Multi-Agent Deep Reinforcement Learning for Task Offloading in UAV-Assisted Mobile Edge Computing

Nan Zhao, *Member, IEEE*, Zhiyang Ye, Yiyang Pei, *Senior Member, IEEE*,
Ying-Chang Liang, *Fellow, IEEE*, and Dusit Niyato, *Fellow, IEEE*

Abstract—Mobile edge computing can effectively reduce service latency and improve service quality by offloading computation-intensive tasks to the edges of wireless networks. Due to the characteristic of flexible deployment, wide coverage and reliable wireless communication, unmanned aerial vehicles (UAVs) have been employed as assisted edge clouds (ECs) for large-scale sparely-distributed user equipment. Considering the limited computation and energy capacities of UAVs, a collaborative mobile edge computing system with multiple UAVs and multiple ECs is investigated in this paper. The task offloading issue is addressed to minimize the sum of execution delays and energy consumptions by jointly designing the trajectories, computation task allocation, and communication resource management of UAVs. Moreover, to solve the above non-convex optimization problem, a Markov decision process is formulated for the multi-UAV assisted mobile edge computing system. To obtain the joint strategy of trajectory design, task allocation, and power management, a cooperative multi-agent deep reinforcement learning framework is investigated. Considering the high-dimensional continuous action space, the twin delayed deep deterministic policy gradient algorithm is exploited. The evaluation results demonstrate that our multi-UAV multi-EC task offloading method can achieve better performance compared with the other optimization approaches.

Index Terms—Mobile edge computing, UAV networks, task offloading, cooperative offloading, deep reinforcement learning.

I. Introduction

WITH the development of mobile applications (i.e., automatic navigation, infrastructures monitoring, online games, etc.), more and more mobile application tasks become computation-intensive and delay-sensitive, especially in Internet-of-Things [1], [2]. However, these tasks may impose a great challenge on user equipment (UE), which have a limited computation and battery capabilities. To address these challenges, multi-access edge computing (MEC) [3] is considered to be an extension of cloud computing for data computation and communication in mobile networks. Instead of transmitting the computation requests to the central computing stations, MEC places servers at the mobile network edges (i.e., cellular base stations or WiFi access points) with computation and storage resources. It will be more convenient for servers to offer computing services to deal with intensive computation tasks of UEs, leading to lower service latency and better service quality.

Nevertheless, there is still a challenging issue for UEs to obtain the reliable computation services. On one hand, many UEs execute computation-intensive applications in remote or mountainous areas, where communication infrastructures are always distributed sparsely with poor communication conditions and uncertain MEC environments [4]. On the other hand, there may be massive users to require computation-intensive services simultaneously. With limited storage and computation resources, it will be difficult for MEC servers to offer their computation services, especially in hotspot areas [5]. Fortunately, due to the advantages of flexible deployment and large coverage, unmanned aerial vehicles (UAVs) have been applied to assist MEC systems to execute the computation-intensive tasks [6], [7]. By establishing LoS links with ground UEs, the UAVs can act as the "flying MEC servers" to offer considerable offloading services with low network overhead and execution latency.

Although prior works in the UAV-assisted networks mainly focus on communication aspects [8], [9], there is still some research on UAVs-assisted MEC systems, such as trajectory design [10]–[12], resource management [13]–[15], computation offloading [16]–[18]. However, most existing works considered the scenario of single UAV for computation offloading.

Due to the limited computation and energy capacities, one UAV may provide the quite limited performance of task offloading. It will be more suitable to investigate the scenario with multiple UAVs and multiple edge clouds (ECs) collaboratively. Moreover, almost all the existing studies focused on the static UAV-assisted MEC systems with fixed UEs. Practically, UEs always move around during computing, which makes it difficult to obtain the optimal strategy. Furthermore, since UAVs need to fly to certain areas to offer their offloading helps from different taking off points, different trajectories of UAVs may cause various channel qualities, leading to different communication delays and energy consumptions. The allocated amounts of computation tasks of UAVs may also influence on computation delay and energy with the limited on-board resource. Thus, it will be necessary to jointly consider the issues of trajectories, computation tasks allocation, and communication resource management to obtain the minimum execution delays and energy consumptions. Unfortunately, with the non-convex nature and non-stationary environment, it may be difficult to obtain the global optimal policy without exact and complete information about the environment.

Recently, some research has tried to deal with the joint optimization issue in UAV-assisted MEC systems via reinforcement learning (RL) [19]–[22]. By exploring the dynamic MEC environments, RL can make intelligent decision under uncertainty. In [23], a hierarchical game-theoretic and RL framework was proposed for computational offloading with multiple service providers. Zhu *et al.* studied the learning-based computation offloading mechanism to minimize the average mission response time [24]. In [25], the authors presented a deep RL (DRL) approach to plan flying path for UAV-mounted MEC systems. In [26], a DRL approach was investigated to minimize energy consumption by optimizing the dynamic trajectory control strategy. In [27], Ren *et al.* proposed an efficient scheduling strategy via hierarchical RL for the large-scale UAV-assisted MEC. In [28], a multi-agent DRL method was studied for trajectory planning in multi-UAV assisted MEC systems. However, if the number of UAVs or UEs or ECs is large, the state (i.e., UAVs' positions) and action (i.e., UAVs' movements, tasks allocation, and resource management) may grow exponentially, leading to the poor convergence efficiency.

To deal with the above challenges, this paper investigates the collaborative UAV-assisted MEC systems, where multiple UAVs and multiple ECs are designed to offload computation tasks of UEs. The UEs' tasks offloading optimization problem is formulated to obtain the minimum execution delays and energy consumptions. A cooperative multi-agent DRL (MADRL) approach is proposed to obtain the trajectories, computation task allocation, and communication resource management at UAVs. The major contributions of our work are the following:

- We investigate a collaborative task offloading strategy in the multi-UAV multi-EC MEC systems, where UAVs and ECs offload computation tasks of UEs collaboratively. Cooperative MADRL method for this scenario has never been investigated. The task offloading optimization problem is formulated to obtain the minimum total system
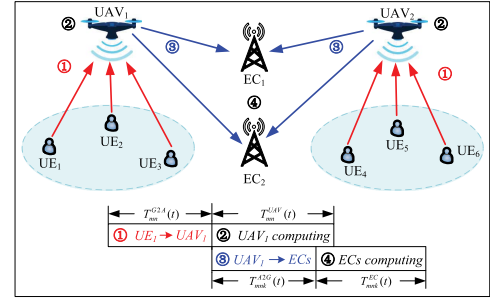

Fig. 1. Multi-UAV assisted MEC system with M UEs, N UAVs, and K ECs.

cost by jointly designing the trajectories, computation task allocation, and communication resource management of UAVs.

- We formulate the highly complex non-convex optimization problem as an MDP, which is then solved by a novel cooperative MADRL framework with each UAV acting as an agent. Considering the high-dimensional continuous action space, the TD3 algorithm is designed to find the efficient UAVs' movements, task offloading allocation, and communication resource management based on dynamic MEC environments.

- We conduct numerical simulations and demonstrate that the proposed collaborative UAV-EC offloading scheme outperforms other optimization approaches, especially in terms of adaptability to UEs' mobility, robustness to the change of communication and computation resources, and flexibility to the dynamicity of computation tasks.

The rest of this paper is organized as follows. In Section II, we provide system model and problem formulation. Section III proposes MADRL framework to address task offloading issues. We present simulation results in Section IV and conclusion in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Fig. 1 presents the multi-UAV assisted MEC system with M UEs, N UAVs, and a set of K ECs. Each UE m needs to periodically handle computation-intensive tasks $W_m = (D_m, C_m, \lambda_m)$, where D_m is the size of task data, C_m is the number of CPU cycles, and λ_m is the arrival rate of the tasks. Considering the limited computation capacities, UEs cannot perform local computing. Then, UAVs are deployed to offer MEC services to ground UEs. Practically, the UAVs are planned carefully to avoid overlapping trajectories to conserve energy and avoid collision. Therefore, we assume that each UAV is deployed to offer MEC services for ground UEs within one corresponding sub-area and that there are no overlaps between each sub-area. Moreover, it is assumed that all UAVs are connected to a single ground cloud server via the wireless backhaul links.

In the multi-UAV assisted MEC system, limited by factors such as size, weight, and power, the UAVs can provide limited computation and communication resources. Unlike UAVs, the ECs always consists of MEC servers with more resources of computation and communication. Therefore, this paper considers four main components of task offloading
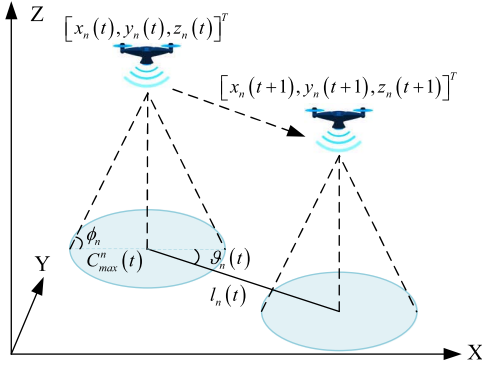
Fig. 2. Multi-UAV assisted MEC system with M UEs, N UAVs, and K ECs.

process: 1) ground-to-air (G2A) transmission from UEs to UAVs; 2) computation at the UAVs; 3) Air-to-ground (A2G) transmission from UAVs to ECs; and 4) computation at the ECs.

A. UAVs Movement

As shown in Fig. 2, the 3D coordinate of UAV n is denoted as $\boldsymbol{\omega}_n(t) = [x_n(t), y_n(t), z_n(t)]^T$, where $x_n(t)$, $y_n(t)$, and $z_n(t)$ are the X, Y, Z coordinates of UAV n at time t, respectively. Denote $v_n(t) = [x_n(t), y_n(t)]^T$ as the 2D coordinate of UAV n. Assume that UAV n flies the distance $l_n(t)$ with the angle direction $\vartheta_n(t) \in [0, 2\pi)$ in the horizontal flight. Then, we have

$$x_n(t+1) = x_n(t) + l_n(t)cos(\vartheta_n(t)), \quad (1)$$
$$y_n(t+1) = y_n(t) + l_n(t)sin(\vartheta_n(t)). \quad (2)$$

Additionally, according to [26], [29], assume that UAV n has a maximum elevation angle ϕ_n. Then, at time t, the maximum horizontal radius of UAV n $C_{max}^n(t)$ can be obtained

$$C_{max}^n(t) = z_n(t)tan(\phi_n). \quad (3)$$

Due to its limited horizontal-flight and vertical-flight speeds, UAVs always have limited flight distances, which can be given by

$$Z_{min} \leq z_n(t) \leq Z_{max}, \quad (4)$$
$$l_n(t) = \|v_n(t+1) - v_n(t)\| \leq L_{max}^h, \quad (5)$$
$$\Delta z_n(t) = |z_n(t+1) - z_n(t)| \leq L_{max}^v, \quad (6)$$

where Z_{min} and Z_{max} denote the minimum and maximum heights, respectively; $\Delta z_n(t)$ denotes the vertical travel distance; L_{max}^h and L_{max}^v are the maximum horizontal and vertical distances of the UAVs, respectively.

Moreover, in order to guarantee that UAVs move within the served rectangle-shaped area, the following move constraint must be satisfied, that is,

$$0 \leq x_n(t) \leq X_{max}, \quad (7)$$
$$0 \leq y_n(t) \leq Y_{max}, \quad (8)$$

where X_{max} and Y_{max} are the side lengths of the rectangle-shaped area, respectively.

To ensure that the coverage of arbitrary two UAVs cannot overlap with each other, the following overlapping constraint must be satisfied,

$$\|v_n(t) - v_j(t)\| \geq [C_{max}^n(t) + C_{max}^j(t)], \quad \forall n, j, n \neq j. \quad (9)$$

Similarly, to avoid collision between any two UAVs, the distance of UAVs should be no less than a minimum distance D_{min}. Then, we have the following collision constraint

$$\|\boldsymbol{\omega}_n(t) - \boldsymbol{\omega}_j(t)\| \geq D_{min}, \quad \forall n, j, n \neq j. \quad (10)$$

Note that if UEs are located within the coverage of certain UAV, the UEs will be served by the same UAV. Let UAV n serve $M_n(t)$ UEs at time t. We denote $\rho_m^n(t)$ as a binary *service-association* vector. $\rho_m^n(t) = 1$ when UE m is served by UAV n, and $\rho_m^n(t) = 0$ otherwise. Assume that each UE can only be served by at most one UAV at any time. That is,
$$\sum_{n=1}^{N} \rho_m^n(t) \leq 1, \forall m, \forall t.$$

B. G2A Transmission From UEs to UAVs

Here, we denote $\boldsymbol{\omega}_m(t) = [x_m(t), y_m(t), 0]^T$ as the location of UE m, where $x_m(t)$ and $y_m(t)$ are the X and Y coordinates, respectively. The distance between UAV n and UE m can be given by

$$d_{mn}(t) = \|\boldsymbol{\omega}_n(t) - \boldsymbol{\omega}_m(t)\|. \quad (11)$$

Similar to [24], [26], [27], assume that the ground UEs can communicate with their serving UAV via the orthogonal frequency-division multiple access. Then, the interference between different UEs in the coverage of each UAV can be ignored. Due to the high altitude of UAVs, the LoS channel is much more dominant than other channel impairments such as shadowing or small-scale fading. The Doppler shift caused by the high mobility of UAVs can be assumed to be perfectly compensated at the UEs [15]. Then, the G2A channel gain between UE m and UAV n can be denoted by the free-space path loss model, which is given by

$$h_{mn}(t) = \frac{g_0}{[d_{mn}(t)]^2}, \quad (12)$$

where g_0 denotes the power gain with the reference distance of 1 meter.

During the task offloading process, the uplink bandwidth B_u is assumed to be allocated to each UE equally. Then, the G2A data rate between UE m and UAV n is

$$R_{mn}(t) = \frac{B_u}{M_n(t)}\log_2\left[1 + \frac{h_{mn}(t)P_m}{\sigma_u^2}\right], \quad (13)$$

where P_m is the transmit power of UE m, σ_u^2 is the additive white Gaussian noise power at each UAV.

Considering that all tasks are offloaded to UAVs through the G2A channel, the G2A transmission delay between UE m and UAV n can be defined as the task data size D_m divided by the corresponding transmission data rate $R_{mn}(t)$, that is,

$$T_{mn}^{G2A}(t) = \frac{D_m}{R_{mn}(t)}. \quad (14)$$

Similarly, the G2A transmission energy consumption between UE m and UAV n can be defined as

$$E_{mn}^{G2A}(t) = P_n^r T_{mn}^{G2A}(t) = \frac{D_m P_n^r}{R_{mn}(t)}, \tag{15}$$

where P_n^r is the receiving power of UAV n.

C. Computation at the UAVs

After receiving the entire input data from UEs, each UAV decides how much tasks computed locally. We define $\gamma_{mk}^n(t) \in [0,1]$ and $\gamma_{m0}^n(t) \in [0,1]$ as the proportion of tasks of UE m executed at EC k and UAV n, respectively. The computation delay of UAV n handling the task of UE m is given by

$$T_{mn}^{UAV}(t) = \frac{\gamma_{m0}^n(t) D_m C_m}{f_{mn}(t)}, \tag{16}$$

where $f_{mn}(t)$ denotes the computation resource allocated from UAV n to UE m. For simplicity, the UAV's computation resource F_u is allocated to each served UE equally, that is $f_{mn}(t) = F_u/M_n(t)$. If $\gamma_{m0}^n(t) = 0$, all tasks of UE m are processed at ECs, while $\gamma_{m0}^n(t) = 1$, all tasks of UE m are computed at UAV n.

Then, considering the computation time $T_{mn}^{UAV}(t)$ and the power consumption [30], the energy consumption of UAV n handling the task of UE m can be obtained as

$$E_{mn}^{UAV}(t) = \kappa \left[f_{mn}(t)\right]^3 T_{mn}^{UAV}(t), \tag{17}$$

where $\kappa \geq 0$ is the effective switched capacitance.

D. A2G Transmission From UAVs to ECs

Here, $\boldsymbol{\omega}_k = [x_k, y_k, 0]^T$ is the fixed location of EC k, where x_k and y_k are the coordinates of EC k, respectively. Then, the distance between UAV n and EC k can be given by

$$d_{kn}(t) = \|\boldsymbol{\omega}_n(t) - \boldsymbol{\omega}_k\|. \tag{18}$$

Considering that certain UAV offloads some tasks to ECs for further computing, the A2G channel gain between UAV n and EC k can be defined as

$$h_{nk}(t) = \frac{g_0}{[d_{kn}(t)]^2}. \tag{19}$$

Then, the transmission data rate between UAV n and EC k is given by

$$R_{nk}(t) = B_k \log_2 \left[1 + \frac{h_{nk}(t) P_n^t(t)}{\sigma_e^2}\right], \tag{20}$$

where B_k is the bandwidth pre-assigned to EC k and $0 \leq P_n^t(t) \leq P_{max}$ denotes the transmit power of UAV n at time t, P_{max} is the maximum transmission power of each UAV, and σ_e^2 is the additive white Gaussian noise power at each EC.

Considering the task data size of ECs and the transmission data rate $R_{mn}(t)$, the A2G transmission delay between UE m and EC k through UAV n can be defined as

$$T_{mnk}^{A2G}(t) = \frac{\gamma_{mk}^n(t) D_m}{R_{nk}(t)}. \tag{21}$$

Similarly, the A2G transmission energy consumption between UE m and EC k through UAV n can be obtained as

$$E_{mnk}^{A2G}(t) = P_n^t T_{mnk}^{A2G}(t) = \frac{\gamma_{mk}^n(t) D_m P_n^t}{R_{nk}(t)}. \tag{22}$$

E. Computation at the ECs

ECs begin to handle the computation tasks when obtaining the task data from UAVs. Considering the task proportion $\gamma_{mk}^n(t)$, the computation delay at EC k can be given by

$$T_{mnk}^{EC}(t) = \frac{\gamma_{mk}^n(t) D_m C_m}{f_{mk}(t)}, \tag{23}$$

where $f_{mk}(t)$ denotes the computation resource allocated to UE m. Here, the total computation resource F_k^e of EC k is allocated to each UE equally, that is $f_{mk}(t) = F_k^e/M$.

F. Problem Formulation

When the computation tasks of all UEs are completed, the energy consumption of UAV n can be obtained as

$$E_n(t) = \sum_{m=1}^M \rho_m^n(t) \lambda_m \left[E_{mn}^{G2A}(t) + E_{mn}^{UAV}(t) + E_{mnk}^{A2G}(t)\right], \tag{24}$$

where λ_m denotes the arrival rate of tasks.

Moreover, considering that the communication and computation modules are often separated at the UAVs, the computation at the UAVs can be processed simultaneously with the task transmission to ECs. Then, the execution delay of UAV n is given by

$$T_n(t) = \sum_{m=1}^M \rho_m^n(t)$$
$$\times \left[T_{mn}^{G2A}(t) + \max_k \{T_{mn}^{UAV}(t), T_{mnk}^{A2G}(t) + T_{mnk}^{EC}(t)\}\right]. \tag{25}$$

Then, similar to [13], [23], we denote the weighted sum of energy consumption $E_n(t)$ and execution delay $T_n(t)$ as the system cost of UAV n, that is,

$$U_n(t) = w_1 E_n(t) + w_2 T_n(t), \tag{26}$$

where w_1 and w_2 are the weights to indicate the different significance on energy consumption and execution delay, respectively. $w_1 \geq w_2$ indicates the energy-saving scenarios while $w_1 < w_2$ is for the delay-sensitive cases.

Thus, by jointly optimizing the UAVs' position $\boldsymbol{\omega}_n(t)$, task partition ratios ($\gamma_{m0}^n(t)$ and $\gamma_{mk}^n(t)$), and transmit power ($P_n^t(t)$), the task offloading optimization problem can be designed to minimize the total system cost, which is formulated as

$$\min_{\substack{\boldsymbol{\omega}_n(t), \gamma_{m0}^n(t), \\ \gamma_{mk}^n(t), P_n^t(t)}} \sum_{n=1}^N U_n(t), \tag{27a}$$

$$s.t. \quad 0 \leq \gamma_{mk}^n(t) \leq 1, \tag{27b}$$

$$0 \leq \gamma_{m0}^n(t) \leq 1, \tag{27c}$$

$$\gamma_{m0}^n(t) + \sum_k \gamma_{mk}^n(t) = 1, \forall n \qquad (27d)$$

$$0 \le P_n^t(t) \le P_{max}, \qquad (27e)$$

$$(4) - (10), \qquad (27f)$$

where (27b), (27c), and (27d) denote the offloading tasks constraints of UEs, (27e) is the constraint about the transmit power of UAVs, (4)-(10) describe the movement constraints of UAVs.

Generally, it is challenging to solve the non-convex optimization problem (27). Certain unknown variables (i.e., UEs' location and channel condition) may influence the energy consumption and execution delay, especially in the dynamic network induced by UAVs' mobility. Moreover, considering the decision with the large solution space, it will be intractable to obtain the optimal strategy by traditional optimization schemes. To address these challenges, an RL method will be investigated to learn the near-optimal policy with little environment information in the next section.

III. MADRL FOR TASK OFFLOADING OPTIMIZATION PROBLEM

Here, we first re-model the above problem as a multi-agent extension of the MDP, which is then solved by an MADRL method.

A. MDP Formulation

In UAV-assisted MEC systems, UAVs determine their position, transmit power and task partition ratios to obtain the minimum total system cost. Considering that UAVs' actions (i.e., UAVs' movements) may influence the environmental state, the total system cost is determined by the current state of system environment and the joint actions of all UAVs. Moreover, the former state and previous actions jointly trigger the system environment into a new stochastic state [31]. In this case, the task offloading optimization issue (27) can be formulated as a multi-agent Markov decision process (MDP) $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_n\}_{n \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}_n\}_{n \in \mathcal{N}}, \delta \rangle$. $\mathcal{N}$ is the agent set, $\mathcal{S}$ is the state set of all agents, $\mathcal{A}_n$ is the action space of agent n, $\mathcal{P}$ represents the state transition probability, $\mathcal{R}_n$ is the reward function of agent n, and $\delta \in [0, 1]$ denotes the discount factor.

1) Agent Set $\mathcal{N}$: Each UAV acts as an agent to learn its scheme of position, transmission power and task partition ratios and obtain the minimum total system cost. Thus, $\mathcal{N} = \{1, \ldots, N\}$.

2) State Space $\mathcal{S}$: According to the task offloading optimization problem, the state $s(t)$ is composed of the 3D coordinate positions of UAVs, that is,

$$s(t) = \{\boldsymbol{\omega}_1(t), \boldsymbol{\omega}_2(t) \ldots, \boldsymbol{\omega}_N(t)\}. \qquad (28)$$

3) Action Space $\mathcal{A}_n$: Since each UAV is required to determine its movements (horizontal fly distance $l_n(t)$, horizontal direction angle $\vartheta_n(t)$, and vertical fly distance $\Delta z_n(t)$), transmission power $P_n^t(t)$ and task partition ratios $\gamma_{mk}^n(t)$, the action space $a_n(t)$ of UAV n can be given by

$$a_n(t) = \{l_n(t), \vartheta_n(t), \Delta z_n(t), P_n^t(t), \gamma_{mk}^n(t), \forall k\}. \qquad (29)$$

According to the constraints of the minimum optimization problem (27), we can have the value ranges of each element in $a_n(t)$, that is, $l_n(t) \in [0, L_{max}^h]$, $\vartheta_i(t) \in [0, 2\pi)$, $\Delta z_n(t) \in [-L_{max}^v, L_{max}^v]$, $P_n^t(t) \in [0, P_{max}]$, and $\gamma_{mk}^n(t) \in [0, 1]$. Also, we can observe that the action space $\mathcal{A}_n$ of UAV n is a continuous set. Moreover, with the number of UEs and ECs increasing, the size of action spaces exponentially increases.

4) Reward Function $\mathcal{R}_n$: To solve the formulated task offloading optimization problem (27), the N agents should cooperatively minimize the total system cost while satisfying certain constraints, such as the overlapping and collision constraints. Then, the reward function $\mathcal{R}_n(t)$ of UAV n is defined as the negative of the system cost $U_n(t)$ if all constraints are satisfied. Otherwise, if certain constraints are not satisfied, there will be the corresponding penalties in the reward function $\mathcal{R}_n(t)$. Moreover, to guarantee UAVs provide computing service to all UEs, the coverage constraint of UAVs should be satisfied. If certain UE is beyond the UAVs' coverage, there will be a penalty in the reward function. Thus, based on the above consideration, the reward function of UAV n is given by

$$\mathcal{R}_n(t) = \begin{cases} -U_n(t), & \text{if satisfying constraints,} \\ -\eta_1 - \eta_2 - \eta_3 & \\ [M - \sum_{n=1}^N M_n(t)], & \text{otherwise,} \end{cases} \qquad (30)$$

where η_1, η_2, and η_3 denote the penalties related with the overlapping constraint (9), the collision constraint (10), and the coverage constraint, respectively. If the horizontal distance of any two UAVs does not meet the overlapping constraints (9), each of the two UAVs will experience a penalty η_1. Moreover, if the distance between any two UAVs does not satisfy the collision constraints (10), there will be a penalty η_2 in the reward functions of the two UAVs. Finally, when any UEs are not covered by UAVs, all UAVs will incur the penalty $\eta_3[M - \sum_{n=1}^N M_n(t)]$.

B. Multi-Agent DRL Algorithm

To solve the above multi-agent MDP, considering the high-dimensional continuous action space of the task offloading optimization problem, the multi-agent TD3 (MATD3) approach is proposed, shown in Fig. 3. Each UAV adopts a TD3 algorithm [32], which comprises one actor network with weights μ_n and two critic networks with weights θ_n^1 and θ_n^2. With the two critic networks, each UAV can deal with the overestimation problem of the Q-values in the one-critic framework. In addition, to improve the learning stability, the *target actor network* with weights μ_n' and *target critic networks* with weights $\{\theta_n^{i'}\}_{i=1,2}$ are adopted.

Different from other multi-agent RL algorithms, where each agent tries to maximize its reward function $\mathcal{R}_n(t)$, a cooperative multi-agent RL architecture is adopted to achieve the maximum expected discounted reward with the sum reward of all UAVs, which is defined as

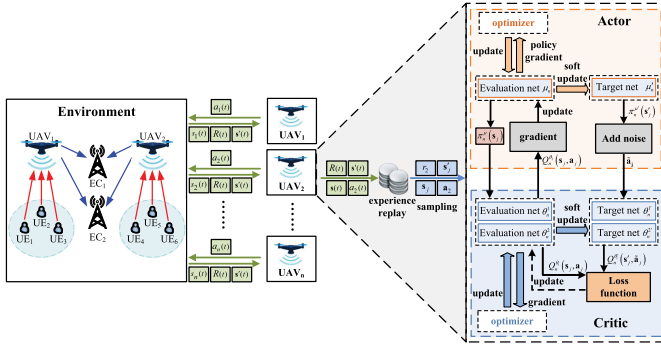$$\mathcal{R}(t) = \sum_{n=1}^N \mathcal{R}_n(t). \qquad (31)$$

Fig. 3. The MATD3 framework in Multi-UAV assisted MEC system.

Moreover, considering the non-stationarity of the network environment, to guarantee convergence, the strategy based on centralized training and decentralized execution is adopted [33]. Specifically, in the centralized training stage, the *evaluation critic networks* and *target critic networks* are designed to obtain a global view and deployed in the ground cloud server. *Evaluation critic networks* are at the state $s_n(t)$ and action $a_n(t)$ of other agents via communication. Then, all UAVs utilize global state $s(t)$ and joint actions $a(t) = \{a_1(t), a_2(t), \ldots, a_N(t)\}$ so that the policy of other UAVs can be estimated and Q-function $Q_n^{\theta_i}(s(t), a(t))$ can be obtained for all UAVs. Also, based on the estimated policy of other UAVs, each UAV can adjust the local actor policy $\pi_n^\mu : \mathcal{S} \to \mathcal{A}_n$ to achieve the global optimal policy $\pi^\mu = \{\pi_1^\mu, \pi_2^\mu, \ldots, \pi_N^\mu\}$. Then, the network environment is considered to be stationary to each UAV during the centralized offline training stage. During the decentralized execution stage, the critic networks of UAVs are no longer required, and the weights of the actor networks are fixed. Each UAV executes its policy using the trained *evaluation actor network* $\pi_n^\mu(s(t))$ with the learned weight μ_n, based only on its local state information $s_n(t)$. Considering that the UAVs do not communicate with each other, this will greatly reduce communication overhead and enable its scalability to multi-UAV assisted MEC system.

The MATD3 approach for the task offloading optimization problem is summarized in Algorithm 1. We first initialize weights of the six neural networks and replay buffer $\mathcal{B}$ in all UAVs. In each episode, each UAV selects action based on its *evaluation actor network* $\pi_n^\mu(s(t))$ with random noise ξ. According to the action taken above, all UAVs execute the three-dimensional movements (horizontal fly distance $l_n(t)$, horizontal direction angle $\vartheta_n(t)$, and vertical fly distance $\Delta z_n(t)$), transmission power $P_n^t(t)$ and task partition ratios $\gamma_{mk}^n(t)$. When moving out of the range of the served area, certain UAV may fly with a random horizontal angle. Moreover, if moving beyond the limit vertical height, UAVs keep flying at the boundary height (Z_{min} or Z_{max}). Also, when covering certain hotspots, the corresponding UAVs keep their 3D positions and only change their transmission power and task partition ratios. By executing the above actions, all UAVs will receive the next state $s'(t)$, joint action $a(t)$ and immediate reward $\mathcal{R}(t)$.

To stabilize training process and improve sample efficiency, each UAV stores the current experience $(s(t), s'(t), a(t), \mathcal{R}(t)$

Algorithm 1 MATD3 Approach for Task Offloading Problem

- Initialize each UAV's actor networks with weights μ_n and μ_n', respectively.
- Initialize each UAV's critic networks with weights $\{\theta_n^i\}_{i=1,2}$ and $\{\theta_n^{i'}\}_{i=1,2}$, respectively.
- Initialize each UAV's replay buffer $\mathcal{B}$.
- **for each** *episode* **do**
- Initialize the state $s(t)$ and $t = 1$.
- **while** $t < T_p$ **do**
- Each UAV selects action $a_n(t) = \pi_n^\mu(s_n(t)) + \xi$.
- All UAVs set their movements, transmission power and task partition ratios according to the joint action $a(t)$.
- All UAVs obtain the reward $\mathcal{R}(t)$ and the next state $s(t+1)$ and joint action $a(t)$ via communication.
- Store $(s(t), s'(t), a(t), \mathcal{R}(t))$ in $\mathcal{B}$ for all $n \in \mathcal{N}$.
- $s(t) \leftarrow s'(t)$.
- **for** $n = 1, \ldots, N$ **do**
- Sample a random mini-batch of (s_j, s_j', a_j, r_j) for all UAVs from $\mathcal{B}$.
- Update weights $\{\theta_n^i\}_{i=1,2}$ of *evaluation critic networks* by minimizing loss function $L(\theta_n^i)$ in (35).
- **If** $t \bmod d$ **then**
- Update weights μ_n of *evaluation actor network* with (32).
- Update weights of the three target networks in (37).
- **end If**
- **end for**
- **end for**
- **end for**

---

in the *replay buffer* $\mathcal{B}$ with size M_r [34]. For each UAV, sample a random mini-batch of $\{s_j, s_j', a_j, r_j\}$ with size M_b from $\mathcal{B}$. Then, by feeding s_j into the *evaluation actor network* to generate the policy $\pi_n^\mu(s_j)$, each UAV can update the weight of *evaluation actor network* using policy gradient strategy [35], that is,

$$\nabla_{\mu_n} J(\mu_n)$$
$$= \frac{1}{M_b} \sum_{j=1}^{M_b} \nabla_{\mu_n} \pi_n^\mu(s_n^j) \nabla_{a_n} Q_n^{\theta_1}(s_j, a_1^j, a_n, \ldots, a_N^j)|_{a_n = \pi_n^\mu(s_n^j)}.$$
(32)

Moreover, to prevent over-fitting on the narrow peaks of Q-values, the random noise $\tilde{\epsilon}$ is added to *target actor network*, which can achieve a smoother state-action value estimation. The modified target actions $\tilde{a}_j$ is given by

$$\tilde{a}_j = \pi_n^{\mu'}(s_j') + \tilde{\epsilon}, \tag{33}$$

where $\tilde{\epsilon} \sim clip\left(N\left(0, \hat{\sigma}^2\right), -1, 1\right)$ is the noise with mean 0 and standard deviation $\hat{\sigma}$ and clipped. Then, target values y_j can be obtained as

$$y_j = r_j + \delta \min_{i=1,2} Q_n^{\theta_i'}\left(s_j', \tilde{a}_j\right), \quad i = 1, 2. \tag{34}$$

Then, based on the policy $\pi_n^\mu(s_j)$, the two *evaluation critic network* will concurrently obtain the two Q-values

$Q_n^{\theta_1}(\boldsymbol{s}_j, \pi_n^\mu(\boldsymbol{s}_j))$ and $Q_n^{\theta_2}(\boldsymbol{s}_j, \pi_n^\mu(\boldsymbol{s}_j))$ by minimizing the loss function $L(\theta_n^i)$, which is defined as

$$L(\theta_n^i) = \frac{1}{M_b} \sum_{j=1}^{M_b} [y_j - Q_n^{\theta_i}(\boldsymbol{s}_j, \boldsymbol{a}_j)]^2, \quad i = 1, 2. \quad (35)$$

Next, according to (32) and (35), each UAV can update the weights of the three evaluation networks using the following equations:

$$\mu_n \leftarrow \mu_n - \lambda \nabla_{\mu_n} J(\mu_n),$$
$$\theta_n^i \leftarrow \theta_n^i - \lambda \nabla_{\theta_n^i} L(\theta_n^i), \quad i = 1, 2, \quad (36)$$

where λ denotes the learning rate. To reduce errors resulting from temporal difference learning, each UAV updates the weights of *evaluation actor network* at a lower frequency than that of *evaluation critic networks*. Here, each UAV chooses to update the *evaluation actor network* every d time-steps.

Thus, in order to stabilize the training process, by copying the weights of corresponding *evaluation networks*, each UAV updates the weights of the three target networks every d time-steps through

$$\mu_n' = \tau \mu_n + (1 - \tau)\mu_n',$$
$$\theta_n^{i'} = \tau \theta_n^i + (1 - \tau)\theta_n^{i'}, \quad i = 1, 2, \quad (37)$$

where τ denotes the updating rate.

Finally, we discuss the complexity analysis of our proposed MATD3 algorithm. As for the communication complexity, in the centralized training procedure, the ground cloud server needs to frequently communicate with UAVs to obtain the state about the 3D coordinate positions of UAVs. Since the total dimension of UAVs' positions is $3N$, the communication complexity is $\mathcal{O}(N)$. While in the decentralized execution process, each UAV obtains its action locally, leading to no communication between UAVs. Hence, the overall communication complexity of our proposed MATD3 algorithm is $\mathcal{O}(N)$.

Moreover, in the centralized training process, each UAV estimates the Q-function values with critic networks, where the sizes of the inputs and outputs are $3N + N(4 + MK)$ and 1, respectively. In addition, each UAV determines its action based on its actor networks with the input size $3N$ and the output size $N(4 + MK)$. While in the decentralized execution procedure, each UAV obtains its action from its actor networks with the input size 3 and the output size $4 + MK$. According to [36], given the fully-connected neural network with fixed numbers of hidden layers and neurons, the computational complexity of the back-propagation algorithm is proportional to the product of the input size and the output size. For the critic network, the centralized training backprop complexity is $\mathcal{O}(NMK)$ while for the actor network, the decentralized execution procedure is $\mathcal{O}(N^2 + NMK)$. Therefore, the overall complexity is $\mathcal{O}(N^2 + NMK)$.

IV. PERFORMANCE EVALUATION

In this section, numerical experiments are conducted to evaluate the performance of our proposed MATD3. Here, a multi-UAV assisted MEC system is considered with 2 fixed
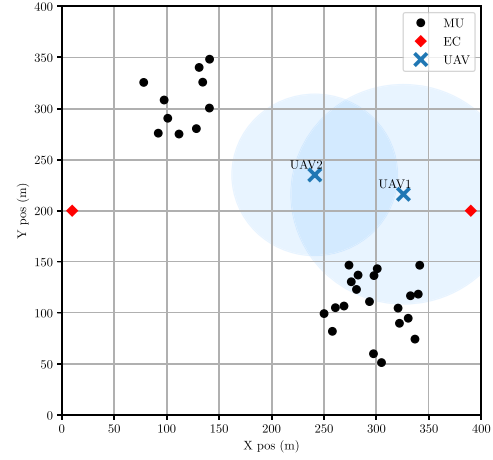

Fig. 4. Locations of 30 UEs, 2 ECs and 2 UAVs in multi-UAV assisted MEC system.

TABLE I
NETWORK ENVIRONMENT PARAMETERS

Parameters	Value
Size of input data D_m	[1,5] Mbits
Number of CPU cycles C_m	[100,200] cycles/bit
Arrival rate of tasks λ_m	1 task/sec
Maximum height of UAVs Z_{max}	100 m
Minimum height of UAVs Z_{min}	50 m
Minimum horizontal distance L_{max}^h	49 m
Minimum vertical distance L_{max}^v	12 m
Minimum distance of UAVs D_{min}	100 m
elevation angle ϕ_n	42.44° [29]
Path loss exponent g_0	-50dB
Uplink channel bandwidth B_u	10 MHz
Bandwidth preassigned to ECs B_k	0.5 MHz
Maximum transmit power of UAVs P_{max}	5 W
Transmit power of UE P_m	0.1 W
Receiving power of UAVs P_n^r	0.1 W
Computation resource F_u	3 GHz
Computation resource F_e^k	[6,9] GHz
Effective switched capacitance κ	10^{-28}
Noise power σ_u^2	-100 dBm
Noise power σ_e^2	-100 dBm
Weights w_1 and w_2	1
Penalty cofficient of UAVs' overlapping η_1	5
Penalty of UAVs' collision η_2	5
Penalty of UAVs' coverage η_3	5

ECs in an area of $400 \times 400 \ m^2$. The 30 UEs are randomly distributed within two hotspots, as illustrated in Fig. 4. The two UAVs are randomly located to offer their computing offloading help to the ground UEs. The size of input data D_m is generated randomly within $[2, 10]$, and number of CPU cycles C_m are uniform randomly chosen from $[100, 200]$. The main simulation parameter settings are summarized in Table I. The proposed MATD3 framework has two-hidden-layer neural networks with 400 and 300 neurons. Table II presents the main hyperparameters of the model.

A. Training Efficiency of MATD3 Scheme

In this section, the training performance of our proposed MATD3 optimization method is analyzed. The optimal location and computing task allocation of UAVs are also present in this multi-UAV assisted MEC system. The training curves of our proposed MATD3 optimization method is shown in

TABLE II
HYPERPARAMETERS OF MATD3 MODEL

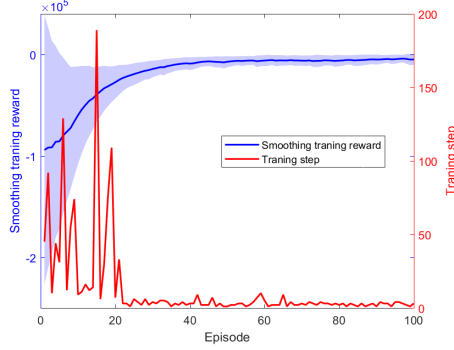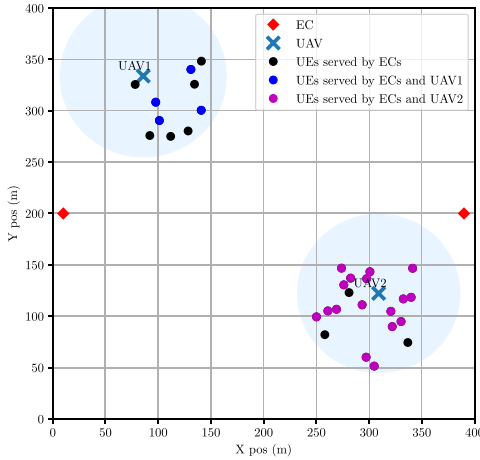| Parameter	Value
Total episodes	100
Time step T_p	200
Updating rate τ	0.005
Mini-batch size M_b	100
Discount rate δ	0.99
Learning rate λ	0.0001
Size of $\mathcal{B}$	100000
Optimizer	AdamOptimizer

Fig. 5. Training curves of MATD3.

Fig. 6. Optimal location of the UAVs.

Fig. 5. The training steps are very large at the beginning of learning. As the number of episodes increases, learning steps converge to less than 10 within 30 episodes, which makes the convergence speed tend to increase. Moreover, as the number of episodes increases, the two UAVs cover the area of served UEs more rapidly. Then, the value of [penalty in the reward function will tend to zero, leading to the convergence of the training reward.

Figures 6 and 7 present the corresponding optimal location and computing task allocation of UAVs, respectively. From Fig. 6, we can observe that each UAV can be almost located in the center of one hotspot, which make UAVs provide computing offloading efficiently. Moreover, the dodgerblue shade represents the coverage of UAVs. The higher the UAV's location is, the larger its coverage becomes. Considering the collision avoiding constraints of UAVs and channel condition, our proposed method can obtain the optimal location of UAVs to provide offloading opportunities for UEs. Furthermore, according to the optimal task splitting ratio allocation strategy,
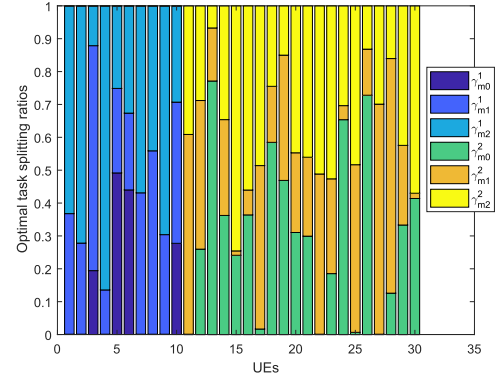

Fig. 7. Optimal task splitting ratios of ECs γ_{mk}^n ($m = 1, 2$ and $n = 1, 2$) for UEs.

certain UEs are served by ECs only, while certain UEs obtain computing offloading services from ECs and UAV.

Then, Fig. 7 presents the optimal task splitting ratio allocation strategy. Since the two UAVs cover the two hotspots respectively, UAV1 (UAV2) does not offer the computing offloading services for the UEs of the hotspot covered by UAV2 (UAV1). In this case, the first 10 UEs are served by UAV1, while the last 20 UEs are served by UAV2. Furthermore, we observe that for UEs (5 and 6) with large size of input tasks, over 40% of the tasks are first processed at UAV1 (i.e., γ_{m0}^1). After that, the remaining tasks will be offloaded to ECs for subsequent executing. While 75% of the last 20 UEs are served by both UAV2 and ECs.

Next, Fig. 8 indicates the effect of the per-device bandwidth on the optimal task partition ratios. The per-device bandwidth B_1 of EC 1 changes from 0.1 to 3 MHz while the other per-device bandwidth B_2 remains 0.5 MHz, and vice versa. With the bandwidth assigned to ECs increasing, the more bandwidth will be assigned to UEs when computing tasks are offloaded from the UAVs to the ECs, leading to the higher downlink data rates. Then, we can achieve the less transmission delay and energy consumption. Moreover, when $B_1 = B_2 = 0.5$, we can achieve the same total system cost in both cases, that is, two lines intersect at the same point. Specifically, EC 1 with the greater weight on total system cost when $B_1 = B_2 = 0.5$ will have a greater impact on reducing total system cost with more assigned bandwidth. While $B_k > 0.5$ with the other fixed 0.5 MHz, the more bandwidth will be assigned to EC k. The case of EC 1 will achieve the less total system cost compared with that of EC 2. However, when $B_k < 0.5$, EC k will receive the less bandwidth. In the case of EC 2, EC 1 has a greater impact on reducing total system cost with $B_1 = 0.5 > B_2$.

Figure 9 plots total system cost with the various computation capacities of and different per-device bandwidths B_1. The computation capacity UAVs F_u increases from 3 to 10 GHz. The bandwidth B_1 of EC 1 increases from 0.5 to 2 MHz with $B_2 = 0.5$. With the growing bandwidth of EC 1, the higher downlink data rate will be obtained, resulting in the less transmission delay, energy consumption and total system cost. Moreover, with the computation capacity UAVs F_u increasing, the more computation resource is allocated UEs, leading to the less computation delay and total system cost.
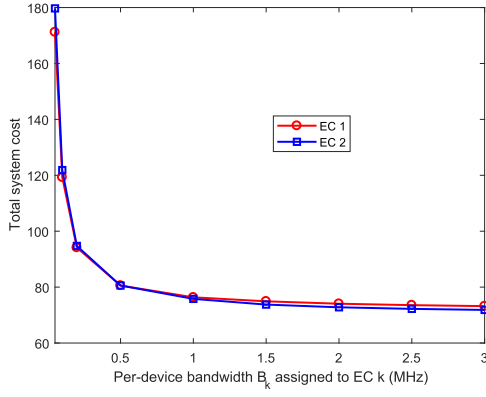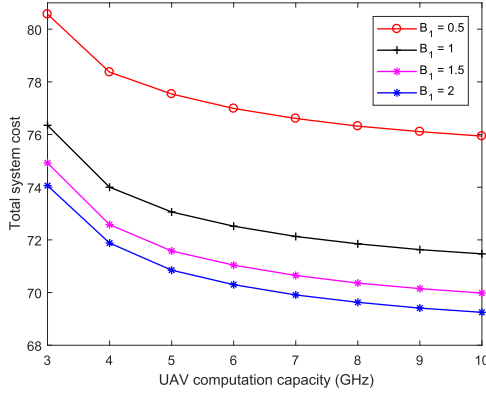
Fig. 8. Total system cost with different per-device bandwidths B_k.

Fig. 9. Total system cost with different computation capacitis of UAVs.

Fig. 10. Total system cost as a function of the UAVs' numbers.

Fig. 11. Energy consumption and execution delay vs weight w_2.

To further analyze the scalability of our proposed MATD3 method, we evaluate the performance with different numbers of UAVs and UEs, as shown in Fig. 10. The UEs are distributed over N hotspots uniformly, with the number M increasing from 30 to 80. From Fig. 10, we can observe that as the numbers of UEs M increases, the more computation tasks are required to be processed, which results in the higher total system cost. Moreover, with the number of UAVs N increasing, there will be more UAVs participating in computation offloading. In the case of the same numbers of UEs and tasks, the greater number of participating UAVs is, the smaller total system cost will be achieved. However, if the numbers of UEs and tasks are so small, it may be not suitable to obtain much more UAVs to participate. For example, when $M = 30$, the performance of two UAVs is almost close to that of three UAVs. Furthermore, as we increase the numbers of UEs M to 80 with $N = 3$, the MATD3 method can still deal with the multi-UAV optimization problem. This confirms the high scalability of the MATD3 strategy with respect to the size of UAVs, state and action spaces.

Fig. 11 depicts the relationship between the energy consumption and execution delay of task offloading problem under the weight parameter w_2. The weight w_2 increases from 0.2 to 1.8 with $w_1 = 1$. As can be seen, a small w_2 puts more weight to the energy consumption. With the weight w_2 increasing, the execution delay is more emphasized and more tasks are offloaded to UAVs, which results in less delay and more energy consumption. However, when w_2 is large enough, the execution delay even does not decrease any more since the
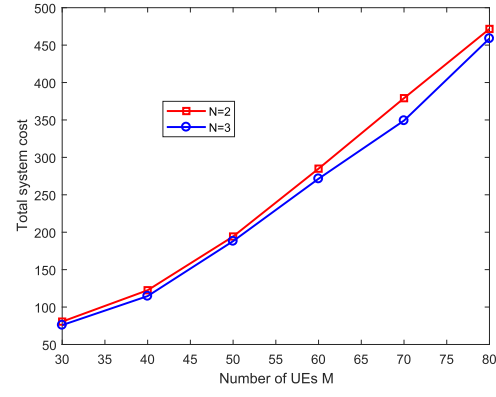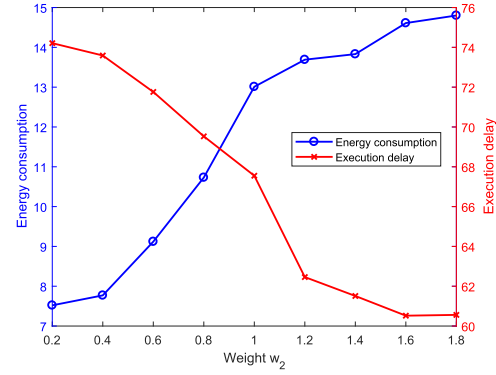
computing capacity that the UAVs can provide is limited and more tasks lead to higher processing delay.

B. Optimization Performance With Various Approaches

In this section, we evaluate the performance with various optimization approaches in both fixed UEs and mobile UEs scenarios. In the mobile UEs scenarios, UEs can walk randomly with a normal distribution movement in each episode.

The MATD3 approach is compared with the following five other optimization methods. The degraded versions of the MATD3 approach with the fixed power scheme ($P_n^t = 3 W$), the fixed hight of UAVs ($z_n = 80m$) are considered, which are denoted as MATD3-FP and MATD3-FH. Multi-agent DDPG (MADDPG) approach is also considered. In the MATD3-EC method, UAVs offload all tasks to ECs for processing directly. In the random scheme, all UAVs randomly select each element of action space within the constraints, that is, the horizontal flying distance $l_n(t) \in [0, L_{max}^h]$, the flying angle $\vartheta_i(t) \in [0, 2\pi]$, the vertical flying distance $\Delta z_n(t) \in [-L_{max}^v, L_{max}^v]$, the transmission power $P_n^t(t) \in [0, P_{max}]$, and the task splitting ratio $\gamma_{mk}^n(t) \in [0, 1]$.

Figure 12 presents total system cost as a function of uplink channel bandwidths B_u with different optimization methods. As the uplink channel bandwidth B_u increases, the higher uplink data rate from UEs is achieved, which leads to the less G2A transmission delay and energy consumption. Then, the total system cost decreases in all optimization methods. Moreover, compared with the case of $N = 2$, more UAVs will participate in computation tasks with the number of UAVs $N = 3$, resulting in the smaller total system cost in all optimization methods.

(a) Fixed UEs with $N = 2$ (b) Fixed UEs with $N = 3$ (c) Mobile UEs with $N = 3$
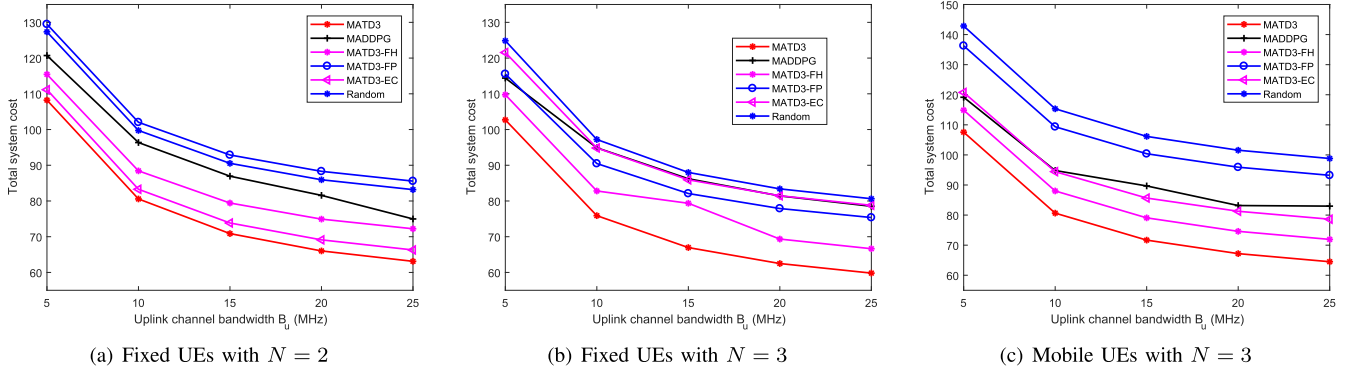
Fig. 12.   Total system cost with different optimization methods and uplink channel bandwidths B_u.

(a) Fixed UEs with $N = 2$ (b) Fixed UEs with $N = 3$ (c) Mobile UEs with $N = 3$
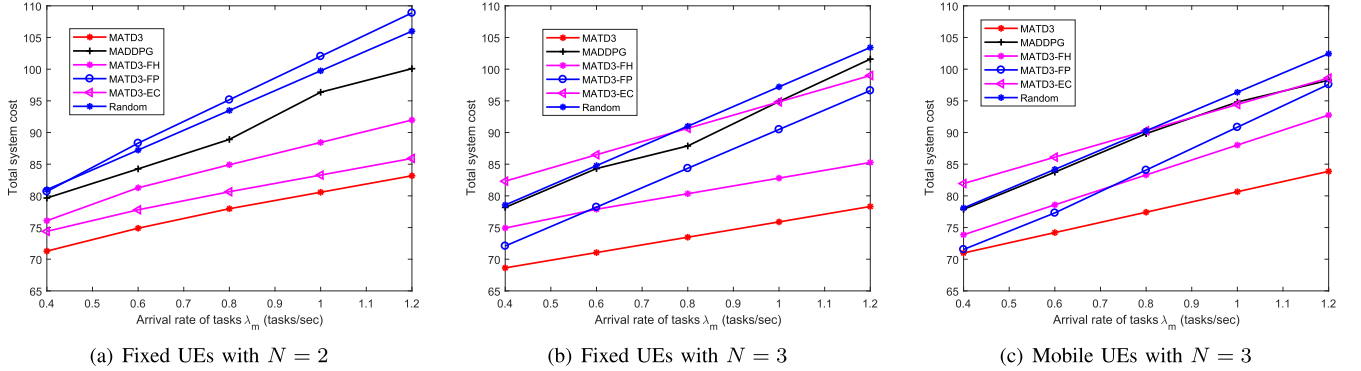
Fig. 13.   Total system cost with different optimization methods and arrival rate of tasks λ_m.

(a) Fixed UEs with $N = 2$ (b) Fixed UEs with $N = 3$ (c) Mobile UEs with $N = 3$
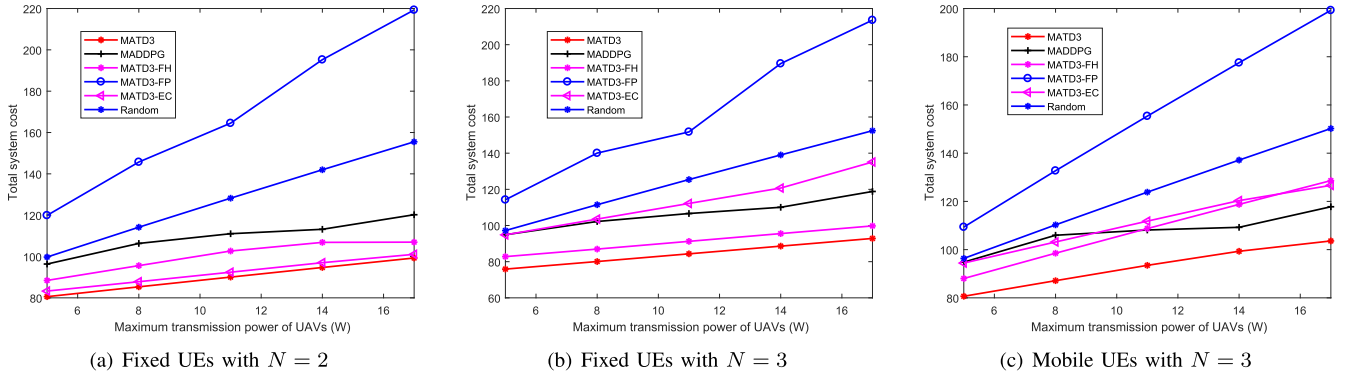
Fig. 14.   Total system cost with different optimization methods and maximum transmission power of UAVs P_{max}.

Furthermore, since the random approach selects a random action to achieve the maximum immediate reward, the large total system cost is experienced with both numbers of UAVs, especially in the mobile UEs scenarios. With the fixed power allocation and fixed height of UAVs, the MATD3-FP and MATD3-FH methods always obtain the larger total system cost compared with our proposed MATD3 approach with both numbers of UAVs. In the case of the MADDPG method, as the number of UAVs increasing, it will be more difficult to obtain the optimal action, leading to the worse performance in the case of the three UAVs. In the MATD3-EC method, without UAVs participating in tasks processing, it always achieves larger total system cost compared with our proposed method. Our MATD3 method can always achieve the smallest total system cost among the six approaches in both fixed UEs and mobile UEs scenarios.

Figure 13 plots total system cost as a function of arrival rate of tasks λ_m with different optimization methods. With the

arrival rate of tasks λ_m increasing, the more total energy needs to be consumed for UAVs, resulting in the higher total system cost decreases in all optimization approaches. In addition, with more UAVs participating in task offloading, the smaller total system cost can be achieved in Fig. 14(b). Moreover, with the relatively high fixed transmission power, the largest total system cost is obtained with the MATD3-FH method in the case of $N = 2$. The random scheme always obtains the large total system cost with the high arrival rate of tasks. Without UAVs participating in tasks processing, it will be challenging for the MATD3-EC method to deal with so much tasks, especially in the case of $N = 3$. Compared with the other four learning approaches, our proposed MATD3 approach can achieve the smallest total system cost with both numbers of UAVs.

Figure 14 shows total system cost as a function of maximum transmission power of UAVs P_{max} with different optimization methods. The MATD3-FP approach is considered with the

fixed power scheme ($P_n^t = P_{max}$). With the maximum transmission power of UAVs P_{max} increasing, we may need to use the higher transmission power of UAVs P_n^t. Considering that the transmission energy consumption is an increasing function of P_n^t, the higher system cost can be obtained as P_{max} increases in all cases. It can be also observed that with more UAVs offering task offloading services, the scenario of $N = 3$ can achieve the smaller total system cost than that of $N = 2$.

Moreover, since the MATD3-FP approach always allocates the fixed transmission power of UAVs with $P_n^t = P_{max}$, it may achieve the maximum downlink transmission energy consumption among the six approaches, especially in the large maximum transmission power of UAVs P_{max}. As for the random method, the relatively higher total system cost is achieved compared with other four learning schemes (MATD3-FH, MADDPG, MATD3-EC, and MATD3). With the fixed height of UAVs, the MATD3-FH method may need the more transmission power of UAVs to guarantee the enough downlink transmission data rate, which results in the larger transmission energy consumption. In the MATD3-EC method, since all UAVs only offload all tasks to ECs for processing directly, the downlink transmission energy consumption accounts for a large proportion in the total system cost. Then, as P_{max} increases, it may achieve the larger total system cost, especially in the case of $N = 3$. Clearly, MADDPG experiences the worse performance with the larger number of UAVs compared with other methods. Our proposed approach greatly outperforms the above four schemes with the smallest total system cost with both numbers of UAVs. Especially when $N = 2$, our proposed approach always obtains the optimal transmission power of UAVs regardless of the maximum transmission power P_{max}.

V. Conclusion

This paper investigated a UAV-assisted MEC system with multiple UAVs and multiple ECs offloading computation tasks of UEs collaboratively. An optimization problem was formulated to obtain the minimum sum of execution delays and energy consumptions by jointly designing the trajectories, computation task allocation, and communication resource management. A cooperative MADRL framework was developed to tackle the non-convexity of the task offloading optimization issue. Considering the high-dimensional continuous action space, MATD3 algorithm was presented to obtain the optimal policy efficiently. Numerical evaluations were given to indicate that the proposed collaborative UAV-EC offloading method can adapt to the mobility of UEs, the change of communication and computation resources, and the dynamicity of computation tasks. The proposed scheme can significantly reduce the total system cost compared with other optimization approaches.

References

[1] G. Yang, Q. Zhang, and Y.-C. Liang, "Cooperative ambient backscatter communications for green Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1116–1130, Apr. 2018.

[2] X. Kang, Y.-C. Liang, and J. Yang, "Riding on the primary: A new spectrum sharing paradigm for wireless-powered IoT devices," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6335–6347, Sep. 2018.

[3] C. Park and J. Lee, "Mobile edge computing-enabled heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1038–1051, Feb. 2021.

[4] Q. Chen, H. Zhu, L. Yang, X. Chen, S. Pollin, and E. Vinogradov, "Edge computing assisted autonomous flight for UAV: Synergies between vision and communications," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 28–33, Jan. 2021.

[5] P. A. Apostolopoulos, G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty," *IEEE Trans. Mobile Comput.*, early access, Mar. 31, 2021, doi: 10.1109/TMC.2021.3069911.

[6] G. Yang, Y.-C. Liang, R. Zhang, and Y. Pei, "Modulation in the air: Backscatter communication over ambient OFDM carrier," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1219–1233, Mar. 2018.

[7] X. Xu, H. Zhao, H. Yao, and S. Wang, "A blockchain-enabled energy-efficient data collection system for UAV-assisted IoT," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2431–2443, Feb. 2021.

[8] N. Zhao, Z. Liu, and Y. Cheng, "Multi-agent deep reinforcement learning for trajectory design and power allocation in multi-UAV networks," *IEEE Access*, vol. 8, pp. 139670–139679, 2020.

[9] G. Yang, R. Dai, and Y. C. Liang, "Energy-efficient UAV backscatter communication with joint trajectory design and resource optimization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 926–941, Feb. 2021.

[10] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "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.

[11] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2020.

[12] Y. Xu, T. Zhang, D. Yang, Y. Liu, and M. Tao, "Joint resource and trajectory optimization for security in UAV-assisted MEC systems," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 573–588, Jan. 2021.

[13] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[14] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229–12239, Oct. 2020.

[15] J. Ji, K. Zhu, C. Yi, and D. Niyato, "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, May 2021.

[16] J. Zhang *et al.*, "Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3688–3699, Apr. 2019.

[17] C. Sun, W. Ni, and X. Wang, "Joint computation offloading and trajectory planning for UAV-assisted edge computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5343–5358, Aug. 2021, doi: 10.1109/TWC.2021.3067163.

[18] C. Zhan, H. Hu, Z. Liu, Z. Wang, and S. Mao, "Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15553–15567, Oct. 2021, doi: 10.1109/JIOT.2021.3073208.

[19] R. S. Sutton, and A. G. Barto, Reinforcement learning: An introduction. MIT Press Cambridge, 1998.

[20] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, May 2019, doi: 10.1109/COMST.2019.2916583.

[21] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.

[22] H. Peng and X. Shen, "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.

[23] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8753–8769, Oct. 2019.

[24] S. Zhu, L. Gui, D. Zhao, N. Cheng, Q. Zhang, and X. Lang, "Learning-based computation offloading approaches in UAVs-assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 928–944, Jan. 2021.

[25] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. S. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020.

[26] 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. Mobile Comput.*, early access, Feb. 16, 2021, doi: 10.1109/TMC.2021.3059691.

[27] T. Ren *et al.*, "Enabling efficient scheduling in large-scale UAV-assisted mobile edge computing via hierarchical reinforcement learning," *IEEE Internet Things J.*, early access, Apr. 7, 2021, doi: 10.1109/JIOT.2021.3071531.

[28] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 1, pp. 73–84, Mar. 2021.

[29] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.

[30] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[31] F. Ding, X. Zhang, and L. Xu, "The innovation algorithms for multivariable state-space models," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 11, pp. 1601–1608, Oct. 2019.

[32] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, *arXiv:1802.09477*.

[33] T. Yuan, W. D. R. Neto, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, "Dynamic controller assignment in software defined Internet of vehicles through multi-agent deep reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 585–596, Mar. 2021.

[34] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Jun. 2014, pp. 387–395.

[35] F. Ding, L. Xu, D. Meng, X.-B. Jin, A. Alsaedi, and T. Hayat, "Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model," *J. Comput. Appl. Math.*, vol. 369, May 2020, Art. no. 112575.

[36] M. Sipper, "A serial complexity measure of neural networks," in *Proc. IEEE Int. Conf. Neural Netw.*, San Francisco, CA, USA, Mar. 1993, pp. 962–966.

Yiyang Pei (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2007 and 2012, respectively. From 2012 to 2016, she was a Research Scientist with the Institute for Infocomm Research, Singapore. She is currently an Associate Professor with the Singapore Institute of Technology, Singapore. Her current research interests include reconfigurable intelligent surface, dynamic spectrum access, and application of machine learning to wireless communications and networks. She was a recipient of the IEEE Communications Society Stephen O. Rice Prize Paper Award in 2021. She is an Editor of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

Ying-Chang Liang (Fellow, IEEE) was a Professor with The University of Sydney, Australia, a Principal Scientist and a Technical Advisor with the Institute for Infocomm Research, Singapore, and a Visiting Scholar with Stanford University, USA. He is currently a Professor with the University of Electronic Science and Technology of China, China, where he leads the Center for Intelligent Networking and Communications (CINC). His research interests include wireless networking and communications, cognitive radio, symbiotic communications, dynamic spectrum access, the Internet of Things, artificial intelligence, and machine learning techniques.

Dr. Liang is a Foreign Member of Academia Europaea. He was a Distinguished Lecturer of the IEEE Communications Society and the IEEE Vehicular Technology Society. He received the Prestigious Engineering Achievement Award from the Institution of Engineers, Singapore, in 2007, the Outstanding Contribution Appreciation Award from the IEEE Standards Association in 2011, and the Recognition Award from the IEEE Communications Society Technical Committee on Cognitive Networks in 2018. He was a recipient of numerous paper awards, including the IEEE Communications Society Stephen O. Rice Prize Paper Award in 2021, the IEEE Jack Neubauer Memorial Award in 2014, and the IEEE Communications Society APB Outstanding Paper Award in 2012. He was the Chair of the IEEE Communications Society Technical Committee on Cognitive Networks and served as the TPC Chair and the Executive Co-Chair for IEEE GLOBECOM'17. He was a Guest/an Associate Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Signal Processing Magazine*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He was the Associate Editor-in-Chief of the *Random Matrices: Theory and Applications* (World Scientific). He is the Founding Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: Cognitive Radio Series, and the Key Founder and the Editor-in-Chief of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also serving as the Associate Editor-in-Chief for *China Communications*. He has been recognized by Thomson Reuters (now Clarivate Analytics) as a Highly Cited Researcher since 2014.

Nan Zhao (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Wuhan University, Wuhan, China, in 2005, 2007, and 2013, respectively. She is currently a Professor with the Hubei University of Technology, Wuhan, and also works as a Post-Doctoral Research Fellow at the University of Electronic Science and Technology of China. Her current research involves machine learning in wireless communications.

Zhiyang Ye received the bachelor's degree from Nanchang Hangkong University in 2019. He is currently pursuing the master's degree in electrical engineering with the Hubei University of Technology. His main research focuses on machine learning in wireless communications.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang in 1999 and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Full Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of green communications, the Internet of Things, and sensor networks.