A Triple Learner Based Energy Efficient Scheduling for Multi-UAV Assisted Mobile Edge Computing

Jiayuan Chen*, Changyan Yi*, Jialiuyuan Li*, Kun Zhu* and Jun Cai†
*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
†Department of Electrical and Computer Engineering, Concordia University, Montréal, QC, H3G 1M8, Canada
Email: {jiayuan.chen, changyan.yi, jialiuyuan.li, zhukun}@nuaa.edu.cn, jun.cai@concordia.ca

***Abstract*—In this paper, an energy efficient scheduling problem for multiple unmanned aerial vehicle (UAV) assisted mobile edge computing is studied. In the considered model, UAVs act as mobile edge servers to provide computing services to end-users with task offloading requests. Unlike existing works, we allow UAVs to determine not only their trajectories but also decisions of whether returning to the depot for replenishing energies and updating application placements (due to limited batteries and storage capacities). Aiming to maximize the long-term energy efficiency of all UAVs, i.e., total amount of offloaded tasks computed by all UAVs over their total energy consumption, a joint optimization of UAVs' trajectory planning, energy renewal and application placement is formulated. Taking into account the underlying cooperation and competition among intelligent UAVs, we reformulate such problem as three coupled multi-agent stochastic games, and then propose a novel triple learner based reinforcement learning approach, integrating a trajectory learner, an energy learner and an application learner, for reaching equilibriums. Simulations evaluate the performance of the proposed solution, and demonstrate its superiority over counterparts.**

I. Introduction

Recently, the multi-unmanned aerial vehicle (UAV) assisted mobile edge computing (MEC) has attracted a myriad of attentions due to its high-flexibility in providing MEC services for end-users (e.g., IoT devices). Particularly, UAVs with computing resources can dynamically adjust their positions to get close to end-users or fly to the areas that cannot be covered by fixed MEC infrastructures [1]. Thus, compared to the traditional MEC system, the multi-UAV assisted MEC can provide better quality of experience for end-users [2], [3].

Although the multi-UAV assisted MEC is envisioned as a light-weight but highly efficient paradigm for alleviating computation burdens on end-users, it also suffers several inherent restrictions. For instance, computing tasks offloaded from different end-users are required to be processed by specific service applications, while the limited storage capacities of UAVs impede their abilities to store all applications. Additionally, the limited energy capacities of UAVs also hinders the implementation of this paradigm in providing the long-term MEC services. Recent research efforts in this area include trajectory optimization [4], [5], service caching [6], UAV deployment [7], [8], etc. Nevertheless, there are still some critical issues, especially how UAVs' installed applications should be updated (with severely restricted wireless backhauls) and how UAVs' energy replenishment should be jointly scheduled, which are of great importance but have not yet been well investigated.

In this paper, we study a joint optimization of trajectory planning, energy renewal, and application placement for multi-UAV assisted MEC to maximize the long-term energy efficiency of all UAVs, i.e., the total amount of offloaded tasks computed by all UAVs over their total energy consumption, when providing MEC services. Specifically, in the considered system, each UAV working over a target region has to decide its actions after finishing the last one, i.e., a flight direction for serving IoT devices in other areas or returning back to the depot for replenishing its energy and simultaneously updating its application placement (through wired connections), with the aim of maximizing the long-term energy efficiency of all UAVs. Since UAVs are intelligent, we allow each of them to make its own decisions while regulate the underlying cooperation and competition among them. Additionally, we take into account the uncertainty that the future environment information (e.g., positions and task requirements of IoT devices) is unavailable to UAVs. To this end, we reformulate the joint optimization problem as three coupled multi-agent stochastic games, namely, trajectory planning stochastic game (TPSG), energy renewal stochastic game (ERSG) and application placement stochastic game (APSG), and then propose a novel triple learner based reinforcement learning (TLRL) approach to obtain corresponding equilibriums of these games.

The main contribution of this paper are in the following.

- A joint optimization of trajectory planning, energy renewal and application placement for multi-UAV assisted MEC is formulated, where the objective is to maximize the long-term energy efficiency of all UAVs.
- Observing the underlying cooperation and competition among UAVs, the optimization problem is reformulated as three coupled multi-agent stochastic games, i.e., TPSG, ERSG and APSG, and then a novel approach, called TLRL, is proposed to derive corresponding equilibriums.
- Extensive simulations are conducted to show the superiority of the proposed TLRL approach over counterparts.

The rest of this paper is organized as follows: Section II introduces the system model and problem formulation. In Section III, a problem reformulation based on multi-agent stochastic game is proposed and analyzed, along with the developed TLRL approach. Simulation results are provided in Section IV, followed by the conclusion in Section V.

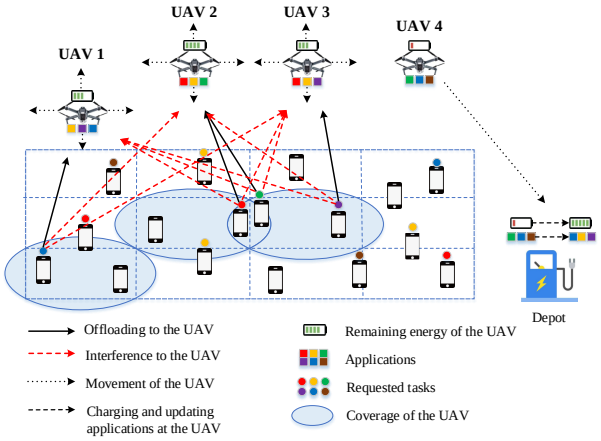

Fig. 1: An illustration of considered multi-UAV assisted MEC.

II. System Model and Problem Formulation

A. Network Model

Consider a multi-UAV assisted MEC system deployed in a target region, as illustrated in Fig. 1, consisting of a group of heterogeneous UAVs (acting as mobile edge servers) $\mathcal{M}$ with cardinality of $|\mathcal{M}| = M$ and a set of randomly scattered IoT devices $\mathcal{N}$ with cardinality of $|\mathcal{N}| = N$. There is a depot located at the edge of the target region, which can be used by UAVs for both energy replenishment and application placement through wired connections. A time-slotted operation framework is studied, in which we define $t \in \{1, 2, ..., T\}$ as the index of time slot. The target region is equally divided into small squared grids with the side length of q, and similar to [9], we assume that the downlink transmission range of each UAV is $\frac{\sqrt{2}}{2}q$, which totally covers a grid (for feeding back computation outcomes). All IoT devices are required to offload their tasks to their associated UAVs via uplink communications using the same frequency band B, and the set of IoT devices served by (or associated with) a certain UAV m is denoted by $\mathcal{G}_m \subseteq \mathcal{N}$. The horizontal coordinates of IoT device $n \in \mathcal{N}$ and UAV $m \in \mathcal{M}$ at time slot t are represented as $\mathcal{I}_n(t) = (x_n^I(t), y_n^I(t))$ and $\mathcal{U}_m(t) = (x_m^U(t), y_m^U(t))$, respectively. Then, the distance of IoT device $n \in \mathcal{N}$ and UAV $m \in \mathcal{M}$ at time slot t can be expressed as

$$d_{m,n}(t) = \sqrt{(x_m^U(t) - x_n^I(t))^2 + (y_m^U(t) - y_n^I(t))^2 + H^2},$$

where H denotes a fixed flight altitude of all UAVs. Following the literature [10], the line-of-sight (LoS) probability between IoT device $n \in \mathcal{G}_m$ and UAV $m \in \mathcal{M}$ at time slot t is given by $\delta_{m,n}(t) = a \cdot exp(-b(arctan(H/d_{m,n}(t)) - a))$, where a and b are constant values depending on the environment. Then, the path loss between IoT device $n \in \mathcal{G}_m$ and UAV $m \in \mathcal{M}$ at time slot t can be expressed as

$$\lambda_{m,n}(t) = 20log(\sqrt{H^2 + d_{m,n}(t)^2}) + \delta_{m,n}(t)(\eta_{LoS} - \eta_{NLoS}) + 20log[(4\pi f)/c] + \eta_{NLoS},$$

where f and c signify the carrier frequency and the speed of light, respectively; η_{LoS} and η_{LoS} are the losses corresponding to the LoS and non-LoS links, respectively.

Since a common frequency band is reused among all links, the signal-to-interference-plus-noise ratio (SINR) at UAV $m \in \mathcal{M}$ with regard to the uplink communication of IoT device $n \in \mathcal{G}_m$ at time slot t can be calculated as

$$\sigma_{m,n}(t) = \frac{\boldsymbol{v}_n(t)(\boldsymbol{w}_m(t)^\top)p_n^{tran}10^{\frac{-\lambda_{m,n}}{10}}}{\sum_{i=1\setminus\{n\}}^{N}\boldsymbol{v}_n(t)(\boldsymbol{w}_m(t)^\top)p_i^{tran}10^{\frac{-\lambda_{m,n}}{10}} + \varphi B},$$

where p_n^{tran} is the transmission power of IoT device n, and φ indicates the power spectral density of noise. At time slot t, we consider that IoT device $n \in \mathcal{G}_m$ can offload no more than one task to its associated UAV m. Let $\boldsymbol{v}_n(t) = \{v_{n,1}(t), v_{n,2}(t), ..., v_{n,c}(t), ..., v_{n,C}(t)\}$, where $c \in \{1, 2, ..., C\}$ is the index of the type of task, and $v_{n,c}(t) = 1$ signifies that IoT device n requests to offload task c, and $v_{n,c}(t) = 0$, otherwise. Meanwhile, the applications placed in UAV m can be defined as $\boldsymbol{w}_m(t) = \{w_{m,1}(t), w_{m,2}(t), ..., w_{m,c}(t), ..., w_{m,C}(t)\}$, where $w_{m,c}(t) \in \{0, 1\}$ signifies whether UAV m places the application type c. Note that, any UAV $m \in \mathcal{M}$ can only process the types of tasks fitting the types of its placed applications. Based on these, the transmission time of IoT devices $n \in \mathcal{G}_m$ in offloading a task to UAV $m \in \mathcal{M}$ can be written as

$$t_{m,n}^{off}(t) = \frac{\boldsymbol{v}_n(t)(\boldsymbol{w}_m(t)^\top)D_n}{Blog_2(1 + \sigma_{m,n}(t))},$$

where D_n is the size of task offloaded by IoT device n.

Within each time slot t, we consider that UAV $m \in \mathcal{M}$ hovers over the center of a certain grid to provide MEC services with time duration t^{hover}, and $t_{m,n}^{off}(t) < t^{hover} < |t|$, $\forall n \in \mathcal{G}_m, \forall m \in \mathcal{M}$, which means that t^{hover} is large enough for UAV m to receive any task offloaded by any IoT device and is shorter than the duration of a time slot. Then, the size of tasks computed by UAV $m \in \mathcal{M}$ can be expressed as

$$Task_m^{comp}(t) = \begin{array}{l}\min\{\sum_{n\in\mathcal{G}_m}\boldsymbol{v}_n(t)(\boldsymbol{w}_m(t)^\top)D_n, \\ (t^{hover} - \min\{t_{m,n}^{off}(t)_{n\in\mathcal{G}_m}\})f_m^U\},\end{array}$$

where f_m^U is the computing capacity of UAV m (in the number of CPU cycles per second), and $(t^{hover} - \min\{t_{m,n}^{off}(t)_{n\in\mathcal{G}_m}\})$ indicates that UAV m starts edge computing since the first task is totally received. Correspondingly, the energy consumption of UAV $m \in \mathcal{M}$ for computing tasks at slot t is calculated as

$$E_m^{comp}(t) = \xi(f_m^U)^2 Task_m^{comp}(t),$$

where ξ shows the capacitance coefficient of UAV $m \in \mathcal{M}$.

Furthermore, let $\kappa_m(t) \in \{0, 1\}$ stand for the decision that UAV $m \in \mathcal{M}$ chooses to whether return to the depot at the beginning of each time slot t. If UAV $m \in \mathcal{M}$ decides to not return to the depot (denoted by $\kappa_m(t) = 1$), it will select a direction among forward, backward, left and right, and then move to the center of another adjacent grid with a constant velocity V. The propulsion energy consumption (consisting of the energy consumption of horizontal moving and hovering) of the UAV m can be expressed as $E_m^{pro} = P_m^{pro}(V)\frac{q}{V} + P_m^{pro}(0)t^{hover}$, where P_m^{pro} is the propulsion power model of

UAVs, and its descriptions follows from [11] and are omitted here. If UAV $m \in \mathcal{M}$ decides to return to the depot (denoted by $\kappa_m(t) = 0$), the energy consumption of UAV m moving between the target region and the depot with the constant velocity V can be written as $E_m^{dep} = 2 \cdot P_m^{pro}(V)\frac{d_{m,dep}(t)}{V}$, where $d_{m,dep}(t)$ is the distance between UAV m and the depot at time slot t. At the depot, UAV m can quickly renew its energy and also update its application placement for better serving IoT devices. Note that, the total size of applications placed at UAV $m \in \mathcal{M}$ should be smaller than its storage capacity S_m, that is $\sum_{c=1}^{C} \mu_c w_{m,c}(t) \leq S_m$, where μ_c stands for the size of application type c. Additionally, to guarantee the quality of service (QoS) of IoT devices, each type of application should be placed in at least one UAV hovering over the target region at each time slot t, i.e., $\sum_{m=1}^{M} w_{m,c}(t)\kappa_m(t) \geq 1, \forall c \in C$. After replenishing energy and updating application placement, UAV m will back to the original region and continue to provide MEC services.

B. Problem Formulation

In this work, we aim to maximize the energy efficiency of all UAVs, i.e., total amount of offloaded tasks computed by all UAVs over their total energy consumption, and we have

$$E^{effi}(t) = \frac{\sum\limits_{m=1}^{M} \kappa_m(t) Task_m^{comp}(t)}{\sum\limits_{m=1}^{M} (\kappa_m(t)(E_m^{comp}(t)+E_m^{pro})+(1-\kappa_m(t))E_m^{dep})} \quad (1)$$

Then, the joint optimization of UAVs' trajectory planning, energy renewal and application placement is formulated as

$$[\mathcal{P}1]: \max_{\mathcal{U}_m(t),\boldsymbol{w}_m(t),\kappa_m(t)} \lim_{T\to+\infty} \frac{1}{T}\sum\nolimits_{t=1}^{T} E^{effi}(t) \quad (2)$$

$$s.t., \ \kappa_m(t) \in \{0,1\}, \forall m \in \mathcal{M}, \quad (3)$$

$$w_{m,c}(t) \in \{0,1\}, \forall m \in \mathcal{M}, \forall c \in C, \quad (4)$$

$$\sum\nolimits_{c=1}^{C} \mu_c w_{m,c}(t) \leq S_m, \forall m \in \mathcal{M}, \quad (5)$$

$$\sum\nolimits_{m=1}^{M} w_{m,c}(t)\kappa_m(t) \geq 1, \forall c \in C, \quad (6)$$

$$|\mathcal{U}_m(t) - \mathcal{U}_m(t-1)|^2 \kappa_m(t) = q^2, \forall m \in \mathcal{M}, \quad (7)$$

$$(x_m^U(t) - x_m^U(t-1))(y_m^U(t) - y_m^U(t-1))\kappa_m(t) = 0, \quad (8)$$

$$|\mathcal{U}_m(t) - \mathcal{U}_{m'}(t)|\kappa_m(t) \geq q, \forall m \neq m', \quad (9)$$

where constraint (5) means that the total size of applications placed at each UAV should be less than its storage capacity; constraint (6) states that QoS of serving IoT devices should be met; constraints (7) and (8) imply that each UAV can only move to the center of adjacent grid if it does not return to the depot; constraint (9) indicates that each grid can only be covered by one UAV to avoid potential collisions. In the following section, we will first analyze problem $[\mathcal{P}1]$, and then propose a novel approach to derive the solution.

III. Problem Reformulation and Solution

A. Problem Reformulation

Since UAVs are intelligent, to solve problem $[\mathcal{P}_1]$, we can allow each UAV to make its own decisions while regulating the underlying cooperation and competition among them. Specifically, UAVs are expected to cooperatively conduct the trajectory planning, energy renewal and application placement to maximize the energy efficiency of all UAVs while guaranteeing QoS of IoT devices. Meanwhile, allowing UAVs to make decisions themselves may also lead to competitions in trajectory planning, energy renewal and application placement among them. Additionally, considering the uncertainty that the future environment information (e.g., task requirements of IoT devices) is not available to UAVs, to this end, we reformulate $[\mathcal{P}1]$ as three coupled multi-agent stochastic games as follows.

$[\mathcal{P}1]$ is reformulated as three coupled multi-agent stochastic games, i.e., TPSG $\langle \mathcal{M}, \mathcal{S}^{TPSG}, \mathcal{A}^{TPSG}, \mathcal{P}^{TPSG}, \mathcal{R}^{TPSG} \rangle$, ERSG $\langle \mathcal{M}, \mathcal{S}^{ERSG}, \mathcal{A}^{ERSG}, \mathcal{P}^{ERSG}, \mathcal{R}^{ERSG} \rangle$ and APSG $\langle \mathcal{M}, \mathcal{S}^{APSG}, \mathcal{A}^{APSG}, \mathcal{P}^{APSG}, \mathcal{R}^{APSG} \rangle$, where $\mathcal{M}$ indicates the set of agents (i.e., UAVs in this paper), $\mathcal{S}$ stands for the environment states, $\mathcal{A}$ represents the set of joint actions of all agents, $\mathcal{P}$ signifies the set of state transition probabilities, and $\mathcal{R}$ is the set of reward functions. Particularly, for TPSG, each UAV $m \in \mathcal{M}$ will choose an action individually based on the current environment states $s^{TPSG}(t) \in \mathcal{S}^{TPSG}$ at each time slot t, and then form a joint action $\boldsymbol{a}^{TPSG}(t) \in \mathcal{A}^{TPSG}$. After executing the joint action, rewards will be obtained according to $\mathcal{R}^{TPSG}$, and the environment states will turn to be next ones following $\mathcal{P}^{TPSG}$. The descriptions of ERSG and APSG are similar to TPSG, and are omitted here. Note that, TPSG, ERSG and APSG are inherently coupled. In the following subsection, we propose a novel approach, called TLRL, to obtain equilibriums of these three coupled multi-agent stochastic games.

B. TLRL Approach

The transitions of states actions of TPSG, ERSG, and APSG satisfy the Markov property, because all joint actions, i.e., $\boldsymbol{a}^{TPSG}(t)$, $\boldsymbol{a}(t)^{ERSG}$ and $\boldsymbol{a}(t)^{APSG}$, at time slot t only depend on the environment states at time slot t, i.e., $\boldsymbol{s}^{TPSG}(t)$, $\boldsymbol{s}^{ERSG}(t)$ and $\boldsymbol{s}^{APSG}(t)$, and thereby, in this paper, we characterize each UAV's strategic decision process in TPSG, ERSG and APSG by three Markov decision processes (MDPs).

MDP for each UAV in TPSG: With the aim of finding the optimal trajectories for all UAVs, the individual decision making problem for each UAV $m \in \mathcal{M}$ in TPSG can be modelled as an MDP $(\mathcal{S}^{TPSG}, \mathcal{A}_m^{TPSG}, \mathcal{R}_m^{TPSG}, \mathcal{P}^{TPSG})$.

1) Environment State for Each UAV in TPSG: The environment state $s^{TPSG}(t) \in \mathcal{S}^{TPSG}$ for UAV $m \in \mathcal{M}$ in TPSG at time slot t consists of all UAVs' positions $\mathcal{U}_m(t)$, $m \in \mathcal{M}$ and application placement $\boldsymbol{w}_m(t)$, $m \in M$, which can be expressed as $s^{TPSG}(t) = (\mathcal{U}_m(t), \boldsymbol{w}_m(t))_{m\in\mathcal{M}}$.

2) Action for Each UAV in TPSG: At time slot t, UAV $m \in \mathcal{M}$ chooses an action $a_m^{TPSG}(t) \in \mathcal{A}_m^{TPSG}$, where $\mathcal{A}_m^{TPSG}$ is the set consisting of four possible actions, i.e., moving forward, backward, left or right.

3) Reward of Each UAV in TPSG: The immediate reward of UAV $m \in \mathcal{M}$ at time slot t is given by

$$\mathcal{R}_m^{TPSG}(t) = \frac{\kappa_m(t) Task_m^{comp}(t)}{E_m^{comp}(t) + E_m^{pro}}, \quad (10)$$

where the numerator indicates the size of tasks computed by UAV m at time slot t, and the denominator represents the energy consumption of UAV m at time slot t.

4) State Transition Probabilities of UAVs in TPSG: The state transition probability from state s^{TPSG} to $s^{TPSG'}$ by taking the joint action $\boldsymbol{a}^{TPSG}(t)$=($a_1^{TPSG}(t)$, $a_2^{TPSG}(t)$, ..., $a_M^{TPSG}(t)$) can be expressed as $\mathcal{P}^{TPSG}_{s^{TPSG},s^{TPSG'}}(\boldsymbol{a}^{TPSG}(t))$=$Pr(s^{TPSG}(t+1)$=$s^{TPSG'}|s^{TPSG}(t)=s^{TPSG},\boldsymbol{a}^{TPSG}(t))$.

MDP for each UAV in ERSG: With the aim of designing the optimal schedule of energy renewal for all UAVs, the individual decision making problem for each UAV $m \in \mathcal{M}$ in ERSG can be modelled as an MDP $(\mathcal{S}^{ERSG}, \mathcal{A}_m^{ERSG}, \mathcal{R}_m^{ERSG}, \mathcal{P}^{ERSG})$.

1) Environment State for Each UAV in ERSG: The environment state $s^{ERSG}(t) \in \mathcal{S}^{ERSG}$ for UAV $m \in \mathcal{M}$ in ERSG at time slot t consists of all UAVs' remaining energy $E_m^{remain}(t)$, $m \in \mathcal{M}$ and positions $\mathcal{U}_m(t)$, $m \in \mathcal{M}$, which can be expressed as $s^{ERSG}(t) = (E_m^{remain}(t), \mathcal{U}_m(t))_{m\in\mathcal{M}}$.

2) Action for Each UAV in ERSG: UAV $m \in \mathcal{M}$ chooses an action $a_m^{ERSG}(t) \in \mathcal{A}_m^{ERSG}$ at time slot t, where $\mathcal{A}_m^{ERSG}$ is the set consisting of two actions, i.e., deciding to return to the depot or not.

3) Reward of Each UAV in ERSG: The immediate reward of UAV $m \in \mathcal{M}$ at time slot t is given by

$$\mathcal{R}_m^{ERSG}(t) = \begin{cases} -10, & \text{if constraint (6) is violated,} \\ \kappa_m(t), & \text{otherwise.} \end{cases} \tag{11}$$

This reward function can prompt UAVs to hover over the target region providing MEC services without violating (6).

The definition of state transition probabilities of UAVs in ERSG $\mathcal{P}^{ERSG}$ is similar to that in TPSG and is omitted here.

MDP for each UAV in APSG: With the aim of producing the optimal policy for updating the application placement of all UAVs, the individual decision making problem for each UAV $m \in \mathcal{M}$ in APSG can be defined as an MDP $(\mathcal{S}^{APSG}, \mathcal{A}_m^{APSG}, \mathcal{R}_m^{APSG}, \mathcal{P}^{APSG})$.

1) Environment State for Each UAV in APSG: The environment state $s^{APSG}(t) \in \mathcal{S}^{APSG}$ for UAV $m \in \mathcal{M}$ at time slot t consists of applications placed in all UAVs $\boldsymbol{w}_m(t), m \in \mathcal{M}$ and the amount of the task requests from IoT devices covered by UAV m before t, i.e., $\theta_m(t) = \sum_{\tau=1}^{t}\sum_{n\in\mathcal{G}_m}\boldsymbol{v}_n(\tau), m \in \mathcal{M}$, and thus $s^{APSG}(t) = (\boldsymbol{w}_m(t), \theta_m(t))_{m\in\mathcal{M}}$.

2) Action for Each UAV in APSG: UAV $m \in \mathcal{M}$ chooses an action $a_m^{APSG}(t) \in \mathcal{A}_m^{APSG}$ at time slot t, signifying that it selects S_m types of applications from the total C types.

3) Reward of Each UAV in APSG: The immediate reward of UAV $m \in \mathcal{M}$ in APSG at time slot t is given by

$$\mathcal{R}_m^{APSG}(t) = \frac{e(t)}{C}\sum_{\tau=1}^{t}\sum_{n\in\mathcal{G}_m}\boldsymbol{v}_n(\tau)\boldsymbol{w}_m(\tau)^\top, \tag{12}$$

where $e(t)$ indicates the number of application types placed in all UAVs at time slot t. This reward function would guide UAVs to update more popular but diverse applications according to the history of providing MEC services.

The definition of state transition probabilities $\mathcal{P}^{APSG}$ is similar to that in TPSG and is omitted here.

Based on the above three MDP formulations, we develop a novel triple learner (i.e., trajectory learner, energy learner and application learner) based reinforcement learning approach to obtain equilibriums of these three coupled multi-agent stochastic games. Specifically, each UAV runs three Q-learning algorithms to learn the optimal Q values of each state-action pair, and obtain the optimal local policies for trajectory learner, energy learner, and application learner. It is worth noting that, since trajectory planning, energy renewal and application placement are tightly coupled, these three learners have to run in a back-and-forth manner.

1) Settings for Trajectory Learner: The policy π_m^{TPSG}: $\mathcal{S}^{TPSG} \rightarrow \mathcal{A}_m^{TPSG}$ of the trajectory learner in UAV $m \in \mathcal{M}$, meaning a mapping from the environment state set to the action set, signifies a probability distribution of actions $a_m^{TPSG} \in \mathcal{A}_m^{TPSG}$ in a given state s^{TPSG}. Particularly, for UAV m in state $s^{TPSG} \in \mathcal{S}^{TPSG}$, the trajectory policy of the trajectory learner in UAV m can be presented as $\pi_m^{TPSG}(s^{TPSG}) = \{\pi_m^{TPSG}(s^{TPSG}, a_m^{TPSG})|a_m^{TPSG} \in \mathcal{A}_m^{TPSG}\}$, where $\pi_m^{TPSG}(s^{TPSG}, a_m^{TPSG})$ is the probability of UAV m selecting action a_m^{TPSG} in state s^{TPSG}.

In Q-learning, the process of building trajectory policy π_m^{TPSG} is significantly affected by trajectory learner's Q function, and the Q function of the trajectory learner in UAV m is the expected reward by executing action $a_m^{TPSG} \in \mathcal{A}_m^{TPSG}$ in state $s^{TPSG} \in \mathcal{S}^{TPSG}$ under the given policy π_m^{TPSG}, which can be expressed by

$$\begin{aligned} &Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG}, \pi_m^{TPSG}) = \\ &\mathbb{E}(\textstyle\sum_{\tau=0}^{\infty}\gamma^\tau\mathcal{R}_m^{TPSG}(t+\tau+1)|s^{TPSG}(t) = s^{TPSG}, \\ &\boldsymbol{a}(t)^{TPSG} = \boldsymbol{a}^{TPSG}, \pi_m^{TPSG}), \end{aligned} \tag{13}$$

where γ is a constant discounted factor with $\gamma \in [0, 1]$, and the results of (13) are termed as action values, i.e., Q values.

Trajectory learner in UAV $m \in \mathcal{M}$ selects an action $a_m^{TPSG}(t) \in \mathcal{A}_m^{TPSG}$ according to its Q function at slot t. For striking a balance between exploration and exploitation, we consider an ϵ-greedy exploration strategy for the trajectory learner. Specifically, the trajectory learner in UAV $m \in \mathcal{M}$ selects a random action $a_m^{TPSG} \in \mathcal{A}_m^{TPSG}$ in state $s^{TPSG} \in \mathcal{S}^{TPSG}$ with probability ϵ and selects the best action a_m^{TPSG*} with probability $(1-\epsilon)$, where the best action has $Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG*}, \pi_m^{TPSG}) \geq Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG}, \pi_m^{TPSG})$, $\forall \boldsymbol{a}^{TPSG} \in \mathcal{A}^{TPSG}$ with a_m^{TPSG*} being the m-th element of $\boldsymbol{a}^{TPSG*}$. Besides, if the later described energy learner in UAV m selects to return to the depot, the trajectory learner will not choose any action in $\mathcal{A}_m^{TPSG}$. Then, the probability of selecting action $a_m^{TPSG} \in \mathcal{A}_m^{TPSG}$ in state s^{TPSG} can be expressed by

$$\pi_m^{TPSG}(s^{TPSG}, a_m^{TPSG}) = \begin{cases} 0, \text{if UAV } m \text{ decides to return to the depot,} \\ 1-\epsilon, \text{if } Q_m^{TPSG}(s^{TPSG}, \cdot, \cdot) \text{ of } a_m^{TPSG} \text{ is the highest,} \\ \epsilon, \text{otherwise.} \end{cases}$$

In the Q value update step of Q-learning, the trajectory

learner in each UAV $m \in \mathcal{M}$ follows the update rule:

$$\begin{aligned} &Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG}, t+1) = \\ &Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG}, t) + \beta^{TPSG}(\mathcal{R}_m^{TPSG}(t) + \\ &\gamma \max_{\boldsymbol{a}^{TPSG'} \in \mathcal{A}^{TPSG}} Q_m^{TPSG}(s^{TPSG'}, \boldsymbol{a}^{TPSG'}, t) \\ &- Q_m^{TPSG}(s^{TPSG}, \boldsymbol{a}^{TPSG}, t)), \end{aligned} \tag{14}$$

where β^{TPSG} denotes the learning rate in TPSG.

2) Settings for Energy Learner: The policy of energy learner in UAV $m \in \mathcal{M}$ is expressed as π_m^{ERSG}: $\mathcal{S}^{ERSG} \rightarrow \mathcal{A}_m^{ERSG}$.

Here, the Q function of the energy learner in UAV $m \in \mathcal{M}$ can be expressed by

$$\begin{aligned} &Q_m^{ERSG}(s^{ERSG}, \boldsymbol{a}^{ERSG}, \pi_m^{ERSG}) = \\ &\mathbb{E}(\textstyle\sum_{\tau=0}^{\infty} \gamma^\tau \mathcal{R}_m^{ERSG}(t+\tau+1) | s^{ERSG}(t) = s^{ERSG}, \\ &\boldsymbol{a}(t)^{ERSG} = \boldsymbol{a}^{ERSG}, \pi_m^{ERSG}). \end{aligned} \tag{15}$$

The energy learner in UAV $m \in \mathcal{M}$ selects an action $a_m^{ERSG} \in \mathcal{A}_m^{ERSG}$ (i.e., whether returning to the depot) also according the ϵ-greedy exploration strategy. Then, we have

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

The energy learner in UAV $m \in \mathcal{M}$ follows the update rule:

$$\begin{aligned} &Q_m^{ERSG}(s^{ERSG}, \boldsymbol{a}^{ERSG}, t+1) = \\ &Q_m^{ERSG}(s^{ERSG}, \boldsymbol{a}^{ERSG}, t) + \beta^{ERSG}(\mathcal{R}_m^{ERSG}(t) + \\ &\gamma \max_{\boldsymbol{a}^{ERSG'} \in \mathcal{A}^{ERSG}} Q_m^{ERSG}(s^{ERSG'}, \boldsymbol{a}^{ERSG'}, t) \\ &- Q_m^{ERSG}(s^{ERSG}, \boldsymbol{a}^{ERSG}, t)), \end{aligned} \tag{16}$$

where β^{ERSG} denotes the learning rate in ERSG.

3) Settings for Application Learner: The policy of application learner in UAV $m \in \mathcal{M}$ is π_m^{APSG}: $\mathcal{S}^{APSG} \rightarrow \mathcal{A}_m^{APSG}$.

Here, the Q function of the application learner in UAV $m \in \mathcal{M}$ can be expressed by

$$\begin{aligned} &Q_m^{APSG}(s^{APSG}, \boldsymbol{a}^{APSG}, \pi_m^{APSG}) = \\ &\mathbb{E}(\textstyle\sum_{\tau=0}^{\infty} \gamma^\tau \mathcal{R}_m^{APSG}(t+\tau+1) | s^{APSG}(t) = s^{APSG}, \\ &\boldsymbol{a}(t)^{APSG} = \boldsymbol{a}^{APSG}, \pi_m^{APSG}). \end{aligned} \tag{17}$$

The application learner in UAV $m \in \mathcal{M}$ selects an action $a_m^{APSG} \in \mathcal{A}_m^{APSG}$ also according the ϵ-greedy exploration strategy. Then, we have

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

The update rule of application learner in UAV $m \in \mathcal{M}$ is

$$\begin{aligned} &Q_m^{APSG}(s^{APSG}, \boldsymbol{a}^{APSG}, t+1) = \\ &Q_m^{APSG}(s^{APSG}, \boldsymbol{a}^{APSG}, t) + \beta^{APSG}(\mathcal{R}_m^{APSG}(t) + \\ &\gamma \max_{\boldsymbol{a}^{APSG'} \in \mathcal{A}^{APSG}} Q_m^{APSG}(s^{APSG'}, \boldsymbol{a}^{APSG'}, t) \\ &- Q_m^{APSG}(s^{APSG}, \boldsymbol{a}^{APSG}, t)), \end{aligned} \tag{18}$$

where β^{APSG} denotes the learning rate in APSG.

In summary, the proposed TLRL approach is detailedly illustrated in Algorithm 1.

Algorithm 1: TLRL Approach

```
for m = 1 to M do
    Initialize Q values Q_m^TPSG = Q_m^ERSG = Q_m^APSG = 0;
Set the maximal iteration counter LOOP and loop = 0;
for loop < LOOP do
    t = 0;
    for m = 1 to M do
        Send Q_m^TPSG, Q_m^ERSG and Q_m^APSG to other UAVs;
    while t ≤ T do
        Observe state s^TPSG, s^ERSG and s^APSG;
        for m = 1 to M do
            UAV m selects a_m^ERSG according to π_m^ERSG;
            if UAV m returns to the depot then
                UAV m selects a_m^APSG according to π_m^APSG;
            else
                UAV m selects a_m^TPSG according to π_m^TPSG;
        Obtain rewards R_m^TPSG, R_m^ERSG and R_m^APSG;
        Update Q_m^TPSG, Q_m^ERSG and Q_m^APSG according to (14),
          (16) and (18), respectively;
        Send Q_m^TPSG, Q_m^ERSG and Q_m^APSG to other UAVs;
        Set t = t + 1;
    Set loop = loop + 1.
```

TABLE I: Simulation Parameters

Param.	Value	Param.	Value	Param.	Value
M	3	B	10 MHz	C	10
N	300	D_n	$[2, 5]$ MB	V	20 m/s
t^{hover}	5 s	ξ	10^{-18}	f	3 GHz
q	100 m	p_n^{tran}	$[0.2, 0.5]$ W	H	120 m
S_m	6 GB	μ_c	$[1, 3]$ GB	φ	-174 dBm/Hz
a, b	9.6117, 0.1581	f_m^U	2 $Mbps$	Target region	1000 m×1000 m

IV. SIMULATION RESULTS

In this section, simulations are conducted to evaluate the performance of the proposed TLRL approach for $[\mathcal{P}1]$. Table I lists the values of all simulation parameters, and the propulsion power model follows [11]. Similar settings have also been employed in [9], [12].

For comparison purpose, we introduce an energy efficient oriented trajectory planning (EOTP) algorithm and an existing algorithm called decentralized multiple UAVs cooperative reinforcement learning (DMUCRL) [9] algorithm as benchmarks: EOTP determines the trajectories of all UAVs with the aim of maximizing the energy efficiency but asks UAVs to return to the depot for energy renewal only when their batteries are exhausted, and EOTP does not enable the update of application placement; DMUCRL is originally designed to maximize the energy efficiency of UAVs in downlink content sharing by controlling all UAVs to work collaboratively based on a double Q-learning (each UAV contains a trajectory learner and an energy learner).

Fig. 2 investigates the energy efficiency with different IoT devices' transmission power under DMUCRL, EOTP and the proposed TLRL. It can be observed that the energy efficiency first increases and then becomes stable with the increase of IoT devices' transmission power. This is because, with a larger

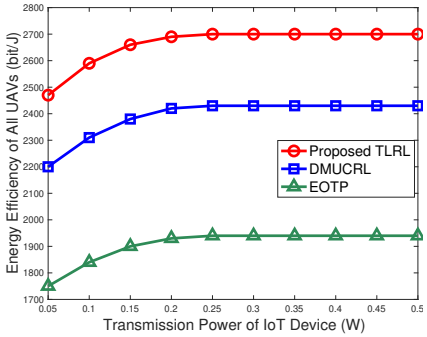

Fig. 2: Energy efficiency w.r.t. transmission power of IoT devices.

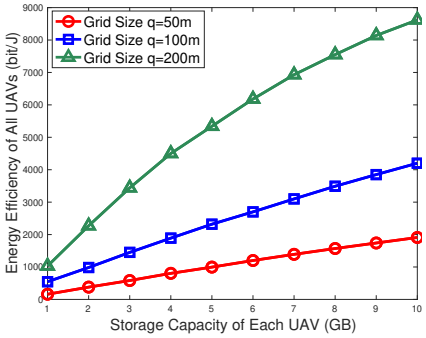

Fig. 3: Energy efficiency w.r.t. storage capacity of each UAV.

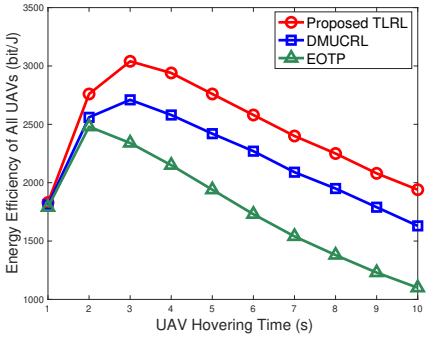

Fig. 4: Energy efficiency w.r.t. UAV hovering time.

transmission power, IoT devices would offload more tasks to their associated UAVs, and thereby increasing the amount of tasks processed by UAVs. However, since the computing capacity of each UAV is still limited, such increasing trend slows down as the limit is approaching. More importantly, this figure shows that the proposed TLRL outperforms both DMUCRL and EOTP. The reason is that i) each UAV under EOTP returns to the depot directly once its energy is exhausted regardless of other UAVs (lacking the cooperation in providing all kinds of MEC services to IoT devices); ii) each UAV's applications are fixed placed under DMUCRL, making it capable of serving very limited IoT devices; and iii) our proposed TLRL well addresses the shortcomings of DMUCRL and EOTP by jointly optimizing all UAVs' trajectory planning, energy renewal and application placement.

Fig. 3 shows all UAV's energy efficiency with different UAV storage capacities under different grid size settings.Specifically, UAVs can adjust their downlink transmission ranges so as to adjust the size q of grids. It can be seen from Fig. 3 that the larger the grid size is, the higher energy efficiency of all UAVs is obtained. This is because with a larger grid size, more IoT devices are included in a grid, and thereby each UAV can potentially process more offloaded tasks. Besides, it is also shown that the energy efficiency of all UAVs increases monotonically with the storage capacity of each UAV. The reason is that with the increase of storage capacity, more types of applications can be placed in each UAV, so that more tasks may be processed.

Fig. 4 illustrates the energy efficiency of all UAVs with different UAV hovering time under DMUCRL, EOTP and the proposed TLRL. It can be observed that, the energy efficiency of all UAVs first increases with the UAV hovering time, and then decreases. This is because with the growth of UAV hovering time, more offloaded tasks from IoT devices can be computed by UAVs during hovering. However, when all tasks have been completely processed by UAVs, they will become idle and consume hovering energy over the target region until hovering time expires. Additionally, it is also shown that the proposed TLRL outperforms both DMUCRL and EOTP, and the explanations for this are similar to those for Fig. 2.

V. Conclusion

In this paper, an energy efficient scheduling problem for multi-UAV assisted MEC has been studied. With the aim of maximizing the long-term energy-efficiency of all UAVs, a joint optimization of UAVs' trajectory planning, energy renewal and application placement is formulated. By taking the inherent cooperation and competition among UAVs, we reformulate such optimization problem as three coupled multi-agent stochastic games, and then propose a novel TLRL approach for reaching equilibriums. Simulation results show that, compared to counterparts, the proposed TLRL approach can significantly increase the energy efficiency of all UAVs.

References

[1] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 10, pp. 3536–3550, Oct. 2020.

[2] Q. Song, S. Jin, and F. Zheng, "Completion time and energy consumption minimization for UAV-enabled multicasting," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 3, pp. 821–824, Jun. 2019.

[3] H. Wang, J. Wang *et al.*, "Completion time minimization with path planning for fixed-wing UAV communications," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 7, pp. 3485–3499, Jul. 2019.

[4] J. Ji, K. Zhu *et al.*, "Energy consumption minimization in UAV-assisted mobile-edge computing systems: Joint resource allocation and trajectory design," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8570–8584, 2021.

[5] J. Zhang, L. Zhou, F. Zhou, B.-C. Seet, H. Zhang, Z. Cai, and J. Wei, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, 2020.

[6] G. Zheng, C. Xu, M. Wen, and X. Zhao, "Service caching based aerial cooperative computing and resource allocation in multi-UAV enabled MEC systems," *IEEE Trans. Veh. Technol.*, pp. 1–14, 2022.

[7] Y. Zhao, Z. Li, N. Cheng, R. Zhang, B. Hao, and X. Shen, "UAV deployment strategy for range-based space-air integrated localization network," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[8] L. Yang, H. Yao *et al.*, "Multi-UAV deployment for MEC enhanced IoT networks," in *Proc. IEEE ICCC*, 2020, pp. 436–441.

[9] 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, Oct. 2021.

[10] 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 Things J.*, vol. 7, no. 7, pp. 5906–5921, Jul. 2020.

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 4, p. 2329–2345, Apr. 2019.

[12] 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, May. 2022.