capacities of UAVs hinder the practical implementation of this paradigm for providing the long-term MEC services. Moreover, the constrained storage capacities of UAVs hinder their capability to accommodate all applications to meet the varied task requirements. Recent research efforts in this area include cooperative trajectory planning [9], [10] and collaborative task delegation [11]–[13], etc. Nevertheless, there are still several crucial challenges, especially how UAV swarms can cater to dynamic service requirements of IoT devices, and how UAV swarms can be dynamically clustered based on their spatial positions and updated application placement, which are imperative but have not yet been well investigated and are exceedingly difficult due to the following factors. i) Since the MEC service demands of IoT devices vary dynamically, UAV swarms with fixed clustering makes it challenging to balance the computing workload among different swarms. This prompts us to dynamically schedule the clustering of UAVs to collaboratively provide MEC services for IoT devices with balanced workload. ii) UAVs (especially the leaders) are battery-constrained and have to fly to the depot for energy replenishment if necessary, meaning that the leader UAV interrupts computing services, reducing system performance. This motivates us to develop a more efficient approach to dispatch UAVs to return to the depot for energy replenishment. iii) The limited storage capacities of UAVs (both leaders and followers) impede their abilities to store all applications to fulfill diverse task requirements of IoT devices, which inspires us to update application placement of UAVs and enable UAVs to help with each other through task delegations (particularly within the swarm).

In this paper, we investigate the joint optimization of UAV swarm dynamic clustering and scheduling, considering energy replenishment, application placement, trajectory planning and task delegation for UAV swarm assisted MEC. The objective is to optimize the long-term energy efficiency of all UAVs, defined as the number of tasks processed by all UAVs divided by their total energy consumption when offering MEC service.

J. Li, C. Yi, J. Chen, Y. Shi, T. Zhang, R. Wang and K. Zhu are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 211106, China. (E-mail: {jialiuyuan.li, changyan.yi, jiayuan.chen, shyou, zhangt, wangran, zhukun}@nuaa.edu.cn).

X. Li is with the Hunan Provincial General University Key Laboratory of IoT Intelligent Sensing and Distributed Collaborative Optimization, Hunan University of Technology and Business, Changsha, 410205, China. (E-mail: lxl@hutb.edu.cn).

Corresponding authors: Changyan Yi and Xiaolong Li.

This paper has been presented in part at the IEEE WCNC 2024 [1].

In the considered system, leader UAVs lead the swarm in moving to other areas, replenishing energy and updating their application through wired connections. Furthermore, follower UAVs dynamically change swarms and decide whether to process tasks locally or delegate tasks to their followed leader UAV. Addressing the optimization problem is challenging due to several reasons. First, while UAVs are intelligent and capable of making autonomous decisions based on the state information, the system objective of improving total energy efficiency necessitates cooperative strategies among all UAVs, while enabling UAVs to autonomously make individual decisions may potentially result in competitions and suboptimal outcomes. For instance, leader UAVs might prioritize their own computation offloading requests and postpone energy replenishment, disregarding the needs of other UAVs. Moreover, they may focus on placing popular applications without considering the quality of service (QoS) requirements of IoT devices, thereby degrading system performance. Additionally, selfish movement decisions of leader UAVs towards regions with intensive computation requirements can result in collisions among UAV swarms. Follower UAVs might prefer joining UAV swarms that serve more IoT devices, leading to a higher number of follower UAVs occupying a leader UAV's computing resources. Second, we also consider that UAVs do not have access to future environment information.

To tackle these challenges, we reformulate the joint optimization problem as a series of complex multi-agent stochastic games: energy replenishment stochastic game (ERSG), application planning stochastic game (APSG), trajectory planning stochastic game (TPSG), dynamic clustering stochastic game (DCSG), and task delegation stochastic game (TDSG). This formulation allows us to comprehensively describe strategic interactions among UAVs and facilitates a refined problem structure for finding solutions. However, due to the tight coupling among these multi-agent stochastic games, solving them directly remains difficult. To obtain the corresponding equilibrium for these interconnected multi-agent stochastic games, we design a novel reinforcement learning (RL) based UAV swarm dynamic coordination (RLDC) algorithm, with the objective of generating the long-term optimal energy efficient decisions for providing quality services for IoT devices. For clarity, the main contributions of this paper are summarized in the following.

- A joint optimization problem of dynamic clustering and scheduling in UAV swarm assisted MEC is formulated, with the objective of maximizing the long-term energy efficiency.
- Through the observation of cooperation and competition among UAVs as well as the environmental uncertainty, we reformulate the optimization problem as a series of interconnected multi-agent stochastic games, called ERSG, APSG, TPSG, DCSG and TDSG.
- To efficiently obtain the corresponding equilibrium for these interconnected games, we propose a novel algorithm, called RLDC. Additionally, we theoretically prove it existing Nash Equilibrium (NE), and analyze the convergence and complexity of the RLDC algorithm.
- Extensive simulations are performed to demonstrate the superiority of the RLDC algorithm over counterparts. The simulation results validate the effectiveness and efficiency of the RLDC algorithm in achieving optimized solutions for the UAV swarm assisted MEC system.

The rest of this paper is organized as follows: Section II reviews the related work and highlights the novelties of this paper. Section III presents the system model and problem formulation of the considered UAV swarm assisted MEC. In Section IV, a problem reformulation based on multi-agent stochastic games is developed. Section V proposes the RLDC algorithm for optimizing dynamic UAV swarm clustering and scheduling. Simulation results are presented in Section VI, followed by the conclusion in Section VII.

II. Related Work

In recent years, there has been a significant increase in attention towards adopting UAV swarms as edge servers for IoT devices in MEC systems, which can be attributed to the rapid development of UAV technology. For instance, Wang et al. in [14] proposed an optimal collaborative computing offloading method to solve the collaborative task offloading problem in edge computing based on UAV swarms, aiming to minimize the overall task processing delay. Huang et al. in [15] proposed a grouping and role partitioning algorithm to solve the high latency that can result from multi-hop transmissions between UAVs. Seid et al. in [16] proposed a multi-agent reinforcement learning based drone cluster algorithm to provide computing task offloading and resource allocation services for IoT devices to minimize overall network computing costs while ensuring quality of service (QoS) for IoT devices or UEs in IoT networks. Liao et al. in [2] proposed a heuristic algorithm to solve the problem of minimizing UAV swarm energy consumption with the constraints of UAV flight speed and swarm stability in an iterative manner. Fragkos et al. in [17] designed an autonomous MEC server selection scheme for UAV data offloading based on stochastic learning automata theory and developed non-cooperative games to determine which UAV data to offload to selected MEC servers. However, in the majority of these studies, dynamic clustering in UAV swarm assisted MEC was neglected.

To enhance the energy efficiency of UAV swarm assisted MEC, energy replenishment, trajectory planning, task delegation, and application placement have been discussed in existing work. For energy replenishment, Liang et al. in [18] applied magnetic coupled resonant wireless power transmission technology, which makes mobile users collect abundant energy from wireless charging stations in a short period of time. In terms of trajectory planning, Mou et al. in [19] designed a UAV swarm trajectory algorithm that performs specific coverage tasks within patches to solve the coverage problem of three-dimensional irregular terrain. For task delegation, Li et al. in [11] proposed a layered network architecture based on UAV swarms to jointly delegate sensing task and computing to improve computing resource utilization. For application placement, to the best of our knowledge, only a few existing research focus on application placement [4], [20]. Moreover, the energy efficiency optimization of UAV swarm

TABLE I
A TABLE COMPARING OUR WORK WITH THE EXISTING WORKS.

Reference	Dynamic clustering	Energy replenishment	Application placement	Trajectory planning	Task delegation
[14]	✗	✗	✗	✗	✓
[15]	✗	✗	✗	✗	✗
[16]	✗	✗	✗	✗	✓
[2]	✗	✗	✗	✗	✓
[17]	✗	✗	✗	✓	✓
[18]	✗	✗	✗	✗	✓
[19]	✗	✓	✗	✗	✓
[11]	✗	✗	✗	✓	✗
[4]	✗	✓	✓	✓	✗
Our work	✓	✓	✓	✓	✓

TABLE II
IMPORTANT NOTATIONS IN THIS PAPER

Symbol	Meaning
$\mathcal{M}$	Set of leader UAVs
$\mathcal{N}$	Set of follower UAVs
$\mathcal{K}$	Set of IoT devices
$\mathcal{L}$	Positions set of leader UAVs
$\mathcal{F}$	Positions set of follower UAVs
$\mathcal{I}$	Positions set of IoT devices
T	The number of time slot
l	Length of the large grid
q	Length of the small grid
V	UAV Velocity
C	The number of the task types
$\mathcal{Z}_n$	Set of IoT devices served by follower UAV n
B	Channel bandwidth
λ	Path loss
φ	LoS probability
γ	SINR
μ	Instantaneous achievable rate
ϖ	Power spectral density of noise
ξ	Effective capacitance coefficient
$\boldsymbol{\omega}^L$	Applications placement of leader UAVs
$\boldsymbol{\omega}^F$	Applications placement of follower UAVs
$\boldsymbol{\varepsilon}$	Set of leader UAVs returning to the depot
$\boldsymbol{\delta}$	Set of follower UAVs following which leader UAVs
$\boldsymbol{\phi}$	Set of whether delegating tasks to leader UAVs

assisted MEC involving multiple decision variables can be formulated as a joint optimization problem. Nevertheless, the joint optimization of dynamic clustering and scheduling has not been previously investigated.

In summary, unlike prior existing work, this paper specifically explores the following issues related to UAV swarm assisted MEC systems. First, a joint optimization problem of dynamic clustering and scheduling in UAV swarm assisted MEC is formulated. Second, we reformulate the optimization problem as a series of interconnected multi-agent stochastic games, and subsequently propose a novel algorithm to determine the corresponding equilibrium. To highlight the novelties of our work, we summarize the differences between our work and existing works regarding UAV swarm-assisted MEC in Table I.

III. System Model and Problem Formulation

A. Overview

Consider a deployment scenario of UAV swarm-assisted MEC in a target region, as shown in Fig. 1. The system consists of a group of leader UAVs, denoted as $\mathcal{M}$ with

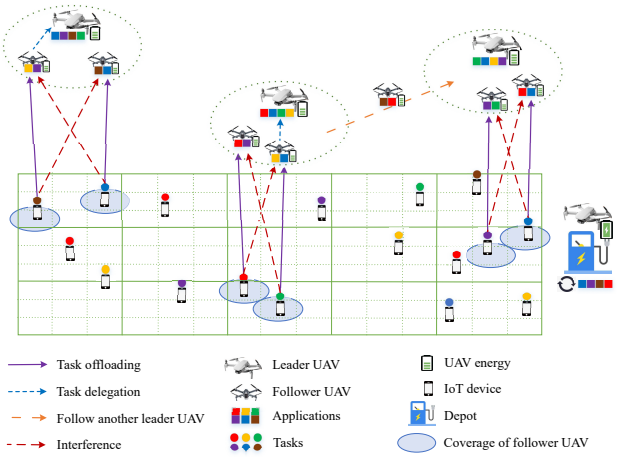

Fig. 1. An illustration of the considered UAV swarm assisted MEC system.

$|\mathcal{M}| = M$, a group of follower UAVs, denoted as $\mathcal{N}$ with $|\mathcal{N}| = N$, and a set of IoT devices scattered randomly on the ground, denoted as $\mathcal{K}$ with $|\mathcal{K}| = K$. At the edge of the target region, a depot is deployed to serve leader UAVs with energy replenishment and application placement update via wired connections. We investigate a time-slotted operation framework, which is characterized by $t \in \{1, 2, ..., T\}$. The target region is uniformly divided into large grids with side length l to limit the activities scope of each swarm. Meanwhile, the large grids are further uniformly partitioned into small grids with side length q to specify the activities of follower UAVs. Similar to [21], the downlink transmission range of each follower UAV is set as $\sqrt{2}q/2$, such that it can cover a small grid for computation outcome feedback. Additionally, we denote the set of IoT devices served by follower UAV $n \in \mathcal{N}$ as $\mathcal{Z}_n$. Since we consider that each UAV swarm shares a frequency band B, IoT devices may introduce interference to other UAVs within the swarm. All important notations are listed in Table II.

To clearly describe the process of UAV swarms providing MEC service over the target region, we illustrate the time slot structure, as shown in Fig. 2. At the beginning of each time slot t, each leader UAV determines whether to return to the depot for replenishing energy and updating applications. If a leader UAV chooses not to return to the depot, it will

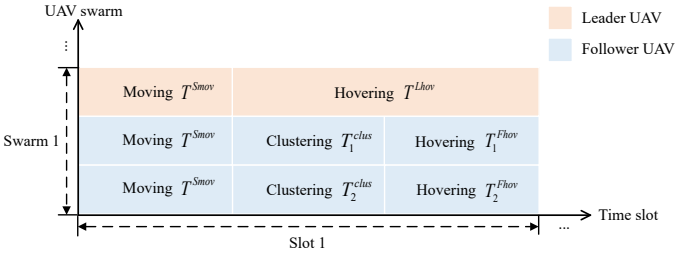

Fig. 2. Time slot structure of UAV swarms providing MEC service over the target region.

autonomously lead its UAV swarm to move to another adjacent large grid with a constant velocity and a direction (forward, backward, left, or right), and this period of time is denoted as T^{Smov}. Then, the leader UAV hovers over the center of the large grid with the time T^{Lhov}. During the period of time T^{Lhov}, each follower UAV independently determines which leader UAV to follow for clustering with the time T^{clus}, and hover over the center of the small grid for processing tasks and task delegation with the time T^{Fhov}. Additionally, since the size of the results is significantly small compared to the offloaded task, the delay and energy consumption associated with delegating results back from follower UAVs to IoT devices are omitted in this paper. Note that if the follower UAV cannot process the tasks received from IoT devices, it will delegate the task to their followed leader UAV for processing. Moreover, to improve the energy efficiency of follower UAVs, leader UAVs provide energy replenishment to the follower UAVs in a swarm through wireless power transfer technology [22]. In contrast, if the leader UAV chooses to return to the depot, it will replenish its energy and update its application placement.

B. Task Offloading and Delegation Model

In this paper, let $\boldsymbol{\mathcal{L}}(t) = \{\mathcal{L}_1(t), \mathcal{L}_2(t)..., \mathcal{L}_M(t)\}$, $\boldsymbol{\mathcal{F}}(t) = \{\mathcal{F}_1(t), \mathcal{F}_2(t)..., \mathcal{F}_N(t)\}$ and $\boldsymbol{\mathcal{I}}(t) = \{\mathcal{I}_1(t), \mathcal{I}_2(t)..., \mathcal{I}_K(t)\}$ denote the set of leader UAVs' positions, the set of follower UAV's positions and set of IoT devices' positions, respectively. Therein, $\mathcal{L}_m(t) = (x_m^L(t), y_m^L(t))$, $\mathcal{F}_n(t) = (x_n^F(t), y_n^F(t))$, $\mathcal{I}_k(t) = (x_k^I(t), y_k^I(t))$ represent their horizontal coordinates at time slot t, respectively. Thus, the distance between follower UAV n and IoT device k, as well as the distance between leader UAV m and follower UAV n at time slot t can be mathematically expressed as $d_{n,k}(t) = \sqrt{(x_n^F(t) - x_k^I(t))^2 + (y_n^F(t) - y_k^I(t))^2 + H_F^2}$ and $d_{m,n}(t) = \sqrt{(x_m^L(t) - x_n^F(t))^2 + (y_m^L(t) - y_n^F(t))^2 + (H_L - H_F)^2}$, where H_L and H_F denote the fixed flight altitudes of leader UAVs and follower UAVs, respectively.

Following the literature [10], the line-of-sight (LoS) probability between follower UAV n and IoT device k at time slot t can be expressed as $\varphi_{n,k}(t) = a \cdot exp(-b(\arctan(H_F/d_{n,k}(t)) - a))$, where a and b are constant values depending on the environment. Based on this, their path loss is given by $\lambda_{n,k}(t) = 20\log\sqrt{H_F^2 + d_{n,k}(t)^2} + \varphi_{n,k}(t)(\eta_{LoS} - \eta_{NLoS}) + 20\log((4\pi f^c)/c^l) + \eta_{NLoS}$, where f^c signifies the carrier frequency, and c^l signifies the speed of light. η_{LoS} and η_{NLoS} denote the losses corresponding to LoS and non-LoS, respectively. Following the literature [23], the path loss between leader UAV m and follower UAV n at time slot t can be expressed as $\lambda_{m,n}(t) = 32.45 + 20\log f^c + 20\log d_{m,n}(t)$.

Given the reuse of a common frequency band across all links in a UAV swarm, the signal-to-interference-plus-noise ratio (SINR) at follower UAV n for the uplink communication of IoT device k, and at leader UAV m for the uplink communication of follower UAV n during time slot t, can be expressed as $\gamma_{n,k}(t) = p_k^I 10^{-\lambda_{n,k}(t)/10} / (\sum_{i \in \mathcal{G}_n \setminus k} p_i^I 10^{-\lambda_{n,k}(t)/10} + \varpi)$ and $\gamma_{m,n}(t) = p_n^F 10^{-\lambda_{m,n}(t)/10} / \varpi$, respectively. p_k^I and p_n^F denote the transmission power of IoT device k and follower UAV n, respectively. ϖ denotes the power spectral density of noise. Besides, it is assumed that follower UAV n can only receive one task offloaded from IoT device $k \in \mathcal{Z}_n$ at time slot t. Hence, the instantaneous achievable rates of IoT device k offloading tasks to follower UAV n, and follower UAV n delegating tasks to leader UAV m at time slot t can be written as $\mu_{n,k}^F(t) = B\log_2(1 + \gamma_{n,k}(t))$ and $\mu_{m,n}^L(t) = B\log_2(1 + \gamma_{m,n}(t))$, respectively. Let $\mathcal{C} = \{1, 2, ..., C\}$ denote the set of task types. Therefore, the time of IoT device k offloading its task whose type is $c \in \mathcal{C}$ to follower UAV n, and the time of follower UAV n delegating its task whose type is c to leader UAV m at time slot t can be expressed as:

$$T_{n,k,c}^{off}(t) = \upsilon_{k,c}(t)\kappa_{k,c}/\mu_{n,k}^F(t) \tag{1}$$

and

$$T_{m,n,c}^{dele}(t) = (1 - \varepsilon_m(t))\delta_{m,n}(t)\phi_{m,n,c}(t)\kappa_{k,c}/\mu_{m,n}^L(t), \tag{2}$$

respectively, where $\kappa_{k,c}$ indicates the size of the task whose type is c offloaded from IoT device k. $\upsilon_{k,c}(t) \in [0, 1]$, and $\upsilon_{k,c}(t) = 1$ means that IoT device k requests its task whose type is c at time slot t, otherwise $\upsilon_{k,c}(t) = 0$. Note that we consider each IoT device generating only one task request at time slot t in this paper. Besides, $\varepsilon_m(t) \in [0, 1]$, and $\varepsilon_m(t) = 1$ means that leader UAV m returns to the depot at time slot t, otherwise $\varepsilon_m(t) = 0$. Meanwhile, $\delta_{m,n}(t) \in [0, 1]$, and $\delta_{m,n}(t) = 1$ means that follower UAV n follows the leader UAV m at time slot t, otherwise $\delta_{m,n}(t) = 0$. Additionally, $\phi_{m,n,c}(t) \in [0, 1]$, and $\phi_{m,n,c}(t) = 1$ means that follower UAV n delegates its task whose type is c to the leader UAV m, otherwise $\phi_{m,n,c}(t) = 0$.

C. UAV Computation Model

As shown in Figure 2, in this paper, we consider $T_{n,k,c}^{off}(t) < T_n^{Fhov}(t)$, indicating that $T_n^{Fhov}(t)$ is sufficiently long for follower UAV n to receive each task offloaded by IoT device $k \in \mathcal{Z}_n$ at time slot t. Besides, the application c placed in follower UAV n and leader UAV m can be defined as $\omega_{n,c}^F(t) \in \{0, 1\}$ and $\omega_{m,c}^L(t) \in \{0, 1\}$, respectively. $\omega_{n,c}^F(t) = 1$ means that follower UAV n places the application which can process task type c, otherwise $\omega_{n,c}^F(t) = 0$. And the definition of $\omega_{m,c}^L(t)$ is similar to that of $\omega_{n,c}^F(t)$. Consequently, the size of tasks processed by follower UAV n and leader UAV m can

be expressed as:

$$D_n^F(t) = \min\{\sum_{k\in\mathcal{G}_n}\sum_{m=1}^{M}\sum_{c=1}^{C}(1-\phi_{m,n,c}(t))\upsilon_{k,c}(t) \omega_{n,c}^F(t)\kappa_{k,c}, f_n^F(T_n^{Fhov}(t)-\min\{\boldsymbol{T}_n^{off}(t)\})\} \tag{3}$$

and

$$D_m^L(t) = \min\{\sum_{n=1}^{N}\sum_{c=1}^{C}\delta_{m,n}(t)\phi_{m,n,c}(t)\upsilon_{k,c}(t) \omega_{m,c}^L(t)\kappa_{k,c}, f_m^L(T_n^{Fhov}(t)-\min\{\boldsymbol{T}_m^{dele}(t)\})\}, \tag{4}$$

respectively, where $\boldsymbol{T}_n^{off}(t)=\{T_{n,1,1}^{off}(t),...,T_{n,k,c}^{off}(t),..., T_{N,K,C}^{off}(t)\}$ and $\boldsymbol{T}_m^{dele}(t)=\{T_{m,1,1}^{dele}(t),...,T_{m,n,c}^{dele}(t),..., T_{M,N,C}^{dele}(t)\}$. f_n^F and f_m^L indicate the computing capacity of follower UAV n and leader UAV m (the CPU cycle rate). $T_n^{Fhov}(t)-\min\{\boldsymbol{T}_n^{off}(t)\}$ and $T_n^{Fhov}(t)-\min\{\boldsymbol{T}_m^{dele}(t)\}$ indicate that follower UAV n and leader UAV m start to process tasks when the first task is totally received, respectively.

D. UAV Propulsion Model

In this paper, we adopt a propulsion power model of rotary-wing UAVs to compute the propulsion powers of leader UAVs and follower UAVs, which is dependent on the velocity v. Specifically, the propulsion power of each UAV can be expressed as follows:

$$P^{pro}(v) = \frac{1}{2}(\frac{S_f}{R_sA})\rho R_sAv^3+(\frac{\varphi_e}{8}\rho R_sA\Omega_e^3R_e^3)(1+\frac{3v^2}{(\Omega_eR_e)^3})+((1+\varsigma_p) \frac{(gM_{UAV})^{\frac{3}{2}}}{\sqrt{2\rho A}})(\sqrt{1+v^4/4(\sqrt{gM_{UAV}/2\rho A})^2}-v^2/2(\sqrt{\frac{gM_{UAV}}{2\rho A}}))^{\frac{1}{2}}, \tag{5}$$

where the parameters in (5) are described in Table III.

E. UAV Energy Model

In this paper, we consider that UAV energy consumption includes task delegation energy consumption, computing energy consumption and propulsion energy consumption. First, the task delegation energy consumption is given by $E_n^{dele}(t)=p_n^FT_{m,n,c}^{dele}(t)$, where p_n^F indicates the transmission power of follower UAV n. Moreover, the computing energy consumption of follower UAV n and leader UAV m can be written as $E_n^{comp}(t)=\xi(f_n^F)^2D_n^F(t)$ and $E_m^{comp}(t)=\xi(f_m^L)^2D_m^L(t)$, where ξ indicates the effective capacitance coefficient. Besides, f_n^F and f_m^L indicate the CPU frequencies of follower UAV n and leader UAV m, respectively.

Additionally, the propulsion energy consumption of follower UAV n and leader UAV m can be expressed as:

$$E_n^{pro}(t)=P^{pro}(v)(((1-\delta_{m,n}(t-1)\delta_{m,n}(t))d_{m,n}(t)+l/v) +P^{pro}(0)T_n^{Fhov}(t) \tag{6}$$

and

$$E_m^{pro}(t)=\varepsilon_m(t)P^{pro}(v)d_m^{return}(t)/v +(1-\varepsilon_m(t))(P^{pro}(v)l/v+P^{pro}(0)T_n^{Lhov}(t)), \tag{7}$$

respectively, where $d_m^{return}(t)$ indicates the distance between leader UAV m and depot at time slot t. Specifically, $E_n^{pro}(t)$ consists of the dynamic clustering energy consumption, horizontal moving energy consumption and the hovering energy consumption of follower UAV n at each time slot t. Meanwhile, $E_m^{pro}(t)$ consists of the energy consumption of returning to the depot, horizontal moving energy consumption and the hovering energy consumption of leader UAV m at each time slot t. For simplicity, we consider that the clustering distance between follower UAV n and its following leader UAV m is denoted as $d_m^{return}(t)$, which is the average distance between follower UAV n and the UAV swram containing leader UAV m.

Furthermore, let $E_m^{charge}(t)$ denote the energy consumption of leader UAV m aerial charging to the follower UAVs in its UAV swarm. For simplicity, we consider that the energy consumed by each follower UAV can be fully replenished by the leader UAV in the swarm at time slot t, which can be written as:

$$E_m^{charge}(t)=\begin{cases}\sum_{n=1}^{N}(\sum_{c=1}^{C}\delta_{m,n}(t)p_n^FT_{m,n,c}^{dele}(t) \\ +\xi(f_n^F)^2D_n^F(t)+E_n^{pro}(t)),\ \varepsilon_m(t)=0, \\ 0,\end{cases} \tag{8}$$

Based on these, let $E_m^{resi}(t)$ and E_m^{total} denote the residual energy and energy capacity of leader UAV m at time slot t, respectively. $E_m^{resi}(t)$ can be formulated as:

$$E_m^{resi}(t)=\begin{cases}E_m^{resi}(t-1)-E_m^{comp}(t)-E_m^{pro}(t)-E_m^{charge}(t), \\ \qquad\qquad \varepsilon_m(t-1)=0\ ,\ \varepsilon_m(t)=0, \\ E_m^{resi}(t-1)-E_m^{return}(t), \qquad \varepsilon_m(t)=1, \\ E_m^{total}, \qquad\qquad \varepsilon_m(t-1)=1\ ,\ \varepsilon_m(t)=0,\end{cases} \tag{9}$$

F. Application Placement Model

In this paper, if leader UAVs choose to return to the depot, they will update their application placement to provide better MEC service. Besides, to guarantee the QoS for IoT devices, it is essential to allocate each type of application to a leader UAV that remains airborne over the target region during each time slot. This allocation can be mathematically represented as follows:

$$\sum_{m=1}^{M}\omega_{m,c}^L(t)\varepsilon_m(t)\geq 1, \forall c\in\mathcal{C}. \tag{10}$$

After replenishing its energy and updating its applications, leader UAV m will return to its original location within the target region and resume its MEC service. It is noteworthy that applications placed in leader UAV m must not exceed its storage capacity, which is given by:

$$\sum_{c=1}^{C}\omega_{m,c}^L(t)\leq S^L, \tag{11}$$

where S^L indicates the maximum number of applications placed in each leader UAV.

G. Problem Formulation

In this paper, we denote $E^{effi}(t)$ as the energy efficiency of all UAVs at time slot t, which means that the size of tasks processed by all UAVs relative to their energy consumption during time slot t. This can be mathematically expressed as:

$$E^{effi}(t)=\frac{\sum_{m=1}^{M}D_m^L(t)+\sum_{n=1}^{N}D_n^F(t)}{\sum_{m=1}^{M}(1-\varepsilon_m(t))(E_m^{resi}(t-1)-E_m^{resi}(t))}. \tag{12}$$

Then we aim to jointly optimize the dynamic clustering and scheduling of the considered UAV swarm assisted MEC

TABLE III
UAV ENERGY MODEL PARAMETERS

Parameter	Descriptions
p_n^F	transmission power of follower UAV n
f_n^F	CPU frequencies of follower UAV n
R_s	Rotor solidity
ς_p	Induced power factor
φ_e	Drag coefficient of the blade
Ω_e	Angular velocity of the blade
S_f	Equivalent flat plate area of the fuselage
ξ	effective capacitance coefficient
f_m^L	CPU frequencies of leader UAV m
M_{UAV}	UAV mass
ρ	Air density
R_e	Rotor radius
A	Rotor disc area
g	Gravity acceleration

system, with the objective of maximizing the long-term energy efficiency of all UAVs, which can be formulated as:

$$[\mathcal{P}1]: \max_{\boldsymbol{\mathcal{L}}(t),\boldsymbol{\omega}^L(t),\boldsymbol{\varepsilon}(t),\boldsymbol{\delta}(t),\boldsymbol{\phi}(t)} \lim_{T\to+\infty} \frac{1}{T}\sum\nolimits_{t=1}^{T} E^{effi}(t) \tag{13a}$$

$$s.t., (10), (11),$$

$$E_m^{resi}(t) \geq E_m^{return}(t), m \in \mathcal{M}, \tag{13b}$$

$$\sum\nolimits_{m=1}^{M} \delta_{m,n}(t) = 1, \forall n \in \mathcal{N}, \tag{13c}$$

$$1 \leq \sum\nolimits_{n=1}^{N} \delta_{m,n}(t) \leq N^{clus}, \forall m \in \mathcal{M}, \tag{13d}$$

$$X^{low} \leq (1-\varepsilon_m(t))x_m^L(t) \leq X^{up}, m \in \mathcal{M}, \tag{13e}$$

$$Y^{low} \leq (1-\varepsilon_m(t))y_m^L(t) \leq Y^{up}, m \in \mathcal{M}, \tag{13f}$$

where $\boldsymbol{\omega}^L(t) = \{\omega_{1,1}^L(t), ..., \omega_{M,C}^L(t)\}$, $\boldsymbol{\varepsilon}(t) = \{\varepsilon_1(t), \varepsilon_2(t), ..., \varepsilon_M(t)\}$, $\boldsymbol{\delta}(t) = \{\delta_{1,1}(t), ..., \varepsilon_{M,N}(t)\}$ and $\boldsymbol{\phi}(t) = \{\phi_{1,1,1}(t), ..., \phi_{M,N,C}(t)\}$. Constraint (13b) indicates that the residual energy consumption can not be less than the returning energy consumption for each leader UAV at time slot t; constraint (13c) indicates that each follower UAV must follow at least one leader UAV; constraint (13d) indicates that each UAV swarm contains at least one and not more than N^{clus} follower UAVs, where N^{clus} indicates the maximum number of follower UAVs that can be accommodated in a UAV swarm; constraint (13e) indicates that the abscissa of each leader UAV cannot exceed the upper boundary X^{up} or fall below the lower boundary X^{low} unless it returns to depot; constraint (13f) indicates that the ordinate of each leader UAV cannot exceed the upper boundary Y^{up} or fall below the lower boundary Y^{low} unless it returns to depot.

Remark 1: Taking into account UAV path planning induced by dynamic clustering may also be interesting in the optimization of the UAV swarm assisted MEC system. In fact, many existing work have studied such an issue, i.e., individual UAV path planning [24]–[26]. To be more specific, each follower UAV flies to a new UAV cluster by determining a series of actions (e.g., moving forward, back, left and right to another adjacent small gird). Meanwhile, the relevant constraints (e.g., collision avoidance constraint) are required to be taken into account in the dynamic clustering problem, which makes the follower UAV's action space very large. Hence, it may require the introduction of more approaches, such as autonomous path planning algorithm based on a tangent intersection and target guidance strategy (APPATT) [24], tangent-based (3D-TG) method [25] and adaptive clustering-based algorithm [26]. Obviously, this is not trivial but beyond the focus of the current paper, and thus we would like to leave this extension and integration in our future work. In the following sections, we first reformulate the optimization problem $[\mathcal{P}1]$ as a series of interconnected multi-agent stochastic games, and subsequently introduce a novel algorithm to obtain the corresponding solution.

IV. Problem Reformulation based on Multi-agent Stochastic Game

A. Game Statement

To solve the UAV swarm dynamic clustering and scheduling optimization in the unknown stochastic environment, the statement of the multi-agent stochastic game is presented. Since UAVs have no prior information on the task requirements of IoT devices, the complete information based dynamic clustering and scheduling methods fail to effectively solve problem [P1] under an unknown environment. At the beginning of each time slot, a new stochastic state emerges in the environment, which is impacted by both the previous state and the actions taken by all UAVs in the preceding time slot. As a result, the state-action transition adheres to the Markov property. Considering the intelligence of UAVs, in order to solve problem [P1], each UAV is allowed to independently make decisions, and the relationships of cooperation and competition exist among them. UAVs are formed as UAV swarms to cooperatively take actions (dynamic clustering and scheduling) to maximize the energy efficiency of the whole UAV swarm assisted MEC system. Meanwhile, UAVs independently make decisions based on individual interest, in which the conflicts of interest among UAVs leads to competition. The competition caused by granting UAVs to make decisions independently is outlined as below.

1) For energy replenishment, each leader UAV may prefer to processing more tasks for maximizing its energy efficiency but is reluctant to return to the depot to replenish its energy until its battery is completely depleted, potentially leading to the collapse of constraint (10).

2) For application placement, in order to optimize leader UAVs' energy efficiency, each leader UAV has a tendency to prioritize the placement of applications that are frequently requested. However, this situation may neglect the QoS requirements of several IoT devices, leading them to experience resource starvation.

3) For trajectory planning, each leader UAV aims to maximize the energy efficiency of its swarm by moving to large grids with demands for intensive computation. However, this situation may result in collisions among UAV swarms.

4) For dynamic clustering, each follower UAV can leave the previous swarm and join a new swarm for processing more tasks. However, this situation may result in many follower UAVs occupying a certain leader UAV's computing resources, making several tasks impossible to be processed in the leader UAV's hovering time.

5) For task delegation, each follower UAV decides whether to delegate tasks to their leader UAV or not, which will result in computing resource competition among follower UAVs F.

Additionally, considering the uncertainty of the future environment information, such as the uncertain task requirements of IoT devices, we reformulate the joint optimization problem $[\mathcal{P}1]$ as a series of strongly interconnected multi-agent stochastic games as described below.

B. Game Formulation

Firstly, we define the multi-agent stochastic game $\mathcal{G}$ as a tuple $\langle \mathcal{U}, \mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{R} \rangle$ based on the discussion above.

1) $\mathcal{U} = \{1, 2, ..., U\}$ denotes the set of agents.

2) $\mathcal{S}$ denotes the set of environment states. $s(t)$ denotes the environment state at time slot t.

3) $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_U\}$ represents the set of joint actions, where $\mathcal{A}_u$ refers to the set of individual actions for agent u. The joint action at time slot t is represented as $\boldsymbol{a}(t) \in \mathcal{A}$, while the individual action of agent u is represented as $a_u(t) \in \mathcal{A}_u$. Hence, the joint action can be expressed as $\boldsymbol{a}(t) = \{a_1(t), ..., a_U(t)\}$.

4) $\mathbb{P}$ denotes the $U \times U$ matrix of state transition probabilities. $p_{ss'}(\boldsymbol{a}(t))$ signifies the probability of transitioning from state s to s' by taking the joint action $\boldsymbol{a}(t) \in \mathcal{A}$.

5) $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_U\}$ indicates the reward function, where r_u represents the set of immediate reward of agent $u \in \mathcal{U}$.

For the formulated stochastic game $\mathcal{G}$, the mapping from the set of states to the set of actions is represented by the policy denoted as $\pi_u : \mathcal{S} \longrightarrow \mathcal{A}_u$. It is worth noting that the expected reward of each agent depends on the joint policy instead of the individual policy. Thus, we introduce the NE to determine the joint policy, which is defined as follows:

Definition 1: An NE refers to a set of optimal policy for multi-agent stochastic game $\mathcal{G}$, denoted as $\boldsymbol{\pi}^* = \{\pi_1^*, \pi_2^*, ..., \pi_U^*\}$, if and only if no player can minimize its expected discount function by unilaterally departing [27], i.e., $\hat{\Theta}_u(\pi_u^*, \boldsymbol{\pi}^*_{\mathcal{U}\setminus\{u\}}) \geq \hat{\Theta}_u(\pi_u, \boldsymbol{\pi}^*_{\mathcal{U}\setminus\{u\}}), \forall u \in \mathcal{U}, \forall \pi_u \in \boldsymbol{\pi}$.

Second, we model the multi-UAV dynamic clustering and scheduling problem as interconnected multi-agent stochastic games. The multi-agent stochastic games are consisted of ERSG $\langle \mathcal{U}, \mathcal{S}^{ER}, \mathcal{A}^{ER}, \mathbb{P}^{ER}, \mathcal{R}^{ER} \rangle$, APSG $\langle \mathcal{U}, \mathcal{S}^{AP}, \mathcal{A}^{AP}, \mathbb{P}^{AP}, \mathcal{R}^{AP} \rangle$, TPSG $\langle \mathcal{U}, \mathcal{S}^{TP}, \mathcal{A}^{TP}, \mathbb{P}^{TP}, \mathcal{R}^{TP} \rangle$, DCSG $\langle \mathcal{U}, \mathcal{S}^{DC}, \mathcal{A}^{DC}, \mathbb{P}^{DC}, \mathcal{R}^{DC} \rangle$ and TDSG $\langle \mathcal{U}, \mathcal{S}^{TD}, \mathcal{A}^{TD}, \mathbb{P}^{TD}, \mathcal{R}^{TD} \rangle$. Specifically, for ERSG, APSG and TPSG, each leader UAV independently selects an action based on the current environmental states $s^{ER}(t) \in \mathcal{S}^{ER}$, $s^{AP}(t) \in \mathcal{S}^{AP}$ and $s^{TP}(t) \in \mathcal{S}^{TP}$, respectively. Subsequently, the joint actions $\boldsymbol{a}^{ER}(t) \in \mathcal{A}^{ER}$, $\boldsymbol{a}^{AP}(t) \in \mathcal{A}^{AP}$ and $\boldsymbol{a}^{TP}(t) \in \mathcal{A}^{TP}$ are formed. After executing these joint actions, rewards are obtained according to $\mathcal{R}^{ER}$, $\mathcal{R}^{AP}$ and $\mathcal{R}^{TP}$, and the environment transitions to its next state by $\mathbb{P}^{ER}$, $\mathbb{P}^{AP}$ and $\mathbb{P}^{TP}$, respectively. Similarly, for DCSG and TDSG, each follower UAV independently selects an action based on the current environmental states $s^{DC}(t) \in \mathcal{S}^{DC}$ and $s^{TD}(t) \in \mathcal{S}^{TD}$, respectively. Subsequently, the joint actions $\boldsymbol{a}^{DC}(t) \in \mathcal{A}^{DC}$ and $\boldsymbol{a}^{TD}(t) \in \mathcal{A}^{TD}$ are formed. After executing these joint actions, rewards are obtained according to $\mathcal{R}^{DC}$ and $\mathcal{R}^{TD}$, and the environment transitions to its next state by $\mathbb{P}^{DC}$ and $\mathbb{P}^{TD}$, respectively.

Note that, ERSG, APSG, TPSG, DCSG and TDSG are inherently interconnected. Specifically, the joint action of ERSG $\boldsymbol{a}^{ER}(t) \in \mathcal{A}^{ER}$ determining whether leader UAVs lead their swarms to move to another adjacent large grid or update application placement at time slot t changes the states $\mathcal{S}^{TP}$ and $\mathcal{S}^{AP}$, respectively. Moreover, since different UAV swarms' trajectories influence the dynamic clustering time of follower UAVs, the joint action of TPSG $\boldsymbol{a}^{TP}(t) \in \mathcal{A}^{TP}$ changes the state $\mathcal{S}^{DC}$. Additionally, since the decisions of follower UAVs choosing which leader UAV to follow influence UAV swarms' formations for further delegating tasks, the joint action of DCSG $\boldsymbol{a}^{DC}(t) \in \mathcal{A}^{DC}$ changes the state $\mathcal{S}^{TD}$. Finally, since the computing energy consumption of leader UAVs influences their residual energy, the joint action of TDSG $\boldsymbol{a}^{TD}(t) \in \mathcal{A}^{TD}$ changes the state $\mathcal{S}^{ER}$. In the following section, we propose a novel algorithm called RLDC to obtain equilibrium of these interconnected multi-agent stochastic games.

***Lemma** 1:* For ERSG, the optimal policy for leader UAV $m \in \mathcal{M}$ can be denoted by π_m^{ER*}, namely, $\{\pi_1^{ER*}, \pi_2^{ER*}, ..., \pi_M^{ER*}\}$ forms the NE.

Proof: Please refer to Appendix A. ■

According to Definition 1 and Lemma 1, we aim to identify a Nash equilibrium (NE) strategy for each agent u at any given state $s(t)$. It is worth noting that even though the environmental information available to each agent may be imperfect, they have the chance to learn the optimal policies through repeated interactions with the environment [28]. Since ERSG, APSG, TPSG, DCSG and TDSG can converge to their optimal policies by the RLDC algorithm respectively, which has been theoretically analyzed in Theorem 1 of Section V, the interconnected multi-agent stochastic games can obtain the optimal policy consisting of the optimal policies of ERSG, APSG, TPSG, DCSG and TDSG. Therefore, the proposed RLDC algorithm can achieve the equilibrium point for the interconnected multi-agent stochastic games.

V. Reinforcement Learning Based UAV Swarm Dynamic Coordination Algorithm

In this section, we first introduce a finite-state Markov decision process (MDP) to characterize the game process of each leader UAV and follower UAV. Then, we propose the RLDC algorithm to maximize the expected long-term reward of the considered UAV swarm-assisted MEC system, where each learner operates in an unknown stochastic environment and does not know the reward and transition functions in advance. Since the state and action transitions satisfy the Markov property in ERSG, APSG, TPSG, DCSG and TDSG, we characterize the strategic decision processes of each leader UAV and follower UAV by a series of respective MDPs.

MDP for each leader UAV in ERSG: To design an optimal schedule for energy replenishment of all leader UAVs in ERSG, the individual decision-making problem for each leader UAV $m \in \mathcal{M}$ can be modeled as an MDP $(\mathcal{S}^{ER}, \mathcal{A}_m^{ER}, \mathcal{R}_m^{ER}, \mathbb{P}^{ER})$.

1) Environment state for each leader UAV in ERSG: To reduce the size of the state space in ERSG, we divide the energy of leader UAVs into several levels. Specifically, the energy level of leader UAV m can be written as $E_m^{level}(t) = \lceil E_m^{resi}(t)/E^{unit} \rceil$, where E^{unit} indicates the UAV energy unit. Hence, the environment state $s^{ER}(t) \in \mathcal{S}^{ER}$ for each leader UAV $m \in \mathcal{M}$ at time slot t can be written as $s^{ER}(t) = \boldsymbol{E}^{level}(t)$, where $\boldsymbol{E}^{level}(t) = \{E_1^{level}(t), E_2^{level}(t), ..., E_M^{level}(t)\}$ indicates the set of all leader UAVs' energy levels.

2) Action for each leader UAV in ERSG: Leader UAV $m \in \mathcal{M}$ selects an action $a_m^{ER}(t) \in \mathcal{A}_m^{ER}$ at time slot t, where $\mathcal{A}_m^{ER}$ denotes the action set of leader UAV m consisting of two actions, i.e., returning to the depot or not.

3) Reward of each leader UAV in ERSG: The immediate reward $r_m^{ER}(t) \in \mathcal{R}_m^{ER}$ of leader UAV $m \in \mathcal{M}$ at time slot t is given by:

$$r_m^{ER}(t) = \begin{cases} -10, & \text{if constraint (10) is violated,} \\ \varepsilon_m(t), & \text{otherwise.} \end{cases} \tag{14}$$

4) State Transition Probabilities of Leader UAVs in ERSG: The state transition probability from state s^{ER} to state $s^{ER'}$ by taking the joint action $\boldsymbol{a}^{ER}(t) = (a_1^{ER}(t), a_2^{ER}(t), ..., a_M^{ER}(t))$ can be written as $p_{s^{ER}, s^{ER'}}^{ER}(\boldsymbol{a}^{ER}(t)) = Pr(s^{ER}(t+1) = s^{ER'} | s^{ER}(t) = s^{ER}, \boldsymbol{a}^{ER}(t))$. Moreover, the descriptions of state transition probabilities of APSG $\mathbb{P}^{AP}$, TPSG $\mathbb{P}^{TP}$, DCSG $\mathbb{P}^{DC}$ and TDSG $\mathbb{P}^{TD}$ are similar to that in ERSG, and are omitted in this paper for conciseness.

MDP for each leader UAV in APSG: To produce an optimal schedule for application placement of all leader UAVs in APSG, the individual decision-making problem for each leader UAV $m \in \mathcal{M}$ can be modeled as an MDP $(\mathcal{S}^{AP}, \mathcal{A}_m^{AP}, \mathcal{R}_m^{AP}, \mathbb{P}^{AP})$.

1) Environment state for each leader UAV in APSG: The environment state $s^{AP}(t) \in \mathcal{S}^{AP}$ for each leader UAV $m \in \mathcal{M}$ consists of whether leader UAVs returning to the depot and all leader UAVs' applications placement at time slot t, which can be expressed as $s^{AP}(t) = \{\boldsymbol{\varepsilon}(t), \boldsymbol{\omega}^L(t)\}$.

2) Action for each Leader UAV in APSG: Leader UAV m selects an action $a_m^{AP}(t) \in \mathcal{A}_m^{AP}$, where $\mathcal{A}_m^{AP}$ signifies the action set of leader UAV m consisting of $C!/((C-S^L)*S^L!)$ actions.

3) Reward of each leader UAV in APSG: The immediate reward $r_m^{AP}(t) \in \mathcal{R}_m^{AP}$ of leader UAV $m \in \mathcal{M}$ at time slot t can be written as: $r_m^{AP}(t) = \sum_{\tau=1}^{t} \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{k \in \mathcal{Z}_n} \delta_{m,n}(\tau) \upsilon_{k,c}(\tau) \omega_{m,c}^L(\tau)$, where $r_m^{AP}(t)$ denotes the number of the tasks processed by leader UAV m before time slot t.

MDP for each leader UAV in TPSG: To find an optimal schedule for application placement of all leader UAVs in TPSG, the individual decision-making problem for each leader UAV $m \in \mathcal{M}$ can be modeled as an MDP $(\mathcal{S}^{TP}, \mathcal{A}_m^{TP}, \mathcal{R}_m^{TP}, \mathbb{P}^{TP})$.

1) Environment state for each leader UAV in TPSG: The environment state $s^{TP}(t) \in \mathcal{S}^{TP}$ for each leader UAV $m \in \mathcal{M}$ consists of all leader UAVs' positions $\mathcal{L}(t)$ and UAV association set $\boldsymbol{\delta}(t)$ at time slot t, which can be expressed as $s^{TP}(t) = \{\boldsymbol{\mathcal{L}}(t), \boldsymbol{\delta}(t)\}$.

2) Action for each leader UAV in TPSG: Leader UAV $m \in \mathcal{M}$ selects an action $a_m^{TP}(t) \in \mathcal{A}_m^{TP}$ at time slot t, where $\mathcal{A}_m^{TP} = \{\text{forward, backward, left, right}\}$ indicates the action set of leader UAV m moving to an adjacent large grid in one direction.

3) Reward of each leader UAV in TPSG: The immediate reward $r_m^{TP}(t) \in \mathcal{R}_m^{TP}$ of leader UAV $m \in \mathcal{M}$ at time slot t can be written as:

$$r_m^{TP}(t) = \begin{cases} -10, & \text{if constraint (13e) or (13f) is violated,} \\ E^{effi}(t), & \text{otherwise.} \end{cases} \tag{15}$$

MDP for each follower UAV in DCSG: To produce an optimal schedule for application placement of all leader UAVs in DCSG, the individual decision-making problem for each leader UAV $m \in \mathcal{M}$ can be modeled as an MDP $(\mathcal{S}^{DC}, \mathcal{A}_n^{DC}, \mathcal{R}_n^{DC}, \mathbb{P}^{DC})$.

1) Environment state for each leader UAV in DCSG: The environment state $s^{DC}(t) \in \mathcal{S}^{DC}$ for each follower UAV $n \in \mathcal{N}$ consists of all leader UAVs' positions $\mathcal{L}(t)$ and UAV association set $\boldsymbol{\delta}(t)$ at time slot t, which can be expressed as $s^{DC}(t) = \{\boldsymbol{\mathcal{L}}(t), \boldsymbol{\delta}(t)\}$.

2) Action for each leader UAV in DCSG: At time slot t, follower UAV $n \in \mathcal{N}$ selects an action $a_n^{DC}(t) \in \mathcal{A}_n^{DC}$, where $\mathcal{A}_n^{DC}$ signifies the action set of follower UAV n consisting of M.

3) Reward of each leader UAV in DCSG: The immediate reward $r_n^{DC}(t) \in \mathcal{R}_n^{DC}$ of follower UAV $n \in \mathcal{N}$ in DCSG at time slot t can be written as:

$$r_n^{DC}(t) = \begin{cases} -10, & \text{if constraint (13d) is violated,} \\ E^{effi}(t), & \text{otherwise.} \end{cases} \tag{16}$$

MDP for each follower UAV in TDSG: To find an optimal schedule for application placement of all leader UAVs in TDSG, the individual decision-making problem for each leader UAV $m \in \mathcal{M}$ can be modeled as an MDP $(\mathcal{S}^{TD}, \mathcal{A}_n^{TD}, \mathcal{R}_n^{TD}, \mathbb{P}^{TD})$.

1) Environment state for each follower UAV in TDSG: The environment state $s^{TD}(t) \in \mathcal{S}^{TD}$ for each follower UAV $n \in \mathcal{N}$ consists of leader UAV m's application placement $\boldsymbol{\omega}_m^L(t)$, follower UAV n's application placement $\boldsymbol{\omega}^F(t)$ and UAV association set $\boldsymbol{\delta}(t)$ at time slot t, which can be written as $s^{TD}(t) = \{\boldsymbol{\omega}_m^L(t), \boldsymbol{\omega}^F(t), \boldsymbol{\delta}(t)\}$.

2) Action for each follower UAV in TDSG: At time slot t, follower UAV $n \in \mathcal{N}$ selects an action $a_n^{TD}(t) \in \mathcal{A}_n^{TD}$, where $\mathcal{A}_n^{TD}(t)$ indicates the action set of follower UAV n consisting of two feasible actions, i.e., whether delegating its tasks to the leader UAV or not.

3) Reward of each follower UAV in TDSG: The immediate reward $r_n^{TD}(t) \in \mathcal{R}_n^{TD}$ of follower UAV $n \in \mathcal{N}$ in TDSG at time slot t can be written as: $r_n^{TD}(t) = \sum_{m=1}^{M} \sum_{c=1}^{C} ((D_n^F(t) + \delta_{m,n}(t) D_m^L(t))/(p_n^F T_{m,n,c}^{dele}(t) + \xi(f_n^F)^2 D_n^F(t) + \delta_{m,n}(t) \xi(f_m^L)^2 D_m^L(t)))$, where the numerator denotes the size of tasks processed by follower UAV n and leader UAV m in a UAV swarm, and the denominator represents the energy consumption of task delegation and task processing.

Based on the formulation of these MDPs, we propose a novel RLDC algorithm, where Q learning is utilized to obtain

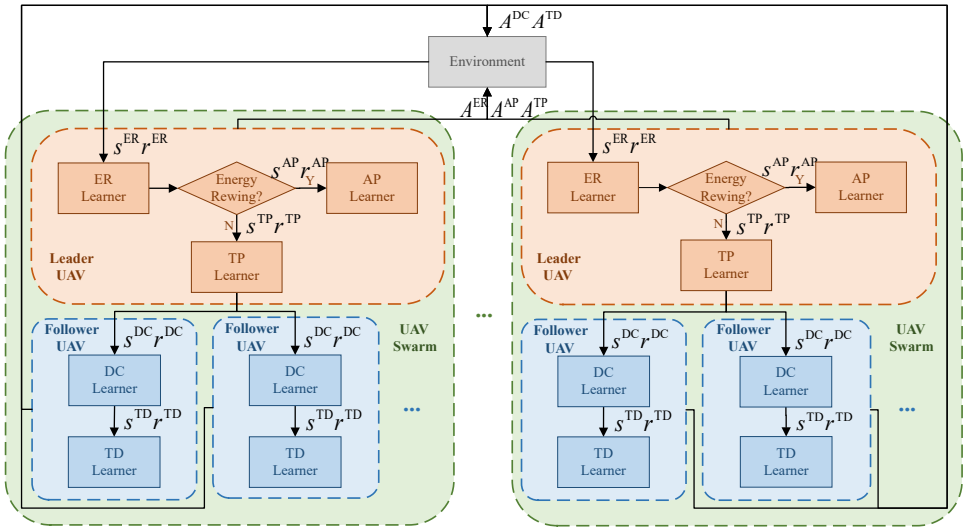

Fig. 3. An illustration of the RLDC algorithm.

the solution. In the proposed RLDC algorithm, each UAV initially conducts uniform exchanges of its historical Q-values with other UAVs. Subsequently, each UAV makes its current decision and updates the reward based on its current Q-value and the historical Q-values of other UAVs. Finally, each UAV updates its Q-value for the subsequent exchange with other UAVs based on its own decision, reward, and the historical Q-values of others. For inter-UAV information interaction, a central controller and a dedicated channel are enabled to be responsible for managing UAV information, including the reception and distribution of Q-values. Notably, to maintain the timeliness of decision-making, the dedicated channel refrains from participating in UAV decision-making, focusing solely on information exchange. Therefore, each UAV's decision-making remains decentralized, with each UAV considering only its own strategy optimization. Given that the data size of the information exchanged via the dedicated channel is considerably small (only Q values), while the transmission rate of the dedicated channel between different UAVs can be relatively large (commonly over 70 Mbps [29]), the delay of the information exchange in such a system is negligible, especially compared to that of the UAV decision-making process. Similar settings have been widely employed in the literature on game-theoretic analysis and optimization for UAV systems [21] and [30].

To be more specific, we may adopt distributed channel access (DCA) mechanisms to the dedicated channel, where multiple UAVs share a common channel for exchanging information [31]. Since most traditional DCA mechanisms are based on random access schemes, we may consider the most popular random access scheme, namely carrier-sense multiple access with collision avoidance (CSMA/CA) [32], which abides by listen-before-talk (LBT) protocol for regulating UAV to continually sense the channel before initiating a transmission. Again note that, since this is a dedicated channel for information exchange, the traffic load will be considerably low. Similar settings are given in [29] and [31].

Furthermore, each leader UAV includes an energy replenishment learner (ER learner), an application placement learner (AP learner) and a trajectory planning learner (TP learner). Meanwhile, each follower UAV includes and a dynamic clustering learner (DC learner) and a task delegation learner (TD learner). At the beginning of each time slot, each UAV first shares its Q-values of ERSG, APSG and TPSG to other UAVs. Subsequently, each leader UAV takes an action in ER learner. If the leader UAV flies to the depot for energy replenishment, it will take an action in AP learner. Otherwise, it will take an action in TP learner. Afterwards, each follower UAV takes an action in DC learner. Then, each leader UAV takes an action in TD learner. Finally, each UAV updates its Q-values based on its rewards and other UAVs' shared Q values.

Settings for ER learner: The policy of ER learner in leader UAV m is expressed as $\pi_m^{ER} : \mathcal{S}^{ER} \longrightarrow \mathcal{A}_m^{ER}$, which signifies a probability distribution of actions $a_m^{ER} \in \mathcal{A}_m^{ER}$ in a given state s^{ER}. Specifically, for leader UAV m in state $s^{ER} \in \mathcal{S}^{ER}$, the energy replenishment policy can be denoted as $\pi_m^{ER}(s^{ER}) = \{\pi_m^{ER}(s^{ER}, a_m^{ER}) | a_m^{ER} \in \mathcal{A}_m^{ER}\}$, where $\pi_m^{ER}(s^{ER}, a_m^{ER})$ is the probability of leader UAV m choosing action a_m^{ER} in state s^{ER}.

The Q function of the ER learner for leader UAV m is the expected reward resulting from executing action $a_m^{ER} \in \mathcal{A}_m^{ER}$ in state $s^{ER} \in \mathcal{S}^{ER}$ under the given policy π_m^{ER}, which can be expressed by $Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}) = \mathbb{E}(\sum_{\tau=0}^{\infty} \sigma^\tau r_m^{ER}(t+\tau+1) | s^{ER}(t) = s^{ER}, \boldsymbol{a}(t)^{ER} = \boldsymbol{a}^{ER}, \pi_m^{ER})$, where the constant discounted factor σ is defined with a value range of $[0, 1]$. This equation yields the action value, commonly referred to as the Q value. It takes into account the aggregation of immediate rewards in the current time slot to ascertain the long-term reward.

For striking a balance between exploration and exploitation, in this paper, an exploration strategy based on ϵ-greedy is taken into account for the ER learner. Specifically, the ER learner for leader UAV m selects a random action $a_m^{ER} \in \mathcal{A}_m^{ER}$ in state $s^{ER} \in \mathcal{S}^{ER}$ with probability ϵ, and chooses the optimal action a_m^{ER*} with probability $(1-\epsilon)$, where the best action has $Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER*}, \pi_m^{ER}) \geq Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}), \forall \boldsymbol{a}^{ER} \in$

$\mathcal{A}^{ER}$ with a_m^{ER*} being the m-th element of $\boldsymbol{a}^{ER*}$. Then, the probability of selecting action $a_m^{ER} \in \mathcal{A}_m^{ER}$ in state s^{ER} can be expressed by:

$$\pi_m^{ER}(s^{ER}, a_m^{ER}) = \begin{cases} 1-\epsilon, & \text{if } Q_m^{ER}(s^{ER},\cdot,\cdot) \text{ of } a_m^{ER} \text{ is the highest,} \\ \epsilon, & \text{otherwise.} \end{cases} \tag{17}$$

In the Q value update step of Q-learning, the ER learner for UAV m follows the update rule $Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t+1) = Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t) + \eta \sum_{m' \in M \backslash m}(Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t) - Q_{m'}^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t)) + \nu(r_m^{ER}(t) + \max_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} \sigma Q_m^{ER}(s^{ER'}, \boldsymbol{a}^{ER'}, t) - Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t))$, where η and ν denote the learning rates. $s^{ER'}$ and $\boldsymbol{a}^{ER'}$ denote the next environment state and the next joint action, respectively.

Since the settings of other learners are similar to those of the ER learner, they are omitted here for conciseness. In summary, Algorithm 1 provides a detailed illustration of the RLDC algorithm. To better understand the implementation of UAV swarm dynamic clustering and scheduling in the RLDC algorithm, we can take a look at an example: The RLDC algorithm obtains the optimal strategy through several iterations, where each iteration involves solving the long-term optimization problem of UAV swarm dynamic clustering and scheduling. At each time slot, first, leader UAV $m \in \mathcal{M}$ shares Q values Q_m^{ER}, Q_m^{TP} and Q_m^{AP} with other leader UAVs, which means that the all leader UAVs simultaneously learn to update the Q value from various state-action pairs [21]. Meanwhile, follower UAV $n \in \mathcal{N}$ shares Q values Q_n^{DC} and Q_n^{TD} with other follower UAVs. Second, leader UAV m selects an action a_m^{ER} according to policy $\pi_m^{ER}(s_m^{ER}|\cdot)$ and then obtains reward R_m^{ER}. If leader UAV m chooses to fly back to the depot for energy replenishment, it will select an action a_m^{AP} according to policy $\pi_m^{AP}(s_m^{AP}|\cdot)$ for updating its applications and then obtains reward R_m^{AP}. Otherwise, leader UAV m will select an action a_m^{TP} according to policy $\pi_m^{TP}(s_m^{TP}|\cdot)$ for leading its swarm to another adjacent large grid and then obtains reward R_m^{TP}. Third, follower UAV n selects an action a_n^{DC} according to policy $\pi_n^{DC}(s_n^{DC}|\cdot)$ for choosing a leader UAV to follow and then obtains reward R_n^{DC}. Meanwhile, follower UAV n selects an action a_n^{TD} according to policy $\pi_n^{TD}(s_n^{TD}|\cdot)$ for whether delegating tasks to its leader UAV and then obtains reward R_m^{ER}. Forth, the energy efficiency of all UAVs E^{effi} and Q values Q_m^{ER}, Q_m^{TP}, Q_m^{AP}, Q_n^{DC} and Q_n^{TD} are updated. Finally, the energy efficiency of all UAVs E^{effi} is added and the average energy efficiency of all UAVs is calculated at each iteration.

The Convergence of the RLDC Algorithm: As recognized in [33], [34], when the limits of the Q value $\lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}, t)$, $\lim_{t\to\infty} Q_m^{AP}(s^{AP}, \boldsymbol{a}^{AP}, \pi_m^{AP}, t)$, $\lim_{t\to\infty} Q_m^{TP}(s^{TP}, \boldsymbol{a}^{TP}, \pi_m^{TP}, t)$, $\lim_{t\to\infty} Q_m^{DC}(s^{DC}, \boldsymbol{a}^{DC}, \pi_m^{DC}, t)$ and $\lim_{t\to\infty} Q_m^{TD}(s^{TD}, \boldsymbol{a}^{TD}, \pi_m^{TD}, t)$ converge to the optimal Q value $Q^{ER^*}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER})$, $Q^{AP^*}(s^{AP}, \boldsymbol{a}^{AP}, \pi_m^{AP})$, $Q^{TP^*}(s^{TP}, \boldsymbol{a}^{TP}, \pi_m^{TP})$, $Q^{DC^*}(s^{DC}, \boldsymbol{a}^{DC}, \pi_m^{DC})$ and $Q^{TD^*}(s^{TD}, \boldsymbol{a}^{TD}, \pi_m^{TD})$ respectively, the RLDC approach is converged.

Algorithm 1: RLDC Algorithm

```
Initialize Q value: Q_m^ER = Q_m^AP = Q_m^TP = Q_n^DC = Q_n^TD = 0, ∀m ∈ M, n ∈ N.
Set the maximal iteration counter LOOP, loop = 0 and sum = 0.
for loop < LOOP do
    Set t = 0.
    while t ≤ T do
        for m = 1 to M do
            Share Q values Q_m^ER, Q_m^TP and Q_m^AP with leader UAV m' ∈ M\m.
            Observe states s^ER(t), s^AP(t) and s^TP(t).
            Select a_m^ER(t) according to π_m^ER(s^ER, ·).
            if ε_m(t) = 1 then
                Select a_m^AP(t) according to π_m^AP(s^AP, ·).
            else
                Select a_m^TP(t) according to π_m^TP(s^TP, ·).
        for n = 1 to N do
            Share Q values Q_n^DC and Q_n^TD with follower UAV n' ∈ N\n.
            Observe states s^DC(t) and s^TD(t).
            Select a_n^DC(t) according to π_n^DC(s^DC, ·).
            Select a_n^TD(t) according to π_n^TD(s^TD, ·).
        Obtain the E^effi(t) and the rewards R_m^ER(t), R_m^AP(t), R_m^TP(t), R_n^DC(t) and R_n^TD(t).
        Update the Q values Q_m^ER(t), Q_m^AP(t), Q_m^TP(t), Q_n^DC(t) and Q_n^TD(t).
        Set t = t + 1.
    Set sum = sum + Σ_{t=1}^{T} E^effi(t)
    Set loop = loop + 1.
Output: sum/loop
```

Lemma 2: A random iterative process $\Delta^{t+1}(x) = (1-\nu^t(x))\Delta^t(x) + \eta^t(x)\Phi^t(x)$ converges to zeros with a probability of 1 under the following conditions:

1) The state space is finite.

2) $\sum_t \nu^t(x) = \sum_t \eta^t(x) = \sum_t(\nu^t(x))^2 = \sum_t(\eta^t(x))^2 = \infty$ and $E\{\eta^t(x)|\Lambda^t\} \leq E\{\nu^t(x)|\Lambda^t\}$.

3) $||E\{\Phi^t(x)|\Lambda^t\}||_w \leq \chi||\Delta^t||_w$, where $\chi \in (0,1)$.

4) $Var\{\Phi^t(x)|\Lambda^t\} \leq \Lambda(1+||\triangle^t||_W)^2$, where Λ is a constant.

Proof: Please refer to Appendix B. ■

Theorem 1: In ERSG, we can obtain: $\mathcal{P}(lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}, t) = Q_m^{ER^*}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER*})) = 1, \forall m \in \mathcal{M}, s^{ER} \in \mathcal{S}^{ER}, \boldsymbol{a}^{ER} \in A^{ER}$.

Proof: Please refer to Appendix C. ■

The theorem and proof of Q_m^{AP}, Q_m^{TP}, Q_n^{DC} and Q_n^{TD} is analogous to that of Q_m^{ER}, and thus their detailed procedures are omitted here for conciseness.

Through the theoretical analysis, we show that the optimal NE exists in the proposed RLDC algorithm, and the NE point can be obtained by updating Q-value (i.e. updating Q matrix one by one). Specifically, in Theorem 1, we first prove $\mathcal{P}(lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}, t) = \overline{Q}_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER})) = 1$. Then, according to the proof of Lemma 2, we have $\mathcal{P}(lim_{t\to\infty} \overline{Q}_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}, t) = Q_m^{ER^*}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER*})) = 1$. Hence, we obtain $\mathcal{P}(lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER}, t) = Q_m^{ER^*}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER*})) = 1$, which indicates that leader UAV m can obtain the optimal policy π_m^{ER*} in ERSG. Similarly, according to Lemma 1, all leader UAVs can obtain their optimal policies $\{\pi_1^{ER*}, \pi_2^{ER*}, ..., \pi_M^{ER*}\}$, which indicates that the system can reach NE point in ERSG. Given the analogy in theorem and proof between Q_m^{AP}, Q_m^{TP}, Q_n^{DC},

Q_n^{TD} and Q_m^{ER}, it follows that APSG, TPSG, DCSG, and TDSG can each reach their respective NE by the RLDC algorithm. Consequently, the NE of interconnected multi-agent stochastic games encompasses those of ERSG, APSG, TPSG, DCSG, and TDSG. Thus, the system can reach this NE point in interconnected multi-agent stochastic games.

The Complexity of the RLDC Algorithm: Hereafter, we analyze the time complexity and space complexity of the proposed RLDC algorithm, which plays a critical role in UAV swarm assisted MEC system. The time complexity of the proposed RLDC algorithm is dependent on three factors: the maximal iteration counter $LOOP$, the number of time slots T, and the number of follower UAVs N. Thus, the time complexity can be written as $O(LOOP * T * N)$. The space complexity of the proposed RLDC algorithm is determined by the size of information exchanged among all UAVs during dynamic clustering and scheduling. Specifically, the size of this information is determined by the dimension of the state space in various learners. As mentioned before, the dimension of the state space in the ER learner is influenced by the energy levels. The dimension of the state space in the AP learner depends on the application types. Furthermore, the dimension of the state space in the TP learner is affected by the number of possible positions. Similarly, the dimension of the state space in the DC learner is influenced by the number of possible positions. Finally, the dimension of the state space in the TD learner is influenced by the number of possible positions and application types. Therefore, for a given UAV swarm assisted MEC system, the space complexity of the proposed RLDC algorithm remains constant even with increasing number of IoT devices over the target region, which indicates that the proposed RLDC algorithm demonstrates scalability.

VI. SIMULATION RESULTS

In this section, simulations are conducted to evaluate the performance of the proposed RLDC algorithm. We consider a $1000m \times 1000m$ square target region, which includes 3 leader UAVs and 9 follower UAVs. Meanwhile, 500 IoT devices are randomly located in the target region, and their positions are not time-varying. Additionally, the leader UAVs' altitude is $150m$, while the follower UAVs' altitude is $120m$. Table IV lists the values of main simulation parameters. Table V lists the RLDC algorithm settings. Similar settings have also been utilized in previous work such as [21], [35]–[37]. It is worth noting that specific parameters have the potential to vary based on various evaluation scenarios. For the purpose of comparison, we introduce two benchmark algorithms, namely, a fixed UAV swarm algorithm and a no UAV swarm algorithm.

- Fixed UAV swarm algorithm is devised to maximize the energy efficiency of all UAVs without considering dynamic clustering based on the RLDC algorithm, where each leader UAV contains a TP learner, an ER learner and an AP learner, and each follower UAV contains a TD learner.
- No UAV swarm algorithm is devised to maximize the energy efficiency of all UAVs without considering UAV swarm and leader UAV based on the RLDC algorithm, where each UAV is equipped with a TP learner, an ER learner and an AP learner.

TABLE IV
SIMULATION PARAMETERS

Parameter	Value
Carrier frequency f^c	$3\,GHz$
Effective Capacitance Coefficient ξ	10^{-18}
Number of task types C	10
Time slot length t	$30\,s$
Length of the small grid q	$50\,m$
Storage capacity of each leader UAV S^L	6
Storage capacity of each follower UAV S^F	4
Transmission power of follower UAV p^F	$0.2\,W$
Bandwidth B	$10\,MHz$
Power spectral density of noise ϖ	$-174dBm/Hz$
UAV velocity v	$20\,m/s$
Length of the large grid q	$150\,m$
Computing capacity of leader UAV f^L	$4\,Mbps$
Computing capacity of follower UAV f^F	$2\,Mbps$
Transmission power of leader UAV p^L	$2\,W$
The maximum number of follower UAVs	9
Task size of follower UAVs $\kappa_{k,c}$ in a UAV swarm N^{clus}	$10\,Mbits$

TABLE V
RLDC ALGORITHM SETTINGS

Leader UAV		
ER learner (state space size = $\lceil E^{total}/E^{unit} \rceil^M$, action space size = 2)		
AP learner (state space size = 2^{2M+C}, action space size = $C!/((C-S^L)*S^L!)$)		
TP learner (state space size = $((X^{up}-X^{low})/l)^{2M} * ((Y^{up}-Y^{low})/l)^{2M} * 2^N$), action space size = 4)		
Follower UAV		
DC learner (state space size = $((X^{up}-X^{low})/l)^{2M} * ((Y^{up}-Y^{low})/l)^{2M} * 2^N$), action space size = M)		
TD learner (state space size = 2^{2N+2C}, action space size = 2)		
Discounted Factor σ	**Learning Rate η/ν**	**Probability ϵ**
0.9	0.1/0.1	0.1

Fig. 4 illustrates the variation in energy consumption of leader UAVs over time slots, indicating the impact of dynamic clustering on leader UAVs' energy consumption. Since leader UAVs need to expend energy to charge their follower UAVs, an increase in the number of follower UAVs following the leader UAV results in higher energy consumption for the leader UAV. Then, the follower UAV will re-select the leader UAV based on its own position, the distance between itself and the leader UAVs, and other factors. Consequently, the energy consumption of the leader UAV continues fluctuating. It is obvious that the energy consumption of leader UAV 1 decreases rapidly and the energy consumption of leader UAV 2 increases rapidly at time slot 5. This can be attributed to the dynamic clustering, where a follower UAV changes its leader UAV from leader UAV 1 to leader UAV 2, and thereby, the leader UAV 2 has to consume more energy to charge the follower UAV. The explanations of the energy consumption of leader UAV 2 and 3 at time slot 15 and leader UAV 1 and 2 at time slot 28 are similar to those discussed above. Meanwhile, the energy consumption of leader UAV 1 and 3 at time slot 22 is double that in time slot 5, 15 and 28, indicating that two follower UAVs change their leader UAV from leader UAV 3 to leader UAV 1. All these outcomes are intended to demonstrate that the RLDC algorithm is capable of achieving dynamic clustering of UAV swarms.

Fig. 5 depicts all UAVs' energy efficiency as the length of time slot varies. It is evident that the energy efficiency of all

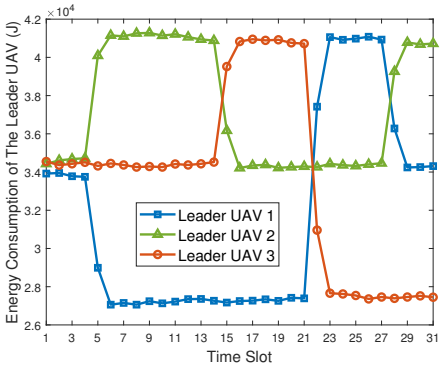

Fig. 4. Comparison of leader UAVs' energy consumption with varying time slots.

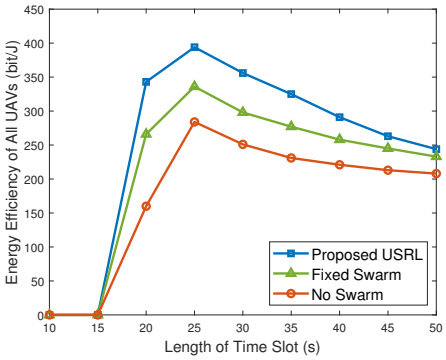

Fig. 5. Comparison of all UAVs' energy efficiency with varying lengths of time slot in fixed UAV swarm algorithm.

UAVs is zero due to the moving time exceeding the length of time slot. As the length of time slot increases, more tasks can be offloaded and processed by follower UAVs or leader UAVs from IoT devices, improving the energy efficiency. However, once all tasks have been processed, the UAVs remain in an idle state and consume energy while hovering over the target region until the time slot expires. Furthermore, the results demonstrate that the proposed RLDC algorithm outperforms both the fixed UAV swarm algorithm and no UAV swarm algorithm. This superiority arises from several reasons: i) in cases where the task requests from IoT devices are dynamically changing, the fixed UAV swarm cannot dynamically cluster according to the ever-changing task requests, and as a result, many tasks cannot be processed; ii) the storage capacity of each UAV is limited, and furthermore, UAVs are unable to delegate the tasks that cannot be processed to other UAVs without UAV swarm.

Fig. 6 examines all UAVs' energy efficiency as the number of IoT devices varies. Obviously, the energy efficiency of all UAVs exhibits a monotonically increasing trend with the increasing number of IoT devices. This can be attributed to the generation of more task requests by IoT devices as their quantity grows. Furthermore, the results demonstrate that the proposed RLDC algorithm surpasses both the fixed UAV swarm and no UAV swarm algorithms, which is consistent with the discussion in Fig. 5.

Fig. 7 illustrates all UAVs' energy efficiency as the UAV velocity varies. Obviously, the energy efficiency of all UAVs initially increases and then decreases with the UAV velocity increasing. Since as the velocity of UAVs increases, there is a reduction in the time taken for movement. As a result, UAVs have more time available for hovering and processing tasks. The decrease in energy efficiency of all UAVs can be attributed to two main factors: increased propulsion energy consumption as velocity increases and insufficient tasks to be processed. It is evident that the performance superiority of the proposed RLDC algorithm over the other two algorithms, which is consistent with the discussion in Fig. 5.

Fig. 8 demonstrates all UAVs' energy efficiency with the varying storage capacities of leader UAV. It can be observed that the performance with small grid size 50m outperforms that with 25m and 75m. The reason is that the length of small grid 25m accommodates fewer IoT devices, leading to decreased number of tasks processed by UAVs. In contrast, while the length of small grid 75m may accommodate a greater number of IoT devices, it also results in a substantial increase in the energy consumption of all UAVs as the moving distance of UAV swarm increases. Furthermore, the results also indicate that all UAVs' energy efficiency increases as the storage capacity of leader UAV grows, which can be attributed to the increased capability of processing various types of applications.

Fig. 9 shows the average task processing latency with the varying transmission power of follower UAVs. It can be seen that the average task processing latency increases as the length of small grid increases. This phenomenon can be attributed to the longer length of small grid leading to the longer average distance between leader UAVs and follower UAVs, which results in longer average task delegation time. Moreover, the average task processing latency consistently decreases with the increasing transmission power of follower UAVs, which can be attributed to higher transmission power leading to shorter task delegation time.

Fig. 10 depicts the average task processing latency with the varying computing capacities of leader UAV. It can be observed that the average task processing latency increases as the length of small grid increases. The explanation for this trend is similar to those presented in Fig. 9. Furthermore, the average task processing latency exhibits a monotonically decreasing trend with the computing capacity of leader UAV increasing. This trend can be attributed to the fact that as the computing capacity of leader UAV increases, task processing time of of leader UAV decreases, and the average task processing latency decreases accordingly.

VII. Conclusion

In this paper, with the aim of maximizing the long-term energy efficiency of the UAV swarm assisted MEC system, a joint optimization problem of UAV swarm dynamic clustering and scheduling is formulated. Considering the cooperation and competition among intelligent UAVs as well as the environment uncertainty, the optimization problem is reformulated as

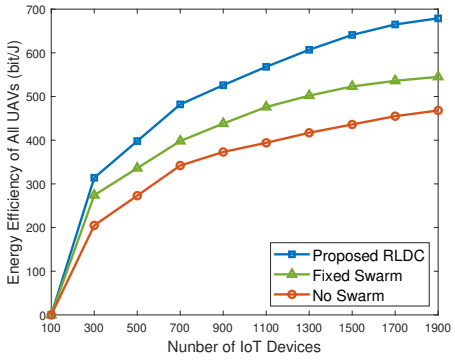

Fig. 6. Comparison of all UAVs' energy efficiency with varying numbers of IoT devices in fixed UAV swarm algorithm.

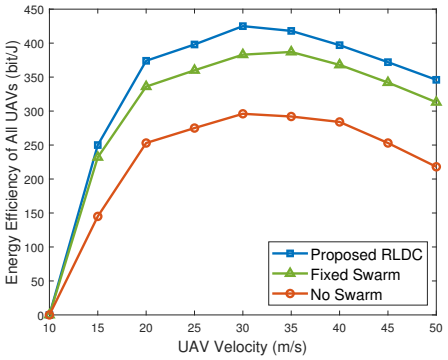

Fig. 7. Comparison of all UAVs' energy efficiency with varying UAV velocities in fixed UAV swarm algorithm.

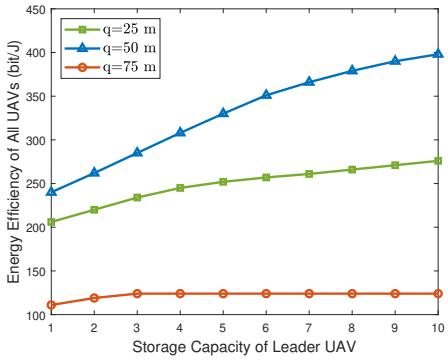

Fig. 8. Comparison of all UAVs' energy efficiency with varying storage capacities of leader UAV.

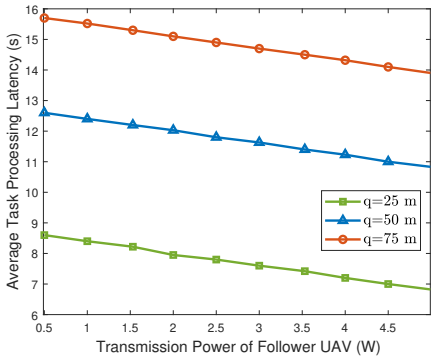

Fig. 9. Comparison of average task processing latency with different transmission power of follower UAV.

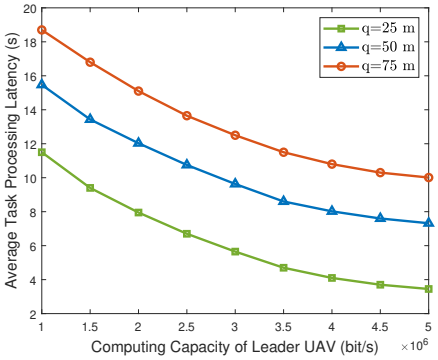

Fig. 10. Comparison of average task processing latency with different computing capacities.

a series of interconnected multi-agent stochastic games, and theoretically prove the existence of the corresponding NE. Furthermore, we propose a novel RLDC algorithm for obtaining such an equilibrium. Simulation results show that, compared to counterparts, the proposed RLDC algorithm significantly increases the energy efficiency of the UAV swarm assisted MEC system.

ACKNOWLEDGMENTS

This work was supported by the Major Program Project of Xiangjiang Laboratory under grant No. XJ2022001 and XJ2023001, IoT Intelligent Sensing Support Project for Science and Technology Innovation Teams in Hunan Province, Hunan Provincial Natural Science Foundation of China under grant No. 2023JJ40237, and the State Key Laboratory of Massive Personalized Customization System and Technology under grant No. H&C-MPC-2023-04-01, National Natural Science Foundation of China (NSFC) under grants No. 62176122, No. 62132007 and No. 62171218, and Postgraduate Research & Practice Innovation Program of NUAA under grant No. xcxjh20231601, and by Postgraduate Research & Practice Innovation Program of Jiangsu Province under grants No. KYCX22 0372.

APPENDIX

A. *Proof of Lemma 1*

According to [38], if any problem can be proved to be a multi-period stage game, it always exists the NE. Therefore, the key for proving the existence of NE is whether our proposed problem is a multi-period stage game. First of all, by the formulation of utility functions and corresponding strategies in Section III, our proposed problem can be defined as a stochastic game, which is a generalized form of repeated

game involving various state transition probabilities [39]. Furthermore, it is well known that any repeated game can be seen as a series of multiple stage games. Consequently, we can conclude that our proposed problem is a stochastic game consisting of a set of multiple stage games, each characterized by distinct stage transition probabilities, and is equivalent to the multi-period stage game [40]. In order to clearly express such NE in our specifically considered problem (as we just described), we introduce Definition 2 (i.e., the multi-UAV stage game expression) and Definition 3 (i.e., the NE expression) as follows:

Definition 2: A multi-UAV stage game can be defined as $(\widetilde{\Upsilon}_1, \widetilde{\Upsilon}_2, ..., \widetilde{\Upsilon}_M)$ [39], where $\widetilde{\Upsilon}_m$ denotes the reward of leader UAV $m \in \mathcal{M}$ over the joint action space [39]. Thus, $\widetilde{\Upsilon}_m$ is:

$$\widetilde{\Upsilon}_m = \{r_m(t)(a_1(t), a_2(t), ..., a_M(t))|a_m(t) \in \mathcal{A}_m\}. \quad (18)$$

Definition 3: Let ϑ_{-m}^{ER} represent the product of all leader UAVs' policies except the leader UAV $m \in \mathcal{M}$, $\vartheta_{-m}^{ER} \equiv \pi_1^{ER}, ..., \pi_{m-1}^{ER} \cdot \pi_{m+1}^{ER}, ..., \pi_M^{ER}$. Thus, in the multi-UAV stage game $(\widetilde{\Upsilon}_1^{ER}, \widetilde{\Upsilon}_2^{ER}, ..., \widetilde{\Upsilon}_M^{ER})$, the NE consists of a joint policy $\{\pi_1^{ER}, \pi_2^{ER}, ..., \pi_U^{ER}\}$, when the inequality is satisfied [39]:

$$\pi_m^{ER}\vartheta_{-m}^{ER}\widetilde{\Upsilon}_m^{ER} \geq \boldsymbol{\pi}_m^{ER}\vartheta_{-m}^{ER}\widetilde{\Upsilon}_m^{ER}, \forall m \in \mathcal{M}, \forall \pi_m \in \boldsymbol{\pi}_m. \quad (19)$$

In Definition 3, the NE consists of a set of policies $\{\pi_1^{ER}, \pi_2^{ER}, ..., \pi_U^{ER}\}$, when the policy π_m^{ER} of the leader UAV $m \in \mathcal{M}$ can maximize its utility function. Hence, we can infer that $\pi_m^{ER*}\vartheta_{-m}^{ER*}\widetilde{\Upsilon}_m^{ER} \geq \boldsymbol{\pi}_m^{ER}\vartheta_{-m}^{ER*}\widetilde{\Upsilon}_m^{ER}$, and $\{\pi_1^{ER*}, \pi_2^{ER*}, ..., \pi_M^{ER*}\}$ forms the NE, where π_m^{ER*} is the optimal policy of leader UAV $m \in \mathcal{M}$. In addition, it is worth noting that, in our paper, the utility function is defined as the expected discounted reward function. If an NE is reached, every leader UAV $m \in \mathcal{M}$ will choose to take the NE strategy and will not unilaterally deviate from the NE, thereby ensuring that the utility function $Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER})$ remains unaltered, and consequently, the reward remains unchanged [27].

B. Proof of Lemma 2

At time slot t, the iteration process of the RLDC algorithm for a given state-action pair $(s^{ER}, \boldsymbol{a}^{ER})$ can be represented by $\{Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t+1)\}$, which can be written as $\overline{Q}^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t) = \frac{1}{M}\sum_{m=1}^{M} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t), \forall t \geq 0$.

In this proof, the action and state within the bracket are omitted for conciseness, i.e., $Q_m^t = Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t)$, $\overline{Q}^t = \overline{Q}^{ER}(s^{ER}, \boldsymbol{a}^{ER}, t)$, $r_m^t = r_m(s^{ER}, \boldsymbol{a}^{ER}, s^{ER'}, t)$, and $Q_m^{t'} = Q_m^{ER}(s^{ER'}, \boldsymbol{a}^{ER'}, t)$. According to the equation above, we can obtain:

$$\overline{Q}^{t+1} = (1-\nu^t)\overline{Q}^t + \frac{\nu^t}{M}\sum\nolimits_{m=1}^{M}(\mathcal{R}_m^t + \sigma \max\nolimits_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} Q_m^{t'}). \quad (20)$$

By deducting Q^* from both sides of equation (20), we have:

$$\overline{Q}^{t+1} - Q^* = (1-\nu^t)(\overline{Q}^t - Q^*) + \nu^t(\tfrac{1}{M}\textstyle\sum_{m=1}^{M}(\mathcal{R}_m^t + \sigma \max\nolimits_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} Q_m^{t'}) - Q^*). \quad (21)$$

It is important to highlight that the temporal difference algorithm, as discussed in (21), can be viewed as a stochastic process outlined in Lemma 1 with $\triangle^{t+1} = \overline{Q}^t - Q^*$, $\Phi^t = \frac{1}{M}\sum_{m=1}^{M}(\mathcal{R}_m^t + \sigma \max_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} Q_m^{t'}) - Q^*$ and $\nu^t = \eta^t$. Consequently, the condition 1) and 2) in Lemma 1 are satisfied. To meet the requirements of the condition 3) and 4) in Lemma 1, we provide the proof of the temporal difference algorithm in equation (21).

Based on Proposition 5.1 in [34], operator $\mathcal{F}(\cdot)$ can be considered as a contraction mapping, and Q^* represents the sole fixed point of $\mathcal{F}(\cdot)$. The expression for $\mathcal{F}(\cdot)$ is:

$$\mathcal{F}(Q) = \sum_{s^{ER'} \in \mathcal{S}^{ER}} P_{s^{ER}s^{ER'}}^{ER}(\boldsymbol{a}^{ER})(\tfrac{1}{M}\sum_{m=1}^{M}\mathcal{R}_m^t(s^{ER}, \boldsymbol{a}^{ER}, s^{ER'}) + \sigma \max_{a^{ER'} \in \mathcal{A}^{ER}} Q(s^{ER'}, \boldsymbol{a}^{ER'})). \quad (22)$$

Thus, we have $\mathcal{F}(Q^*) = Q^*$ and $||\mathcal{F}(Q_1(s^{ER}, \boldsymbol{a}^{ER})) - \mathcal{F}(Q_2(s^{ER}, \boldsymbol{a}^{ER}))||_\infty = ||Q_1(s^{ER}, \boldsymbol{a}^{ER}) - Q_2(s^{ER}, \boldsymbol{a}^{ER})||_\infty$. This further gives $E\{\Phi^t\} = \sum_{s^{ER'} \in \mathcal{S}^{ER}} P_{s^{ER}s^{ER'}}^{ER}(\boldsymbol{a}^{ER})(\frac{1}{M}\sum_{m=1}^{M}\mathcal{R}_m^t + \sigma \max_{a^{ER'} \in \mathcal{A}^{ER}} \overline{Q}^{t'} - Q^*) = \mathcal{F}(\overline{Q}^t) - Q^*$. Then, we have $||E\{\Phi^t\}||_\infty = ||\mathcal{F}(\overline{Q}^t) - \mathcal{F}(Q^*)||_\infty \leq \sigma||\overline{Q}^t - Q^*||_\infty$ based on the properties of a contraction mapping, replacing $||\cdot||_\infty$ with $||\cdot||_W$ satisfies the condition 3) in Lemma 1. With respect to the condition 4) in Lemma 1, we can obtain $E\{\Phi^t\} = \sum_{s^{ER'} \in \mathcal{S}^{ER}} P_{s^{ER}s^{ER'}}^{ER}(\boldsymbol{a}^{ER})(\frac{1}{M}\sum_{m=1}^{M} r_m + \sigma \max_{a^{ER'} \in \mathcal{A}^{ER}} \overline{Q}^{t'} - Q^*) = \mathcal{F}(\overline{Q}^t) - Q^*$, by which $\boldsymbol{Var}\{\Phi^t\} \leq \Lambda(1 + ||\overline{Q}^t - Q^*||_W^2)$ can be rigorously proved for a given constant Λ owing to the fact that $\frac{1}{M}\sum_{m=1}^{M} r_m^t$ is bounded [41]. Therefore, the condition 4) in Lemma 2 is satisfied. The proof of Lemma 2 is completed, and thus we have $\mathcal{P}(lim_{t\to\infty}\overline{Q}(s^{ER}, \boldsymbol{a}^{ER}\pi_m^{ER}, t) = Q^{ER^*}(s^{ER}, \boldsymbol{a}^{ER}, \pi_m^{ER})) = 1$.

C. Proof of Theorem 1

Similar to Lemma 2, the action and state within the bracket are excluded in this proof.

Considering that the Q value of state-action pair $(s^{ER}, \boldsymbol{a}^{ER})$ is updated if and only if the joint action $\boldsymbol{a}^{ER}$ occurs at state s^{ER}, we represent the sequence of updating state-action pairs as $\{j\}, \forall j \geq 0$ in the ER learner. Hence, we have: $\boldsymbol{Q}^{j+1} = (\boldsymbol{W}_M - \eta^j\mathcal{L} - \nu^j\boldsymbol{W}_M)\boldsymbol{Q}^j + \nu^j(\boldsymbol{R}^j + \boldsymbol{V}^j)$, where $\boldsymbol{Q}^{j+1} = (Q_1^{j+1}, ..., Q_m^{j+1})^\top$, and $\boldsymbol{W}_M$ is the $M \times M$ identity matrix. Then, we can obtain $\boldsymbol{R}^j = (\mathcal{R}_1^j, ..., \mathcal{R}_M^j)^\top$ and $\boldsymbol{V}^j = (\sigma \max_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} Q_1^{j'}, ..., \sigma \max_{\boldsymbol{a}^{ER'} \in \mathcal{A}^{ER}} Q_M^{j'})^\top$. Furthermore, we can obtain: $\boldsymbol{Q}^{j+1} - \overline{\boldsymbol{Q}}^{j+1} = (\boldsymbol{W}_M - \eta^j\mathcal{L} - \nu^j\boldsymbol{Y}_M)(\boldsymbol{Q}^j - \overline{\boldsymbol{Q}}^j) + \nu^j(\hat{\boldsymbol{R}}^j + \hat{\boldsymbol{V}}^j)$, where the M-dimensional column vector of ones is denoted as $\mathbf{1}_M$. Then we have $\overline{\boldsymbol{Q}}^j = \overline{Q}^j\mathbf{1}_M$. Additionally, we can derive $\hat{\boldsymbol{R}}^j = (\boldsymbol{W}_M - (\frac{1}{M})\mathbf{1}_M(\mathbf{1}_M)^\top)\boldsymbol{R}^j$ and $\hat{\boldsymbol{V}}^j = (\boldsymbol{W}_M - (\frac{1}{M})\mathbf{1}_M(\mathbf{1}_M)^\top)\boldsymbol{V}^j$. Hence, we obtain:

$$\begin{aligned} &||\boldsymbol{Q}^{j+1} - \overline{\boldsymbol{Q}}^{j+1}|| = \\ &||(\boldsymbol{W}_M - \eta^j\mathcal{L} - \nu^j\boldsymbol{W}_M)\boldsymbol{Q}^j - \overline{\boldsymbol{Q}}^{j+1}|| + ||\nu^j(\boldsymbol{R}^j + \boldsymbol{V}^j)|| \\ &\overset{(\varrho)}{\leq} (1 - X_j + \nu^j)||\boldsymbol{Q}^j - \overline{\boldsymbol{Q}}^j|| + \nu^j(||\hat{\boldsymbol{R}}^j|| + ||\hat{\boldsymbol{V}}^j||), \end{aligned} \quad (23)$$

where the value of (ϱ) is determined according to Lemma 4.4 in [42] while $X_j \to 0$ as $j \to \infty$ with $X_j \in [0, 1]$. As

$\nu^j \to 0$ when $j \to \infty$, it follows that $(1 - X_j + \nu^j) \to 0$ as well. Consequently, we can conclude that $\mathcal{P}(lim_{t\to\infty}||\boldsymbol{Q}^j - \overline{\boldsymbol{Q}}^j|| = 0) = 1$. Namely, $\mathcal{P}(lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}) = \overline{Q}_m^{ER}(s^{ER}, \boldsymbol{a}^{ER})) = 1, \forall m \in \mathcal{M}, s^{ER} \in \mathcal{S}^{ER}, \boldsymbol{a}^{ER} \in \mathcal{A}^{ER}$.

Additionally, according to Lemma 2, we have $\mathcal{P}(lim_{t\to\infty} \overline{Q}_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}) = Q_m^{ER^*}(s^{ER}, \boldsymbol{a}^{ER})) = 1$. Hence, we can obtain $\mathcal{P}(lim_{t\to\infty} Q_m^{ER}(s^{ER}, \boldsymbol{a}^{ER}) = Q_m^{ER^*}(s^{ER}, \boldsymbol{a}^{ER})) = 1$, and this completes the proof of Theorem 1.

REFERENCES

[1] J. Li, J. Chen, C. Yi, T. Zhang, K. Zhu, and J. Cai, "Energy-efficient UAV swarm assisted MEC with dynamic clustering and scheduling," in *Proc. IEEE WCNC*, 2024, pp. 1–6.

[2] Y. Liao, X. Chen, S. Xia, Q. Ai, and Q. Liu, "Energy minimization for UAV swarm-enabled wireless inland ship MEC network with time windows," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 594–608, 2023.

[3] C. Yi, S. Huang *et al.*, "An incentive mechanism integrating joint power, channel and link management for social-aware D2D content sharing and proactive caching," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 789–802, 2018.

[4] J. Li, C. Yi, J. Chen, K. Zhu, and J. Cai, "Joint trajectory planning, application placement, and energy renewal for UAV-assisted MEC: A triple-learner-based approach," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13 622–13 636, 2023.

[5] W. He, H. Yao, T. Mai, F. Wang, and M. Guizani, "Three-stage stackelberg game enabled clustered federated learning in heterogeneous UAV swarms," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9366–9380, 2023.

[6] Y. Liao, X. Chen, S. Xia, Q. Ai, and Q. Liu, "Energy minimization for UAV swarm-enabled wireless inland ship MEC network with time windows," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 594–608, 2023.

[7] Y. Shi, C. Yi, R. Wang, Q. Wu, B. Chen, and J. Cai, "Service migration or task rerouting: A two-timescale online resource optimization for MEC," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 2, pp. 1503–1519, 2024.

[8] Y. Liu, J. Yan, and X. Zhao, "Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4225–4236, 2022.

[9] Y. Miao, K. Hwang, D. Wu, Y. Hao, and M. Chen, "Drone swarm path planning for mobile edge computing in industrial internet of things," *IEEE Trans. Ind. Inf.*, vol. 19, no. 5, pp. 6836–6848, 2023.

[10] K. Wang, X. Zhang, L. Duan, and J. Tie, "Multi-UAV cooperative trajectory for servicing dynamic demands and charging battery," *IEEE Trans. Mob. Comput.*, vol. 22, no. 3, pp. 1599–1614, 2023.

[11] T. Li, S. Leng, Z. Wang, K. Zhang, and L. Zhou, "Intelligent resource allocation schemes for UAV-swarm-based cooperative sensing," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 570–21 582, 2022.

[12] A. Mukherjee, S. Misra, V. S. P. Chandra, and M. S. Obaidat, "Resource-optimized multiarmed bandit-based offload path selection in edge UAV swarms," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4889–4896, 2019.

[13] Y. Shi, Y. Yang, C. Yi, B. Chen, and J. Cai, "Towards online reliability-enhanced microservice deployment with layer sharing in edge computing," *IEEE Internet Things J.*, pp. 1–1, 2024.

[14] Y. Wang, H. Guo, and J. Liu, "Cooperative task offloading in UAV swarm-based edge computing," in *Proc. IEEE GLOBECOM*, 2021.

[15] W. Huang, H. Guo, and J. Liu, "Task offloading in uav swarm-based edge computing: Grouping and role division," in *Proc. IEEE GLOBECOM*, 2021.

[16] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Serv. Manage.*, vol. 18, no. 4, pp. 4531–4547, 2021.

[17] G. Fragkos, N. Kemp, E. E. Tsiropoulou, and S. Papavassiliou, "Artificial intelligence empowered UAVs data offloading in mobile edge computing," in *IEEE ICC*, 2020, pp. 1–7.

[18] L. Liang, Y. Zhao, K. Jian, H. You, and X. Zhang, "Resource allocation strategy for multi-UAV-assisted MEC system with dense mobile users and MCR-WPT," in *Proc. IEEE WCNC*, 2023.

[19] Z. Mou, Y. Zhang, F. Gao, H. Wang, T. Zhang, and Z. Han, "Deep reinforcement learning based three-dimensional area coverage with UAV swarm," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3160–3176, 2021.

[20] Y. Yang, Y. Shi, C. Yi, J. Cai, J. Kang, D. Niyato, and X. Shen, "Dynamic human digital twin deployment at the edge for task execution: A two-timescale accuracy-aware online optimization," *IEEE Trans. Mobile Comput.*, pp. 1–16, 2024.

[21] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, and J. Li, "Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3193–3207, 2021.

[22] O. S. Oubbati, A. Lakas, and M. Guizani, "Multiagent deep reinforcement learning for wireless-powered UAV networks," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 044–16 059, 2022.

[23] H. Mei, K. Yang, Q. Liu, and K. Wang, "Joint trajectory-resource optimization in UAV-enabled edge-cloud system with virtualized mobile clone," *IEEE Internet of Things J.*, vol. 7, no. 7, pp. 5906–5921, 2020.

[24] H. Liu, X. Li *et al.*, "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3061–3073, 2022.

[25] H. Liu, G. Wu, L. Zhou, W. Pedrycz, and P. N. Suganthan, "Tangent-based path planning for uav in a 3-d low altitude urban environment," *IEEE Trans. Intell.Transp. Syst.*, vol. 24, no. 11, pp. 12 062–12 077, 2023.

[26] J. Chen, Y. Zhang, L. Wu, T. You, and X. Ning, "An adaptive clustering-based algorithm for automatic path planning of heterogeneous uavs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16 842–16 853, 2022.

[27] J. Zheng, Y. Cai, N. Lu, Y. Xu, and X. Shen, "Stochastic game-theoretic spectrum access in distributed and dynamic environment," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4807–4820, 2015.

[28] R. Chen, C. Yi *et al.*, "A three-party hierarchical game for physical layer security aware wireless communications with dynamic trilateral coalitions," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 5, pp. 4815–4829, 2024.

[29] A. Mahmood, T. Vu, S. Chatzinotas, and B. Ottersten, "Joint optimization of 3D placement and radio resource allocation for per-UAV sum rate maximization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 13 094–13 105, 2023.

[30] B. Shi, Z. Chen, and Z. Xu, "A deep reinforcement learning based approach for optimizing trajectory and frequency in energy constrained multi-uav assisted mec system," *IEEE Trans. Netw. Serv. Manage.*, 2024.

[31] Z. Guo, Z. Chen, P. Liu, J. Luo, X. Yang, and X. Sun, "Multi-agent reinforcement learning-based distributed channel access for next generation wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1587–1599, 2022.

[32] A. Colvin, "Csma with collision avoidance," *Comput. Commun.*, vol. 6, no. 5, pp. 227–235, 1983.

[33] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 703–710.

[34] S. Kar, J. M. F. Moura, and H. V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1848–1862, 2013.

[35] C. Yi and J. Cai, "Two-stage spectrum sharing with combinatorial auction and stackelberg game in recall-based cognitive radio networks," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3740–3752, 2014.

[36] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, 2022.

[37] Y. Shi, C. Yi *et al.*, "Joint online optimization of data sampling rate and preprocessing mode for edge–cloud collaboration-enabled industrial IoT," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 402–16 417, 2022.

[38] M. S. Munir, S. F. Abedin *et al.*, "Risk-aware energy scheduling for edge computing with microgrid: A multi-agent deep reinforcement learning approach," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3476–3497, 2021.

[39] J. Hu *et al.*, "Nash q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, p. 1039–1069, 2003.

[40] M. Makris and L. Renou, "Information design in multi-stage games," *Working Papers*, 2018.

[41] F. S. Melo, "Convergence of q-learning: A simple proof," *Inst. Syst. Robot.*, pp. 1–4, 2001.

[42] S. Kar, J. M. Moura, and H. V. Poor, "Distributed linear parameter estimation: Asymptotically efficient adaptive strategies," *SIAM J. Control Optim.*, vol. 51, no. 3, pp. 2200–2229, 2013.

Jialiuyuan Li is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include game theory, stochastic game, reinforcement learning, and their applications in various wireless networks, including edge computing, industrial IoT, and UAV systems.

Changyan Yi (S'16-M'18) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Manitoba, MB, Canada,in 2018. He is currently a Professor withthe College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. From September 2018 to August 2019, he was a Research Associate with the University of Manitoba, MB, Canada. His research interests include stochastic optimization, mechanism design, game theory, queueing scheduling and machine learning with applications in resource management and decision making for various networking systems and services.

Jiayuan Chen is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include machine learning (e.g., reinforcement learning) and mechanism design with applications in resource management and decision making for various wireless networks and mobile services, including edge/fog computing, industrial IoT, vehicular/UAV systems, and digital twin.

You Shi received the M.S. degree from the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China, in 2020. He is pursuing a Ph.D. at the College of Computer Science and Technology, Nanjing University of Aeronauticsand Astronautics (NUAA), Nanjing, China. His main research interests include mobile edge computing, online optimization, service deployment, resource management, Internet of Things, 5G and beyond.

Tong Zhang is currently an associate professor at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. In 2014, she received her B.E. degree in Computer Science and Technology from Xi'an Jiaotong University. In 2019, she received her Ph.D degree in Computer Science and Technology from Tsinghua University. Her research interests are primarily centered on flow scheduling in data center networks, task scheduling in real-time edge systems and traffic management in time-sensitive networking.

Xiaolong Li received his B.E. degree from the Harbin Institute of Technology in Harbin, China, in 2003, and his Ph.D. degree from Hunan University in Changsha, China, in 2008. Since January 2017, he has been a Professor at Hunan University of Technology and Business in Changsha. His research interests include deep learning, intelligent transportation systems, and the Internet of Things. Prof. Li won the Best Paper Award at the ChinaCom conference in 2013. He has served as a Technical Program Committee (TPC) member for IEEE GreenCom 2024 and IEEE VTC-Fall 2020.

Ran Wang (M'18) is an Associate professor and Doctoral Supervisor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received his B.E. in Electronic and Information Engineering from Honors School, Harbin Institute of Technology, China in July 2011 and Ph.D. in Computer Science and Engineering from Nanyang Technological University, Singapore in April 2016. He has authored or co-authored over 60 papers in top-tier journals and conferences. He received the Nanyang Engineering Doctoral Scholarship (NEDS) Award in Singapore and the innovative and entrepreneurial Ph.D. Award of Jiangsu Province, China in 2011 and 2017, respectively. He is the recipient of the Second Prize for Scientific and Technological Progress awarded by the China Institute of Communications and he is the ChangKong Scholar of NUAA. His current research interests include telecommunication networking and cloud computing.

Kun Zhu (Member, IEEE) received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2012. He was a Research Fellow with the Wireless Communications Networks and Services Research Group, University of Manitoba, Winnipeg, MB, Canada, from 2012 to 2015. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing. He is also a Jiangsu Specially Appointed Professor. He has published more than 50 technical papers. His research interests include resource allocation in 5G, wireless virtualization, and self-organizing networks. Dr. Zhu won several research awards, including the IEEE WCNC 2019 Best Paper Awards and the ACM China Rising Star Chapter Award. He has served as a TPC for several conferences.