# Distributed Energy-efficient Computation Offloading and Trajectory Planning in Aerial Edge Networks

Xiaoyan Huang, Yiding Wen, Supeng Leng
*School of Information & Communication Engineering*
*University of Electronic Science and Technology of China*
Chengdu, China
xyhuang@uestc.edu.cn,eadingwen@std.uestc.edu.cn,spleng@uestc.edu.cn

Yan Zhang
*Department of Informatics*
*University of Oslo*
Oslo, Norway
yanzhang@ieee.org

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000889

***Abstract*—In this paper, we investigate energy-efficient computation offloading and trajectory planning for aerial edge networks, wherein multiple Unmanned Aerial Vehicles (UAVs) cooperate with each other to provide computing service to the ground users. In particular, we formulate the joint optimization of computation offloading, channel allocation, power control, resource allocation, and trajectory planning as a multi-agent deep reinforcement learning (MA-DRL) problem to minimize the system energy consumption while guaranteeing the latency requirements of computation tasks. Then, we propose a distributed algorithm to enable UAVs to independently make action decisions based on their local observations, still collaboratively learn their policies for system performance. Numerical results demonstrate that our proposed algorithm can effectively plan the trajectory, reduce the system energy consumption, and improve the latency requirements satisfaction ratio.**

***Index Terms*—unmanned aerial vehicle, multi-agent deep reinforcement learning, computation offloading, trajectory planning**

## I. Introduction

With the advantages of on-demand configuration and deployment, unmanned aerial vehicles (UAVs) equipped with mobile edge computing (MEC) servers emerges as a promising solution to provide flexible computing services for ground users to support various computation-intensive and delay-sensitive applications [1] [2]. However, to reap the potential benefits of aerial edge networks, there are several challenges: 1) how to control each UAV's trajectory according to the distribution and demands of users to improve the performance of edge computing services; 2) determine whether each user should offload its computation task considering the limited on-board computing capacity and the latency requirement of computation task, and if so, which UAV should offload to; 3) how to allocate the limited communication and computing resources to the offloading users considering the interference among different UAVs, so as to improve the system performance while satisfying the diversified requirements of computation tasks.

Recently, many research efforts have been made for computation offloading and trajectory planning in aerial edge computing networks. The authors in [3], [4], and [5] addressed the energy consumption minimization problem by decomposing it into multiple subproblems, whereas the successive convex approximation (SCA) and/or decomposition and iteration (DAI) methods are utilized in [6], [7], and [8]. The authors in [9] solved the computation efficiency maximization problem by decomposing it into three subproblems. The authors in [10] solved the sum of the maximum delay minimization problem by a penalty dual decomposition-based algorithm.

The aforementioned centralized optimization method based algorithms often require complete global network information, resulting in large signaling overhead for large-scale networks. Moreover, the centralized algorithms are inapplicable to the decentralized networks without a central controller. To this end, some efforts have been made to leverage deep reinforcement learning (DRL) methods to address the prohibited complexity of the joint optimization of computation offloading and and trajectory planning in highly dynamic wireless environment. The authors in [11] proposed a centralized single-agent deep deterministic policy gradient (DDPG) based algorithm to minimize the energy consumption of all users by optimizing computation offloading and trajectory, without considering channel allocation and power control. The authors in [12] proposed a distributed multi-agent deep deterministic policy gradient (MADDPG)-based method to manage the spectrum, computing, and caching resources of macro eNodeB and UAVs in vehicular networks, aiming at maximizing the number of offloaded tasks, without considering trajectory planning and power control.

In this paper, we focus on distributed design of energy-efficient computation offloading and trajectory planning for aerial edge networks without central controllers. In particular, we formulate the system energy consumption minimization problem of multi-UAV and multi-user system as a multi-agent deep reinforcement learning (MA-DRL) problem to jointly optimize the computation offloading, channel allocation, power control, computation resource allocation, and trajectory planning, taking into account the latency requirements of diverse computation tasks. Then, we propose a MA-DRL based algorithm to enable the UAVs to independently make action decisions based on their local observations, and collaboratively learn their policies. Numerical results indicated

This work was supported in part by the National Natural Science Foundation of China under Grants 61941102 and 62071092.

that the proposed algorithm is capable of effectively planning the trajectory, and outperforms the benchmark algorithms on the performance of system energy consumption and latency guaranteed percentage.

The remainder of this paper is organized as follows. The system model and problem formulation are presented in Section II. The proposed MA-DRL algorithm is introduced in Section III. Numerical results are presented in Section IV. Finally, the paper is concluded in Section V.

## II. System Model and Problem Formulation

### A. Aerial Edge Networks

Consider an aerial edge network, with a set of UAVs equipped with MEC servers $\mathcal{M} = \{1, 2, ..., M\}$ and a set of ground users $\mathcal{N} = \{1, 2, ..., N\}$, as illustrated in Fig.1. During a period of $T$ equal-length time slots, the UAVs fly above the area at altitude $H$, and cooperatively provide computing services to the users. Denote the locations of user $n$ and UAV $m$ at time slot $t \in \mathcal{T} = \{1, 2, ..., T\}$ as $\mathbf{u}_n[t] = (x_n[t], y_n[t], 0)$ and $\mathbf{u}_m[t] = (x_m[t], y_m[t], H)$, respectively.

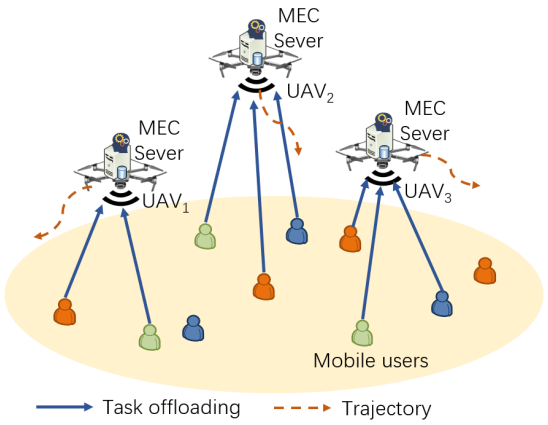



Fig. 1: Aerial edge network

The computation task of user $n$ at time slot $t$ can be described as $Task_n[t] = \{d_n[t], \omega_n[t], \tilde{T}_n[t]\}$, where $d_n[t]$ denotes the data size of the computation task, and $\omega_n[t]$ represents the CPU cycles required to execute the task, and $\tilde{T}_n[t]$ is the maximum latency tolerated by the user. To complete the computation tasks, the users can compute them locally, or offload them to the MEC servers mounted in UAVs, named local computing and edge computing respectively. Define $\alpha_{m,n}[t] \in \{0, 1\}$ as the computation offloading indicator at time slot $t$, where $\alpha_{m,n}[t] = 1$ represents that user $n$ offloads its task to UAV $m$, and $\alpha_{m,n}[t] = 0, \forall m \in \mathcal{M}$ represents that local computing is adopted. In this paper, we consider the binary offloading strategy. Assume that each UAV can serve multiple users simultaneously, and each user can be served by at most one UAV at a time, that is

$$\sum_{m \in \mathcal{M}} \alpha_{m,n}[t] \leq 1, \forall n, t \tag{1}$$

### B. Communication Model

Considering the limited spectrum resource, the UAVs reuse the full spectrum of the system to improve the spectrum utilization efficiency. Let $\mathcal{K} = \{1, 2, ..., K\}$ denote the set of orthogonal channels in the system. Define $b_{m,n,k}[t] \in \{0, 1\}$ as the channel allocation indicator at time slot $t$, where $b_{m,n,k}[t] = 1$ represents that UAV $m$ allocates channel $k$ to user $n$, and otherwise $b_{m,n,k}[t] = 0$. To avoid the interference among the users served by the same UAV, the orthogonal frequency-division multiple access scheme is adopted, that is

$$\sum_{n \in \mathcal{N}} b_{m,n,k}[t] \leq 1, \forall m, k, t \tag{2}$$

Furthermore, each user can be allocated at most one channel for offloading its computation task, that is

$$\sum_{k \in \mathcal{K}} b_{m,n,k}[t] = \alpha_{m,n}[t], \forall m, n, t \tag{3}$$

For the ground-to-air uplink communications between users and UAVs, the probabilistic path loss model [13] is applied taking into account the impacts of both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) conditions, i.e, the average path loss can be given by

$$PL_{m,n}[t] = \hat{p}^{LoS}_{m,n}[t] \phi_3 d^{-2}_{m,n}[t], \forall m, n, t \tag{4}$$

where $\hat{p}^{LoS}_{m,n}[t] = p^{LoS}_{m,n}[t] + \phi_4(1 - p^{LoS}_{m,n}[t])$ is the regularized probability of LoS link with $\phi_4$ being the additional attenuation factor due to the NLoS condition. $p^{LoS}_{m,n}[t] = 1/(1 + \phi_1 exp(-\phi_2(\xi_{m,n}[t] - \phi_1)))$, where $\phi_1$ and $\phi_2$ are two constant system parameters reflecting the propagation environment impact. $\xi_{m,n}[t] = \frac{180}{\pi} \arcsin(H/d_{m,n}[t])$ is the elevation angle in degree, where $d_{m,n}[t]$ is the Euclidean distance between UAV $m$ and user $n$. $\phi_3$ is the reference channel gain at one unit distance in meter.

Let $\psi_{m,n,k}[t]$ denote the small-scale fading of the link between UAV $m$ and user $n$ on channel $k$ at time slot $t$, with $\mathbb{E}|\psi_{m,n,k}[t]|^2 = 1$. Define $p_n[t] \in [0, P^{max}_n]$ as the transmit power of user $n$ at time slot $t$, where $P^{max}_n$ is the maximum transmit power. Therefore, the achievable uplink data rate from user $n$ to UAV $m$ at time slot $t$ is given by

$$U_{m,n}[t] = \sum_{k \in \mathcal{K}} b_{m,n,k}[t] B \log_2 \left(1 + \frac{p_n[t] h_{m,n,k}[t]}{N_0 B + I_{m,k}[t]}\right) \tag{5}$$

where $B$ is the bandwidth of each channel. $h_{m,n,k}[t] = PL_{m,n}[t]|\psi_{m,n,k}[t]|^2$ is the channel gain. $N_0$ is the spectrum density of the additive white Gaussian noise (AWGN). $I_{m,k}[t] = \sum_{m' \neq m} \sum_{n \in \mathcal{N}} b_{m',n,k}[t] p_n[t] h_{m,n,k}[t]$ is the received co-channel interference on channel $k$ for UAV $m$. According to (5), the achievable uplink data rate of a user depends not only on the allocated channel and its transmit power, but also on the inter-cell interference from the offloading users served by the other UAVs.

### C. Computation Model

*1) Local Computing:* Denote the CPU frequency and energy consumption coefficient for per CPU cycle of user $n$ as $C_n$ and $\varepsilon_n$, respectively. Thus, the delay and energy consumption for local computing at time slot $t$ can be respectively given by

$$T_n^{LC}[t] = \omega_n[t]/C_n, \forall n, t. \tag{6}$$

and

$$E_n^{LC}[t] = \omega_n[t]\varepsilon_n, \forall n, t. \tag{7}$$

*2) Edge Computing:* According to the communication model presented in section II-B, the delay and energy consumption for transmitting $Task_n[t]$ to the UAV $m$ at time slot $t$ are respectively given by

$$T_{m,n}^{EC,tr}[t] = d_n[t]/U_{m,n}[t], \forall m, n, t. \tag{8}$$

and

$$E_{m,n}^{EC,tr}[t] = p_n[t]d_n[t]/U_{m,n}[t], \forall m, n, t. \tag{9}$$

Denote the computation capacity of the MEC server in UAV $m$ as $\bar{C}_m$, which is finite and fixed, but may vary over the UAVs. Define $f_{m,n}[t] \in [0, 1]$ as the fraction of computation resource assigned to user $n$ from UAV $m$ at time slot $t$. The computation resource assigned to the users cannot exceed the total available computation resource, that is

$$\sum_{n\in\mathcal{N}} \alpha_{m,n}[t] f_{m,n}[t] \leq 1, \forall m, t \tag{10}$$

Accordingly, the delay for edge computing can be expressed as

$$T_{m,n}^{EC,ex}[t] = \omega_n[t]/(f_{m,n}[t]\bar{C}_m), \forall m, n, t. \tag{11}$$

Given the energy consumption coefficient $\bar{\varepsilon}_m$ for per CPU cycle of the MEC server in UAV $m$, the energy consumption for edge computing is given by

$$E_{m,n}^{EC,ex}[t] = \omega_n[t]\bar{\varepsilon}_m, \forall m, n, t. \tag{12}$$

Therefore, the delay and energy consumption of user $n$ for offloading its task to UAV $m$ at time slot $t$ can be respectively written as

$$T_{m,n}^{EC}[t] = T_{m,n}^{EC,tr}[t] + T_{m,n}^{EC,ex}[t], \forall m, n, t \tag{13}$$

and

$$E_{m,n}^{EC}[t] = E_{m,n}^{EC,tr}[t] + E_{m,n}^{EC,ex}[t], \forall m, n, t. \tag{14}$$

### D. Mobility Model of the UAV

Let $\mathbf{v}_m[t] = (v_m^x[t], v_m^y[t])$ denote the velocity of UAV $m$ at time slot $t$. Within the operating duration, the location and velocity of UAV $m$ at time slot $t$ should satisfy the following mobility constraints:

$$\mathbf{u}_m[0] = \bar{\mathbf{u}}_m, \mathbf{v}_m[0] = \bar{\mathbf{v}}_m, \forall m, \tag{15}$$

$$\mathbf{u}_m[t+1] = \mathbf{u}_m[t] + \mathbf{v}_m[t]\lambda, \forall m, t \tag{16}$$

$$\nu_m^{min} \leqslant \|\mathbf{v}_m[t]\| \leqslant \nu_m^{max}, \forall m, t \tag{17}$$

where $\bar{\mathbf{u}}_m$ and $\bar{\mathbf{v}}_m$ represent the initial location and velocity of UAV $m$, respectively. $\lambda$ is the time slot duration. $\nu_m^{min}$ and $\nu_m^{max}$ are the minimum and maximum speed, respectively,

### E. Problem Formulation

In the aerial edge network, the quality of offloading service provided by the UAVs is tightly coupled with each other due to the co-channel interference. To improve the overall system performance, it remains challenging to jointly design computation and communication resource allocation and trajectory planning to facilitate UAVs to provide computing services to users cooperatively. To this end, the problem of interest in this paper is to jointly optimize the computation offloading decision, channel allocation, power control, computation resource allocation, and UAVs' trajectories, with the objective of minimizing the total energy consumption while satisfying the latency requirements of the users.

$$\begin{aligned}
&\text{P0:} \min_{\boldsymbol{\alpha},\mathbf{b},\mathbf{p},\mathbf{f},\mathbf{v}} \sum_{t=1}^{T} E[t] \\
&s.t. \\
&\quad (1)-(3), (10), (13)-(15) \\
&\quad \text{C1:} \sum_{m=1}^{M} \alpha_{m,n}[t] T_{m,n}^{EC}[t] \\
&\qquad + \left(1 - \sum_{m=1}^{M} \alpha_{m,n}[t]\right) T_n^{LC}[t] \leqslant \tilde{T}_n[t], \forall n, t
\end{aligned} \tag{18}$$

with

$$\begin{aligned}
E[t] = &\sum_{m=1}^{M}\sum_{n=1}^{N} \alpha_{m,n}[t] E_{m,n}^{EC}[t] \\
&+ \sum_{n=1}^{N} \left(1 - \sum_{m=1}^{M} \alpha_{m,n}[t]\right) E_n^{LC}[t]
\end{aligned} \tag{19}$$

where $E[t]$ is the system energy consumption. Constraint C1 is considered to ensure the latency requirements of tasks. The optimization variables consist of the binary variables, i.e., $\alpha_{m,n}[t]$ and $b_{m,n,k}[t]$ , and the real variables, i.e., $p_n[t]$, $f_{m,n}[t]$, and $\mathbf{v}_m[t]$. The mixed integer non-linear programming (MINLP) problem P0 is non-convex due to the interference terms in the achievable data rate in (5), and combinatorial due to the binary variables, such that it can not be solved directly using the traditional optimization methods.

## III. Multi-Agent DRL based Computation Offloading and Trajectory Planning Scheme

### A. Cooperative Markov Game Modeling

In the aerial edge network, each UAV individually decides the set of served users, manages its computation and communication resources, as well as controls its trajectory. Since there exists the co-channel interference in the wireless uplink communications, and each user can only be served by at most one UAV at a time slot, the decisions of different UAVs affect each other. Therefore, it can be modelled as a multi-agent reinforcement learning problem. Each UAV agent interacts with the dynamic environment to gain experiences, improving its policy of computation offloading and trajectory planning.

The formulated optimization problem P0 can be modelled as a partially observable Markov game [14]. At each time $t$, UAV agent $m$ receives an observation $s_m[t]$ of the current environment state, and then takes an action $a_m[t]$. Thereafter, it receives the immediate reward $r_m[t]$ based on the joint action of all agents, and the environment evolves to the next state. The new observation $s_m[t+1]$ is then received by UAV agent $m$ with a transition probability of $p(s_m[t+1]|s_m[t], a_1[t], ..., a_M[t])$. Specifically, the observation, action, and reward function are defined as follows.

*1) Observation:* The local observation $s_m[t]$ of UAV agent $m$ at time slot $t$ consists of the locations of its own and the users, the task profiles, the local instant channel gains of the links between it and each user on each channel, and received interference on each channel.

*2) Action:* Based on the local observation $s_m[t]$, UAV agent $m$ takes an action $a_m[t] = \{\mathbf{v}_m[t], \boldsymbol{\alpha}_m[t], \mathbf{b}_m[t], \mathbf{p}_m[t], \mathbf{f}_m[t]\}$, including its own velocity, computation offloading decision, channel allocation, uplink power control, and computation resource allocation for the users served by itself.

*3) Reward:* According to the optimization problem P0 in (18), a system performance-oriented immediate reward function $r_m[t]$ is designed to encourage the UAVs to cooperate with each other, thus improving the overall system performance, that is

$$r_m[t] = -E[t] + \phi_5 \sum_{n=1}^{N} \log_2 \left( \frac{\tilde{T}_n[t]}{\hat{del}_n[t]} + \phi_6 \right) + \phi_7 K[t] \quad (20)$$

where $\hat{del}_n[t] = \sum_m \alpha_{m,n}[t] T^{EC}_{m,n}[t] + (1 - \sum_m \alpha_{m,n}[t]) T^{LC}_n[t]$ represents the delay of completing the task of user $n$. $\phi_6 \in (0,1)$ is a constant hyperparameter, which is tuned empirically such that $\log_2 \left( \frac{\tilde{T}_n[t]}{\hat{del}_n[t]} + \phi_6 \right)$ is a positive real number if $\hat{del}_n[t] \leqslant \tilde{T}_n[t]$, and otherwise a negative real number. $K[t]$ represents the number of users whose latency requirements are guaranteed at time slot $t$. In (20), the second part describes how far the latency requirements of the users are satisfied by the current actions of all UAV agents. The learning goal is to find a joint computation offloading and trajectory planning policy to maximize the expected cumulative discounted reward, i.e., $J_m = \mathbb{E}\left( \sum_{t=0}^{T} \gamma^t r_m[t] \right)$, where the constant $\gamma \in [0,1)$ is the discount factor.

### *B. Multi-Agent DRL based Algorithm*

To solve the formulated Markov game for aerial edge network, based on the multi-agent deep deterministic policy gradient (MADDPG) [12], we propose a distributed multi-agent deep reinforcement learning (MA-DRL) based computation offloading and trajectory planning scheme. In the proposed MA-DRL algorithm, the UAV agents independently make decisions on computation offloading, channel allocation, power control, resource allocation, and trajectory control based on their local observations. Moreover, the UAV agents refine their strategies through collaborative learning, so as to improve the system energy consumption while guaranteeing the latency requirements of users.

Specifically, each UAV $m$ is modelled as a DDPG agent, consisting of an actor network $\boldsymbol{\mu}_m$ and a critic network $Q_m$, as well as a target actor network $\boldsymbol{\mu}'_m$ and a target critic network $Q'_m$. The input of the actor network is the local observations of the agent itself, and the output is its selected actions. The input of the critic network contains the observations and actions of the agent itself along with the observations and actions of the other agents, and the output is the corresponding Q-value. During the training stage, a mini-batch of $W$ transition tuples $(\mathbf{s}[t], \mathbf{a}[t], \mathbf{r}[t], \mathbf{s}[t+1])$ from an experience replay buffer can be selected to update the critic and actor networks, with the joint observation $\mathbf{s}[t] = \{s_1[t], ..., s_M[t]\}$, the joint action $\mathbf{a}[t] = \{a_1[t], ..., a_M[t]\}$, and the joint reward $\mathbf{r}[t] = \{r_1[t], ..., r_M[t]\}$. To be specific, the critic network of UAV agent $m$ is updated by minimizing the loss function $L_m\left(\theta^Q_m\right)$ given by (21) using an appropriate optimizer, e.g., the stochastic gradient descent method.

$$L_m(\theta^Q_m) = \frac{1}{W} \sum_{j=1}^{W} \left[ y^j_m - Q^{\boldsymbol{\mu}}_m \left( \mathbf{s}^j, a^j_1, ..., a^j_M \right) \right]^2 \quad (21)$$

where $j$ is the index of the sampled mini-batches. $y^j_m$ is the target value calculated by the target critic network, given by

$$y^j_m = r^j_m + \gamma Q^{\boldsymbol{\mu}'}_m (\mathbf{s}'^j, a'_1, ..., a'_M) \mid_{a'_k = \boldsymbol{\mu}'_k(s^j_k)} . \quad (22)$$

Note that the independent learning methods evaluate the quality of the selected action only based on local observations. In contrast, in the proposed multi-agent collaborative learning algorithm, the action-value function $Q^{\boldsymbol{\mu}}_m(\mathbf{s}, \mathbf{a})$ is based on the the joint observation and joint action of all UAV agents, considering the coupling relationship between the actions of UAV agents. Therefore, the critic network of each UAV agent can evaluate the quality of its selected action considering the influence of the other UAV agents' behaviour, so that the UAV agents can improve their policies in a cooperative way for the desired global objective.

Meanwhile, the weight $\theta^{\boldsymbol{\mu}}_m$ of the actor network $\boldsymbol{\mu}_m$ can be updated by the policy gradient scheme with

$$\nabla_{\theta^{\boldsymbol{\mu}}_m} J_m(\theta^{\boldsymbol{\mu}}_m) = \frac{1}{W} \sum_{j=1}^{W} \nabla_{\theta^{\boldsymbol{\mu}}_m} \boldsymbol{\mu}(o^j_m) \nabla_{a_m} Q^{\boldsymbol{\mu}}_m (\mathbf{s}^j, a^j_1, ., a_m, ., a^j_M) \big|_{a_m = \boldsymbol{\mu}(s^j_m)} \quad (23)$$

The proposed MA-DRL based computation offloading and trajectory planning algorithm is summarized as Algorithm 1. According to the centralized learning and distributed implementation framework, the proposed algorithm is divided into the training and execution phases in implementation. Specifically, in the training phase, the historical information of all UAV agents is employed to train the actor and critic networks of each UAV agent, thus promoting the collaboration among the UAV agents. In the execution phase, based on the own observation of the environment, each UAV agent utilizes its trained actor network to decide its trajectory, the set of

served users, channel allocation, power control, and computation resource allocation. The time complexity of Algorithm 1 mainly depends on the number of UAVs and the neural network structure. Assuming that the actor and critic network of each UAV agent contains $J$ and $L$ fully connected layers, respectively. The time complexity can be represented by $\mathcal{O}\left(M \times \left(\sum_{j=0}^{J} n_{A,j} n_{A,j+1} + \sum_{l=0}^{L} n_{C,l} n_{C,l+1}\right)\right)$, where $n_{A,j}$ and $n_{C,l}$ represent the unit number in $j$-th actor net layer and $l$-th critic net layer, respectively.

**Algorithm 1** MA-DRL based Computation Offloading and Trajectory Planning Algorithm

Initialize actor networks, critic networks, and experience replay buffer;
**for** each episode $epi = 1, 2, ...$ **do**
  Initialize the local observation state $\mathbf{s}[0]$;
  **for** each time $t = 1, 2, ...,$ **do**
    Each UAV agent selects action $a_m[t] = \boldsymbol{\mu}_m(o_m[t])$ and executes it;
    Each UAV agent receives the reward $r_m[t]$ with (20) and the new observation $o_m[t+1]$;
    Store the tuple $(\mathbf{s}[t], \mathbf{a}[t], \mathbf{r}[t], \mathbf{s}[t+1])$ in $\mathcal{D}$;
    Sample a random mini-batch of tuples from $\mathcal{D}$;
    $\theta_m^Q \leftarrow \theta_m^Q - \delta \nabla_{\theta_m^Q} L(\theta_m^Q), \forall m$ with (21);
    $\theta_m^{\boldsymbol{\mu}} \leftarrow \theta_m^{\boldsymbol{\mu}} - \delta \nabla_{\theta_m^{\boldsymbol{\mu}}} J(\theta_m^{\boldsymbol{\mu}}), \forall m$ with (23);
    $\theta_m^{Q'} = \tau \theta_m^Q + (1-\tau)\theta_m^{Q'}, \forall m$
    $\theta_m^{\boldsymbol{\mu}'} = \tau \theta_m^{\boldsymbol{\mu}} + (1-\tau)\theta_m^{\boldsymbol{\mu}'}, \forall m$ .
  **end for**
**end for**

## IV. Numerical Results

To verify the performance of the proposed MA-DRL based computation offloading and trajectory planning algorithm (denoted as "MA-DRL"), we consider two benchmark algorithms: 1) independent learning [15] based algorithm (denoted as "I-Learning"), wherein each UAV learns policy independently based on its own observation and interaction with the environment, aiming at maximizing its local reward; 2) cooperative learning based algorithm (denoted as "C-Learning"), wherein the UAVs are independent learners as in I-Learning algorithm, except that the objective is to maximize the system reward.

We considered an area of 1km×1km with 3 UAVs and randomly distributed 10 ground users. The UAVs are placed at three corners of the area initially, with a height of 30m. There are 4 orthogonal channels with a bandwidth of 1MHz. The maximum transmit power of each user is 0.1W. The data size of computation task is randomly distributed between 200KB and 300KB, and the CPU cycle required for each bit of data is randomly distributed between 500 and 1000. The delay demand of each UE is set to 95% of the delay of local computing. The energy consumption coefficients of each UAV and each UE are 10mJ/Gcycle and 1J/Gcycle, respectively. In all the three algorithms, each UAV agent is modeled as a DDPG agent with three-layer fully connected neural network. The sizes of the experience replay buffer and minibatch are set to 10000 and 3000, respectively. The learning rate is set to 0.0001, and the soft update rate of target networks is set to 0.01.

Fig. 2 compares the convergence performance of different algorithms. It can be seen that the proposed MA-DRL algorithm obtains the highest system reward. In I-Learning algorithm, the UAVs competes with each other to maximize their individual local rewards, significantly compromising the overall system performance. Although the learning objectives of C-Learning and MA-DRL algorithms are the same, the MA-DRL algorithm utilizes the global information to train local models to adapt to the non-stationary environment, thus achieving a better reward.

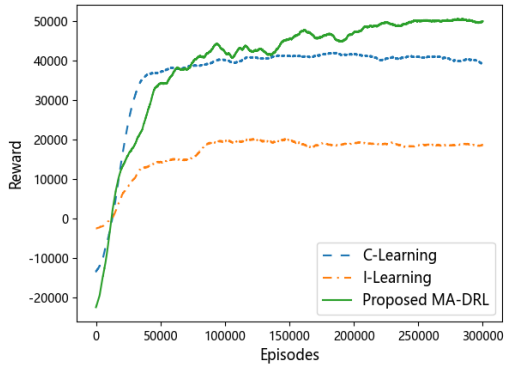
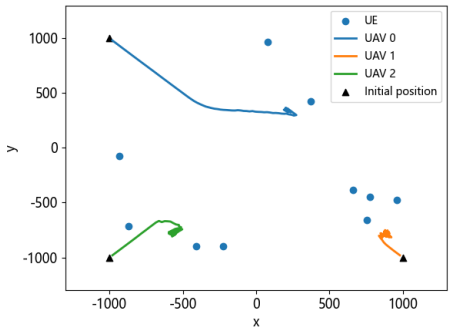


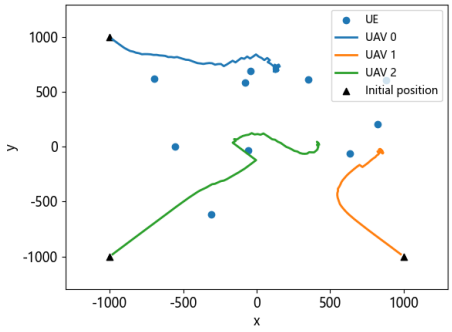

Fig. 2: Convergence performance of different algorithms

Fig. 3 shows the UAVs' trajectories in the MA-DRL algorithm in two typical scenarios with different distribution of the users, i.e., clustered distribution and uniform distribution. In the case of clustered distribution as shown in Fig. 3(a), we can observe that the UAVs fly their way towards the user clusters close to their initial locations, and find the proper spots to provide computing service to the users. In the case of uniform distribution as shown in Fig. 3(b), it can be seen that the UAVs fly towards the center of the target area at the beginning. After a while, the UAVs begin to fly in different directions to serve the different users while mitigating communication interference.

Fig. 4 and Fig. 5 show the statistics of system energy consumption and latency guaranteed percentage for different algorithms. From Fig. 4 and Fig. 5, we can observe that the MA-DRL algorithm outperforms the reference algorithms. Since the delay demand of each user is set to 95% of the delay of local computing and three UAVs starts from the fixed initial locations, not all users can be served by the UAVs in the beginning period, resulting in a latency guaranteed percentage of less than 1. Compared to the I-Learning algorithm, the MA-DRL algorithm promotes the cooperation among UAVs to improve the overall system performance. On average, the MA-DRL algorithm reduces the energy consumption by 15.2%, and increases the latency guaranteed percentage by 5.8%. Compared to the C-Learning algorithm, the MA-DRL




(a) Clustered distribution of users




(b) Uniform distribution of users

Fig. 3: Trajectories of UAVs in different scenarios

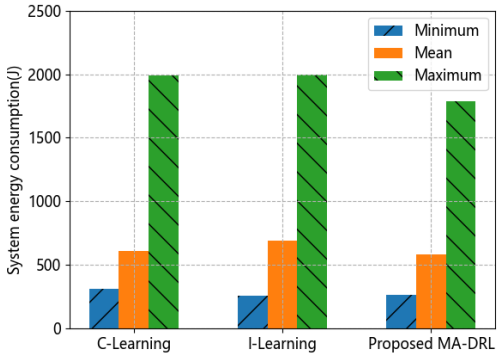



Fig. 4: System energy consumption for different algorithms

algorithm collaboratively trains the models, enabling the UAVs to learn better policies, thereby further improving the system performance. On average, the MA-DRL algorithm reduces the energy consumption by 3.8%, and increases the latency guaranteed percentage by 2.2%.

## V. Conclusion

In this paper, we proposed a distributed MA-DRL based algorithm to jointly optimize computation offloading, channel allocation, power control, computation resource allocation, and trajectory planning, with an objective of minimizing the system energy consumption. The proposed MA-DRL algorithm promotes the cooperation among UAVs through collaborative learning, thus improving the system performance. Numerical results verified the effectiveness of our proposed algorithm.

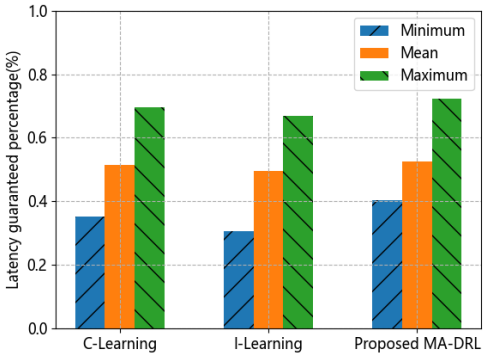



Fig. 5: Latency guaranteed percentage for different algorithms

## References

[1] F. Zhou, et al., "Mobile edge computing in unmanned aerial vehicle networks", *IEEE Wireless Commun.*,vol. 27, no. 1, pp. 140–146, Feb. 2020.

[2] L. Qian, et al., "NOMA assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial internet of things", *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5688-5698, Aug. 2021.

[3] Z. Yang, et al., "Energy efficient resource allocation in UAV-enabled mobile edge computing networks", *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576-4589, Sept. 2019.

[4] X. Hu, et al., "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization", *IEEE Trans.Wireless Commun.*, vol. 18, no. 10, pp. 4738-4752, Oct. 2019.

[5] T. Zhang, et al., "Joint computation and communication design for UAV-assisted mobile edge computing in IoT", *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505-5516, Aug. 2020.

[6] Y. Liu, et al., "UAV-assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization", *IEEE Internet Things J.*, vol.7, no. 4, pp. 2777-2790, Apr. 2020.

[7] M. Li, et al., "Energy efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization", *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp.3424-3438, Mar. 2020.

[8] C. Sun, et al., "Joint computation offloading and trajectory planning for UAV-assisted edge computing", *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5343-5358, Aug. 2021.

[9] J. Zhang, et al., "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing", *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114-2125, Feb. 2020.

[10] Q. Hu, et al., "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems", *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879-1892, Apr. 2019.

[11] L. Wang, et al., "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing", *IEEE Trans. Mobile Computing*, early access, doi:10.1109/TMC.2021.3059691.

[12] H. Peng, et al., "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks", *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[13] A. Al-Hourani, et al., "Optimal LAP altitude for maximum coverage", *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[14] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," *In Proceedings of the eleventh international conference on machine learning*, vol. 157, pp. 157–163, 1994.

[15] J. Cui, et al., "Multi-agent reinforcement learning-based resource allocation for UAV networks", *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729-743, Feb. 2020.