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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach

Chi Harold Liu, *Senior Member, IEEE*, Zheyu Chen, Jian Tang, *Senior Member, IEEE*, Jie Xu and Chengzhe Piao

***Abstract*—Unmanned Aerial Vehicles (UAVs) can be used to serve as aerial Base Stations (BSs) to enhance both the coverage and performance of communication networks in various scenarios, such as emergency communications and network access for remote areas. Mobile UAVs can establish communication links for ground users to deliver packets. However, UAVs have limited communication ranges and energy resources. Particularly, for a large region, they cannot cover the entire area all the time or keep flying for a long time. It is thus challenging to control a group of UAVs to achieve certain communication coverage in a long run, while preserving their connectivity and minimizing their energy consumption. Towards this end, we propose to leverage emerging Deep Reinforcement Learning (DRL) for UAV control, and present a novel and highly energy-efficient DRL-based method, which we call DRL-EC^3 (DRL-based Energy-efficient Control for Coverage and Connectivity). The proposed method 1) maximizes a novel energy efficiency function with joint consideration for communications coverage, fairness, energy consumption and connectivity, 2) learns the environment and its dynamics, and 3) makes decisions under the guidance of two powerful Deep Neural Networks (DNNs). We conduct extensive simulations for performance evaluation. Simulation results have shown that DRL-EC^3 significantly and consistently outperform two commonly-used baseline methods in terms of coverage, fairness and energy consumption.**

***Index Terms*—UAV Control, Deep Reinforcement Learning, Energy Efficiency, Communication Coverage**

I. Introduction

Unmanned Aerial Vehicles (UAVs) can be used to serve as aerial Base Stations (BSs) to enhance both the coverage and performance of communication networks in various scenarios [1], such as emergency communications and network access for remote areas. When communication networks are disrupted by a catastrophic natural disaster, mobile UAVs can be quickly deployed to establish efficient communication links for ground users to deliver packets. For example, in 2011 Great East Japan earthquake, some people were trapped in broken down houses or otherwise cut-off areas, where quite limited number of disaster relief workers patrolled destroyed areas to search for survivors, which will not be possible if communication infrastructure is damaged by the disaster and can be temporarily provided by mobile UAVs [2].

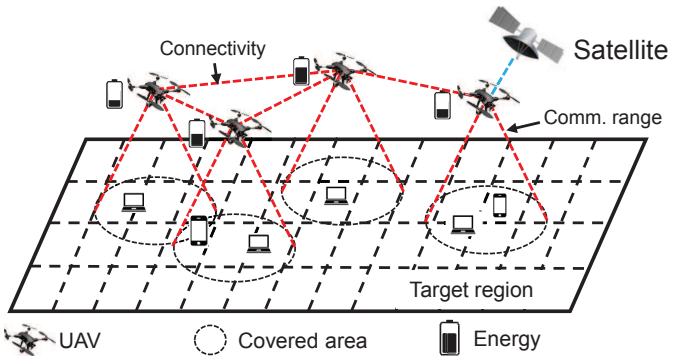


Fig. 1: A UAV network providing communication coverage for ground users in a target region.

Using UAVs as aerial BSs provides several benefits. First, due to their high altitude, aerial BSs have a higher chance of Line-of-Sight (LoS) links to ground users, compared to ground BSs. Second, UAVs are able to provide fast, reliable and cost-effective network access to regions poorly covered by terrestrial networks [3].

In order to provide effective communication coverage in a long run, UAVs with a high degree of mobility needs to work as a team autonomously, which is illustrated in Fig. 1. In such a UAV network, UAVs can work as BSs to provide communication links for ground users using current wireless technologies such as WiFi or LTE. One or a small number of UAVs have long-distance connections (such as satellite links) to external networks (such as Internet), which are called gateways. This task is quite challenging because UAVs have very limited communication range and energy resources, and moreover, a UAV network usually has very limited number of gateways. First, due to limited communication range and relatively high costs (several thousand USDs for each commercial UAV), it is impossible to have sufficient UAVs to cover a large target region all the time. Therefore, UAVs need to move around to ensure each area is covered for a reasonable amount of time. Moreover, fairness is critical for communication coverage since it is not desirable to cover certain areas most of time while leaving the rest barely covered. Second, due to limited energy resources, a UAV cannot keep flying for a

C. H. Liu (corresponding author), Z. Chen, J. Xu and C. Piao are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China. C. H. Liu is also with the Department of Computer Information and Security, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul, South Korea (Email: liuchi02@gmail.com).

J. Tang is with the Department of Computer Science and Engineering, Syracuse University, USA (Email: jtang02@syr.edu).

This paper was financially supported in part by National Natural Science Foundation of China (No. 61772072), and in part by the National Key Research and Development Program of China under Grant 2018YFB1003701.

Manuscript received January 2, 2018; revised April 24, 2018.

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

long time, therefore, they need to be operated in an energy-efficient manner to prolong network lifetime. In addition, due to very limited number of gateways, a UAV network needs to be kept connected all the time; otherwise, those ground users associated with a disconnected non-gateway node will lose their connections to the external network.

To address the above issues, we propose to leverage emerging deep reinforcement learning (DRL) [4], which has been shown to deliver superior performance on a few game-playing tasks recently. We believe DRL provides a promising solution because it can well handle a sophisticated state space and time-varying environment; and it uses powerful Deep Neural Networks (DNNs) to guide decision making, which have been shown to offer state-of-the-art performance on quite a few learning tasks with limited to even zero domain knowledge. However, it is not straightforward to solve the UAV control problem using DRL. The basic DRL technique, deep Q learning, uses a Deep Q Network (DQN) to estimate Q value for each state-action pair, which can only handle a very limited action space. The control problem here is a continuous control problem with an unlimited action space. The commonly-used method for continuous control is the actor-critic method [5]. So we choose to use a state-of-the-art actor-critic method, Deep Deterministic Policy Gradient (DDPG) [6], as the starting point for our design. The control problem here is more complicated than most other control problems since it involves multiple objectives (i.e., coverage, fairness and energy consumption) and a constraint on network connectivity. Even though DRL has made remarkable successes on a few game-playing tasks, it remains unknown if it can succeed on control tasks in complex communication networks, which usually have quite different objectives, constraints, and states and action spaces. To the best of our knowledge, we are the first to leverage DRL for enabling energy-efficient UAV control in the context of providing communication coverage for ground users. Specifically, we present a novel DRL-based method for UAV control, DRL-EC^3 (DRL-based Energy-efficient Control for Coverage and Connectivity), which maximizes a novel energy efficiency function while ensuring effective and fair communication coverage, and network connectivity. Extensive simulation results have also been presented to justify its effectiveness, robustness and superiority in terms of various metrics.

The rest of the paper is organized as follows: Section II presents system model and problem definition. Section III reviews the related research efforts. Section IV introduces necessary preliminaries for DRL. Section V presents the proposed DRL-based method for UAV control. Section VI presents extensive simulation results for performance evaluation, and Section VII describe practical implementation issues. Finally, Section VIII concludes the paper.

II. Related Work

In this section, we review the related works and point out the differences.

UAV networks have been studied recently [7], [8], [9], [10], [11]. In [12], the authors categorized UAV networks into four types: centralized UAV network, UAV ad-hoc network, multi-group UAV network and multi-layer UAV ad-hoc network. In [13], Shibata *et. al.* proposed an information communication system consisting of multiple UAVs. The authors of [14] proposed a framework for optimized deployment and mobility of multiple UAVs for the purpose of energy-efficient uplink data collection from ground IoT devices. Furthermore, by using the mathematical framework of optimal transport theory, in [15], Mozaffari *et. al.* proposed a framework to maximize the average data service that is delivered to users based on the maximum possible hover times. Other related works on UAV communication networks and their applications to data collection include [16], [17], [18], [19], [20], [21], [22], [23], [24].

UAV control has also been studied recently. The authors of [25] developed a novel distributed algorithm for coordination and communications of multiple UAVs engaging multiple targets, where coordination of UAV motion is achieved by implementing a simple behavioral flocking algorithm utilizing a tree topology for distributed flight coordination. In [26], a passivity-based decentralized approach was proposed for bilaterally teleoperating a group of UAVs composing the slave side of the teleoperation system, ensuring high flexibility to the group topology (e.g., possibility to autonomously split or join during the motion). In [19], Dierks *et. al.* proposed a new nonlinear controller for UAV using neural networks, which learns complete dynamics of UAVs online, and outputs feedback. For single UAV control, the authors of [27] proposed a method to figure out an altitude for maximizing coverage region, which can guarantee a minimum outage performance. Although the authors of [28] presented an adaptation of an optimal terrain coverage algorithm, which could ensure a complete coverage of the terrain, a single UAV has to fly more than 10 hours to finish it, which requires a large power supply. The need for a rapid-to-deploy solution to providing wireless cellular services can be realized by UAV-BSs. The authors of [29] studied a 3D UAV-BS placement problem that maximizes the number of covered users with different Quality-of-Service (QoS) requirements. We summarize the differences from these related works as follows:

- None of them have carefully addressed energy efficiency in UAV networking or control, which, however, is the main focus of this paper.
- Unlike a static UAV deployment problem considered in [29], dynamic UAV control is studied here.

Some research efforts have considered energy efficiency for UAV control. In [30], the authors proposed an optimal placement algorithm for UAV-BSs, which maximizes the number of covered users by using the minimum transmission power. The authors of [31] developed a framework to determine the optimal 3D locations of the UAVs in order to maximize the downlink coverage performance with minimum transmission power. In [32], Chen, *et. al.* proposed a framework that leverages user-centric information to deploy cache-enabled UAVs while maximizing users' Quality-of-Experience (QoE) using minimum total transmission power. The authors of [33] presented a solution to UAV energy saving problem, ensuring

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

TABLE I: List of Major Notations

Notation	Explantation
k, K	The index of a PoI, the number of PoIs
i, N	The index of a UAV, the number of UAVs
t, T	Decision epoch, the total number of epochs
R, R'	The communication range, the coverage range
T_k, c_k, f	The number of timeslots in which PoI k is covered, the coverage score of PoI k, fairness index
$r(\cdot)$, $Q(\cdot)$, $\pi(\cdot)$, $L(\cdot)$	Reward function, Q function, policy function, loss function
$\boldsymbol{s}_t$, $\boldsymbol{a}_t$, r_t	State, action and reward at epoch t
c_k^t, b_k^t, e_i^t	The current coverage score of PoI k, the current coverage state of PoI k, the current energy consumption of UAV i at epoch t
θ_i^t, d_i^t	Flying direction and distance of UAV i at epoch t

a continuous tracking of a mobile target. They computed the energy consumption caused by transmitting images and by vertical and horizontal UAV movements. In [34], Di, *et. al.* proposed an energy model which is derived from real measurements to find the power consumption as a function of the UAV dynamic in different operating conditions. Different from these research works, we focus on energy consumption for UAV movements (with consideration for communication coverage and connectivity), rather than energy used for data transmissions [30], [31], [32], [33], which has been well studied in the literature of radio resource management. Moreover, we consider the problem of jointly maximizing coverage and fairness and minimizing energy consumption, which is mathematically different from those problems studied in these related works.

DRL has recently attracted much attention from both industry and academia. In a pioneering work, the authors of [4] introduced a RL framework that uses a DQN as the function approximator, and two new techniques, experience replay and target network to improve learning stability. To solve problems with continuous action spaces, the authors of [6] presented an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate over a continuous action space. Other recent works on DRL for continuous control include [35], [36]. Although DRL has made remarkable successes on a few game-playing tasks, its applicability and effectiveness on complex communication system control remain unexplored.

III. System Model and Problem Statement

We describe the system model and the control problem in this section. First, we provide a list of major notations in Table I.

A. System Model

We consider a network with a group of N UAVs flying horizontally at a certain altitude to provide communication coverage for ground users in a target region, which is illustrated in Fig. 1. Each UAV is aware of its own location. We divide the target region into K cells. To ensure effective coverage, we assume that the center of each cell (rather than the whole cell), which we call a Point-of-Interest (PoI), needs to be covered by at least a UAV for a reasonable amount of time. We consider a communication coverage task that lasts for T timeslots with equal durations. Due to limited number of UAVs, they may not able to cover all the PoIs in every timeslot. So UAVs need to fly around to cover different subsets of PoIs in different timeslots. At the beginning of the task, each UAV takes off at a random origin. In each timeslot, at very beginning, each UAV hovers at its current location or flies horizontally in a direction $\theta \in (0, 2\pi]$ for a distance of d, which consumes $\phi(d)$ energy. The proposed method is not restricted to any particular energy consumption model $\phi(\cdot)$. But the model should at least reflect the fact that moving around leads to more energy consumption than hovering at a location, which monotonically increases with the flying distance. In our simulation, we used a linear model where the energy consumption increases linearly with the flying distance. Note that we are only interested in energy consumed for UAV flying or hovering at the beginning of each timeslot. Once a UAV reaches the desired location, it hovers there and starts to serve as a BS for ground users for the rest of the timeslot. These activities consume energy too, which, however, are our of scope of this paper since communication energy efficiency has been well studied in the literature of radio resource management and we focus on how to control movements of UAVs with consideration for both network connectivity and coverage, which is unique to UAV networks.

Each UAV has a communication range of R. As mentioned above, due to limited number of gateways, network formed by UAVs need to be connected with regards to R all the time. In addition, since each UAV flies at a certain altitude, in terms of communication coverage for ground users, the corresponding range R' is different and usually less than R, which we call *coverage range*. We consider the scenario in which a cloud periodically collects the state (including locations, energy usages, etc) of the UAV network via gateways.

B. Problem Statement

We aim to find a control policy which specifies how each UAV moves in each timeslot. If a PoI falls into the coverage range of a UAV, then we say it is covered. Note that a PoI may be covered by multiple UAVs in a timeslot. Given a control policy, the corresponding *coverage score* of a PoI k is:

$$c_k = \frac{T_k}{T}, k \in \{1, \cdots, K\}, \tag{1}$$

where T_k gives the number of timeslots in which PoI k is covered. Our objective is to maximize the total or average PoI coverage. However, doing so may lead to unfair coverage, that is, in most or even all timeslots, a subset (likely a small subset) of PoIs are covered, while the rest are left uncovered. Hence, we need to address fairness in terms of coverage. The most widely-used metric for fairness is Jain's fairness index [37]. Here, given a control policy, the corresponding *fairness index* is:

$$f = \frac{(\sum_{k=1}^{K} c_k)^2}{K(\sum_{k=1}^{K} c_k^2)}. \tag{2}$$

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

It can be easily seen that $f \in [0, 1]$ and the larger the fairness index, the fairer the coverage. In addition, flying UAVs around leads to energy consumption, which needs to be minimized to prolong network lifetime.

In short, we aim to find a control policy that can 1) maximize the total/average PoI coverage score; 2) maximize the fairness index for coverage, 3) minimize the energy consumption, and 4) ensure UAV network connectivity in every timeslot. It is quite challenging to achieve all of these objectives because on one hand, to provide effective and fair communication coverage, it is preferred to move UAVs around to different cells from time to time such that they can be well spread out in both the temporal and spatial domains; one the other hand, to minimize energy consumption and ensure network connectivity, it is preferred to reduce UAV movements (for energy savings) and make them stay together (for connectivity). Hence, a good solution to this problem is supposed to well address this tradeoff.

IV. Preliminaries

Before presenting the proposed method, we give a necessary background introduction to DRL in this section.

In a standard Reinforcement Learning (RL) setting, an agent interacts with a system environment in discrete decision epochs. At each epoch t, the agent observes state $\boldsymbol{s}_t$, executes action $\boldsymbol{a}_t$, and receives a reward r_t. We are interested in finding a policy $\pi(\boldsymbol{s})$ that maps a state to an action (or a distribution over actions) to maximize the discounted cumulative reward $R_0 = \sum_{t=0}^{T} \gamma r(\boldsymbol{s}_t, \boldsymbol{a}_t)$, where $r(\cdot)$ is the reward function and the discount factor $\gamma \in [0, 1]$.

DRL can be considered as the "deep" version of RL, which uses a DNN (or multiple DNNs) as the approximator of the $Q(\cdot)$ function. If in state $\boldsymbol{s}_t$, the system follows action $\boldsymbol{a}_t$ at epoch t, then:

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}[R_t | \boldsymbol{s}_t, \boldsymbol{a}_t], \tag{3}$$

where $R_t = \sum_{j=t}^{T} \gamma r(\boldsymbol{s}_j, \boldsymbol{a}_j)$ and the $Q(\cdot)$ estimates the expected discounted cumulative reward for each state-action pair. A commonly-used off-policy method follows the greedy policy: $\pi(\boldsymbol{s}_t) = \arg\max_{\boldsymbol{a}_t} Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The DQN is trained by minimizing the following loss function:

$$L(\boldsymbol{\theta^Q}) = \mathbb{E}[y_t - Q(\boldsymbol{s}_t, \boldsymbol{a}_t | \boldsymbol{\theta^Q})], \tag{4}$$

where $\boldsymbol{\theta^Q}$ is the weight vector of the DQN; and y_t is the target value, which can be estimated by [38]:

$$y_t = r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma Q(\boldsymbol{s}_{t+1}, \pi(\boldsymbol{s}_{t+1} | \boldsymbol{\theta^\pi}) | \boldsymbol{\theta^Q}). \tag{5}$$

A DNN has a bad reputation for causing instability or even divergence, which are certainly not desired. DRL usually uses two techniques, experience relay and target network, to resolve this issue. To update the DNN, DRL uses a mini-batch from an experience replay buffer with state transition samples collected during learning (instead of the immediately collected sample). Compared to immediate sampling used in traditional Q-learning, experience replay breaks correlations between sequentially generated samples, thus can avoid divergence and smooth out learning. Moreover, DRL uses an additional target network to estimate target values $< y_t >$ for DNN training. A target network has the same structure as the original DNN, however, its weights are updated slowly with the original DNN's weights every a few epochs and are held fixed in between.

DQN-based DRL only works for control problems with a low-dimensional discrete action space. It is hard to apply a DQN to continuous control because it needs to figure out the action that maximizes the Q function, which is quite difficult. The DQN-based method can only handle tasks with a limited discrete action space. However, UAV control is a continuous control task. The commonly-used method for continuous control is the actor-critic method [5], which can be also used with DNNs to search for the optimal control policy. The basic idea is to maintain the parameterized actor function $\pi(\boldsymbol{s}_t | \boldsymbol{\theta^\pi})$ to derive the best action from a given state, and a critic function $Q(\boldsymbol{s}_t, \boldsymbol{a}_t | \boldsymbol{\theta^Q})$ to model the correlation between Q values and state-action pairs. The above DQN can be used to implement the critic function, which can be trained using the loss function and method described above. According to [6], the actor network can then be updated by the chain rule applied to the cumulative reward J on the actor parameters $\boldsymbol{\theta^\pi}$:

$$\begin{aligned} \nabla_{\boldsymbol{\theta^\pi}} J &\approx \mathbb{E}[\nabla_{\boldsymbol{\theta^\pi}} Q(\boldsymbol{s}, \boldsymbol{a} | \boldsymbol{\theta^Q})|_{\boldsymbol{s}=\boldsymbol{s}_t, \boldsymbol{a}=\pi(\boldsymbol{s}_t|\boldsymbol{\theta^\pi})}] \\ &= \mathbb{E}[\nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a} | \boldsymbol{\theta^Q})|_{\boldsymbol{s}=\boldsymbol{s}_t, \boldsymbol{a}=\pi(\boldsymbol{s}_t)} \\ &\quad \cdot \nabla_{\boldsymbol{\theta^\pi}} \pi(\boldsymbol{s} | \boldsymbol{\theta^\pi})|_{\boldsymbol{s}=\boldsymbol{s}_t}]. \end{aligned} \tag{6}$$

Note that the experience reply and target network introduced above can also be integrated to this approach to ensure stability.

V. Proposed DRL-based Method: DRL-EC^3

In this section, we present the proposed DRL-based method for UAV control, namely, DRL-EC^3. A DRL agent periodically collects the state of the UAV network, finds the best action using DRL-EC^3, and deploys it by sending commands to move UAVs. First, we define the state, action and reward of the DRL agent.

1) State $\boldsymbol{s}_t$ (at decision epoch t): $\boldsymbol{s}_t$ consists of three parts:

- $c_k^t \in [0, 1]$: the current coverage score of each PoI k (Equation 1);
- $b_k^t \in \{0, 1\}$: the current coverage state of each PoI k ($b_k^t = 1$ if it is covered; 0, otherwise.)
- e_i^t: the current energy consumption of UAV i.

Formally, the state $\boldsymbol{s}_t = [c_t^1, \cdots, c_t^K, b_t^1; \cdots, b_t^K; e_t^1, \cdots, e_t^N]$, which has a cardinality of $(2K + N)$. Note that the state is defined in this way because the DRL agent makes decisions mainly based on current coverage and energy consumption. Here the state is composed of both UAV locations and actions, i.e. b_t^k indicate whether or not a specific PoI is covered by a UAV which also reflects the UAV movement status to certain extent.

2) Action $\boldsymbol{a}_t$ (at decision epoch t): an action $\boldsymbol{a}_t$ consists of two parts:

- $\theta_i^t \in (0, 2\pi]$: the flying direction (i.e., angle) for each UAV i;

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

Algorithm 1 DRL-EC^3

1: Randomly initialize critic network $Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta^Q})$ and actor network $\pi(\boldsymbol{s}|\boldsymbol{\theta^\pi})$ with weights $\boldsymbol{\theta^Q}$ and $\boldsymbol{\theta^\pi}$;
2: Initialize target networks $Q'(\cdot)$ and $\pi'(\cdot)$ with weights $\boldsymbol{\theta^{Q'}} := \boldsymbol{\theta^Q}, \boldsymbol{\theta^{\pi'}} := \boldsymbol{\theta^\pi}$;
3: Initialize replay buffer $\mathbf{B}$;
4: **for** episode $:= 1, \cdots, M$ **do**
5: Initialize the environment and receive an initial
6: state $\boldsymbol{s}_1$;
7: **for** epoch $t := 1, \cdots, T$ **do**
8: $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t) + \epsilon\mathcal{N}$,
9: where $\mathcal{N}$ is a random noise and ϵ decays over time;
10: Execute $\boldsymbol{a}_t$, and obtain $\boldsymbol{s}_{t+1}$ and r_t;
11: **for** UAV $i := 1, \cdots, T$ **do**
12: **if** UAV i flies beyond the border **then**
13: $r_t := r_t - p$, where p is a given penalty;
14: Cancel the movement of UAV i and
15: update $\boldsymbol{s}_{t+1}$ accordingly;
16: **end if**
17: **if** i is disconnected **then**
18: $r_t := r_t - p$;
19: **end if**
20: **end for**
21: Store transition sample $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ into $\mathbf{B}$;
22: Sample a random minibatch of H samples
23: $(\boldsymbol{s}_j, \boldsymbol{a}_j, r_j, \boldsymbol{s}_{j+1})$ from $\mathbf{B}$;
24: $y_j := r_j + \gamma Q'(\boldsymbol{s}_{j+1}, \pi'(\boldsymbol{s}_{j+1}|\boldsymbol{\theta^{\pi'}})|\boldsymbol{\theta^{Q'}})$;
25: Update weights $\boldsymbol{\theta^Q}$ of $Q(\cdot)$ by minimizing the loss:
26: $L(\boldsymbol{\theta^Q}) = \frac{1}{H}\sum_{j=1}^{H}(y_i - Q(\boldsymbol{s}_j, \boldsymbol{a}_j))^2$;
27: Update the weights $\boldsymbol{\theta^\pi}$ of $\pi(\cdot)$ using:
28: $\nabla_{\boldsymbol{\theta^\pi}} J \approx \frac{1}{H}\sum_{j=1}^{H} \nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta^Q})|_{\boldsymbol{s}=\boldsymbol{s}_j, \boldsymbol{a}=\pi(\boldsymbol{s}_j)}$
29: $\cdot \nabla_{\boldsymbol{\theta^\pi}} \pi(\boldsymbol{s}|\boldsymbol{\theta^\pi})|_{\boldsymbol{s}_j}$;
30: Update the corresponding target networks:
31: $\boldsymbol{\theta^{Q'}} := \tau\boldsymbol{\theta^Q} + (1-\tau)\boldsymbol{\theta^{Q'}}$;
32: $\boldsymbol{\theta^{\pi'}} := \tau\boldsymbol{\theta^Q} + (1-\tau)\boldsymbol{\theta^{\pi'}}$;
33: **end for**
34: **end for**

- $d_i^t \in [0, 1]$: the flying distance for each UAV i, which is normalized by a maximum distance d_{max}. If $d_i^t = 0$, UAV hovers at the current location (i.e., static), otherwise UAV flies to a certain d_i^t, and when $d_i^t = 1$, it flies to the maximum distance d_{max}.

Formally, the action $\boldsymbol{a}_t = [\theta_i^t, \cdots, \theta_N^t; d_1^t, \cdots, d_N^t]$, which has a cardinality of $2N$. Note that the action is defined in this way because the control policy is used to specify how each UAV flies at each decision epoch. Since both decision variables take continuous values, it is a continuous control task.

3) Reward r_t (at decision epoch t): the reward t is defined as:

$$r_t = \frac{f_t * (\sum_{k=1}^{K} \Delta c_k^t)}{\sum_{i=1}^{N} \Delta e_i^t}, \tag{7}$$

where f_t is the fairness index calculated based on the current coverage scores (Equation 2), $\Delta c_k^t = c_k^t - c_k^{t-1}$ is the incremental coverage score (Equation 2), and $\Delta e_i^t = e_i^t - e_i^{t-1}$ is the incremental energy consumption. It is not trivial to define the reward because three objectives, coverage score, fairness and energy consumption, need to be properly addressed by the reward. $f_t * \Delta c_k^t$ can be considered as the *effective* incremental coverage, which adds a discount to the actual incremental value if such an increment leads to unfairness. Hence, the numerator of the reward gives the gain, while the denominator is the cost (in terms of energy consumption). Overall, the reward can be considered as the energy efficiency (gain that can be brought by a unit of energy). Then maximizing the cumulative reward is equivalent to maximizing the average energy efficiency.

DRL-EC^3 is formally presented as Algorithm 1. As mentioned above, since we are dealing with a continuous control task, we choose to use a state-of-the-art actor-critic method, DDPG [6], as the starting point for our design, whose basic idea has been introduced in Section IV. Our algorithm works as follows.

In the beginning, the algorithm randomly initializes the weights $\boldsymbol{\theta^\pi}$ and $\boldsymbol{\theta^Q}$ of the actor $\pi(\cdot)$ and critic $Q(\cdot)$ networks respectively (Line 1). As mentioned above, we employ target networks $\pi^{'}(\cdot)$ and $Q^{'}(\cdot)$ to improve learning stability. The target networks have the same structures as the original actor or critic networks, whose weights $\boldsymbol{\theta^{\pi'}}$ and $\boldsymbol{\theta^{Q'}}$ are initialized in the same way as their original networks (Line 2), but are updated slowly (Lines 28-30) for the sake of stability. τ is used to control the updating rate and $\tau = 0.001$ in our implementation.

The second part (Lines 8-20) is exploration. During exploration, the algorithm derives an action from the current actor network $\boldsymbol{\theta^\pi}(\cdot)$ and then add a random noise $\epsilon\mathcal{N}$, where $\mathcal{N}$ is a random noise and ϵ decays over time. In our implementation, $\mathcal{N}$ follows a normal distribution with a mean of 0 and a variance of 0.6; and ϵ is initialized to 1 and decays with a rate of 0.9995 over epochs. We also need to take care of an important case where an action leads to violations of the boundary and/or connectivity constraints by assigning a large penalty to the reward (Lines 11-20). Specifically, if an action causes the boundary violation of a UAV, then a penalty p is deducted from the reward; and moreover, the corresponding movement is canceled (i.e., the UAV stays put without making the movement) and the elements related to this UAV in $\boldsymbol{s}_{t+1}$ are updated accordingly. Similarly, if an action causes disconnection of a UAV, a penalty p is simply deducted from the reward. In our implementation, the penalty is set to a large value, which is 100 times the corresponding reward.

The third part is how to update the neural networks (Line 21-32). Similar as in DDPG, we use a replay buffer for updating the actor and critic networks, which is initialized at the beginning with size B (Line 3). Specifically, we first store the collected samples into the replay buffer (Line 21), and then sample a mini-batch of them from the buffer to update the actor and critic networks (Line 21-29). As explain above, the critic network $\boldsymbol{\theta^Q}$ is updated by minimizing a loss function $L(\cdot)$ (Equation 4); and the actor network $\boldsymbol{\theta^\pi}$ is updated by computing the gradient $\nabla_{\boldsymbol{\theta^\pi}}$ (Equation 6). In our implementation, we set the minibatch length $H = 1024$ and the discount factor $\gamma = 0.9$. Then the target networks are slowly updated with a controled updating rate τ (Line 30-32).

In our design and implementation, we used a 2-layer fully-connected feedforward neural network to serve as the actor network, which includes 400 and 300 neurons in the first and second layers respectively, and utilized the ReLU [39] function for activation. In the final output layer, we used $\tanh(\cdot)$ as the activation function to bound the actions. Similarly, for

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

the critic network, we also used a 2-layer fully-connected feedforward neural network with 400 and 300 neurons in the first and second layers respectively, and with ReLU for activation. Besides, we utilized the L_2 weight decay [40] to prevent overfitting. These DNNs (i.e., the actor and critic networks) were implemented using TensorFlow 1.4.

VI. Performance Evaluation

We conducted simulation to evaluate the performance of the proposed DRL-EC^3. In this section, we first describe simulation settings and then present results and analysis.

A. Simulation Settings

In our simulation, we set the target region to be a square area with a size of 10×10 units, where each unit corresponds to 100 meters. We divided this region to 100 cells, each of which has a unit size. Hence, we had $K = 100$ PoIs in the centers of these cells. We set the communication range $R = 5$ units. The energy consumption of a hovering (stationary) UAV during a timeslot (i.e., epoch) is 1 unit. Our simulation runs were performed with Tensorflow 1.4 and Python 3.5 on a Ubuntu 16.04.3 server with 4 NVIDIA TITAN XP GPUs. We trained the proposed DRL-based method for 1000 episodes, each of which has 1000 epochs. After training, we tested it for a period of $T = 1000$ epochs (i.e., timeslots).

We used the following metrics for performance evaluation.

- *Average Coverage Score ($\bar{c}$)*: This is the average PoI coverage score at the end of the testing period. Each PoI's coverage score can be obtained using Equation (1).
- *Fairness Index (f)*: This is the Jain's fairness index with regards to PoI coverage scores at the end of the testing period, which can be calculated using Equation (2).
- *Normalized Average Energy Consumption ($\bar{E}$)*: This is the average UAV energy consumption during the test period, which is normalized by $E_{\max}$. $E_{\max}$ is the maximum possible total energy consumption of a UAV during the test period, which corresponds to the case where the UAV flies the maximum distance in each timeslot.
- *Energy Efficiency (r)*: This is similar to the reward, which is calculated using the following equation:

$$r = \frac{f * \bar{c}}{\bar{E}'} \tag{8}$$

  where f, $\bar{c}$ are the fairness index and average coverage score defined above respectively. $\bar{E}'$ is almost the same as the above normalized average energy consumption except the normalization is done using a fixed energy consumption model.

We compared DRL-EC^3 with two commonly-used baseline methods, Random and Greedy.

- *Random*: This is an extension to a simple random method. At each timeslot, it randomly selects a moving direction within $(0, 2\pi]$ and a flying distance within $[0, 1]$ as the current action for each UAV. If the new location is beyond the target region boundary or any UAV becomes disconnected after executing this action, then all UAVs abandon this action and stay put.
- *Greedy*: This is an extension to a greedy method. At each timeslot, it sequentially chooses the moving direction from $\{1, \cdots, 360\}$ and flying distance from $\{0, 0.5, 1\}$ that can maximize the instantaneous reward for every UAV subject to the region boundary and connectivity constraints described above.

B. Results and Analysis

1) Comparison with Other Solutions: In simulation scenario 1, we show the impact of the UAV coverage range, the number of UAVs and the *energy consumption ratio* (the ratio between energy consumed for flying with the maximum distance to hovering energy) on energy efficiency (Equation 8) in Fig. 2. In scenario 1.a, we fixed the number of UAV to 7, and the energy consumption ratio to 10:5, while we changed the UAV coverage range from 1.75 to 3 with a step size of 0.25. In scenario 1.b, the UAV coverage range and the energy consumption ratio were fixed at 2.5 and 10:5 respectively, while the number of UAVs was changed from 5 to 10. In scenario 1.b, we fixed the number of UAVs and the UAV coverage range at 7 and 3 respectively, while we changed the energy consumption ratio from 10:8 to 10:3.

We can make the following observations from this figure:

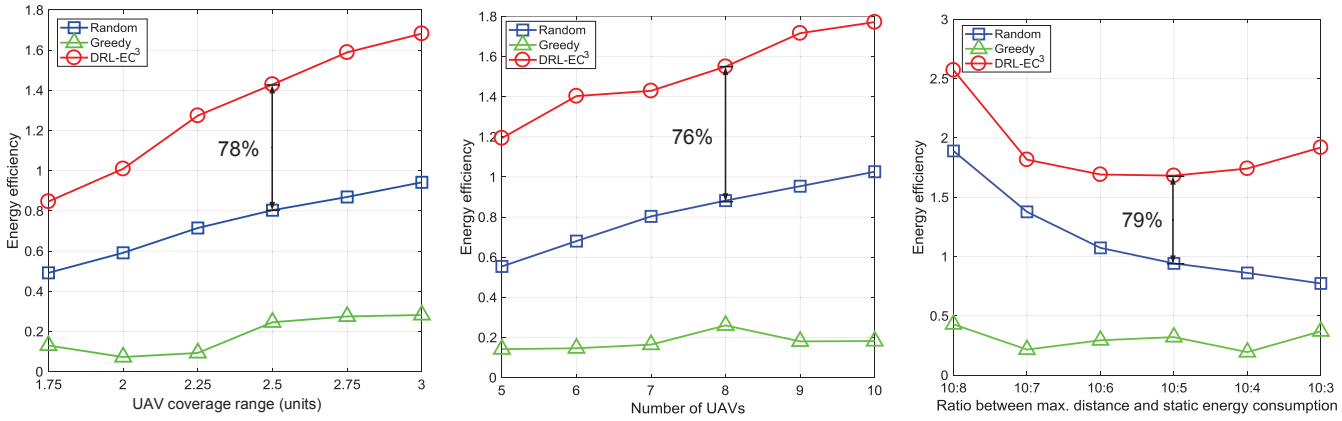
(1) DRL-EC^3 consistently outperforms both baselines in terms of energy efficiency. For example, In Fig. 2(a), when the coverage range is 2.5, DRL-EC^3 achieves an energy efficiency of 1.43, compared to 0.80 given by the best baseline, Random, which represents a 78% improvement. In Fig. 2(b), when the number of UAVs is 8, DRL-EC^3 gives an energy efficiency of 1.55, which makes an improvement of 76% compared to 0.88 given by Random. In Fig. 2(c), DRL-EC^3 achieves an energy efficiency of 1.68 and outperforms Random by 79% when the energy consumption ratio is 10:5. On average, DRL-EC^3 significantly improves energy efficiency by 80% and 671% over Random and Greedy respectively.

(2) From Fig. 2(a), it can be observed that the energy efficiency of DRL-EC^3 increases monotonically with the coverage range. This is because a larger coverage range certainly leads to better coverage (without any additional energy) thus better energy efficiency.

(3) From Fig. 2(b), we can see that the energy efficiency of DRL-EC^3 increases monotonically with the number of UAVs. This shows that the proposed method can make a good use of every UAV. That is, when more UAVs are provided, DRL-EC^3 can well utilize them to improve coverage in an energy-efficient manner.

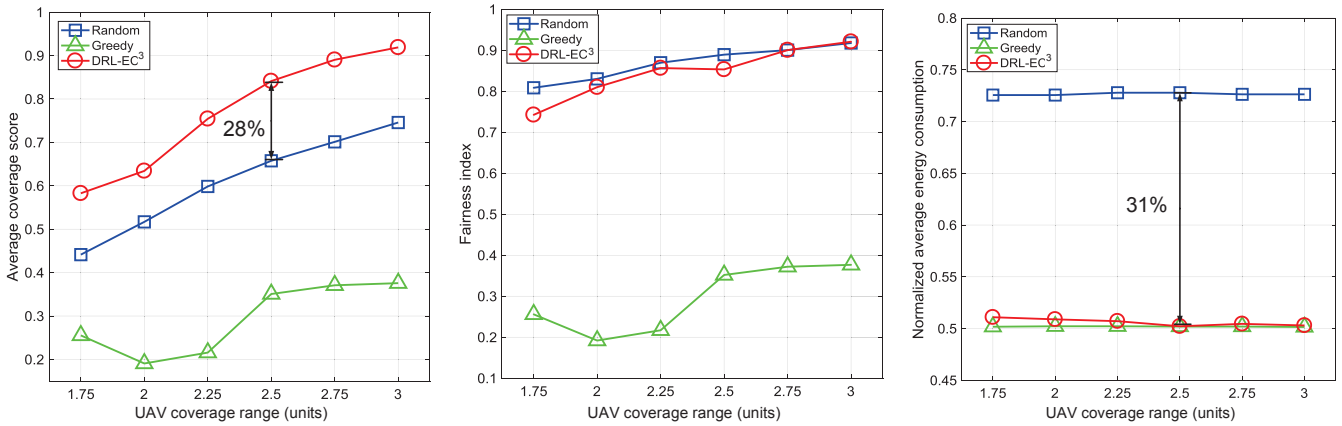
(4) From Fig. 2(c), we can make an interesting observation that the energy efficiency of DRL-EC^3 does not increase/decrease monotonically with the energy consumption ratio. This is because its impact on energy efficiency is fairly complicated. Specifically, a small energy consumption ratio encourages UAVs to move around more, which will likely lead to better coverage and fairness but more energy consumption. On the contrary, a small ratio discourages UAVs' movements, which will likely hurt fairness but save energy; and it is hard to tell how it will affect coverage. It is worth mentioning that DRL-EC^3 consistently outperforms both baseline no matter what the

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications



(a) (b) (c)

Fig. 2: The impact of (a) communication range, (b) number of UAVs, and (c) energy consumption ratio on energy efficiency.



(a) (b) (c)

Fig. 3: The impact of the coverage range on (a) average coverage score, (b) fairness index, and (c) normalized average energy consumption.

energy consumption model (i.e., ratio) becomes, which well justifies its robustness.

(5) We can also see that Random performs consistently better than Greedy. This may be due to two reasons: (a) Greedy sequentially determines an action for every UAV in each timeslot, which may lead to suboptimal solutions. (b) Greedy needs to discretize the solution space for greedy action selection, which may lead to poor performance.

We show the impact of the coverage range on the average coverage score, the fairness index and the normalized average energy consumption in simulation scenario 2, whose settings are the same as those in scenario 1.a. We can make the following observations from Fig. 3:

(1) DRL-EC^3 outperforms both baselines in terms of the average coverage score and the normalized average energy consumption. For example, In Fig. 3(a), when the coverage range is 2.5, DRL-EC^3 achieves an average coverage score of 0.84 compared to 0.66 obtained by Random, which represents a 28% improvement. In Fig. 3(b), DRL-EC^3 and Random achieve almost the same fairness index when the coverage range is 2.5. In Fig. 3(c), when the coverage range is 2.5, DRL-EC^3 achieves a normalized average energy consumption of 0.50 , which represents a 31% reduction compared to 0.73 given by Random. On average, DRL-EC^3 improves the average coverage score by 26% and reduces the average energy consumption by 30%. Moreover, DRL-EC^3 significantly improves the average coverage score and the fairness index by 173% and 206% respectively over Greedy on average.

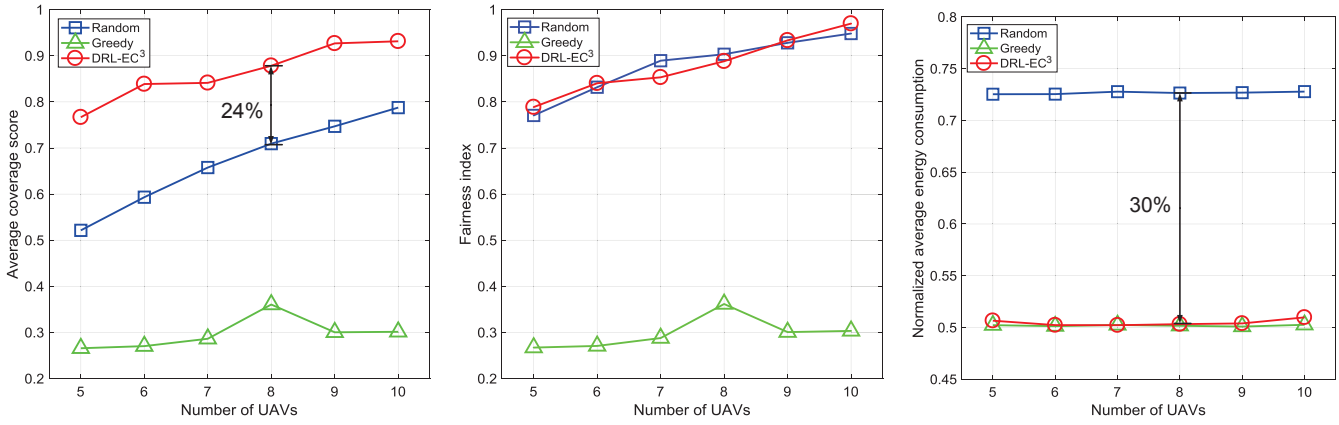
(2) It is expected that the average coverage score given by DRL-EC^3 increases monotonically with the coverage range, as shown in Fig. 3(a). This is easy to understand since a longer coverage range can certainly improve coverage.

(3) In Fig. 3(b), we can see that DRL-EC^3 and Random increase slightly with the coverage range and both of them have comparable fairness indices over 0.8 (in most cases), which means both of them can lead to very fair coverage. Random can certainly achieve good fairness due to its nature of random movements. DRL-EC^3 can provide comparable results, which well justify its effectiveness on fairness.

(4) In Fig. 3(c), we can observe that the coverage range does not make a significant impact on average energy consumption since they obviously have pretty loose correlations.

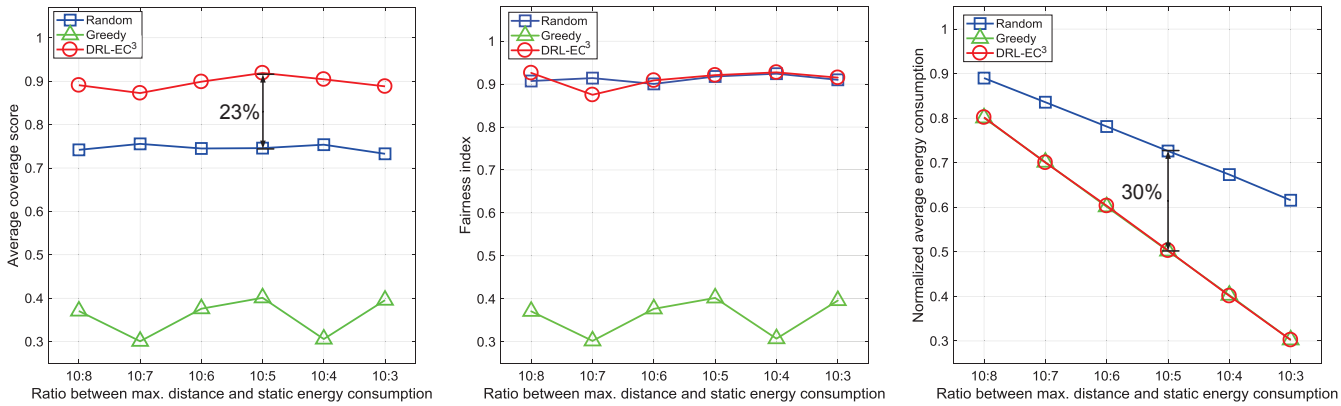
Fig. 4 shows the impact of the number of UAVs on the average coverage score, the fairness index and the normalized average energy consumption in simulation scenario 3, whose settings are the same as those in scenario 1.b. We can make

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications



(a) (b) (c)

Fig. 4: The impact of the number of UAVs on (a) average coverage score, (b) fairness index, and (c) average energy consumption.



(a) (b) (c)

Fig. 5: The impact of the energy consumption ratio on (a) average coverage score, (b) fairness index, and (c) normalized average energy consumption.

the following observations from this figure:

(1) DRL-EC^3 beats both baselines in terms of the average coverage score and the normalized average energy consumption. For instance, In Fig. 4(a), when the number of UAVs is 8, DRL-EC^3 achieves an average coverage score of 0.89, which represents an improvement of 24% over Random. In Fig. 4(b), similar as the last scenario, DRL-EC^3 and Random achieve almost the same fairness when the number of UAV is 8. In Fig. 4(c), DRL-EC^3 gives a normalized average energy consumption of 0.50, which represents a 30% reduction compared to Random, when the number of UAVs is 8. On average, DRL-EC^3 reduces the normalized average energy consumption by 31%, and significantly improves the average coverage score by 30% compared to Random. Moreover, DRL-EC^3 significantly improves the average coverage score and fairness index by 192% and 196% respectively over Greedy on average.

(2) In Fig. 4(a), as the number of UAV increases, the average coverage score given by DRL-EC^3 monotonically improves, since more UAVs can provide more flexibility on covering PoIs thus better coverage. Particularly, when the number of UAVs is sufficiently large (larger than 9), DRL-EC^3 can achieve a very high coverage (more than 90%).

(3) From Fig. 4(b), we can make similar observations about fairness index as those in Fig. 3(b).

(4) From Fig. 4(c), we can make an interesting observation that the average energy consumption does not change much with the number of UAVs, no matter which method is used. More UAVs do not necessarily lead to more energy consumption since more UAVs may lead to shorter distance movements, which can somehow save energy.

Finally, we show the impact of energy consumption ratio on the average coverage score, the fairness index and the normalized average energy consumption in scenario 4 using Fig. 5, whose settings are the same as those in scenario 1.c. We can make following observations form this figure:

(1) We can see that DRL-EC^3 consistently outperforms two baselines in terms of the average coverage and the normalized average energy consumption. For example, In Fig. 5(a), when the energy consumption ratio is 10:5, the average coverage score of DRL-EC^3 is 0.92 compared to 0.75 given by Random, which represents an improvement of 23%. In Fig. 5(b), DRL-EC^3 and Random achieve almost the same fairness index, when the energy consumption ratio is 10:5. In Fig. 5(c), DRL-EC^3 obtains a normalized average energy consumption of 0.50, which represents a 30% reduction compared to Random, when energy consumption ratio is 10:5. On average, DRL-EC^3 reduces normalized average energy consumption by 28%

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

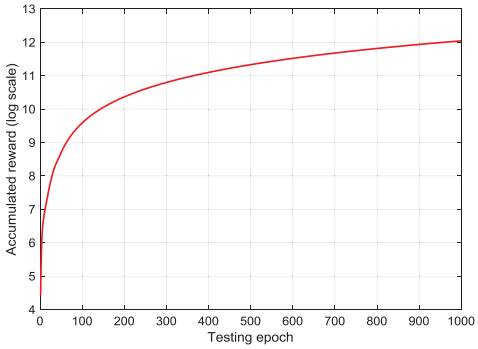


Fig. 6: Accumulated reward over time during testing

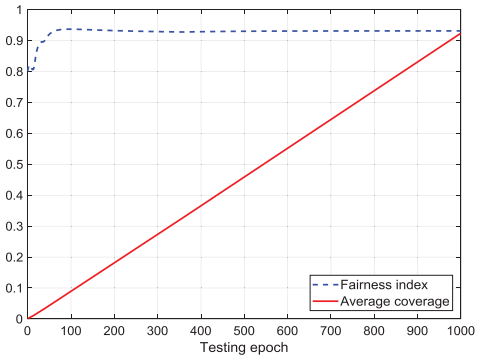


Fig. 7: Average coverage score and fairness index over time during testing

and improves the average coverage score by 20%, compared to Random. Moreover, DRL-EC^3 significantly improves the average coverage score and the fairness index by 153% and 157% respectively over Greedy on average.

(2) From Fig. 5(a) and Fig. 5(b), we can observe that if DRL-EC^3 is used, the energy consumption ratio does not have a significant impact on coverage and fairness, which well justify robustness of the proposed method in terms of the energy consumption model.

(3) From Fig. 5(c), we can see that no matter which method is used, the average energy consumption decreases sharply with the energy consumption ratio. As mentioned above, a small energy consumption ratio encourages UAVs to move around more, which will likely lead to more energy consumption. So this observation is consistent with our observation from Fig. 2(c).

2) Convergence and Impact of Hyper-parameters: We first show the reward, the average coverage and its fairness change over time during testing. In this simulation scenario, we used 7 UAVs and set the coverage range and the energy consumption ratio to 3 and 10:5 respectively.

Fig. 6 shows the accumulated reward over time (epochs). We can see that the accumulated reward (in $\log$ scale) arises monotonically over time. When it reaches 100 epochs (one tenth of the task), the growth slows down. This is because that at the beginning of the task, many PoIs have not yet been covered and the coverage is unfair such that an action can result in significant improvement on the reward. This improvement diminishes when the PoIs are well and fairly covered. A similar observation has also been made in [6]. Fig. 7 shows how the fairness index and coverage change over time. As expected, the coverage increases almost linearly over time and eventually reaches a high value (over 90%). An interesting observation is the fairness index quickly reaches a high value (over 0.9) and stays there for the rest of the testing period. This well justifies that DRL-EC^3 can provides effective and fair coverage.

We next show the impact of some key hyperparameters including the number of neurons of the used actor-critic network, and the discount factor, on average coverage scores, fairness index, normalized average energy consumption and energy efficiency. Results are presented in Table II. We used three different sets of neuron numbers for 2-layer fully-connected feedforward neural network, which is used in actor network, critic network and target network. Structure A has 200 and 100 neurons in the first and second layers, respectively. Structure B has 400 and 300 neurons while Structure C has 600 and 500 neurons, respectively. In each structure, we fixed the number of UAV to 7, UAV coverage range to 2.5 units and energy consumption ratio to 10:5, while the discount factor γ is changed from 0.8, 0.9 to 0.99. When γ is fixed, we observe that energy efficiency keeps increasing when using more neurons. This is because that appropriate number of neurons can improve the capacity of DNNs, leading to find a better solution, i.e. $Q(\cdot)$ and $\pi(\cdot)$. With the same network structure, energy efficiency is also increasing when γ increases. For instance, energy efficiency improves 0.06 when γ changes from 0.9 to 0.99 with Structure B. Although average coverage score and normalized average energy consumption may not change much, there is a remarkable improvement of fairness index which leads to the overall energy efficiency increase. This is because that bigger γ means longer-term consideration of future reward $r(\cdot)$. Since our considered scenario is to maximize the long-term communication coverage, increasing γ helps UAVs to navigate in such a way that future movement will provide more effective communication coverage in a long run.

VII. Discussions

In this section, we discuss two practical implementation issues related to the UAV navigation problem we considered in this paper, decentralized multi-agent control solution and scalability issue.

A. Decentralized Multi-Agent Control Solution

We primarily considered in this paper as a centralized approach, where the decisions (as the action of UAVs) are made by the back-end computational server. In extreme conditions like disaster, communications bandwidth is quite limited and cannot support much information delivery between UAV and server back and forth, and thus decentralized solution is expected. In [41], a multi-agent DDPG (called MADDPG) is proposed as an adaptation of actor-critic methods that considers action policies of other agents and is able to successfully learn policies that require complex multi-agent coordination. However, directly applying it will not work in our scenarios

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/JSAC.2018.2864373, IEEE Journal on Selected Areas in Communications

TABLE II: Impact of different hyperparameters.

discount factor γ	0.8	0.9	0.99	0.8	0.9	0.99	0.8	0.9	0.99	0.8	0.9	0.99
Structure A	$\bar{c}$=0.815	$\bar{c}$=0.813	$\bar{c}$=0.846	f=0.867	f=0.879	f=0.870	$\bar{E}$=0.506	$\bar{E}$=0.509	$\bar{E}$=0.503	r=1.398	r=1.406	r=1.462
Structure B	$\bar{c}$=0.816	$\bar{c}$=0.841	$\bar{c}$=0.835	f=0.898	f=0.853	f=0.905	$\bar{E}$=0.520	$\bar{E}$=0.502	$\bar{E}$=0.508	r=1.409	r=1.429	r=1.489
Structure C	$\bar{c}$=0.819	$\bar{c}$=0.849	$\bar{c}$=0.849	f=0.928	f=0.897	f=0.891	$\bar{E}$=0.526	$\bar{E}$=0.515	$\bar{E}$=0.505	r=1.444	r=1.476	r=1.497
	Average coverage score $\bar{c}$			Fairness index f			Normalized energy consumption $\bar{E}$			Energy efficiency r		

since state, action, and reward are completely different. We can easily extend our proposed DRL-EC^3 solution by allowing each UAV only observes its covered PoI coverage and its own energy consumption to generate a reward, and back-end server collects all UAVs' information at the end of each epoch for critic network evaluation.

B. Scalability Issue

From implementation point of view, our proposed DRL-EC^3 scale well with the increase of number of UAVs and PoIs. This is because that we used only one actor-critic neural network, one experience replay buffer (to store historical samplings), and thus when more UAVs and/or PoIs are considered, the only change is to enlarge the state space and action space, while all the rest of Algorithm 0 remains the same.

VIII. Conclusion

In this paper, we proposed a novel and highly energy-efficient DRL-based method for UAV control, which we call DRL-EC^3 (DRL-based Energy-efficient Control for Coverage and Connectivity). Specifically, DRL-EC^3 maximizes a novel energy efficiency function with joint consideration for communications coverage, fairness, energy consumption and connectivity, based on a recent actor-critic method, DDPG; and makes decisions under the guidance of two Deep Neural Networks (DNNs). We conducted extensive simulation for performance evaluation. Simulation results have shown that DRL-EC^3 significantly and consistently outperforms two commonly-used baseline methods, Random and Greedy, in terms of four metrics, including average coverage score, fairness index, average energy consumption and energy efficiency.

References

[1] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *IEEE Globecom'15*, 2015, pp. 1–6.

[2] L. Zhong, K. Garlichs, S. Yamada, K. Takano, and Y. Ji, "Mission planning for uav-based opportunistic disaster recovery networks," in *IEEE CCNC'18*, 2018, pp. 1–9.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wirel. Comm.*, vol. 15, no. 6, pp. 3949–3963, 2016.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR'16*, 2016.

[7] M. D. Benedetti, F. D'Urso, G. Fortino, F. Messina, G. Pappalardo, and C. Santoro, "A fault-tolerant self-organizing flocking approach for uav aerial survey," *J. Network and Computer Applications*, vol. 96, pp. 14–30, 2017.

[8] P. Pace, G. Aloi, G. Caliciuri, and G. Fortino, "A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects," *ACM/Springer MONET*, vol. 21, no. 4, pp. 708–725, 2016.

[9] Y. Zhang, "Grorec: A group-centric intelligent recommender system integrating social, mobile and big data technologies," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 786–795, 2016.

[10] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: A survey," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 585–598, 2015.

[11] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "Sovcan: Safety-oriented vehicular controller area network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 94–99, 2017.

[12] J. Li, Y. Zhou, and L. Lamont, "Communication architectures and protocols for networking unmanned aerial vehicles," in *IEEE Globecom'13*, 2013, pp. 1415–1420.

[13] Y. Shibata, N. Tanaka, and N. Uchida, "Information communication system consisted of multiple unmanned aerial vehicles on disaster," in *IEEE WAINA'17*, 2017, pp. 627–632.

[14] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications," *IEEE Trans. Wirel. Comm.*, vol. 16, no. 11, pp. 7574 – 7589, 2017.

[15] ——, "Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization," *IEEE Trans. Wirel. Comm.*, vol. 16, no. 12, pp. 8052 – 8066, 2017.

[16] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-based energy-efficient data collection by unmanned vehicles in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1666–1676, 2018.

[17] Y. Ben-Asher, S. Feldman, P. Gurfil, and M. Feldman, "Distributed decision and control for cooperative uavs using ad hoc communication," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 511–516, 2008.

[18] C. Secchi, A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Bilateral teleoperation of a group of uavs with communication delays and switching topology," in *IEEE ICRA'12*, 2012, pp. 4307–4314.

[19] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor uav using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.

[20] C. H. Liu, T. He, K. W. Lee, K. K. Leung, and A. Swami, "Dynamic control of data ferries under partial observations," in *IEEE WCNC'10*, 2010, pp. 1–6.

[21] C. H. Liu, J. Zhao, H. Zhang, S. Guo, K. K. Leung, and J. Crowcroft, "Energy-efficient event detection by participatory sensing under budget constraints," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2490–2501, 2017.

[22] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1435–1446, 2017.

[23] B. Zhang, C. H. Liu, J. Lu, Z. Song, Z. Ren, J. Ma, and W. Wang, "Privacy-preserving qoi-aware participant coordination for mobile crowdsourcing," *Elsevier Computer Networks*, vol. 101, no. 4, pp. 29–41, 2016.

[24] C. H. Liu, J. Fan, P. Hui, J. Wu, and K. K. Leung, "Towards qoi and energy-efficiency in participatory crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4684–4700, 2015.

[25] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and control of multiple uavs," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences*, 2002.

[26] A. Richards and J. How, "Decentralized model predictive control of cooperating uavs," in *43rd IEEE Conference on Decision and Control'04*, vol. 4, 2004, pp. 4286–4291.

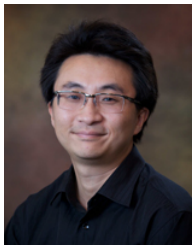
[27] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Ultra reliable uav communication using altitude and cooperation diversity," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 330 – 344, 2018.

[28] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Optimal complete terrain coverage using an unmanned aerial vehicle," in *IEEE ICRA'11*, 2011, pp. 2513–2519.
[29] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3d placement of an unmanned aerial vehicle base station for maximum coverage of users with different qos requirements," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 38–41, 2018.
[30] H. Shakhatreh, A. Khreishah, A. Alsarhan, I. Khalil, A. Sawalmeh, and N. S. Othman, "Efficient 3d placement of a uav using particle swarm optimization," in *IEEE ICICS'17*, 2017, pp. 258–263.
[31] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Comm. Lett.*, vol. 20, no. 8, pp. 1647–1650, 2016.
[32] 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.
[33] M. Elloumi, B. Escrig, R. Dhaou, H. Idoudi, and L. A. Saidane, "Designing an energy efficient uav tracking algorithm," in *IEEE IWCMC'17*, 2017, pp. 127 – 132.
[34] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2015, pp. 111–117.
[35] 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 *International Conference on Machine Learning*, 2016, pp. 1928–1937.
[36] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, "Q-prop: Sample-efficient policy gradient with an off-policy critic," in *ICLR'17*, 2017.
[37] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984, vol. 38.
[38] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *AAAI*, 2016, pp. 2094–2100.
[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[40] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
[41] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *NIPS'17*, 2017, pp. 2094–2100.

Chi Harold Liu (SM'15) receives the Ph.D. degree from Imperial College, UK in 2010, and the B.Eng. degree from Tsinghua University, China in 2006.

He is currently a Full Professor and Vice Dean at the School of Computer Science and Technology, Beijing Institute of Technology, China. He is also the Director of IBM Mainframe Excellence Center (Beijing), Director of IBM Big Data Technology Center, and Director of National Laboratory of Data Intelligence for China Light Industry. Before moving to academia, he joined IBM Research - China as a staff researcher and project manager, after working as a postdoctoral researcher at Deutsche Telekom Laboratories, Germany, and a visiting scholar at IBM T. J. Watson Research Center, USA. His current research interests include the Internet-of-Things (IoT), Big Data analytics, mobile computing, and deep learning. He received the Distinguished Young Scholar Award in 2013, IBM First Plateau Invention Achievement Award in 2012, and IBM First Patent Application Award in 2011 and was interviewed by EEWeb.com as the Featured Engineer in 2011. He has published more than 80 prestigious conference and journal papers and owned more than 14 EU/U.S./U.K./China patents. He serves as the Area Editor for KSII Trans. on Internet and Information Systems and the book editor for six books published by Taylor & Francis Group, USA and China Machinery Press. He also has served as the general chair of IEEE SECON'13 workshop on IoT Networking and Control, IEEE WCNC'12 workshop on IoT Enabling Technologies, and ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He served as the consultant to Asian Development Bank, Bain & Company, and KPMG, USA, and the peer reviewer for Qatar National Research Foundation, and National Science Foundation, China. He is a Senior Member of IEEE.

Zheyu Chen received the B.Eng. degree from the School of Software at Beijing Institute of Technology, China in 2017, and now is a MSc student under the supervision of Prof. Chi Harold Liu. His research interests include airborne communications network and deep learning.



Jian Tang (SM'10) received the PhD degree in computer science from Arizona State University, in 2006. He is an associate professor in the Department of Electrical Engineering and Computer Science, Syracuse University. His research interests lie in the areas of cloud computing, big data and wireless networking. He has published more than 90 papers in premier journals and conferences. He received an NSF CAREER award in 2009, the 2016 Best Vehicular Electronics Paper Award from IEEE Vehicular Technology Society, and Best Paper Awards from the 2014 IEEE International Conference on Communications (ICC) and the 2015 IEEE Global Communications Conference (Globecom) respectively. He has been an editor for the IEEE Transactions on Wireless Communications since 2016, for the IEEE Transactions on Vehicular Technology since 2010 and for the IEEE Internet of Things Journal since 2013. He served as a TPC co-chair for the 2015 IEEE International Conference on Internet of Things (iThings) and the 2016 International Conference on Computing, Networking and Communications (ICNC). He also served as a Area TPC chair for IEEE INFOCOM 2018, Vice TPC Chair for INFOCOM 2019, and TPC members for many international conferences, including IEEE INFOCOM 2010-2017, ICDCS 2015, ICC 2006-2019, Globecom 2006-2019, etc. He is a Senior Member of IEEE.

Jie Xu received the B.Eng. and MSc degrees both from the School of Software at Beijing Institute of Technology, China in 2015 and 2018, respectively, and now is a Software Engineer at Tencent Corp. Ltd focusing on developing artificial intelligence products. He receives the Best Paper Award from IEEE DataCom'16, and his research interests include big data systems, recommender systems and deep learning. He is a student member of IEEE.

Zhengzhe Piao is currently an M.Eng. student at the School of Computer Science, Beijing Institute of Technology, China, and he is working on the problems of fast online video tracking by deep learning and airborne communications networks.