Completion Time Minimization for Data Collection in a UAV-enabled IoT Network: A Deep Reinforcement Learning Approach

Shuai Zhang, *Member, IEEE*, Weiqi Liu, *Graduate Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—In this article, we study the completion time minimization in an unmanned aerial vehicle (UAV)-enabled Internet of Things (IoT) network, where the UAV tries to collect all the data generated by the ground IoT devices for further processing. To simplify the analysis, the continuous time horizon is discretized into several time slots. The duration of each time slot is set to be less than the pre-defined threshold such that the UAV's location can be considered as unchanged during each time slot. In our work, we aim to minimize the completion time of the UAV by optimizing the association scheme of the IoT devices, the location (i.e., the trajectory) and velocity of the UAV at each time slot. However, the formulated problem is challenging to solve by traditional optimization methods considering the unknown number of time slots (which leads to the unknown number of decision variables) and non-convex functions. We thus reformulate it as a Markov decision process (MDP) and propose a deep deterministic policy gradient (DDPG)-based method to efficiently solve it. The DDPG-based algorithm uses deep function approximators instead of finding the action that maximizes the state-action value, and is therefore well suited to solve high-dimensional, continuous control problems. Extensive numerical results are presented to validate the effectiveness of our proposed algorithm.**

***Index Terms*—Unmanned aerial vehicle (UAV), deep reinforcement learning (DRL), trajectory optimization.**

I. Introduction

RECENTLY, Internet of Things (IoT) has attracted increasing interest in applications such as public safety, environment monitoring, intelligent agriculture, smart homes and smart cities [1], [2], [3]. All these applications involve data collection for centralized processing owing to the limited computation power and on-board energy of the IoT nodes. However, the data collection suffers from lack of surrounding terrestrial communication infrastructures or bad channel conditions from the IoT nodes to the base stations (BSs) since the IoT nodes are usually deployed in remote areas.

Drone, or unmanned aerial vehicle (UAV) [4], [5], [6], [7], [8], with the advantage of high mobility and flexible deployment, is capable of moving close to the IoT nodes and establishing the favorable line of sight (LoS) channels with a larger probability as it flies at a higher altitude as compared to the ground BSs. Hence, UAV-enabled IoT is considered as a promising solution to address the above challenges. By properly designing the association scheme and defining the trajectory of the UAV (the UAV's trajectory refers to the path taken by the UAV to perform a task), the ground IoT nodes can choose to transmit the data when the channel condition is better, thus greatly reducing the communication energy consumption.

The miniaturization of the communication equipment and the development of UAVs enable the paradigm of UAV-assisted communications by which the UAV serves as a temporary base station (BS) to replace the malfunctioned macro base station (MBS) or the one that is completely destroyed in a disaster, or as a relay node between the ground IoT nodes and the MBS to improve the throughput performance by leveraging the favorable LoS links. By mounting a receiving node, the UAV can also be dispatched to collect data generated by the IoT nodes, which are deployed in sparse or remote areas. One of the main challenges of the UAV-enabled IoT network is the UAV trajectory design to optimize the objective function (e.g., system throughput maximization, energy consumption minimization or completion time minimization). The formulated optimization problem is generally non-convex since the data rate is non-convex w.r.t. the UAV's location at each time slot (Note that the time horizon is discretized into a finite number of time slots to make the formulated problem more tractable. Otherwise, it will involve an infinite number of optimization variables in the continuous time horizon). To solve the formulated non-convex problem, the successive convex approximation (SCA) method [9], [10], [11] is usually utilized to transform the primal problem into a convex one. Specifically, the above works first prove the non-convex functions to be convex/concave w.r.t. an intermediate variable and then apply the first order Taylor expansion to obtain their corresponding lower/upper bounds. Then, they prove the lower/upper bounds to be convex/concave w.r.t. the UAV's locations. Finally, by replacing the non-convex functions with their convex approximations, the primal problem is transformed into a convex one and therefore can be solved by many off-the-shelf

Manuscript received 16 February 2023; revised 18 April 2023; accepted 23 May 2023. Date of publication 2 June 2023; date of current version 14 November 2023. This work was supported in part by U.S. National Science Foundation under Grant CNS-1814748. The review of this article was coordinated by Dr. Lin X Cai. *(Corresponding author: Shuai Zhang.)*

Shuai Zhang was with the New Jersey Institute of Technology, Newark, NJ 07102 USA. He is now with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 164 40 Stockholm, Sweden (e-mail: shuai2@kth.se).

Weiqi Liu and Nirwan Ansari are with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: wl296@njit.edu; nirwan.ansari@njit.edu).

Digital Object Identifier 10.1109/TVT.2023.3280848

softwares (e.g., CVX). Obviously, the optimality of the solution obtained by this method cannot be guaranteed.

Deep reinforcement learning (DRL) has been proved to be an effective tool to solve problems which cannot be easily handled by traditional optimization methods [12], [13], [14], [15], [16]. Reinforcement learning, which learns while interacting with the environment, is capable of obtaining the near-optimal solution in a trial and error manner [17], [18], [19], [20]. However, it is infeasible to explore all states or actions as the state space or action space increases rapdily. An artificial neural network (ANN), which approximates the optimal state-value function by minimizing the loss function (or the reward prediction error) [21], can generalize what it has learned from past experience. Once the learning procedure is completed, the ANN with the weights of neurons in each layer fixed is ready for carrying out the application.

Various works, using traditional optimization methods or DRL-based methods, have been done to study the UAV-enabled IoT network. However, some challenges still need to be tackled.

1) What are the optimal trajectory and velocity of the UAV?

The trajectory design of the UAV directly influences the maximal achievable rates of the IoT nodes at each time slot. An optimized trajectory should approach close to each IoT node to reduce the data offloading time and therefore the whole completion time. Even the trajectory of the UAV is given (note that the UAV's trajectory in the discrete time horizon is determined by the UAV's location at each time slot), the velocity of the UAV during each time slot, which determines the duration of this time slot, also affects how many data can be collected. Moreover, the trajectory and velocity of the UAV are coupled together and should be jointly optimized to minimize the completion time.

2) What is the optimal association policy of the IoT nodes?

During the whole completion time, the IoT nodes need to determine which time slots to offload their generated data. A proper association scheme should try to transmit the data when the channel conditions are good to improve the data rate and thus increase the amount of offloaded data. Even given the trajectory of the UAV, different association policies result in different completion time. In the discrete time horizon, the trajectory of the UAV is composed of several location points. Once the UAV's trajectory is determined, the distance between each pair of successive location points is also given. Then, the time required for the UAV to fly from one location point to the next location point, i.e., the duration of this time slot, is determined by the UAV's flying velocity. The trajectory and velocity of the UAV jointly determine the duration of each time slot, and further the completion time of the journey. Therefore, the trajectory and velocity of the UAV and the association policy of the IoT nodes should be jointly optimized to minimize the completion time.

To address the above challenges and cope with the continuous and unlimited state and action spaces, we propose a deep deterministic policy gradient (DDPG)-based algorithm to solve the formulated problem. To our best knowledge, no one has studied the completion time minimization problem by designing the UAV's trajectory with unknown number of time slots and unequal durations of the time slots. Existing works assume either the number of time slots is known or the duration of each time slot is constant (the same). Therefore, our generalization of these two assumptions is more practical and could potentially achieve a better solution, which is also validated in the simulation results. We summarize the main contributions of this article as follows:

1) We formulate a completion time minimization problem in a UAV-enabled IoT network, where the data generated by the IoT nodes are offloaded to the UAV for centralized processing. Different from prior works, we assume the number of time slots is unknown (i.e., the number of decision variables or the action space is unlimited), which poses a great challenge to solve the problem via traditional optimization methods. We also assume the duration of each time slot is not equal or fixed. Most existing works [22], [23] assume the time durations are equal or fixed. These two assumptions avoid the optimization of time durations, thus reducing the number of optimization variables. The purpose of doing so is to bypass the optimization of time length and reduce complexity. Our nullification of these two assumptions make the problem more practical yet more challenging to solve.

2) We reformulate the primal problem as a Markov decision process (MDP). To deal with the heterogeneity of the action space (note that the trajectory and velocity of the UAV are continuous while the association indicator is integer), we relax the association variables into continuous ones between 0 and 1 and then quantize them to 0 or 1. To further cope with the continuous and unlimited action and state spaces, we propose a DDPG-based algorithm to obtain the near-optimal solution.

3) The performance of the proposed algorithm is evaluated extensively by demonstrating the impact of different parameters. In addition, the influence of the hyperparameters of the neural network is also given. The DDPG-based algorithm achieves better performance as compared with two other benchmark algorithms.

The remainder of this article is organized as follows. We summarize the related works in Section II. The system model of data collection in a UAV-enabled IoT network and the completion time minimization problem are presented in Section III. In Section IV, we reformulate the problem as an MDP and propose a DDPG-based algorithm to obtain the near-optimal solution. In Section V, simulations are conducted to demonstrate the effectiveness of the proposed algorithm. Finally, Section VI concludes this article.

II. Related Works

There have been various works investigating the trajectory design in a UAV-enabled IoT network with different objectives. However, the trajectory planning of the UAV under practical assumptions (i.e., unknown number of time slots and/or unequal durations of the time slots) remains challenging. Wang et al. [24] proposed to employ wireless power transfer (WPT) and wireless information transfer (WIT) to charge ground terminals (GTs) and collect their generated data, respectively. They aimed to minimize the completion time by designing the trajectory of the UAV and the data offloading scheme of the GTs. To solve the formulated non-convex problem, the SCA method is used to convert the non-convex constraints into convex ones. Xu et al.

[25] investigated the deployment of multiple UAVs to assist mobile edge computing (MEC) with the objective of minimizing the task completion under both partial offloading mode and binary offloading mode. To tackle the non-convexity of the formulated problem, they proposed a SCA-based algorithm which decomposes the primal problem into three subproblems and solves them in an alternating manner. Cai et al. [26] proposed a dual-UAV-enabled communication system where one UAV flies around the target area to provide coverage service to the ground users while the other one jams the eavesdropper to secure the communications with the desired users. They tried to maximize the minimum secrecy rate of all users by jointly optimizing the two UAVs' trajectories and the user association. A penalty concave-convex procedure based algorithm is designed to address the challenging optimization problem. Gupta et al. [27] studied the deployment of a UAV to provide coverage service to the ground users in an on-demand manner. They aimed to maximize the sum-rate of all users subject to the UAV's limited energy and flight duration constraints. Huang et al. [28] proposed to combine non-orthogonal multiple access (NOMA) with a rotary wing UAV to achieve better performance of wireless networks. The ground users are first divided into multiple clusters by the K-means algorithm and the UAV's trajectory is then optimized by the genetic algorithm. However, their developed algorithms do not apply to the cases where users cannot be easily clustered into different groups. Su et al. [29] studied a novel intelligent reflecting surface (IRS)-assisted UAV communication system where the IRS is mounted on a fixed-wing UAV to adaptively reflect the signal transmitted from the BS. The spectral efficiency (SE) maximization and energy efficiency (EE) maximization problems are formulated by jointly optimizing the active beamforming, passive beamforming scheme and the UAV trajectory. SCA-based algorithms are designed to tackle the non-convexity of these two problems.

DRL, with its decision-making ability and the advantage of solving problems with high-dimension or even unlimited/continuous action spaces by leveraging ANNs, has been proposed as an effective tool to address the trajectory design problem in UAV-enabled communications. The DRL method tries to obtain a policy that maximizes the long term reward by flexibly interacting with the environment. Nguyen et al. [30] proposed a DRL-based algorithm to obtain an optimal trajectory and resource allocation scheme in a UAV-assisted IoT network, subject to the limited on-board power. However, they assumed the action space (i.e., the UAV's flight direction) of the MDP only consists of six discrete directions, but this is not practical in real application scenarios. Ding et al. [31] investigated the fair throughput maximization for UAV-assisted communications by designing the 3D trajectory of the UAV and the band allocation among users. They further proposed a deep deterministic policy gradient (DDPG)-based algorithm to solve the formulated NP-hard problem. However, they assumed the duration of each time slot to be equal. Saxena et al. [32] proposed a flow-level model to evaluate the performance of UAV base station networks. Based on the model, a DRL-based algorithm is designed to maximize the flow- and system-level metrics by learning the near optimal UAV trajectory. However, they assume the UAV can only move to six available directions with pre-given fixed number of trajectory steps. Hsu and Gau [33] studied the trajectory planning in a multiple UAVs-enabled communication network where the UAVs deliver objects in the forward path and collect data in the backward path. They utilized the RL approach to learn collision avoidance conditions and the traditional optimization theory to obtain the shortest backward path, respectively.

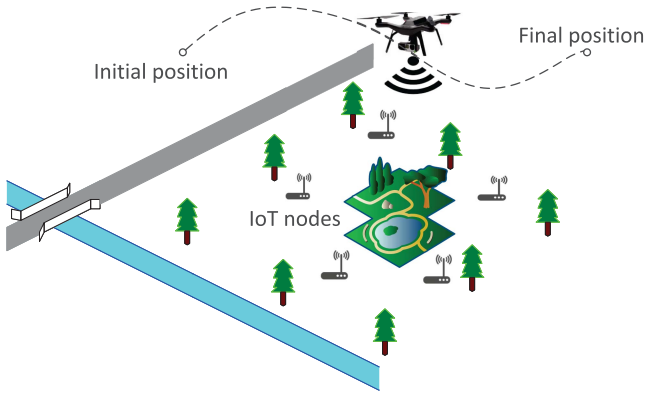

Fig. 1. The UAV-enabled IoT network.

Note that all the works mentioned above assume pre-given number of time slots and/or fixed time durations to make the formulated problem more tractable. Theoretically, with the assumption of a pre-given number of time slots, one can still adjust the total completion time of the UAV by changing the duration of each time slot. However, the duration of each time slot cannot be arbitrarily set, considering the assumption of a small enough time duration to ensure roughly fixed location of the UAV during a certain time slot.

Different from the above works, we assume the number of time slots to be unknown and the duration of each time slot to be non-fixed. Note that the first assumption leads to an unknown number of variables and the second assumption brings in more decision variables which are set to be known *a priori* in some existing works [22], [23] to make the formulated problem easier to solve. These two assumptions make the problem more practical yet more difficult to solve. Our objective is to minimize the completion time by jointly optimizing the trajectory and velocity of the UAV and the association scheme of the IoT nodes while considering the data collection completion and the UAV's maximum allowed velocity constraints.

III. System Model

As shown in Fig. 1, a UAV is dispatched to collect the data generated by the ground IoT nodes in the target area for further processing. We assume the locations of the ground IoT nodes are known *a priori* and fixed in the completion time. Note that the locations of the IoT nodes can be calculated by obtaining the received signal strength (RSS) of the UAV transmitted from the IoT nodes. Denote the 3D coordinate of the i-th IoT node as $(x_i, y_i, 0), \forall i \in \mathcal{I}$, where $\mathcal{I}$ is the set of ground IoT nodes. To simplify the problem, the continuous time horizon is divided into T (T is unknown and needs to be determined)

time slots and the UAV's location is assumed to be fixed during each time slot. In this way, we only need to decide on a finite number of variables in the discrete space, instead of an infinite number of variables in the continuous space. At each time slot, we assume that only one IoT node can be connected to the UAV, but during the entire data collection process, multiple time slots may be allocated to an IoT node, and these slots may not be consecutive. We further assume the duration of each time slot is small enough such that the distance the UAV moves is short enough to ensure the assumption that the UAV's position can be seen as unchanged during each time slot. Denote δ_t, $1 \le t \le T$, as the duration of the t-th time slot. Note that the duration of each time slot is unequal and can be calculated by

$$\delta(t) = \frac{q(t) - q(t-1)}{v(t)},\ 1 \le t \le T, \tag{1}$$

where $q(t) = (x(t), y(t), H)$ and $v(t)$ are the 3D location and velocity of the UAV at the t-th time slot, respectively. H denotes the fixed flight height of the UAV. The duration of each time slot is assumed to be small enough such that the UAV's location is considered to be unchanged within each time slot. Once the data generated by all IoT nodes are successfully collected by the UAV, the whole mission is said to be completed and the corresponding completion time can be obtained as $\sum_{t=1}^{T} \delta(t)$.

In this section, we present the channel pathloss and data rate model between the IoT nodes and the UAV. Then, we formulate the problem with the aim of minimizing the completion time of the UAV, subject to the data collection and the UAV's maximum allowed velocity constraints.

A. Pathloss model between the IoT nodes and the UAV

We assume the channels between the ground IoT nodes and the UAV are line of sight (LoS)-dominated. Thus, the pathloss between IoT node i and the UAV at time slot t can be obtained by using the free space path loss model [34], [35]

$$\begin{aligned}\xi_i(t) &= \beta_0 \left[d_i(t)\right]^2 \\ &= \beta_0 \left[(x(t) - x_i)^2 + (y(t) - y_i)^2 + H^2\right],\end{aligned} \tag{2}$$

where β_0 is the pathloss value at the reference distance of 1 m, and $d_i(t)$ is the distance between the i-th IoT node and the UAV at time slot t. Then, the maximum achievable rate when the i-th IoT node offloads data to the UAV at time slot n can be calculated as

$$R_i(t) = B \log_2 \left(1 + \frac{P}{\xi_i(t)\sigma^2}\right), \tag{3}$$

where P and B represent the transmit power and bandwidth, respectively. σ^2 denotes the environment noise power at the receiver. We assume only one IoT node can be associated with the UAV at a certain time slot, i.e.,

$$\sum_{i=1}^{|\mathcal{I}|} u_i(t) \le 1,\ 1 \le t \le T, \tag{4}$$

where $u_i(t) = 1$ implies that the i-th IoT node chooses to offload data to the UAV at time slot n and $u_i(t) = 0$ otherwise.

B. Problem formulation

In this section, we formulate the UAV trajectory and velocity planning and IoT nodes association problem for data collection in a UAV-enabled IoT network. Our objective is to minimize the completion time with the consideration of data collection completion and the UAV's maximum allowed velocity. Specifically,

$$\mathbf{P0}: \min_{q(t), v(t), u_i(t)} \sum_{t=1}^{T} \delta(t) \tag{5}$$

$$s.t. \sum_{t=1}^{T} \delta(t) R_i(t) u_i(t) \ge D_i, \forall i \in \mathcal{I}, \tag{6}$$

$$\sum_{i=1}^{|\mathcal{I}|} u_i(t) \le 1,\ 1 \le t \le T, \tag{7}$$

$$0 < \delta(t) \le \delta^m,\ 1 \le t \le T, \tag{8}$$

$$0 < v(t) \le V^m,\ 1 \le t \le T, \tag{9}$$

$$q(0) = Q, \tag{10}$$

$$u_i(t) = \{0, 1\}, \forall i \in \mathcal{I},\ 1 \le t \le T, \tag{11}$$

where D_i denotes the data size of the i-th IoT node, and Q is the initial location where the UAV is released to collect data generated by the IoT nodes. V^m and δ^m denote the maximum velocity of the UAV and the maximum allowed duration of each time slot, respectively. Constraint (6) guarantees the data offloading completion of each IoT node. Constraint (11) imposes u_i to be a binary variable. Note that the unknown T makes problem **P0** intractable since the number of decision variables is unknown. Additionally, **P0** is non-convex because $R_i(t)$ is non-convex w.r.t. $q(t)$. Moreover, the integer variables $u_i(t)$ make **P0** even more challenging to solve. Note that **P0** can be seen as a sequential decision-making problem. However, the location and velocity of the UAV at each time slot and the association scheme are coupled together and should be jointly optimized to obtain the optimal solution, i.e., **P0** cannot be solved by a step-by-step manner. Therefore, it is difficult to solve **P0** using traditional optimization methods [36]. DRL, with its decision-making and powerful data processing capability, can obtain a near-optimal solution of **P0** by transforming the variation sequence to an MDP. Specifically, a DDPG-based algorithm is proposed to address the continuous state and action spaces.

IV. Proposed DDPG-based Algorithm

P0 can be seen as a sequential decision-making problem in a time-varying IoT environment, where the future state depends only on the current state, and not on the past states. These two features, being independent of past states and having a dynamic environment, perfectly fit the definition of MDP. Therefore, we can reformulate it as an MDP and then solve it by utilizing the proposed DDPG-based algorithm.

A. MDP Formulation

Note that **P0** is intractable by using traditional optimization methods due to the unknown number of decision variables. However, the MDP suits **P0** well because the state in the current interval depends only on the state of the preceding interval and the action made in that interval. In order to apply the DDPG-based algorithm, we first define the state space $\mathcal{S}$, the action space $\mathcal{A}$ and the reward $\mathcal{R}$ for UAV trajectory and velocity planning and association design as follows.

1) *State Space:* At the beginning of time slot t, the system state $s(t)$, which is fed into the neural network to determine the next action, includes the current location of the UAV and the amount of data remaining for each IoT node, i.e.,

$$s(t) = [x(t), y(t), D_1(t), D_2(t), \cdots, D_{|\mathcal{I}|}(t)], \tag{12}$$

where $D_i(t)$ is the remaining data size of IoT node i at time slot t. Therefore, the system state space can be expressed as

$$\mathcal{S}(t) = \{s(t) | 0 \le D_i(t) \le D_i, \forall i \in \mathcal{I}, 1 \le t \le T\}. \tag{13}$$

2) *Action Space:* The action of the IoT network $a(t)$ at time slot t determines the location and velocity of the UAV and the association scheme of the ground IoT nodes at time slot $t+1$. Hence, $a(t)$ can be defined as

$$a(t) = [q(t+1), v(t+1), u_1(t+1), \cdots, u_{|\mathcal{I}|}(t+1)]. \tag{14}$$

Note that Constraints (8) and (9) have to be satisfied while taking action $a(t)$. Constraint (8) is equivalent to

$$0 \le \frac{q(t) - q(t-1)}{v(t)} \le \delta^m, \ 1 \le t \le T. \tag{15}$$

Hence, the action space at time slot t can be defined as

$$\mathcal{A}(t) = \{a(t) | (9), (15), 1 \le t \le T\}. \tag{16}$$

3) *Reward:* Note that the reward of the system equals to the negative of the system cost. The generated system cost is defined as the duration of the time slot by taking action $a(t)$, i.e.,

$$c(t) = \delta(t) + M, \tag{17}$$

where M is the penalty incurred if Constraint (8) or (9) is violated. In the dynamic UAV-enabled network, the system state evolves from one time slot to another based on the actions that are already taken in previous time slots. Hence, the data collection evolves as

$$D_i(t+1) = \begin{cases} 0, & \text{if } u_i(t) = 1 \text{ and } \delta(t) R_i(t) \ge D_i(t), \\ D_i(t) - \delta(t) R_i(t), & \text{if } u_i(t) = 1 \text{ and } \delta(t) R_i(t) \le D_i(t), \\ D_i(t), & \text{if } u_i(t) = 0. \end{cases} \tag{18}$$

B. Actor-Critic Deep Reinforcement Learning

As shown in Fig. 2, the Actor-Critic network is a widely adopted approach to solve the continuous control problem which consists of a policy network (actor) and a value network (critic). In the training process, the policy network controls the agent

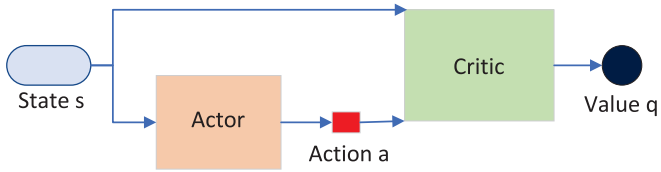

Fig. 2. Actor-Critic network.

(UAV) and output action $\boldsymbol{a}$ based on the system state $\boldsymbol{s}$ while the value network evaluates this action and sends the feedback to the policy network for improvement. Once the training process is finished, the value network will be abandoned and the policy network will be used to control the agent. Note that different from the policy network (which is random) in the general DQN, that of DDPG is deterministic.

The output of the value network $q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta_v})$ is the approximation of the state-action value function $Q(\boldsymbol{s}, \boldsymbol{a})$, where $\boldsymbol{\theta_v}$ is the parameters of the value network. $q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta_v})$ is a real value that can evaluate the performance of the actor's policy. Specifically, a larger value of $q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta_v})$ implies a better performance of the policy network.

The training of the policy network is an off-policy method, i.e., the behavior policy could be different from the target policy. The training objective of the policy network is to update parameters $\boldsymbol{\theta_p}$ in order to increase the value of $q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta_v})$. The objective function is set as the expectation of $q(\boldsymbol{s}, \boldsymbol{\mu}(S; \boldsymbol{\theta_p}), \boldsymbol{\theta_v})$, i.e.,

$$J(\boldsymbol{\theta_p}) = \mathbb{E}_S \left[q(S, \boldsymbol{\mu}(S; \boldsymbol{\theta_p}), \boldsymbol{\theta_v}) \right] \tag{19}$$

where $\boldsymbol{\mu}(\boldsymbol{s}; \boldsymbol{\theta_p}) = \boldsymbol{a}$ is the output of the policy network based on the current state $\boldsymbol{s}$ and parameters $\boldsymbol{\theta_p}$. Note that we take the expectation with respect to state $\boldsymbol{s}$ to eliminate the randomness. Therefore, the training of the policy network can be formulated as an optimization problem with the aim of maximizing the average score of the actor given any state condition, i.e.,

$$\max_{\boldsymbol{\theta_p}} J(\boldsymbol{\theta_p}). \tag{20}$$

Note that the weights $\boldsymbol{\theta_p}$ are obtained with the weights of the value network $\boldsymbol{\theta_v}$ being fixed. $\boldsymbol{\theta_p}$ are updated by using the *policy gradient method* [21]. Given observation state s_j at timestep j, the sampled policy gradient can be calculated by

$$g_j \triangleq \nabla_{\boldsymbol{\theta_p}} q(s_j, \boldsymbol{\mu}(s_j; \boldsymbol{\theta_p}); \boldsymbol{\theta_v}), \tag{21}$$

which has been proven to be an unbiased estimation of $\nabla_{\boldsymbol{\theta_p}} J(\boldsymbol{\theta_p})$. By applying the chain rule, the gradient of q with respect to θ_p can be calculated as

$$\frac{\partial q}{\partial \theta_p} = \frac{\partial a}{\partial \theta_p} \cdot \frac{\partial q}{\partial a}. \tag{22}$$

Hence, the deterministic policy gradient g_j can be rewritten as

$$g_j = \nabla_{\boldsymbol{\theta_p}} \hat{\boldsymbol{a}}_j \cdot \nabla_{\boldsymbol{a}} q(s_j, \hat{\boldsymbol{a}}_j; \boldsymbol{\theta_v}), \tag{23}$$

where $\hat{\boldsymbol{a}}_j = \boldsymbol{\mu}(s_j; \boldsymbol{\theta_p})$. We thus obtain the algorithm to update the weights of the policy network,

$$\boldsymbol{\theta_p} \leftarrow \boldsymbol{\theta_p} + \eta \cdot \nabla_{\boldsymbol{\theta_p}} \hat{\boldsymbol{a}}_j \cdot \nabla_{\boldsymbol{a}} q(s_j, \hat{\boldsymbol{a}}_j; \boldsymbol{\theta_v}), \tag{24}$$

where η is the learning rate, which can be seen as the step size in traditional optimization methods. With the update of the weights of the policy network, the value network will give a higher average score to the output of the policy network.

The objective of the training of the value network is to reduce the gap between the prediction of the policy network $q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta_v})$ and true value function $Q(\boldsymbol{s}, \boldsymbol{a})$. In other words, the training procedure increases the accuracy of the score given by the critic (i.e., the value network). In each training episode, the observed reward after executing certain action will help correcting the critic since the observed reward is unbiased.

Similar to the standard Actor-Critic method, we use the value network to approximate the temporal difference (TD) target. We first randomly obtain a tuple (s_j, a_j, r_j, s_{j+1}) from the replay buffer and let the value network predict two value functions as follows:

$$\hat{q}_j = q(s_j, a_j; \boldsymbol{\theta_v}), \tag{25}$$

$$\hat{q}_{j+1} = q(s_{j+1}, \mu(s_{j+1}; \boldsymbol{\theta_p}); \boldsymbol{\theta_v}). \tag{26}$$

Then, we calculate the TD target $\hat{y}_j = r_j + \gamma \cdot \hat{q}_{j+1}$. Define the loss function as

$$L(\boldsymbol{\theta_v}) = \frac{1}{2}\left[\hat{q}_j - \hat{y}_j\right]^2. \tag{27}$$

The gradient of $L(\boldsymbol{\theta_v})$ with respect to $\boldsymbol{\theta_v}$ can be calculated as

$$\nabla_{\boldsymbol{\theta_v}} L(\boldsymbol{\theta_v}) = (\hat{q}_j - \hat{y}_j)\nabla_{\boldsymbol{\theta_v}} q(s_j, a_j; \boldsymbol{\theta_v}). \tag{28}$$

The parameters of the value network will be updated as

$$\boldsymbol{\theta_v} \leftarrow \boldsymbol{\theta_v} + \eta \cdot \nabla_{\boldsymbol{\theta_v}} L(\boldsymbol{\theta_v}). \tag{29}$$

After each iteration, the value of the loss function $L(\boldsymbol{\theta_v})$ will be decreased and the prediction of the value network $\hat{q}_j = q(s_j, a_j; \boldsymbol{\theta_v})$ will approach closer to the TD target $\hat{y}_j$.

1) *Experience Replay:* The employment of consecutive samples for updating the network will lead to high variance, instability, and even divergence of the model during training since consecutive samples are strongly correlated. Specifically, using consecutive samples can cause the agent to converge to a local optimum instead of the global optimum. The agent may learn to exploit the patterns in the highly-correlated data rather than finding the optimal policy by exploring more. As a result, the model performs well on the training data but poorly on new, unseen test data or in real-world applications. The experience replay can reduce the correlation of updates because the successive updates are performed based on samples randomly obtained from the replay memory. Instead of being used for only once and then abandoned, a tuple can stay in the replay memory for multiple time steps and thus can be employed for more than one update, which can potentially improve the data efficiency. Specifically, as shown in Fig. 3, the agent stores a tuple $e_j = (s_j, a_j, r_j, s_{j+1})$ in the replay memory in each time step. Since the capacity of the replay memory is limited, the oldest experience will be deleted from the buffer and replaced with the newest experience.

2) *Target Network:* The standard Q learning generally has the shortcoming of error propagation due to the utilization of bootstrap. Specifically, the estimation of the DQN is used to update the parameters of the DQN. For instance, an update that

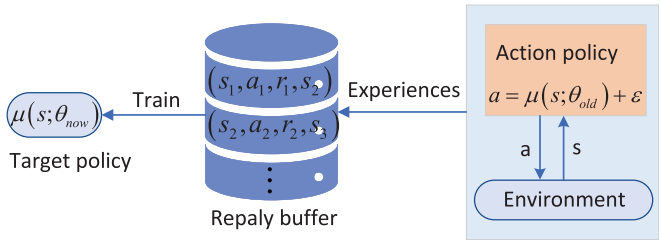

Fig. 3. Experience replay.

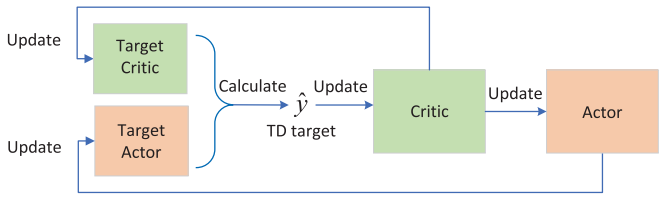

Fig. 4. DDPG network.

increases $Q(s_j, a_j)$ will also increase $Q(s_{j+1}, a)$ for all actions a and possibly propagates to other states. The other shortcoming of DQN is overestimation brought by the maximization operation when calculating the TD target. This overestimation can be tolerated if it is uniformly distributed (i.e., each action shares the same amount overestimation). The agent can still obtain the optimal policy by choosing the action that incurs the highest Q value. Unfortunately, the distribution of the overestimation is usually non-uniform because the probability of each action that appears in the replay memory is not equal. Therefore, the overestimation incurred often leads to divergence or even oscillation of the algorithm.

To reduce the overestimation and avoid the error propagation, the target network is introduced. As shown in Fig. 4, the DDPG network includes two additional target networks that clone the architecture of the actor network and critic network, respectively. Note that the two target networks share different parameters with the two primal networks. The adoption of target networks allows the primal networks to use an older set of experiences and thus stops the propagation brought by bootstrapping.

The DDPG-based algorithm to jointly determine the UAV's trajectory and velocity and the IoT nodes association scheme is embodied in Algorithm 1. The computational complexity of the proposed algorithm is determined by the operations in each iteration. With a fully connected neural network with W hidden layers, the corresponding time complexity can be calculated as [19] $\mathcal{O}(n_I n_1 + \sum_{w=1}^{W-1} n_w n_{w+1} + n_W n_O)$, where n_I, n_O and $n_w, \forall w \in \{1, 3, \cdots, W\}$ denote the number of neurons in the input layer, the output layer and the w-th hidden layer, respectively. Hence, the complexity of each iteration in Algorithm 1 can be expressed as $\mathcal{O}(|\mathcal{S}(t)|(n_1^a + n_1^c) + \sum_{w=1}^{W-1}(n_w^a n_{w+1}^a + n_w^c n_{w+1}^c) + n_W^a|\mathcal{A}(t)| + n_W^c)$, where superscript a and c represent the abbreviations of $actor$ and $critic$, respectively. $|\mathcal{S}(t)|$ and $|\mathcal{A}(t)|$ stand for the cardinality of state space and action space, respectively.

Algorithm 1:

Relax $u_i(t)$ into continuous spaces as $u_i(t) \in [0, 1]$.
Initialize the value (Critic) network and policy (Actor) network by randomly setting the values of $\boldsymbol{\theta_v}$ and $\boldsymbol{\theta_p}$, respectively.
Initialize the target value network and policy network with parameters $\boldsymbol{\theta'_v}$ and $\boldsymbol{\theta'_p}$, respectively.
Initialize the replay buffer.
for episode=1, K **do**
  Initialize a noise process $\mathcal{N}$ for action exploration.
  Set t=1.
  repeat
    Obtain $a_t = \mu(s_t; \boldsymbol{\theta_p}) + \mathcal{N}_t$.
    Set the $u_i(t)$ with the maximum value to 1 and the others to 0.
    Take action a_t and obtain the reward r_t and the next state s_{t+1}.
    Store the tuple (s_t, a_t, r_t, s_{t+1}) in the replay buffer.
    Randomly obtain N transitions from the replay buffer.
    Obtain $y_j = r_j + \gamma \cdot q'(s_{j+1}, \mu'(s_{j+1}; \boldsymbol{\theta'_p}); \boldsymbol{\theta'_v})$.
    Update the value network by minimizing the loss $\frac{1}{N}\sum_j [y_j - q'(s_j, a_j; \boldsymbol{\theta_v})]^2$.
    Update the policy network based on (24).
    Update the target networks based on:
$$\boldsymbol{\theta'_p} = \tau\boldsymbol{\theta_p} + (1-\tau)\boldsymbol{\theta'_p}$$
$$\boldsymbol{\theta'_v} = \tau\boldsymbol{\theta_v} + (1-\tau)\boldsymbol{\theta'_v}$$
    Update $t = t + 1$.
  until All the data are collected.
end for

TABLE I
SIMULATION PARAMETERS

Parameters	Definitions	Values
β_0	Pathloss at the reference distance	30 dB
σ^2	Noise power	-80 dBm
B	Total available bandwidth	10 MHz
P	Transmit power of IoT nodes	0.3 W
δ^m	Max duration of each time slot	10 s
V^m	Max velocity of UAV	20 m/s
C	Replay buffer capacity	10000
L	Mini-batch size	64
M	Penalty value	500
η	Learning rate	0.002
γ	discount factor	0.9
τ	soft update parameter	0.01
f_1	Activate function	ReLU
f_2	Optimizer	Adam

V. NUMERICAL RESULTS

We consider a 1000 m × 1000 m area with 10 IoT nodes in the simulations. The flying height of the UAV H is set as 60 m. The start location of the UAV is set as $Q = (0, 0)$. The data size of IoT devices are randomly chosen from 1 Mb to 10 Mb. Unless otherwise stated, other simulation parameters are shown in Table I. We compare our proposed DDPG-based algorithm with two benchmark algorithms, i.e., the *straight line flight* algorithm and the *fixed time slot duration* algorithm. In the first algorithm, the UAV follows a straight line trajectory and collects data from the nearest IoT node before it is associated to the next IoT node. In the second algorithm, only the trajectory of the UAV is optimized with fixed time slot duration equals to 5 s. The simulations are conducted by Python using the TensorFLow platform.

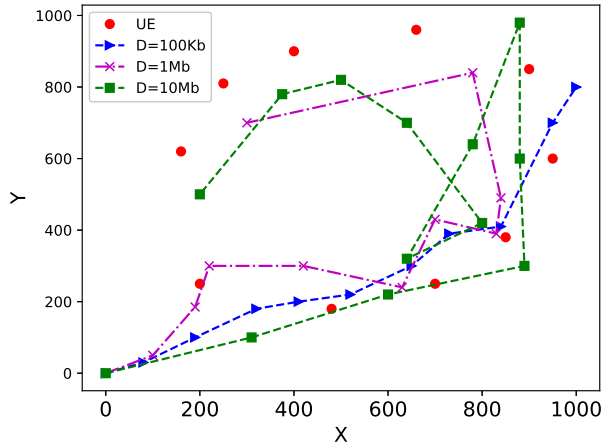

Fig. 5. Trajectories with different average data sizes.

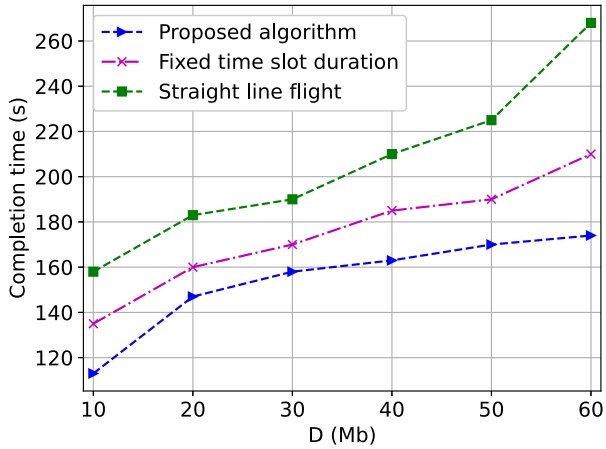

Fig. 6. Completion time with different average data sizes.

Fig. 5 illustrates the UAV trajectories under different average data sizes. It can be observed that the UAV travels larger distances and approaches closer to the location of each IoT node with the increase of the average data size of the IoT nodes. Meanwhile, the number of time slots also increases to 12 (as compared to 10 time slots when D=100Kb and D=1Mb) when the average data size is 10Mb. This is because the UAV needs to fly closer to each IoT node to improve the channel conditions. It can thus increase the data rate and therefore reduces the data collection time of each IoT node. For those whose data cannot be fully collected in a single time slot (note the fly back of the UAV when D=10 Mb), multiple time slots will be allocated to guarantee completion of the data collection. However, these allocated time slots are not necessarily consecutive unless it can reduce the total collection time.

Fig. 6 compares the proposed algorithm with the other two benchmark algorithms. We can observe that the completion time

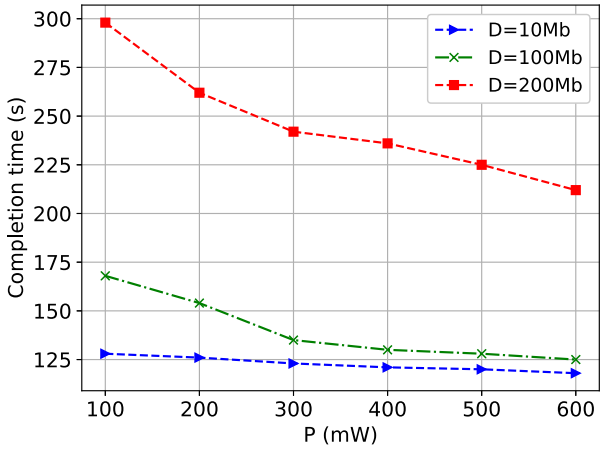

Fig. 7. Completion time vs transmit power with different average data sizes.

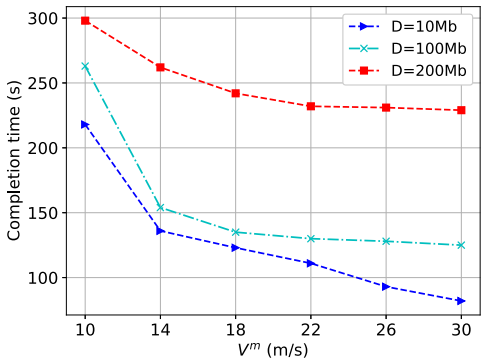

Fig. 8. Completion time vs V^m with different average data sizes.

of these three algorithms increases with the increase of the average data size. This is because the UAV needs to increase the duration of each time slot to finish the data collection of the corresponding associated IoT node. The proposed algorithm achieves the best performance since it optimizes both the trajectory and the velocity of the UAV. The *fixed time slot duration* algorithm achieves better performance (as compared to the *straight line flight* algorithm) since it can improve the channel conditions by approaching closer to the IoT nodes, thus reducing the data collection time. The *straight line flight* algorithm can only increase the duration of each time slot to collect the increased amount of data even though the channel conditions are poor.

Fig. 7 shows the completion time versus transmit power with different average data sizes. The completion time is reduced in all three cases with the increase of the transmit power because larger transmit power leads to larger data rate and thus reduces the data collection time of each IoT node. Note that the time reduction in the cases of D=10 Mb and D=100 Mb when the transmit power is larger than 300 mW is not as significant as that of D=200 Mb. This is because the bottleneck of the first two cases is the maximum velocity of the UAV. Specifically, the data collection time of each IoT node is smaller than the flight time to reach the next IoT node. Therefore, the completion time cannot be further reduced by increasing the transmit power.

Fig. 8 illustrates the completion time versus the maximum velocity of the UAV with different average data sizes. We can see the completion time in all three cases decrease with the increase

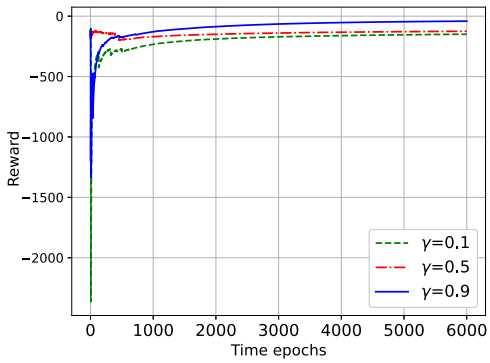

Fig. 9. Reward vs time epochs with different discount factors.

of the maximum velocity when $V^m \leq 22$ m/s. However, the completion time of D=100 Mb and D=200 Mb stay roughly unchanged because the UAV are not flying at the maximum allowed velocity such that the IoT nodes have enough time to offload their generated data. In comparison, the completion time for the case with less amount of data can be reduced by increasing V^m since the duration of each time slot can be reduced while guaranteeing the data collection completion. In Fig. 9, a larger discount factor incurs a larger reward (i.e., smaller completion time) because larger discount means more importance is put on the influence of future actions and thus a long-term average reward is obtained.

VI. Conclusion

In this article, we consider the completion time minimization in a UAV-enabled IoT network by designing the trajectory and velocity of the UAV and the association scheme of the IoT nodes. Different from existing works, the number of time slots is unknown and the duration of each time slot may vary. Unfortunately, the formulated problem based on the two assumptions is non-convex and has infinite number of decision variables. To solve the formulated problem, we thus reformulated it as an MDP and proposed a DDPG-based algorithm. Different from the DQN-based algorithm which selects the action from the discrete action space that maximizes the state-action value function, the DDPG-based algorithm uses an actor-critic approach based on the DPG algorithm and therefore can solve continuous control problems. We have conducted extensive numerical experiments to corroborate the effectiveness of the proposed algorithm under different parameters.

References

[1] X. Sun and N. Ansari, "Dynamic resource caching in the IoT application layer for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 606–613, Apr. 2018.
[2] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, "Internet-of-things (IoT)-Based smart agriculture: Toward making the fields talk," *IEEE Access*, vol. 7, pp. 129551–129583, 2019.
[3] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
[4] X. Sun, N. Ansari, and R. Fierro, "Jointly optimized 3D drone mounted base station deployment and user association in drone assisted mobile access networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2195–2203, Feb. 2020.
[5] Y. Li and L. Cai, "UAV-Assisted dynamic coverage in a heterogeneous cellular system," *IEEE Netw.*, vol. 31, no. 4, pp. 56–61, Aug. 2017.

[6] N. Ansari, Q. Fan, X. Sun, and L. Zhang, "SoarNet," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 37–43, Dec. 2019.
[7] X. Liu and N. Ansari, "Resource allocation in UAV-Assisted M2M communications for disaster rescue," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 580–583, Apr. 2019.
[8] S. Zhang and N. Ansari, "3D drone base station placement and resource allocation with FSO-Based backhaul in hotspots," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3322–3329, Mar. 2020.
[9] C. Zhan, H. Hu, X. Sui, Z. Liu, and D. Niyato, "Completion time and energy optimization in the UAV-enabled mobile-edge computing system," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7808–7822, Aug. 2020.
[10] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.
[11] 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 Things J.*, vol. 7, no. 8, pp. 6898–6908, Aug. 2020.
[12] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, Sep. 2020.
[13] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.
[14] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1749–1758.
[15] H. Yang et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nature Commun.*, vol. 13, no. 1, pp. 1–12, 2022.
[16] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan. 2021.
[17] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2016.
[19] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 6th Int. Conf. Learn. Representations*, 2016, pp. 1–14.
[20] M. Volodymyr, K. Kavukcuoglu, D. Silver, A. Graves, and I. Antonoglou, "Playing A tari with deep reinforcement learning," in *Proc. NIPS Deep Learn. Workshop*, 2013.
[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: http://arxiv.org/abs/1707.06347
[22] M. Liu, J. Yang, and G. Gui, "DSF-NOMA: UAV-Assisted emergency communication technology in a heterogeneous Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5508–5519, Jun. 2019.
[23] L. Xie, J. Xu, and R. Zhang, "Throughput maximization for UAV-Enabled wireless powered communication networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1690–1703, Apr. 2019.
[24] Z. Wang, G. Zhang, Q. Wang, K. Wang, and K. Yang, "Completion time minimization in wireless-powered UAV-Assisted data collection system," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1954–1958, Jun. 2021.
[25] Y. Xu, T. Zhang, J. Loo, D. Yang, and L. Xiao, "Completion time minimization for UAV-assisted mobile-edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12253–12259, Nov. 2021.
[26] Y. Cai, F. Cui, Q. Shi, M. Zhao, and G. Y. Li, "Dual-UAV-Enabled secure communications: Joint trajectory design and user scheduling," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1972–1985, Sep. 2018.
[27] N. Gupta, S. Agarwal, and D. Mishra, "Trajectory design for throughput maximization in UAV-Assisted communication system," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1319–1332, Sep. 2021.
[28] Q. Huang, W. Wang, W. Lu, N. Zhao, A. Nallanathan, and X. Wang, "Resource allocation for multi-cluster NOMA-UAV networks," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8448–8459, Dec. 2022.
[29] Y. Su, X. Pang, S. Chen, X. Jiang, N. Zhao, and F. R. Yu, "Spectrum and energy efficiency optimization in IRS-Assisted UAV networks," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6489–6502, Oct. 2022.
[30] K. K. Nguyen, T. Q. Duong, T. Do-Duy, H. Claussen, and L. Hanzo, "3D UAV trajectory and data collection optimisation via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2358–2371, Apr. 2022.
[31] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A. deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.
[32] V. Saxena, J. Jaldén, and H. Klessig, "Optimal UAV base station trajectories using flow-level models for reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1101–1112, Dec. 2019.
[33] Y.-H. Hsu and R.-H. Gau, "Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 306–320, Jan. 2022.
[34] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for Multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
[35] C. Jeong and S. H. Chae, "Simultaneous wireless information and power transfer for multiuser UAV-Enabled IoT networks," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8044–8055, May, 2021.
[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Shuai Zhang (Member, IEEE) received the B.E. and M.E. degrees from the Department of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, in 2014, 2017, and 2022, respectively. He is currently a Postdoctoral Researcher with the KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include next-generation wireless system design and optimization, Internet of Things, mobile edge computing, optical networks, UAV-assisted communications, and artificial intelligence in wireless networks.

Weiqi Liu (Graduate Student Member, IEEE) received the M.S. degree in electrical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, where he is currently working toward the Ph.D. degree in electrical engineering. His research interests include UAV communications, wireless communications, mobile edge computing, and Internet of Things.

Nirwan Ansari (Fellow, IEEE) received the BSEE degree (*summa cum laude* with a perfect GPA) from New Jersey Institute of Technology, Newark, NJ, USA, the MSEE degree from the University of Michigan, Ann Arbor, MI, USA, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA. He is currently a Distinguished Professor of electrical and computer engineering with the New Jersey Institute of Technology (NJIT), Newark, NJ, USA. He is also a Fellow with the National Academy of Inventors. He has authored or coauthored three books and (co-)authored more than 700 technical publications, with more than half of them published in widely cited journals/magazines. He has also been granted more than 40 U.S. patents. His research interests on green communications and networking, cloud computing, drone-assisted networking, and various aspects of broadband networks. He has Guest-Edited numerous special issues covering various emerging topics in communications and networking. He is the Editor-in-Chief of *IEEE Wireless Communications Magazine* and has been on the Editorial/Advisory board of more than 10 journals. He was elected on the IEEE Communications Society (ComSoc) Board of Governors as the Member-at-Large, has chaired some ComSoc technical, and steering committees, and is the Director with ComSoc Educational Services Board. He was also on many committees such as the IEEE Fellow Committee. He has actively participated in inorganizing numerous IEEE International Conferences/Symposia/Workshops. He was the recipient of multiple Best Paper awards, the NCE Excellence in Research Award, several ComSoc TC Technical Recognition awards, the NJ Inventors Hall of Fame Inventor of the Year Award, the Thomas Alva Edison Patent Award, the Purdue University Outstanding Electrical and Computer Engineering Award, the NCE 100 Medal, the NJIT Excellence in Research Prize and Medal, designation as a COMSOC Distinguished Lecturer, and several excellence in teaching awards among many recognitions.