Dynamic UAV Deployment for Differentiated Services: A Multi-Agent Imitation Learning Based Approach

Xiaojie Wang, Zhaolong Ning, Song Guo, *Fellow, IEEE*, Miaowen Wen, Lei Guo, and H. Vincent Poor, *Life Fellow, IEEE*

Abstract—Unmanned Aerial Vehicles (UAVs) have been utilized to serve on-ground users with various services, e.g., computing, communication and caching, due to their mobility and flexibility. The main focus of many recent studies on UAVs is to deploy a set of homogeneous UAVs with identical capabilities controlled by one UAV owner/company to provide services. However, little attention has been paid to the issue of how to enable different UAV owners to provide services with differentiated service capabilities in a shared area. To address this issue, we propose a multi-agent imitation learning enabled UAV deployment approach to maximize both profits of UAV owners and utilities of on-ground users. Specially, a Markov game is formulated among UAV owners and we prove that a Nash equilibrium exists based on the full knowledge of the system. For online scheduling with incomplete information, we design agent policies by imitating the behaviors of corresponding experts. A novel neural network model, integrating convolutional neural networks, generative adversarial networks and a gradient-based policy, can be trained and executed in a fully decentralized manner with a guaranteed ϵ-Nash equilibrium. Performance results show that our algorithm has significant superiority on average profits, utilities and execution time compared with other representative algorithms.

Index Terms—UAV deployment, differentiated services, imitation learning, decentralized training, Nash equilibrium.

1 Introduction

The upcoming network era initiated by the Fifth Generation of Mobile Communications (5G) is expected to connect massive numbers of devices ubiquitously and seamlessly. According to the report in [1], mobile traffic worldwide will reach 1 ZB/month in 2028, equivalent to 200 GB/month for 5 billion global users. This poses significant challenges to the current network infrastructure with urgent demands on computing abilities and capacities. However, for network operators, it is infeasible to deploy infrastructure everywhere due to installation and maintenance costs. With the advantages of flexibility and mobility, Unmanned Aerial Vehicles (UAVs) have evolved into a promising vehicular paradigm, and can be utilized to extend wireless networks by providing services for on-ground users, such as data collection, rapid network access, edge computing and content caching. According to the report in [2], the global revenue of UAV-based hardware and services will be up to 12.6 billion by 2025 with merely 792 million in 2017.

To achieve the promise of the UAV-based services, a fundamental issue is how to deploy those UAVs in an efficient way to satisfy the requirements of on-ground users, including maximizing the on-demand coverage area as well as the covered number of users, and satisfying different quality-of-service requirements. Although those coexisting UAVs can provide large service coverage for on-ground users by making efficient utilization of UAV idle resources, it is difficult to handle differentiated services provided by various UAV owners in the network. That is, UAV owners provide similar services but with different capabilities. For example, company A provides UAV-based edge computing services with computing capability 1.5 GHz per UAV, while company B with 2.5 GHz per UAV. Higher service capability always requires stronger hardware, which also has a higher cost. Thus, company B is willing to offer a higher service price than that of company A. This brings up the questions: **can the two companies coexist in the network to jointly provide services for users? If so, how can they offer proper service prices and UAV quantities to maximize their own profits?**

1.1 Motivation

UAVs have been developed as an orchestration framework for a wide range of industrial and commercial scenarios. Many countries have established rules for the usage of UAVs with commercial purposes. For example, in 2015, rules for UAV operations were developed by the European safety aviation authority. The federal aviation administration of USA also announced guidelines for commercial UAV operations in 2015 [3]. Many countries and companies have registered for allowance certificates to fly commercial UAVs. For instance, Google Wing has pushed researches on UAV

- *X. Wang, Z. Ning (Corresponding author) and L. Guo are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. Email: xiaojie.kara.wang@ieee.org, z.ning@ieee.org, guolei@cqupt.edu.cn.*
- *S. Guo (Corresponding author) is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: song.guo@polyu.edu.hk.*
- *M. Wen is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. E-mail: eemwwen@scut.edu.cn.*
- *H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. E-mail: poor@princeton.edu.*

delivery system in the United States [4]. In Estonia, UAVs have been utilized to monitor overhead power lines for power-supply companies [5]. Since more and more companies have invested in UAV-based applications, the scenario that different kinds of UAVs controlled by different companies with distinct service capabilities becomes common.

One typical application example is different network operators can provide heterogeneous network services by UAVs for on-ground users, such as 4G and 5G network services [6]. Just like in a live basketball game, audiences with different network requirements can purchase different value-added services from those network operators [7]. Another example is the network operators can utilize UAVs to support differentiated services for 6G, since merely 20% of the land area is covered by mobile communications [8]. To both minimize the deployment cost and provide ubiquitous connections, UAVs are promising to provide flexible network coverage and differentiated services for urban hotspots and rural areas in shortage of network access.

Consequently, due to the practical application scenarios, the potential solutions for differentiated services provided by UAVs with various service providers are important and necessary, motivating us to investigate this topic.

1.2 Challenges

Although many researches have investigated differentiated services provided by Internet service providers [9], [10], they are not suitable for UAV-based networks due to their unique service providers. **To the best of our knowledge, we are the first to investigate differentiated services with various service providers in the UAV-based network.** It is rather challenging to resolve such an issue due to the following reasons:

First, it is difficult to maximize profits of UAV owners and utilities of users simultaneously. Differentiated services provided by distinct UAV owners make users have different preferences, and they may even pursue other services instead of their original preferences when resource quantities and prices are updated. Thus, it is hard to model user utilities with differentiated services, and also difficult to establish the relationship between the total provided resources of UAV owners and user utilities.

Second, for UAV owners, they cannot observe the policies of others beforehand, resulting in their partial observations. Thus, it is difficult to decide the optimal provided UAV quantities and service prices to reach the Nash equilibrium and guarantee the fairness among multiple UAV owners with incomplete system information.

Third, dynamic user requirements make the UAV deployment issue complex, and call for online scheduling algorithms. The authors in [11] propose a trajectory control algorithm for UAVs flying over the target area and providing computing resources for on-ground users based on multi-agent Deep Reinforcement Learning (DRL). Compared with traditional algorithms, it has a superiority on the performance in terms of consumed energy of user devices, fairness among user devices and that among loads of UAVs. However, on one hand, it does not consider differentiated services and profits of UAV owners. On the other hand, although DRL has been widely utilized for online scheduling, it always has poor performance at the initial thousands of iterations. Thus, novel algorithms with both fast convergence speeds and good performance need to be designed.

1.3 Contributions

To solve the above challenges, this paper proposes a multi-agent imitation learning enabled UAV deployment algorithm, named MILU, to maximize both profits of UAV owners and utilities of on-ground users. Specifically, imitation learning is an efficient machine learning method to deal with online scheduling, since it has a faster convergence speed and is more sample efficient. It allows the agent to imitate the behaviors of experts (formed by their state-action trajectories) that are effective to solve the original problem. However, the expert policies cannot be directly applied in an online manner due to their high time complexity. Thus, a high-efficient learning model should be trained to realize imitation from experts. Our contributions can be summarized as follows:

- We establish the system model based on the analysis and formulation of user utilities as well as UAV owner profits. In addition, we formulate the UAV deployment issue as an optimization problem with the purpose of maximizing profits of UAV owners and utilities of on-ground users simultaneously.
- To solve the formulated problem, we first analyze the interactions of UAV owners based on full system observations, and derive the Nash equilibrium condition. With that condition, expert policies and demonstrations in our imitation learning based UAV deployment scheme can be formed.
- For online scheduling with partial observations of UAV owners, we train agent policies through a novel neural network model, integrating Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs) and the gradient-based policy, to approach the expert performance from the beginning of the algorithm iteration. Specifically, our designed model can be both trained and executed in a fully-decentralized manner without obtaining actual policies of opponents.
- We demonstrate the effectiveness of our proposed algorithm from both theoretical and experimental perspectives. An ϵ-Nash equilibrium can be guaranteed, and real-word datasets are utilized for the evaluation of differentiated services provided by UAVs. Performance results show that our algorithm has superiority on average utilities of users, average profits and fairness among UAV owners, with significant improvements compared with other representative solutions.

The rest of this paper is structured as follows: in Section 2, we review the related work and introduce imitation learning briefly; we present the system model and formulate the studied problem in Section 3; in Section 4, we design an imitation learning enabled UAV deployment algorithm, followed by performance evaluation in Section 5; finally, we conclude our work in Section 6.

2 Related Work And Background

In this section, we review the state-of-the-art researches about UAV deployment and illustrate the background of imitation learning.

2.1 UAV Deployment

Existing studies on UAV deployment can be classified into two categories from the number of utilized UAVs, i.e., one UAV deployment and multiple UAV deployment to provide various services, including edge computing, caching and feasible network access. For example, a moving UAV endowed with computing resources is utilized to provide services for mobile applications, aiming at satisfying the quality-of-service requirements of users based on successive convex approximation [12]. A UAV data collection scheme is proposed in [13], where UAV trajectories are optimized based on the simulated annealing algorithm, to reduce redundant collected data with the minimum energy consumption. Generally, optimization algorithms for one UAV deployment are always centralized due to the unique server, and hardly extend to the scenario with multiple UAVs because of the complexity brought by node dimensions.

For multiple UAVs, authors in [14] study the deployment of UAV-based services by designing an approximation algorithm based on game theory, to minimize the social service cost. A UAV clustering method is designed in [15] to enable multi-task offloading. The communication, caching and computing resources are jointly optimized by a model-free learning algorithm, which has a low convergence speed. Caching-enabled UAVs are investigated in [16], where the quality-of-experience of users is optimized by a designed machine learning framework of conceptor-based echo state networks. However, all UAVs belong to one UAV owner, and they are assumed to be in full cooperation. The authors in [17] consider safe and fast configurations of UAV backhaul in a dynamic environment. Specifically, a convex optimization algorithm is proposed to optimize UAV locations and traffic routing. Nevertheless, these algorithms are all centralized, requiring full system knowledge. They are not suitable for our considered scenario, since there are competitions among multiple service providers with partial observations of system states.

A distributed control framework for realizing mobile crowdsensing by UAVs is proposed in [18], where DRL is utilized to select real-time actions for UAVs. Similar to our work, interactions among multi-agents are modeled and explored. However, it has poor system performance in the initial stage of algorithm execution. Mobile edge computing networks, consisting of Base Stations (BSs) as well as UAVs operated by multiple service providers, are considered in [19]. A game theoretic and reinforcement learning framework is proposed to maximize the long-term payoff of BSs and reach a Nash equilibrium among different BSs in a distributed manner. Different from our work, a quasi-stationary environment is considered. A software-defined control framework is designed in [20], where the UAV network can be controlled in a distributed manner and scalable fashion. Control decomposition theories are applied to generate sub-control problems that can be solved by each UAV.

Although different kinds of distributed algorithms are proposed for UAV management, they cannot be applied in our system. On one hand, the traditional distributed convex optimization cannot guarantee a good performance from the long-term perspective, since it merely concentrates on the performance optimization in each time slot. On the other hand, to overcome the drawbacks of poor performance at the initial stage of DRL, imitation learning is more suitable for online scheduling applications with a fast convergence speed and good performance by imitating expert behaviors. In addition, multi-agent imitation learning can learn to achieve a Nash equilibrium for involved agents without tedious bargaining processes in traditional game theories. To the best of our knowledge, **we are the first to investigate the UAV deployment issue by multi-agent imitation learning**, with the purpose of maximizing both utilities of on-ground users and profits of different UAV owners.

2.2 Background of Imitation Learning

As an efficient machine learning technique, imitation learning allows the learning agent to imitate the behaviors from expert demonstrations with the purpose of achieving good performance. It has been widely utilized in robotic motion planning and automatic driving. Two kinds of roles are involved in imitation learning, i.e., experts and learning agents. The expert can provide demonstration $\mathbb{D}$, including expert policy π_E formed by I sampled trajectories. State-action pairs in the i_{th} trajectory can be represented by $\left\langle \left(s_i^0, a_i^0\right), \left(s_i^1, a_i^1\right), ..., \left(s_i^{\mathbb{H}}, a_i^{\mathbb{H}}\right)\right\rangle$. For simplicity and without loss of generality, all trajectories are assumed to have same length $\mathbb{H}$. The agent can train its own policies according to the expert demonstration, and then gradually improve its policies by interacting with the surrounding environment. Generally, traditional imitation learning is always regarded as one kind of supervision learning. In this case, the agent cannot always make the right decision when it encounters situations never met before. This is because the expert demonstration can only be provided with fixed iterations, and cannot contain all states that the agent may encounter [21], [22].

Generative Adversarial Imitation Learning (GAIL) is proposed in [23], which can overcome the compound error caused by limited expert demonstrations. It makes the distributions of state-action pairs visited by the agent close to those of expert trajectories. To shape the distribution of state-action pairs generated by the agent policy, GANs are harnessed to train the learning model. Generally, GAN involves two participants, i.e., generator G and discriminator D. Generator G is utilized to generate data, the distribution of which is analogous to true data distribution Z. Discriminator D tries to distinguish whether a sample is from the data generated by generator G or true data distribution Z. Consequently, GAN tries to achieve the following objective: $\min_G \max_D V(G,D) = E_x\left[\log D\left(x\right)\right] + E_Z\left[\log\left(1 - D\left(G(Z)\right)\right)\right]$, with the purpose of optimizing both generator G and discriminator D.

In GAIL, the learning agent can be regarded as generator G, attempting to generate state-action distributions through imitating those of the expert. Discriminator D learns to distinguish actions generated by the agent and the expert.

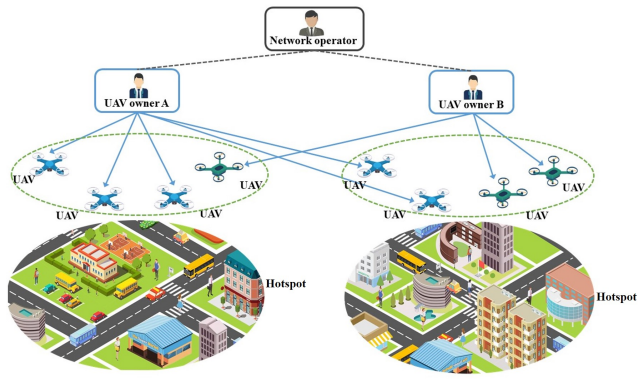


Fig. 1. An illustrative system model of UAV deployment.

Based on the competition between the above two players, the performance of the learning agent in GAIL can be largely improved. Overall, GAIL aims to learn efficient policies through adversarial generated training by mimicking expert demonstrations, i.e.,

$$\hat{\pi} = \arg\min_{\theta}\max_{\omega}\left[E_{\pi_\theta}\left[\log\left(D_\omega\left(s^t,a^t\right)\right)\right] + E_{\pi_E}\left[\log\left(1-D_\omega\left(s^t,a^t\right)\right)\right]\right] - \lambda H(\pi), \tag{1}$$

where λ is a control parameter, and variable $H(\pi)$ is the γ-discounted causal entropy of policy π. Symbols θ and ω are the training parameters for the policy and GAN, respectively. Equation $H(\pi) = E_\pi\left[-\log\pi\left(a^t|s^t\right)\right]$ is founded, which enhances the exploration operation during the learning process.

3 System Model and Problem Formulation

In this section, we first present the system model, and then formulate the optimization problem.

3.1 System Model

As shown in Fig. 1, we consider a scenario that UAVs as flying servers provide mobile services (such as communication, caching and edge computing) to on-ground users. The system contains on-ground user terminals, UAV owners, UAVs, BSs and the network operator. UAVs are controlled by the corresponding UAV owner and provide differentiated services for users to relieve the burdens of the traditional wireless networks. Herein, differentiated services refer to the same kind of service with different service abilities and prices, as exampled in Section 1. Users with higher requirements of service abilities may be content to pay for the higher-quality services with higher prices, and vice versa. UAV owners control their UAVs as well as schedule them by efficient strategies, and can connect to the network operator as in [24]. BSs can sense the number of arrival users nearby. We assume that the network operator can obtain the information of the whole system status, and can send user demands to UAV owners. Then, they can schedule the managed UAVs correspondingly.

3.1.1 Wireless Network Model

There are H hotspots in the considered system, denoted by $h \in \{1,...,H\}$. Those hotspots can be regarded as user density areas that a quantity of users requiring network services, such as shopping mall, sports ground, and downtown. UAVs are deployed over there to relieve the burdens of traditional cellular networks. As shown in Fig. 1, we set hotspots by the circled areas as examples, and they can be regarded as no overlap between any two hotspots due to their geographic regions [24]. For each hotspot, multiple users exist and a set of UAVs can be deployed to serve them. The number of deployed UAVs for each hotspot depends on the users' service requirements.

The time horizon is divided into multiple time slots with the same interval, denoted by $t \in \{0,1,...\}$. In each time slot, user demands of each hotspot keep stable, while can evolve into another value when the next time slot comes. The consideration is reasonable since the time duration of each time slot can be small enough. UAV owners can offer differentiated services with different service abilities, represented by $L = \{l_1,...,l_k,...l_K\}$, where K is the total number of UAV owners. The capability of each UAV owned by UAV owner k is equal to b_k. Each UAV owner has a sufficient number of UAVs, and places a reasonable number of UAVs in the sky of those hotspots. We consider that UAVs can be charged to guarantee their service time. At the beginning of each time slot, after service providers decide the number of deployed UAVs for each hotspot, the UAVs without enough energy are replaced by candidate UAVs with sufficient energy, and those UAVs in shortage of energy are charged by the charging station in case of future usage. The main notations are illustrated in Table 1.

In time slot t, on-ground users randomly arrive in hotspot h with number $m_h(t)$. User i in hotspot h generates service task $\Lambda_{hi}(t) = \left\{d_{hi}(t), \{\iota_{hik}(t)\}_{k=1}^{K}\right\}$ with possibility ϱ_{hi}, where $d_{hi}(t)$ is the required service capacity, and $\iota_{hik}(t) \in [0,1]$ is the preference of user j for service k in hotspot h and time slot t. To satisfy users' service requirements, UAVs belonging to one UAV owner hover in the sky of hotspot h and form a mesh network, acting as a cloudlet. Those UAVs can communicate with each other and transmit tasks to each other for load balancing[1]. The UAVs belonging to different UAV owners do not communicate with each other. When user i has a service request, he/she can first broadcast it to UAVs directly. The nearest UAV of his/her preferred service accepts it, and then user i can send the task to that UAV. Specifically, the communication between UAVs and users (their terminals) is based on Orthogonal Frequency Division Multiplexing (OFDM).

3.1.2 Utilities of Users

For user i, we consider its budget for pursuing services in hotspot h as e_{hi}. Based on that, we have the following definition:

Definition 1 (The total budget of hotspot h)**.** *The total budget of hotspot h in time slot t can be computed by the sum of user*

1. There are many researches having studied load balancing among multiple UAVs, such as [25] and [26], so that it is not our main focus.

TABLE 1
Main notations

Notation	Description
l_k	The identification of service k;
$f_{hk}(t)$	The PDF of users' preferences for service k in hotspot h of time slot t;
$q_{hk}(t)$	The provided quantity of service resources of UAV owner k in hotspot h and time slot t;
$u_h(t)$	The total user utilities of hotspot h in time slot t;
α	The degree of differentiation/substitutability among those provided services;
$m_h(t)$	The number of arrival users in hotspot h and time slot t;
ϱ_{hi}	The possibility for user i to generate a task in hotspot h;
g_0	The cost of deploying one UAV in one hotspot;
g_s	The hovering energy consumption for one UAV in one time slot;
g_c	The service energy consumption for one UAV in one time slot;
b_k	The capability of each UAV owned by UAV owner k;
$p_k(t)$	The price of service $l_k \in S$ for all hotspots provided by UAV owner k;
$\Gamma_{hk}(t)$	The profit of UAV owner k in hotspot h and time slot t;
$c_{hk}(t)$	The total cost for UAV owner k in hotspot h and time slot t;
$u(t)$	The total utility for on-ground users in the covered area in time slot t;
e_h	The maximum value of the payment for on-ground users in hotspot h;
s^t	The network state in time slot t;
o_k^t	The observation of UAV owner k in time slot t;
a_k^t	The action taken by UAV owner k in time slot t;
r_k^t	The received reward of UAV owner k after taking action a_k^t at state s^t;
Ω_k^E	The dataset formed by trajectories of expert policy k;
π_k	The policy of UAV owner k;
$\rho_{\pi_k,\pi_{-k}}$	The occupancy measure from the perspective of agent k;
$\bigtriangledown J_\pi(\theta_k)$	The gradient of the policy network of agent k;
$A_k(o_k, a_k, \hat{a}_{-k})$	The advantage function of agent k;
$V_k(o_k^t, a_k, \hat{a}_{-k})$	The state value in time slot t based on observation o_k^t and estimated opponent action $\hat{a}_{-k}$;
$\bigtriangledown J_D(\omega_k)$	The gradient for the discriminator network with parameter ω_k;
L_{ε_k}	The loss function for the opponent network with parameter ε_k.

budgets in time slot t, i.e., $e_h(t) = \sum_{i=1}^{m_h(t)} e_{hi}$, which can be simplified by e_h.

For service l_k, the UAV owner provides price $p_k(t)$ for all hotspots (this is common for market products in real-world scenarios), and offers service resources with quantity $q_{hk}(t)$ for hotspot h in time slot t. We consider that each user has preference $\iota_{hjk}(t)$ toward service l_k based on its own budget e_{hi}, and provide the following definition similar to [2]:

Definition 2 (Aggregated user preference)**.** *The aggregated user preference for service $l_k \in S$ in hotspot h and time slot t can be denoted by $f_{hk}(t)$, which can be computed based on the personal user preference, i.e.,*

$$f_{hk}(t) = \frac{\sum_{i=1}^{m_h(t)} \iota_{hik}(t)}{m_h(t)}. \tag{2}$$

Then, the total service requirement for service l_k in hotspot h and time slot t can be computed by:

$$d_h(t) = \sum_{i=1}^{m_h(t)} d_{hi}(t) \times \varrho_{hi}. \tag{3}$$

Instead of modeling each user's utility, we form the aggregated user utility in hotspot h and time slot t, since it can reflect the relationship of total user demands with service quantities in an efficient way. As a result, when there are more choices and available service quantities, the aggregated user utility can be increased. As a result, we define the aggregated user utility function based on the Constant Elasticity of Substitution (CES) function [27], which can reflect the relationship between the supply and the demand in the real-word market:

$$u_h(t) = \sum_{k=1}^{K} f_{hk}(t) \left[\frac{q_{hk}(t)}{d_h(t)}\right]^\alpha, \tag{4}$$

where α represents the degree of differentiation/substitutability among those provided services in the system, satisfying $0 < \alpha < 1$. From equation (4), we can observe that when the ratio of $q_{hk}(t)/d_h(t)$ becomes large, more resources can be served for users. In addition, the following theorem is founded:

Theorem 1 (Concavity of the aggregated user utility)**.** *The aggregated user utility in hotpot h defined in equation (4) is concave with service quantity $q_{hk}(t)$ provided by UAV owner k in hotspot h.*

The above theorem can be proved by verifying the second order derivative of equation (4), and a maximum value can be found in its feasible region $[0, q_{hk}^{max}]$.

3.1.3 Profits of UAV owners

For UAV owners, they intend to maximize their profits by providing services for on-ground users. Thus, the obtained profit from users and the cost for providing such services should be considered. For the cost of each UAV owner, it contains two parts: installation and energy costs. For the former, the unit cost can be represented by g_0, which refers to the cost of deploying one UAV in one hotspot and can be spread among time slots. The latter includes the cost for hovering and service energy consumption [2]. When a UAV hovers over one hotspot, the unit hovering energy consumption can be represented by g_s in each time slot. If the UAV provides services, its unit cost for service energy consumption can be denoted by g_c in each service unit. Then, the total cost for UAV owner k in hotspot h and time slot t can be computed by:

$$c_{hk}(t) = (g_0 + g_s)\left\lceil \frac{q_{hk}(t)}{b_k} \right\rceil + g_c q_{hk}(t), \tag{5}$$

where expression $\lceil q_{hk}(t)/b_k \rceil$ calculates the required number of UAVs for service k in hotspot h. Thus, the first part of equation (5) denotes the installation cost and the hovering energy cost in time slot t, and the second part is the energy cost for providing services in time slot t.

Then, the profit of UAV owner k in hotspot h and time slot t can be obtained by:

$$\Gamma_{hk}(t) = p_k(t) q_{hk}(t) - c_{hk}(t), \tag{6}$$

where the first part is the service profit of UAV owners obtained from on-ground users, and the second part is the total cost of UAV owners in time slot t.

3.2 Use Cases

Generally, computation-offloading and caching-based applications are the major focus in mobile edge computing networks. Herein, we provide two use cases to illustrate the utility settings for both UAV owners and users.

1) For computation-offloading applications: the required capacity for user i can be the required CPU cycle related to the computation task, i.e., $d_{hi}(t) = \mathfrak{c}_{hi}(t)$. To complete a task, the computing delay is the dominated factor to impact the system performance in our considered UAV-based network, when the distance between UAVs and on-ground users is close[2]. Thus, we mainly consider the allocations of computing resources, and the total required CPU cycles for computing offloading tasks in time slot t is $d_h(t) = \sum_{i=1}^{m_h(t)} \mathfrak{c}_{hi}(t) \times \varrho_{hi}$. The provided service quantity $q_{hk}(t)$ is the aggregated CPU cycles for all UAVs by UAV owner k in time slot t. For the energy cost of UAVs, g_c can be regarded as the unit energy consumption cost for each CPU cycle, and b_k is the maximum CPU cycle for one UAV of service provider k. According to the defined energy consumption equation in [28], unit energy cost g_c can be computed by $g_c = g_u \kappa_k (b_k)^2$, where κ_k is the coefficient related to energy efficiency of UAVs owned by service provider k, and g_u is the charge fee for unit energy.

2) For caching-based applications: the required capacity can be set by the required transmission rate, i.e., $d_{hi}(t) = \mathfrak{s}_{hi}(t)$. It is the main factor to affect the quality-of-experience of users who prepare to cache their contents in servers or download contents from cached servers. Then, the total required transmission rate for transmitting contents in time slot t is $d_h(t) = \sum_{i=1}^{m_h(t)} \mathfrak{s}_{hi}(t) \times \varrho_{hi}$. The provided service quantity $q_{hk}(t)$ is the aggregated transmission rate for all UAVs by UAV owner k in time slot t. For the energy cost of UAVs, g_c can be the unit energy cost related to the wireless transmission rate. The wireless transmission rate is related to channel bandwidth $\mathbb{B}$, wireless channel gain h_{ji}^t between UAV j and user i, noise power σ_j at UAV j and transmission power $\mathbb{P}_{ji}$ from UAV j to user i according to the Shannon formula. In addition, the consumed transmission energy has a positive relationship with transmission power $\mathbb{P}_{ji}$, thus g_c can be formed by $g_c = g_u \varphi\left(\mathbb{B}, \sigma_j, h_{ji}^t, \mathbb{P}_{ji}\right)$, where $\varphi(\cdot)$ is a function related to its input.

3.3 Problem Formulation

We intend to maximize both the total profits of UAV owners and the utilities of on-ground users by properly deploying UAVs with differentiated services. For on-ground users, their total utility in the covered area can be computed by:

$$u(t) = \sum_{h=1}^{H} u_h(t) = \sum_{h=1}^{H} \sum_{k=1}^{K} f_{hk}(t) q_{hk}(t)^{\alpha}. \tag{7}$$

2. Herein, we merely consider the computing delay for simplicity. Other delays, such as transmission delay, can also be considered by establishing similar relationships between those services and the utility function.

Then, we formulate **the utility-maximization problem for on-ground users** as follows:

$$\text{P1:} \quad \max_{q_{hk}} u(t), k \in \{1, .., K\}, h \in \{1, ..., H\}, \tag{8}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} p_k(t) q_{hk}(t) \le e_h, h \in \{1, ..., H\}, \tag{8a}$$

where constraint (8a) guarantees that the payment of on-ground users in hotspot h cannot exceed maximum value e_h, and also ensures that the prices offered by UAV owners cannot be arbitrarily high.

For UAV owners, they intend to maximize their own profits. Then, we formulate **the long-term profit-maximization problem for UAV owners** as follows:

$$\text{P2:} \quad \max_{q_{hk}, p_k} \Gamma_k = \sum_{t=0}^{\infty} \sum_{h=1}^{H} \Gamma_{hk}(t), k \in \{1, .., K\}. \tag{9}$$

We need to solve Problems P1 and P2 simultaneously. However, it is rather challenging, because: a) price p_k and quantity q_{hk} of services in the two problems are coupled and affect each other, making them cannot be handled independently; b) though the arrival user flow in each hotspot follows the Poisson process, the user preference and total service requirements for each service are not known beforehand, making Problem P1 incomputable; on one hand, the nonlinear character and the unknown parameters of the optimization function make the traditional optimization method disabled; on the other hand, the traditional game-theory based algorithm assign prices for players by selecting the best approaches based on the profits they can obtain. In our system, UAV owners' profits are dependent on unknown user preferences, thus repeated iterations with the environment should be conducted for UAV owners to learn their best strategies, making traditional game-theoretic approaches inefficient; c) for online learning, UAV owners should make their own decisions independently without knowing others' policies, which complicates the solving process of the two problems.

Generally, multi-agent inverse reinforcement learning algorithms are leveraged for the online training with unknown rewards, where a learned reward function can be obtained based on experts' training samples. However, it is only suitable for the situation that the reward function can be decoupled from the environment. In our system, UAV owners' rewards are heavily dependent on users' preferences and budget, thus multi-agent inverse reinforcement learning algorithms are not suitable for our system.

Correspondingly, we propose an imitation learning-enabled UAV deployment algorithm to resolve the above two problems comprehensively as described in the following section, and its advantages are: a) agents can imitate the behaviors of corresponding experts to improve their learning speed; b) agents can interact with the environment to train their policies and further improve their performance; c) even partial system states are known, agent policies can still coverage; and d) the actual policies of opponents are not necessarily known, which can largely reduce the complexity of the training algorithm.

4 Imitation Learning Enabled UAV Deployment

4.1 Algorithm Overview

The formulated problems in Subsection 3.3 are coupled and interdependent. To settle them, we first analyze the interactions among UAV owners to derive the Nash equilibrium condition based on full system states in Subsection 4.2. For online scheduling, we specify the Markov game by defining states, observations, actions, transition possibilities and rewards related to our considered system interactions in Subsection 4.3. Then, we design expert policies offline based on full observations of system states in Subsection 4.4. In Subsection 4.5, we present the designed agent policies, which can be trained online merely with partial observations through a designed neural network model based on expert demonstrations. To enable fully-decentralized training, we design an opponent model for each agent to predict the possible actions taken by its opponents instead of obtaining their actual actions. At last, we prove that an ϵ-Nash equilibrium can be reached among those UAV owners in an online manner.

4.2 UAV Owner Interaction Analysis

We first analyze interactions among UAV owners and derive the Nash equilibrium condition based on the full observation of system states. From Problems P1 and P2, we can observe that two variables should be derived, i.e., quantity $q_{hk}(t)$ and price $p_k(t)$. Then, we try to find whether the two variables can be established by a direct relationship, so that the computation complexity of the formulated problems can be further reduced. Since $u_h(t)$ in Problem P1 is a strictly concave function, objective function (8) should have a maximum value based on constraint (8a). Thus, we can obtain the following theorem by processing Problem P1:

Theorem 2 (Relationship between service prices and quantities)**.** *The prices of differentiated services offered by UAVs can be formed by a function of the provided resource quantities as follows:*

$$p_k(t) = \frac{e_h f_{hk}(t) \left[q_{hk}(t)\right]^{\alpha-1}}{\sum_{k'=1}^{K} f_{hk'}(t) \left[q_{hk'}(t)\right]^{\alpha}}. \tag{10}$$

The proof can be found in Appendix A of Supplemental File.

From Theorem 1, we can observe that the service price not only depends on the quantities of service resources provided by UAV owner k, but also relies on those of other UAV owners. Then, equation (6) can be transformed into:

$$\Gamma_{hk}(t) = \frac{e_h f_{hk}(t) \left[q_{hk}(t)\right]^{\alpha}}{\sum_{k'=1}^{K} f_{hk'}(t) \left[q_{hk'}(t)\right]^{\alpha}} - c_{hk}(t). \tag{11}$$

Problem P2 can be solved by reaching a Nash equilibrium among different UAV owners. Since variable $p_k(t)$ in P2 can be represented by $q_{hk}(t)$ based on Theorem 1, we merely need to compute the value of $q_{hk}(t)$ for the Nash equilibrium. If optimal quantity $q_{hk}^*(t)$ satisfies the following theorem, the Nash equilibrium can be reached.

Theorem 3 (Nash equilibrium condition)**.** *The Nash equilibrium can be reached, if the offered resources of UAV owners in time slot t satisfy the following condition:*

$$\begin{aligned} &\left(\frac{\alpha e_h}{q_{hk}(t)} Q_k - 2A_k Q_k\right) \Psi_1\left(q_{h,-k}(t)\right) - A_k Q_k^2 \\ &= \Psi_2\left(q_{h,-k}(t)\right), \end{aligned} \tag{12}$$

where symbols $A_k = (g_0 + g_s + g_c b_k)/b_k$, $Q_k = f_{hk}(t)\left[q_{hk}(t)\right]^{\alpha}$, $1 \le k \le K$, functions

$$\Psi_1\left(q_{h,-k}(t)\right) = \sum_{k'=1, k' \neq k}^{K} Q_{k'}, \tag{13}$$

and

$$\Psi_2\left(q_{h,-k}(t)\right) = A_k \left[\sum_{\substack{k'=1,\\ k' \neq k}}^{K} Q_{k'}^2 + 2 \sum_{\substack{k''=1,\\ k'' \neq k}}^{K-1} Q_{k''} \sum_{\substack{j=k''+1,\\ j \neq k}}^{K} Q_j \right]. \tag{14}$$

The proof can be found in Appendix B of Supplemental File.

From Theorem 2, we can obtain the implicit condition to reach a Nash equilibrium. This is because when total number K of UAV owners is large, it is impossible to obtain an explicit condition related to optimal resource quantity q_{hk}^* with so many coupled variables. However, we can obtain the uniqueness of the Nash equilibrium as follows:

Theorem 4 (Uniqueness of the Nash equilibrium)**.** *When equation (12) is satisfied, there is a unique Nash equilibrium for the profit competition among different UAV owners.*

The proof can be found in Appendix C of Supplemental File.

We can also deduce the range of q_{hk}^* based on Theorem 3 by the following theorem:

Theorem 5. *When the Nash equilibrium is satisfied, optimal resource quantity q_{hk}^* satisfies $0 \le q_{hk}^* < \alpha e_h / 2A_k$.*

The proof can be found in Appendix D of Supplemental File.

Although there is no explicit form of the Nash equilibrium condition when K is large, we can find the explicit form when $K = 2$. In the following theorem, we provide that case.

Theorem 6. *When there are two UAV owners in the system, i.e., $K = 2$, the Nash equilibrium is unique, and can be reached with the offered resources of each UAV owner in time slot t as follows:*

$$q_{hk}^*(t) = \frac{\alpha e_h f_{h,K+1-k}(t) \left[A_k\right]^{\alpha-1} f_{hk}(t) \left[A_{K+1-k}\right]^{\alpha}}{\left\{f_{h,K+1-k}(t) \left[A_k\right]^{\alpha} + f_{hk}(t) \left[A_{K+1-k}\right]^{\alpha}\right\}^2}, \tag{15}$$

where $1 \le k \le K$.

The proof can be found in Appendix E of Supplemental File.

Based on the above theorem, we can obtain the maximum provided service resources as follows:

Theorem 7. *The maximum value of provided service resources of hotspot h can be computed by $q_{hk}^{max} = \alpha e_h / 4A_k$, when $K = 2$.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3116236, IEEE Transactions on Mobile Computing

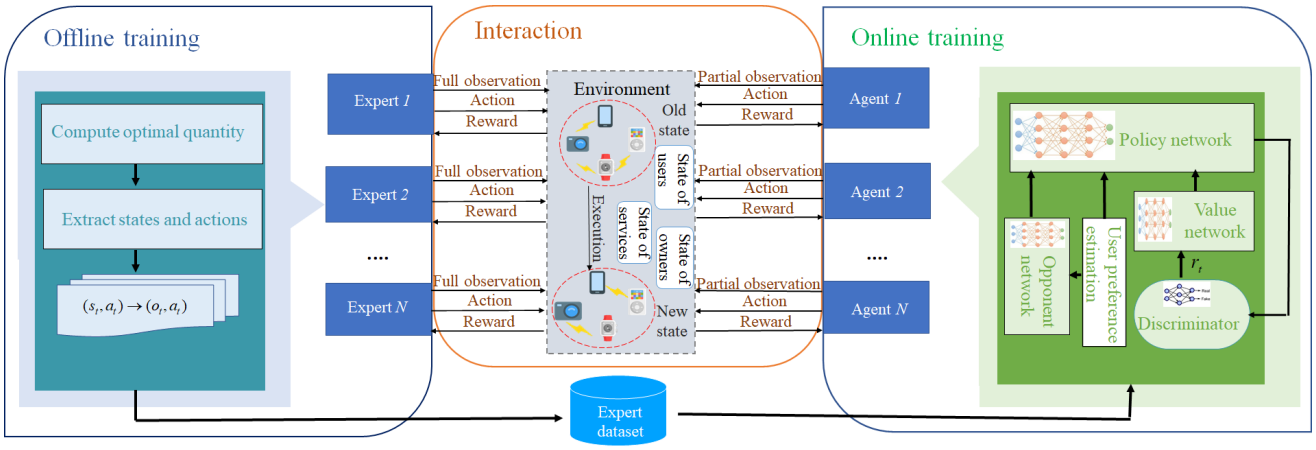


Fig. 2. The structure of the designed algorithm.

The proof can be found in Appendix F of Supplemental File.

The above theorems derive the relationship between service quantity $q_{hk}(t)$ and price $p_k(t)$, and analyze the Nash equilibrium condition for Problem $P2$. Then, we can obtain the optimal value for Problem $P1$ based on the following theorem:

Theorem 8. *Optimal service quantity q^*_{hk} for Problem $P2$ is also the optimal value for Problem $P1$.*

The proof can be found in Appendix G of Supplemental File.

From Theorems 2-4, we can obtain the Nash equilibrium point and maximum provided service quantities in each time slot. However, the above analyses are based on complete system information. That is, each UAV owner knows the user demands in each time slot and the policies of others beforehand, and can make the optimal deployment decision. However, UAV owners are almost impossible to obtain others' policies when they make decisions. To enable online UAV deployment based on partial observations and make the system performance approach that based on complete system information, we present the designed imitation learning based UAV deployment approach in the following subsections. In addition, we merely need to solve Problem P2 by finding the optimal service quantities for UAV owners, which is reflected in Theorem 8.

4.3 Markov Game Formulation

Imitation learning is an efficient learning method allowing experts to pass on experiences for agents and has a fast convergence speed from the beginning of algorithm iterations, thus it is suitable for our online UAV deployment problem. We first transfer the profit-maximization issue defined in Subsection 3.3 into a Markov game.

The profit-maximization issue for different UAV owners can be modeled as a Markov game, regarded as an extension of Markov decision process. The UAV owners can be regarded as K different learning agents, and the game is represented by tuple $\langle K, S, O, A, P, \mathbb{R}, \gamma \rangle$, the elements of which are explained as follows:

a) State: S is the state set of the modeled Markov game, where $S \triangleq \{s^t = (S_1, S_2, S_3)\}$, $t \in \{0, 1, ...\}$. Three elements are included: S_1 represents the state of users, containing service task $\Lambda_{hi}(t)$ of user i, task generation possibility ϱ_{hi}, aggregated user preference $f_{hk}(t)$ and maximum payment e_h of on-ground users; S_2 denotes the state of UAV owners, where unit cost A_k of UAV owners, sustainability degree α and service capability b_k of UAVs are included.

b) Observation: For each UAV owner, full network state s^t cannot be observed while merely partial network state is available, denoted by $O \triangleq \{o^t = \{o^t_k\}\}_{1 \le k \le K}$, where o^t_k is the observation of UAV owner k. In our considered system, aggregated user preference $f_{hk}(t)$, and the policies of other UAV owners are not known to each UAV owner.

b) Action: Set $A \triangleq \left\{a^t = \{a^t_k = \Delta q_{hk}(t)\}_{1 \le k \le K, 1 \le h \le H}\right\}$ denotes the actions taken by UAV owners, where $\Delta q_{hk}(t)$ is the additional service quantity that UAV owner k should deploy over hotspot h in time slot t. Then, the service quantity that owner k should provide can be computed by $q_{hk}(t) = q_{hk}(t-1) + \Delta q_{hk}(t)$. The value of $\Delta q_{hk}(t)$ can be either below or above 0, representing the UAV owner can either improve or reduce the provided services. According to Theorem 3, the maximum value of $q_{hk}(t)$ is $\alpha e_h / 4A_k$, thus $-\alpha e_h / 4A_k \le \Delta q_{hk}(t) \le \alpha e_h / 4A_k$.

c) State transition probability: $P : S \times A \times S \rightarrow [0, 1]$ denotes the state transition probability distribution, and $\rho_0 : S \rightarrow [0, 1]$ is the distribution of initial state s^0. Based on probability $P(s^{t+1}|s^t, a^t)$, the state is transferred into s^{t+1} from s^t by taking action a^t.

d) Reward: $r^t_k : S \times A \rightarrow \mathbb{R}$ represents the immediate received reward of UAV owner k after taking action a^t_k at state s^t. We define $r^t_k = \Gamma_{hk}(t)$, and utilize $-k$ to denote the set of UAV owners except k. The objective of UAV owner k becomes to maximize its own total expected profits $R^t_k \triangleq E\left[\sum_{\tau=0}^{t} \gamma^\tau r^\tau_k\right]$ by solving the formulated Markov game, where variable $\gamma \in [0, 1]$ is the discounted factor.

4.4 Expert Policies

We assume that there are K experts in the network, and they can observe the full network state and know the policies of others, acting as oracles in the system. This is a general assumption in imitation learning algorithms, and the expert policies in our considered system can be obtained by offline computation based on history network requirements and

costs of UAV owners. Thus, the expert demonstration can be collected as follows:

First, for the state in each time slot, experts compute the optimal quantities of provided services according to Theorem 2 in the sight of oracles. Thus, for expert k, its full observation s_k^t and partial observation o_k^t in the view of its corresponding agent, its action a_k^t and the actions of its opponents a_{-k}^t can be obtained. Second, the considered time slots can be divided into U batches, and each batch contains B observation-action pairs. For each observation-action pair (o_k^t, a_k^t, a_{-k}^t), its value is recorded. Then, expert policy trajectories can be collected by dataset $\Omega_k^E = \left\{\Omega_{ki} = \left\{(o_k^t, a_k^t, a_{-k}^t)\right\}_{t=1}^{B}\right\}_{i=1}^{U}$.

4.5 Agent Policies

Since UAV owners in the network cannot observe the full network state and the policies of others beforehand, it is difficult to conduct online scheduling for UAV deployment. To conquer the above difficulties, the UAV owners in the imitation learning act as agents and adopt policies by imitating expert demonstrations to make their performance approach that of the experts. Thus, multi-agent imitation learning can be applied, where agent k imitates the behaviors of corresponding expert k. To improve the performance of imitation learning in our system, we first analyze how to estimate opponents' policies, and then present the whole training process in detail.

4.5.1 Opponent policy estimation

According to Theorem 3 in [29], multi-agent imitation learning can be regarded as an occupancy measure matching problem with reward regularizer ψ, and the optimal policy can be expressed as:

$$\pi_k^* = \arg\min_{\pi_k} -\lambda H(\pi_k) + \psi^*\left(\rho_{\pi_k,\pi_{-k}} - \rho_{\pi_k^E}\right). \quad (16)$$

The occupancy measure represents the unnormalized distribution of observation-action pairs corresponding to the interactions of agents caused by joint policy π. From the perspective of agent k, the occupancy measure is written as:

$$\rho_{\pi_k,\pi_{-k}} = \pi_k(a_k|s)\pi_{-k}(a_{-k}|s)\sum_{t=0}^{\infty}\gamma^t P(s^t = s|\pi), \quad (17)$$

where $\pi_k(a_k|s)\pi_{-k}(a_{-k}|s)$ is the equivalent deformation of $\pi(a_k, a_{-k}|s)$. Equation (16) implies that agents try to minimize the gaps between the distributions of state-action pairs navigated by their own policies and those triggered by expert policies. Nevertheless, on one hand, the decisions on the provided quantities of one agent deeply depend on those of others in our considered system; on the other hand, the agents cannot observe full system state s, and only partial observation o is known. Similar to [30], we define the occupancy measure related to agent k based on its observation o_k and the policies of opponents by:

$$\begin{aligned}\rho_{\pi_k,\pi_{-k}} &= \pi(a_k, a_{-k}|o) \\ &= \pi_k(a_k|o, a_{-k})\pi_{-k}(a_{-k}|o)\sum_{t=0}^{\infty}\gamma^t P(o_k^t = o|\pi).\end{aligned} \quad (18)$$

From equation (18), we notice that the agents not only need to train their own policies, but also should predict the actions based on current observations. To improve the imitation accuracy of agents, we employ GANs to train the learning model, ensuring the distributions of observation-action pairs triggered by agents approach those triggered by experts. Thus, according to [30], the trained policy can be computed by finding saddle point (π_k, D_k) of the following optimization objective:

$$\begin{aligned}\text{P3:}\quad \min_{\pi_k}\max_{D_k} &-\lambda H(\pi_k) + E_{\pi_E}\left[\log D_k(o_k, a_k, a_{-k})\right] \\ &+ E_{\pi_k,\pi_{-k}}\left[\log(1 - D_k(o_k, a_k, a_{-k}))\right]. \quad (19)\end{aligned}$$

4.5.2 Training process

To solve the above problem, the training process of agent k can be found in Algorithm 1, and the detailed process can be specified as follows:

a) *Neural network initialization*: UAV owners need to train their own policies by minimizing the distributions of their observation-action pairs with those of experts. To realize decentralized training, the prediction of opponents' policies is necessary for the action selection of the agent. We establish four neural networks for each UAV owner, i.e., discriminator network with optimization parameter ω_k, opponent network with ε_k, policy network with θ_k, and value network with ϕ_k. The discriminator network is utilized to distinguish whether an observation-action pair is generated by the corresponding expert or agent policies. The opponent network can predict the actions of opponents based on the observations of the current agent. The policy network trains the corresponding agent policy, while the value network is utilized to score the agent policy.

b) *Action execution*: In time slot t, agent k selects action a_k^t based on current policy π_k. First, agent k records its observation o_k^t, and inputs it into the opponent network while outputting the estimation of opponent actions $\hat{a}_{-k}^t$. Then, observation o_k^t and opponent actions $\hat{a}_{-k}^t$ are input into the policy network, and the output is $a_k^t = \pi_k(a_k|o_k, \hat{a}_{-k})$. We compute reward r_k^t, i.e., current profit $\Gamma_{hk}(t)$, based on the decisions of all agents.

c) *Batch data collection*: To train the policy network, we collect agent trajectories in mini-batch. For each record in the mini-batch of agent k, it contains the observation, the action, the predicted opponent actions, and the output of the discriminator network. Similar to the expert dataset, the batch size is set to B. The collected batch data can be utilized to train the neural networks by minimizing their losses.

d) *Network training*: We apply the actor-critic algorithm [31] to train the policy network, which is widely utilized in reinforcement learning algorithms, such as Deep Deterministic Policy Gradient (DDPG) [32]. The actor selects actions based on the output of the neural network, and the critic scores the actions generated by the actor. Then, the actor can improve its policies based on the evaluation of the critic. In our designed system, the policy network plays the role of the actor, while the value network acts as the critic.

Since the agent does not know the policies of its opponents, it utilizes the estimated action to train its policy

Algorithm 1 Pseudo-code of training processes for agents

Input: Expert trajectories $\{\Omega_k^E\}_{k=1}^K$; batch size B; initial policies with policy parameters $\{\theta_k\}_{k=1}^K$, discriminator parameters $\{\omega_k\}_{k=1}^K$, value parameters $\{\phi_k\}_{k=1}^K$ and opponent parameters $\{\varepsilon_k\}_{k=1}^K$.
Output: Learned policy $\{\pi_{\theta_k}\}_{k=1}^K$.

for round $i = 1, 2, ..$ **do**
  for UAV owner $k = 1, 2, ..., K$ **do**
    Get observation-action pair Ω_{ki} from Ω_k^E.
    Sample the interactions among UAV owners with size B of χ_k based on policy π_k and the opponent model with parameter ε_k.
    Solve Problem P3 by the following steps:
    Update ε_k by minimizing the loss in equation (23) based on observation-action pairs $(o_k, a_{-k}) \in \chi_k$.
    Update ω_i by gradient (22) based on observation-action $(o_k, a_k, \hat{a}_{-k})$, and $\hat{a}_{-k}$ is sampled from opponent model.
    Compute estimated reward: $\hat{r}_k(o_k, a_k, \hat{a}_{-k}) = \log(D_{\omega_k}(o_k, a_k, \hat{a}_{-k})) - \log(D_{1-\omega_k}(o_k, a_k, \hat{a}_{-k}))$.
    Compute the advantage function of UAV owner k based on equation (21).
    Update ϕ_k by minimizing the following loss: $L_{\phi_k} = E\left[\| \hat{R}_k - V_k(o_k^t, a_k, \hat{a}_{-k}) \|^2\right]$.
    Update θ_k by computing the policy gradient based on equation (20).
  end for
end for

based on the policy gradient approach. The gradient can be computed by:

$$\begin{aligned}\bigtriangledown J_\pi(\theta_k) =& E_{o_k, a_k \sim \pi_{\theta_k}, \hat{a}_{-k} \sim \hat{\pi}_{\varepsilon_k}} [\bigtriangledown_{\theta_k} \log \pi_{\theta_k}(a_k|o_k, \hat{a}_{-k}) \\ & A_k(o_k, a_k, \hat{a}_{-k})] - \lambda \bigtriangledown_{\theta_k} H(\pi_{\theta_k}),\end{aligned} \tag{20}$$

where $\hat{a}_{-k}$ is the estimated opponent actions generated by policy $\hat{\pi}_{\varepsilon_k}$ from the opponent network. Expression $A_k(o_k, a_k, \hat{a}_{-k})$ is the advantage function of agent k, and can be computed by:

$$\begin{aligned}A_k(o_k, a_k, \hat{a}_{-k}) = & \sum_{j=1}^{B} (\gamma^j \hat{r}_k(o_k^{t+j-1}, a_k^{t+j-1}, \hat{a}_{-k}^{t+j-1}) \\ & + \gamma^k V_k(o_k^{t+B}, a_k^{t+B}, \hat{a}_{-k}^{t+B})) \\ & - V_k(o_k^t, a_k^t, \hat{a}_{-k}^t),\end{aligned} \tag{21}$$

where $V_k(o_k^t, a_k^t, \hat{a}_{-k}^t)$ is the state value in time slot t based on observation o_k^t and estimated opponent action $\hat{a}_{-k}^t$, i.e., $V_k(o_k^t, a_k^t, \hat{a}_{-k}^t) = E_{o_k^0 = o}[\hat{R}_k^t]$. Since we employ GAN to improve our policies, the cumulative output of the discriminator can be utilized as predicted reward $\hat{R}_k^t$ to help the critic to score the policies generated by the actor. For the discriminator, its gradient can be obtained by:

$$\begin{aligned}&\bigtriangledown J_D(\omega_k) = \\ &E_{o_k, a_k \sim \pi_{\theta_k}, \hat{a}_{-k} \sim \hat{\pi}_{\varepsilon_k}} [\bigtriangledown_{\omega_k} \log(1 - D_{\omega_k}(o_k, a_k, \hat{a}_{-k}))] \\ &+ E_{o_k, a_k, \hat{a}_{-k} \sim \Omega_k^E} [\bigtriangledown_{\omega_k} \log D_{\omega_k}(o_k, a_k, \hat{a}_{-k})].\end{aligned} \tag{22}$$

Algorithm 2 Pseudo-code of the MILU algorithm

Input: State of users S_1, state of UAV owners S_2 and state of provided services S_3.
Output: Profits of UAV owners $\Gamma_{hk}(t)$, and utilities of users $u(t)$.

for time slot $t = 0, 1, 2, ..$ **do**
  Estimate the density of users' preferences by network operators.
  for UAV owner $k = 1, 2, ..., K$ **do**
    Get observation o_k^t.
    for hotspot $h = 1, 2, ..., H$ **do**
      Get action a_{hk}^t by the learning model based on Algorithm 1.
      Compute the quantities of service resources provided for hotspot h.
      Compute profits of UAV owners $\Gamma_{hk}(t)$.
      Compute utilities of users $u(t)$.
    end for
  end for
end for

Thus, we can update the discriminator network based on the above gradient. The opponent network can estimate the current actions based on the local observation of agent k, and it tries to minimize the following loss to train its policies:

$$L_{\varepsilon_k} = E\left[\| \pi_{\varepsilon_k}(\hat{a}_{-k}|o_k) - \pi_{-k}(a_{-k}|o_k) \|^2\right]. \tag{23}$$

Based on the above modeling of the opponent network, the agent policies can be trained in a fully decentralized manner without online interactions of opponents.

The presented training process can enable agents to learn efficient policies for deploying proper quantities of UAVs above hotspots. Though the agent policies are trained based on the estimation of opponent actions without knowing the actual decisions, they can still reach an ϵ-Nash equilibrium, which can be regarded as a sub-optimal Nash equilibrium, and the value function of agent k should satisfy:

$$V_k(o_k, a_k^*, a_{-k}^*) \geq V_k(o_k, a_k, a_{-k}^*) - \epsilon. \tag{24}$$

For our designed imitation learning based UAV deployment algorithm, the pseudo-code is shown in Algorithm 2 and the following theorem is derived:

Theorem 9. *If UAV owners all adopt the designed imitation learning based UAV deployment algorithm to train their policies for determining the quantities of provided services, their profits can reach an ϵ-Nash equilibrium from a long-period perspective.*

The proof can be found in Appendix H of Supplemental File.

The overall complexity of the designed MILU algorithm for each agent can be computed by Theorem 10:

Theorem 10. *The overall complexity of the designed MILU algorithm for each learning agent in the execution process is $\mathcal{O}\left(\left(\sum_{z=1}^{Z} n_z \cdot n_{z-1}\right) HT\right)$.*

The proof can be found in Appendix I of Supplemental File.

5 Performance Evaluation

5.1 Experimental Settings

Our experiments are conducted based on Python 3.6 and Tensorflow 2.1. We utilize the real-world map of Hangzhou, China, and select 50 locations as hotspot centers, where users can enjoy the services provided by those UAVs. We analyze traffic flows with a radius of 200m within each selected location. UAVs are uniformly distributed over those hotspots to provide services for on-ground users, and can provide videos to on-ground users. A real dataset in [33] is employed to characterize the quality of videos, including 19 kinds of videos with 5 qualities, i.e., $\{483, 247, 130, 72, 46\}$ MB. Due to the limited storage capacities of UAVs, they cannot storage all qualities of videos like edge servers. We consider that one UAV caches all videos with merely one quality, and provides one transmission rate for users. Then, the on-ground users can purchase services from one UAV owner. In each time slot, UAV owners learn to decide service quantities and prices for all hotspots.

To simulate a realistic scenario, we set the wireless channel parameters according to [24], where the wireless channel noise is -110 dBm, and the bandwidth of wireless channel is 10 MHz. According to the investigation of UAVs in [34–36], the unit price of energy consumption for video transmission can be set by 5.55×10^{-8} Euro/minute, and the required loiter power for each UAV is 200 Watt; the UAV production cost is $\{0.0075, 0.005, 0.004\}$ Euro/minute, and the maintenance cost is $\{0.0114, 0.0075, 0.0054\}$ Euro/minute; the transmission rate of wireless channel can be set by $\{4911, 4563, 4011\}$ MB/minute according to [35]. As demonstrated by [37], the real-world traffic topology has no significant change in 10 minutes, thus we set the duration of each time slot by 10 minutes.

We utilize multi-layer perceptions for imitation learning, with four fully connected layers for policy and value networks, respectively. We utilize Kronecker-Factored Approximate Curvature (KFAC) optimizer [38] and Asynchronous Advantage Actor Critic (A3C) technology [39] to train the learning model. For expert policies, we collect observation-action pairs from 100 to 400 episodes [29].

For the designed MILU algorithm, we consider two situations, i.e., two and three existing UAV owners as examples, since they are two typical cases that the Nash equilibrium condition can be expressed in explicit and implicit forms, respectively. In addition, three representative algorithms are compared with the designed MILU algorithm:

- Expert policy: Experts try to solve Problem P4 by the analyzed result in subsection 4.2 based on centralized management. The whole instantaneous system state can be observed by each expert. When two experts exist, they can obtain their Nash equilibrium based on Theorem 6. When three experts exist, they can solve the formulated problem based on the interior-reflective Newton method [40].
- OMD-based solution [41]: A gradient-based policy, which is widely utilized for the Markov game. Agents update their actions by taking steps towards the gradients of their profit functions. We utilize it in our UAV placement issue to maximize the profits of UAV owners.

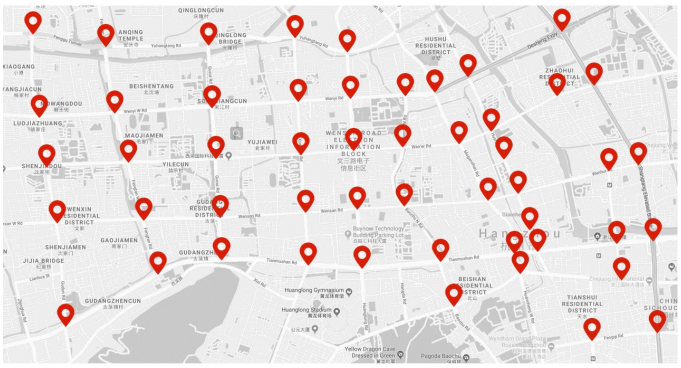


Fig. 3. The illustration of hotspots in Hangzhou, China.

- Multi-agent Deep Deterministic Policy Gradient (MDDPG)-based solution: Similar to [32], we utilize MDDPG in our considered system, where each UAV owner trains its policy based on the actor-critic algorithm with its local observations directly.

5.2 Performance Results

5.2.1 *Impacts of user budgets*

Fig. 4 shows different performance of expert policies, MILU MDDPG-based solution, and OMD-based solution with various values of user budgets. Figs. 4(a)-4(c) show the system performance with two UAV owners, and Figs. 4(d)-4(f) are that with three UAV owners. The average profits illustrated in Fig. 4(a) refer to the long-term average profits that UAV owners win. It is obvious that the performance of expert policy is the best, and the designed MILU algorithm merely has a small gap with that of expert policy. However, the performance of MDDPG-based solution and OMD-based solution is much worse than that of the other two algorithms.

This is because the designed MILU algorithm allows learning agents to mimic the behaviors of experts by minimizing their observation-action distributions and those of experts. Although the MDDPG-based solution also allows each UAV owner to learn for determining its action, it can merely learn based on partial observations without the guideline of experts, resulting in poor performance. The OMD-based solution tries to obtain an estimated gradient in each iteration to help UAV owners make decisions. The gradient is largely dependent on an arbitrary convex function, which has a heavy impact on the algorithm performance. When the user budget increases, the trend of average profits also rises. The reason is that users can pursue more services when their budgets increase, and user utilities as well as profits of UAV owners are positive with user budgets.

Fig. 4(b) shows the average number of provided UAVs in each hotspot, where similar trends with Fig. 4(a) can be found. For instance, when the user budget is 2, the average numbers of provided UAVs of expert policy, MILU, the MDDPG-based solution and the OMD-based solution are 11, 8, 5 and 3, respectively. When the user budget increases to 4, those of the four algorithms are 22, 16, 9 and 4, respectively. This is because expert policy can find the

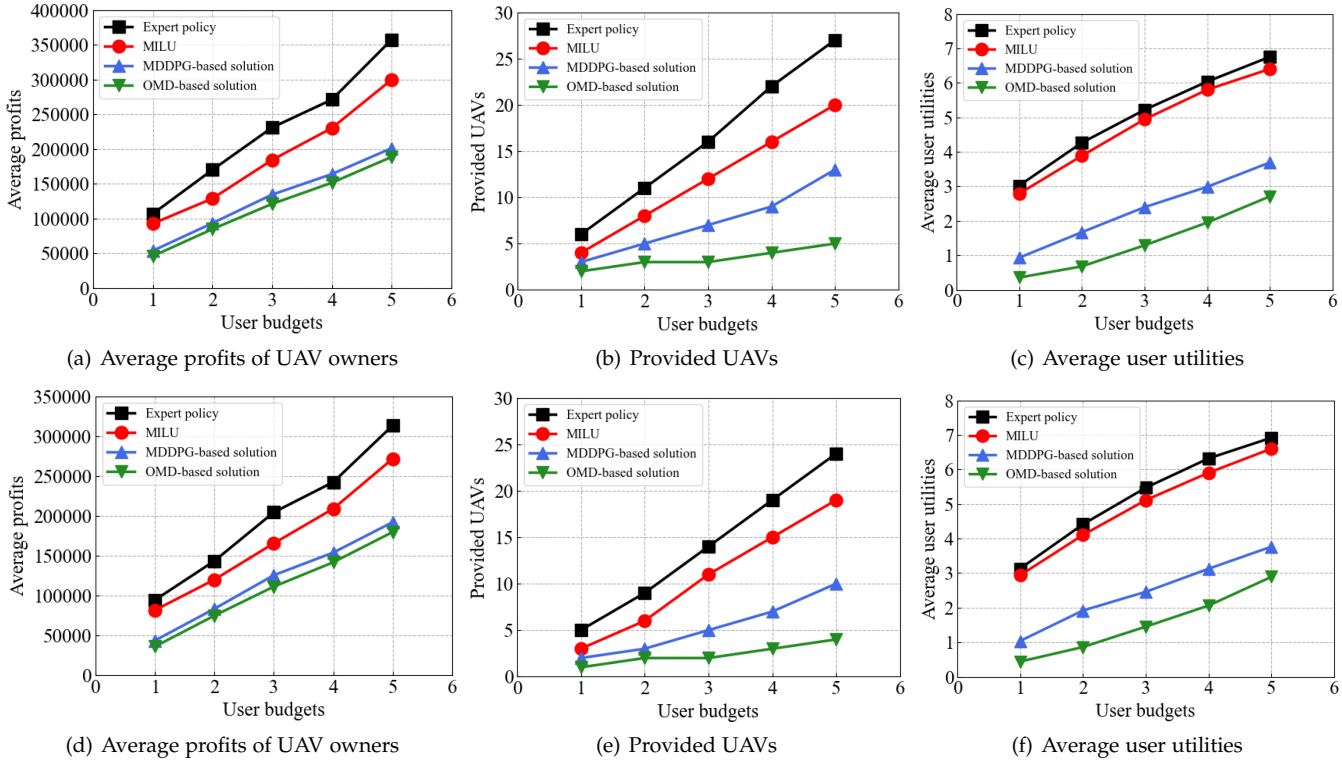


Fig. 4. Performance with different user budgets: a), b) and c) with two UAV owners; c), d) and e) when three UAV owners.

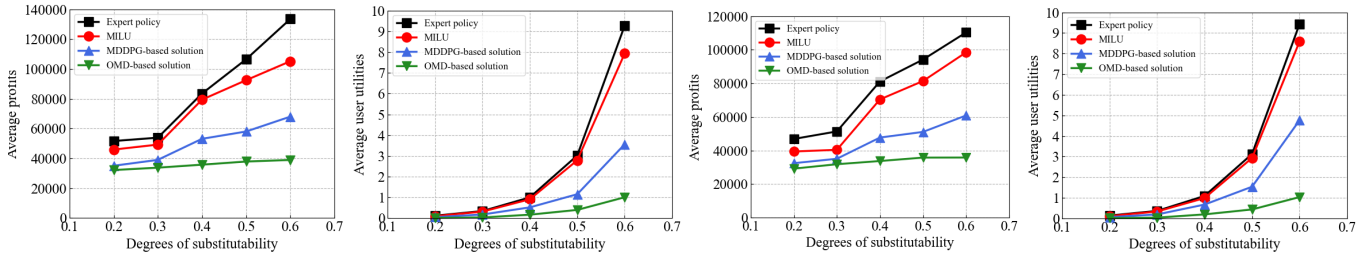
optimal solution based on centralized information, while the designed MILU algorithm obtains a sub-optimal solution caused by the insufficient number of UAVs. The MDDPG-based solution is inefficient in our considered system model since it can be merely trained based on partial observations of learning agents without any global information. The OMD-based solution performs much worse and cannot find the suitable quantities of provided UAVs based on its updated gradient. The performance of average user utilities is illustrated in Fig. 4(c). Expert policy can also guarantee the user utilities based on its optimal choices, while the performance of MILU has a small gap with that of the corresponding expert policy.

Figs. 4(d)-4(f) have similar trends with Figs. 4(a)-4(c), while the performance of average profits shown in former figures is better than that in latter ones. This is because when there are more UAV owners competing in the game, on-ground users have more choices to meet their demands, and profits of UAV owners become less. In addition, average user utility and provided number of UAVs of MILU with two UAV owners are worse than those of MILU with three UAV owners. This is because users have more choices to be served when the number of UAV owners is large in the system.

5.2.2 *Impacts of substitutability degree*

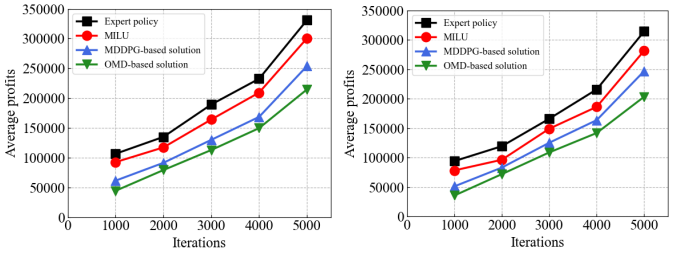
The performance of expert policy, MILU, the MDDPG-based solution and the OMD-based solution with different degrees of substitutability is illustrated in Fig. 5. Figs. 5(a) and 5(b) show the performance with two UAV owners, and Figs. 5(c) and 5(d) are that with three UAV owners. If the degree of substitutability is high, one service can be replaced by other services with high ratios, i.e., users can accept other similar services provided by UAV owners even with high prices, and verse visa. From Fig. 5(a), we can observe that the performance of MILU is much better than that of the MDDPG-based solution and OMD-based solution, and close to that of expert policy. This is because expert policy can find the best policy to reach the Nash equilibrium based on the full information of the system. However, MILU schedules available resources based on partial observations of the system. In addition, more UAV owners result in more fierce competition, and less profits can be gained by each UAV owner. When the degree of substitutability increases, the performance trends of the four algorithms also rise. The reason is that the profit function of UAVs has a positive correlation with the degree of substitutability. UAV owners can increase their prices to a certain extent with the purpose of reaching the Nash equilibrium.

The average user profits are illustrated in Fig. 5(b). We notice that when the degree of substitutability becomes large, the average user utilities increase. For example, when the degree is 0.3, average user utilities achieved by expert policy, MILU, MDDPG-based solution and OMD-based solutions are 0.344, 0.318, 0.174 and 0.037, respectively. When the degree increases to 0.6, the corresponding values are 9.27, 7.93, 3.561 and 1.01, respectively. This is because the fierce competition among different UAV owners affects their costs and prices. When users have more choices, they prefer to purchase services that can maximize their own utilities. Meanwhile, the designed MILU algorithm not only minimizes the gaps between observation-action distributions of agents and those of experts, but also estimates the resource quantities provided by other UAV owners to improve their



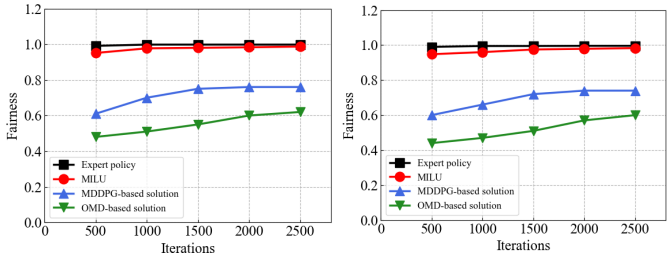
(a) Average profits of UAV owners (b) Average user utilities (c) Average profits of UAV owners (d) Average user utilities

Fig. 5. Performance with different degrees of substitutability: a) and b) with two UAV owners; c) and d) with three UAV owners.



(a) Average profits of two UAV owners (b) Average profits of three UAV owners

Fig. 6. Performance with different iterations.



(a) Faireness with two UAV owners (b) Faireness with three UAV owners

Fig. 7. Performance of fairness.

performance. However, the OMD-based solution merely utilizes an estimated gradient shaped by an arbitrary update function to help UAV owners improve their performance, which has a heavy dependence on the initial input values. Although the learning agents in the MDDPG-based solution train their own learning models based on actor-critic mechanism similar with ours, their actions are not optimal based on their independent training processes without the global information. In addition, Figs. 5(c) and 5(d) have similar trends with Figs. 5(a) and 5(b), which have the same reasons with Figs. 4(d)-4(f).

5.2.3 Impacts of iterations

Fig. 6 displays the performance of expert policy, MILU, MDDPG-based and OMD-based solutions with different numbers of iterations. Fig. 6(a) is the trend of average profits of the four algorithms with two UAV owners. It is obvious that the performance of MILU algorithm is very close to that of expert policy, e.g., the performance gap between expert policy and MILU is merely around 10%. This is because learning agents in MILU mimic the behaviors of experts, and can train their learning models offline based on the observation-action pairs of experts. Meanwhile, the average profit increases when the number of iterations becomes large. For example, when the number of iterations is 2000, average profits of the four algorithms are 134876, 117226, 91321 and 79154. When the number of iterations increases to 4000, those of the four algorithms are 232521, 209246, 168455 and 149542. This is because with time goes by, the accumulated profits increase for all the four algorithms. Fig. 6(b) is average profits of the involved three UAV owners with different iterations, the trend of which is similar with Fig. 6(a), while the overall performance is slightly worse than that of Fig. 6(a).

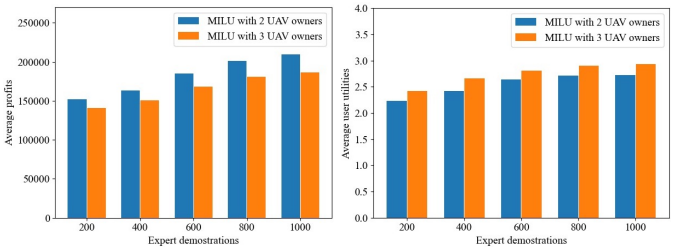
5.2.4 Fairness

Similar to [11], the fairness of profits among different UAV owners can be obtained by:

$$F(t) = \frac{\left(\sum_{k=1}^{K}\sum_{t'=0}^{t}\sum_{h=1}^{H}\Gamma_{hk}(t')\right)^2}{K\sum_{k=1}^{K}\left(\sum_{t'=0}^{t}\sum_{h=1}^{H}\Gamma_{hk}(t')\right)^2}, \quad (25)$$

where $F(t)$ reflects the fairness level of UAV owners' profits. When the profits are balanced among UAV owners, the value of $F(t)$ is close to 1. We evaluate the performance of fairness in Fig. 7, where Fig. 7(a) shows the fairness of profits between two UAV owners, and Fig. 7(b) is that among three UAV owners. From Fig. 7(a), we can observe that the fairness achieved by expert policy is the best, since it can obtain the global information and the optimal service quantities for UAV owners. The performance of MILU algorithm is the second, since learning agents can train their models by sampling observation-action pairs from expert demonstration. The performance of MDDPG-based and OMD-based solutions is far from that of expert policy and MILU algorithm. This is because the MDDPG-based solution lets learning agents to learn their own policies without the interaction of each other, leading to poor fairness with their local observations. The OMD-based solution allows UAV owners to improve their performance based on a shaped gradient. Similar with the MDDPG-based solution, the global information is missed and the gradient may deviate from the optimal direction.

Similar with Fig. 7(a), Fig. 7(b) shows the fairness of profits among three UAV owners. We can observe that the fairness of the four algorithms improves with the increasing number of iterations. For example, when the number of iterations is 1000, the fairness of expert policy, MILU, MDDPG-



(a) Average profits of UAV owners (b) Average user utilities

Fig. 8. Performance with different number of expert demonstrations.

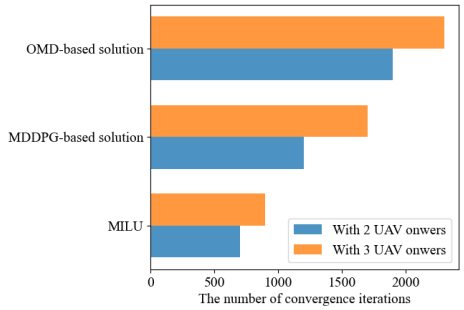


Fig. 9. Convergence iterations.

based and OMD-based solutions is 0.995, 0.96, 0.66 and 0.47, respectively. When the number of iterations increases to 2000, the fairness of the four algorithms is 0.996, 0.979, 0.74 and 0.57, respectively. This is because learning agents have more knowledge of the system based on the iteration with the environment.

5.2.5 Impacts of expert demonstrations

As shown in Fig. 8, the impact of expert demonstrations on the system performance is illustrated with MILU in two cases, i.e., with two and three UAV owners, since they are based on imitation learning and utilize expert demonstrations for policy training. The horizontal axis refers to the number of samples contained in the expert demonstration. Fig. 8(a) is the performance trend of average profits of UAV owners. We notice that when the number of samples in the expert demonstration becomes big, the average profits of UAV owners also become large. This is because agents can have more sampled observation-action pairs to mimic the behaviors of experts. When the number of samples in the expert demonstration increases to 800, the performance trend tends to be gentle. The reason is the number of samples is enough for the agent to minimize the distribution of their observation-action pairs and that of experts.

The average user utilities based on different numbers of samples in the expert demonstration are shown in Fig. 8(b). The user utilities increase when the number of samples in the expert demonstration becomes large. The reason is similar with that of Fig. 8(a). When the number of samples in the expert demonstration reaches to 800, the agent is capable to reach the minimum performance gap with that of experts based on the integration of GAN and the policy gradient method.

5.2.6 Convergence

Fig. 9 shows the convergence iterations of MILU, MDDPG-based and OMD-based solutions with two and three UAV owners, respectively. We can observe that MILU has the fast convergence speed, while the speeds of MDDPG-based and OMD-based solutions are lower. For example, when two UAV owners exist, the convergence iterations of MILU, MDDPG-based and OMD-based solutions are 700, 1200 and 1900, respectively. This is because MILU algorithm learns efficient policies not only by interacting with the environment, but also follows experts' demonstrations, relieving the drawbacks of partial observations. As a result, MILU algorithm can converge fast. However, MDDPG-based and OMD-based solutions do not have efficient mechanisms to improve their performance by obtaining more knowledge of the system, thus they can merely learn based on their local observations, leading to lower convergence speeds.

When there are three UAV owners, the number of convergence iterations is larger than that with two UAV owners. For example, the convergence iteration of MILU is 700 with two UAV owners, and becomes 900 with three UAV owners. This is because when there are more UAV owners in the system, more calculation and competitions are required, taking more iterations for convergence.

6 Conclusion

We established a UAV-based system model to enable differentiated services provided by different UAV operators. With the purpose of both maximizing the utilities of users and the profits of UAV owners, we proposed an imitation learning enabled UAV deployment algorithm. Initially, we analyzed the Nash equilibrium condition with full knowledge of the system state, based on which we derived expert policies utilized in our imitation learning enabled algorithm. Then, agent policies were designed by minimizing the gaps between their distributions of observation-action pairs and those of experts, which can be trained and executed in a fully decentralized manner even with partial observations. Performance results showed that our algorithm has significant advantages on various metrics, such as average profits, average user utilities and execution time, compared with other representative algorithms.

In the future work, we prepare to extend our system model to a 3D environment, where not only differentiated services but also spatial impacts, such as shadowing and blockage [42], are considered. As a result, the trajectory design for UAVs as well as the optimal provided services for different service providers need to be jointly optimized.

References

[1] F. Khan, "Multi-comm-core architecture for terabit-per-second wireless," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 124–129, 2016.

[2] X. Wang and L. Duan, "Dynamic pricing and capacity allocation of UAV-provided mobile services," in *Proc. IEEE INFOCOM*, pp. 1855–1863, 2019.

[3] Uavs.org, "Unmanned aerial vehicle systems association commercial applications." https://www.uavs.org/commercial, 2016.

[4] A. Hanscom and M. Bedford, "Unmanned aircraft system (UAS) service demand 2015–2035, literature review & projections of future usage," *Res. Innov. Technol. Admin.*,

US Dept. Transp., Washington, DC, USA, Tech. Rep. DOT-VNTSC-DoD-13-01, 2013.

[5] N. Liba and J. Berg-Jürgens, "Accuracy of orthomosaic generated by different methods in example of UAV platform MUST Q," in *Proc. IOP Conference Series: Materials Science and Engineering*, vol. 96, p. 8, 2015.

[6] Z. Ning, P. Dong, X. Wang, X. Hu, J. Liu, L. Guo, B. Hu, R. Kwok, and V. C. Leung, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.3025116*, 2020.

[7] C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1342–1355, 2008.

[8] J. Clement, "Internet usage worldwide-statistics & facts," 2020.

[9] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *Proc. IEEE/ACM IWQoS*, pp. 1–6, 2017.

[10] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 12–26, 2002.

[11] 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 Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 73–84, 2020.

[12] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2017.

[13] X. Liu, Y. Liu, N. Zhang, W. Wu, and A. Liu, "Optimizing trajectory of unmanned aerial vehicles for efficient data acquisition: A matrix completion approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1829–1840, 2019.

[14] X. Xu, L. Duan, and M. Li, "Strategic learning approach for deploying UAV-provided wireless services," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2019.2953726*, 2019.

[15] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming," *IEEE Network*, vol. 33, no. 3, pp. 42–48, 2019.

[16] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[17] R. K. Sheshadri, E. Chai, K. Sundaresan, and S. Rangarajan, "SkyHaul: An autonomous gigabit network fabric in the sky," *arXiv preprint arXiv:2006.11307*, 2020.

[18] C. H. Liu, Z. Dai, Y. Zhao, J. Crowcroft, D. O. Wu, and K. Leung, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 130–146, 2021.

[19] 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 of Things Journal*, vol. 6, no. 5, pp. 8753–8769, 2019.

[20] L. Bertizzolo, S. D'Oro, L. Ferranti, L. Bonati, E. Demirors, Z. Guan, T. Melodia, and S. Pudlewski, "SwarmControl: An automated distributed control framework for self-optimizing drone networks," in *Proc. IEEE INFOCOM*, pp. 1768–1777, 2020.

[21] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.3012509*, 2020.

[22] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[23] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.

[24] T. Kimura and M. Ogura, "Distributed collaborative 3D-deployment of UAV base stations for on-demand coverage," in *Proc. IEEE INFOCOM*, pp. 1748–1757, 2020.

[25] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6898–6908, 2020.

[26] P. Wu, F. Xiao, H. Huang, and R. Wang, "Load balance and trajectory design in multi-UAV aided large-scale wireless rechargeable networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13756–13767, 2020.

[27] A. Agliari, A. K. Naimzada, and N. Pecora, "Nonlinear dynamics of a cournot duopoly game with differentiated products," *Applied Mathematics and Computation*, vol. 281, pp. 1–15, 2016.

[28] T. Zhu, J. Li, Z. Cai, Y. Li, and H. Gao, "Computation scheduling for wireless powered mobile edge computing networks," in *Proc. IEEE INFOCOM*, pp. 596–605, 2020.

[29] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," in *Proc. Advances in Neural Information Processing Systems*, pp. 7461–7472, 2018.

[30] M. Liu, M. Zhou, W. Zhang, Y. Zhuang, J. Wang, W. Liu, and Y. Yu, "Multi-agent interactions modeling with correlated policies," in *Proc. ICLR*, pp. 1–20, 2020.

[31] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Proc. Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

[32] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1262–1276, 2019.

[33] "Video trace library." http://trace.eas.asu.edu/, 2010.

[34] K. Kavvadias, "Energy price spread as a driving force for combined generation investments: A view on Europe," *Energy*, vol. 115, pp. 1632–1639, 2016.

[35] W. J. Fredericks, M. D. Moore, and R. C. Busan, "Benefits of hybrid-electric propulsion to achieve 4x cruise efficiency for a VTOL UAV," in *Proc. International Powered Lift Conference*, p. 4324, 2013.

[36] E. Borgogno Mondino and M. Gajetti, "Preliminary considerations about costs and potential market of remote sensing from UAV in the italian viticulture context," *European Journal of Remote Sensing*, vol. 50, no. 1, pp. 310–319, 2017.

[37] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[38] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *Proc. ICML*, pp. 2408–2417, 2015.

[39] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. ICML*, pp. 1928–1937, 2016.

[40] T. F. Coleman and Y. Li, "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, no. 1-3, pp. 189–224, 1994.

[41] Z. Zhou, P. Mertikopoulos, A. L. Moustakas, N. Bambos, and P. Glynn, "Mirror descent learning in continuous games," in *Proc. IEEE CDC*, pp. 5776–5783, 2017.

[42] 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 Transactions on Mobile Computing, DoI:10.1109/TMC.2021.3059691*, 2021.

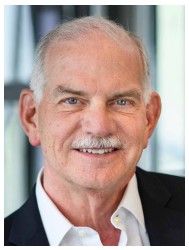
Xiaojie Wang (M'19) received the PhD degree from Dalian University of Technology, Dalian, China, in 2019. After that, she was a postdoctor in the Hong Kong Polytechnic University. Currently, she is a distinguished professor with the College of Communication and Information Engineering, the Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests are wireless networks, mobile edge computing and machine learning. She has published over 40 scientific papers in international journals and conferences, such as IEEE JSAC, IEEE TMC, IEEE TPDS and IEEE COMST.

Zhaolong Ning (M'14-SM'18) received the Ph. D. degree from Northeastern University, China in 2014. He was a Research Fellow at Kyushu University from 2013 to 2014, Japan. Currently, he is a full professor with the College of Communication and Information Engineering, the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include Internet of things, mobile edge computing, deep learning, and resource management. He has published over 120 scientific papers in international journals and conferences. Dr. Ning serves as an associate editor or guest editor of several journals, such as IEEE Transactions on Industrial Informatics, IEEE Transactions on Social Computational Systems, The Computer Journal and so on.

Song Guo (M'02-SM'11-F'20) is a Full Professor in the Department of Computing at The Hong Kong Polytechnic University. He also holds a Changjiang Chair Professorship awarded by the Ministry of Education of China. His research interests are mainly in the areas of big data, edge AI, mobile computing, and distributed systems. He co-authored 4 books, co-edited 7 books, and published over 500 papers in major journals and conferences. He is the recipient of over 12 Best Paper Awards from IEEE/ACM conferences, journals and technical committees. His work was also recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. Prof. Guo's research has been sponsored by RGC, NSFC, MOST, industry, etc. He is the Editor-in-Chief of IEEE Open Journal of the Computer Society and the Chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He was an IEEE ComSoc Distinguished Lecturer and a member of IEEE ComSoc Board of Governors. He has also served for IEEE Computer Society on Fellow Evaluation Committee, Transactions Operations Committee, Editor-in-Chief Search Committee, etc. Prof. Guo has been named on editorial board of a number of prestigious international journals like IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Cloud Computing, IEEE Internet of Things Journal, etc. He has also served as chairs of organizing and technical committees of many international conferences. Prof. Guo is an IEEE Fellow, a Highly Cited Researcher (Web of Science), and an ACM Distinguished Member.

Miaowen Wen (SM'18) received the Ph.D. degree from Peking University, Beijing, China, in 2014. From 2012 to 2013, he was a Visiting Student Research Collaborator with Princeton University, Princeton, NJ, USA. He is currently an Associate Professor with South China University of Technology, Guangzhou, China, and a Hong Kong Scholar with The University of Hong Kong, Hong Kong. He has published a Springer book entitled Index Modulation for 5G Wireless Communications and more than 80 journal papers. His research interests include a variety of topics in the areas of wireless and molecular communications. Dr. Wen was the recipient of four Best Paper Awards. Currently, he is serving as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and the Physical Communication (Elsevier), and a Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING.

Lei Guo received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a full professor with Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical papers in international journals and conferences. He is an editor for several international journals. His current research interests include communication networks, optical communications, and wireless communications.



H. Vincent Poor (M'77-SM'82-F'87)) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. During 2006 to 2016, he served as Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the forthcoming book *Machnie Learning and Wireless Communications* (Cambridge University Press, 2021).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.