Three-Dimensional Continuous Movement Control of Drone Cells for Energy-Efficient Communication Coverage

Peng Yang, Xianbin Cao, *Senior Member, IEEE*, Xing Xi, Wenbo Du, Zhenyu Xiao, and Dapeng Wu, *Fellow, IEEE*

***Abstract*—This paper is concerned with the efficient movement control of multiple drone cells for communication coverage. Although many works have been developed to cope with this problem, but only few of them have investigated the three-dimensional (3-D) continuous movement control of multiple drone cells. In this paper, a problem of 3-D continuous movement control of multiple drone cells is formulated with an objective of maximizing the energy-efficient communication coverage of drone-cell networks while preserving the network connectivity. To mitigate this challenging problem, an energy-efficiency and continuous-movement-control algorithm (E^2CMC), which is based on an emerging deep reinforcement learning method, is proposed. In E^2CMC, an energy-efficiency reward function considering the energy consumption, the sum of quality-of-service (QoS) requirements of users as well as the coverage fairness is first designed. Next, E^2CMC learns to adjust drone cells' locations continuously by interacting with an environment in a sequence of observations, actions, and rewards. Furthermore, E^2CMC will reduce the reward drastically as long as the networks are disconnected. Simulation results verify the superiority of the proposed learning algorithm on deploying multiple drone cells to provide the communication coverage.**

***Index Terms*—Multiple drone-cells, continuous movement control, communication coverage, deep reinforcement learning.**

I. Introduction

FUTURE wireless networks need both high agility-and-resilience and the capability of fast communication service recovery [1]. When encountering either unexpected or temporary situations (e.g., natural disasters, sports events), extreme densities of users may crowd in a region. It then will be timely and/or financially infeasible to invest in telecommunication infrastructures that can achieve revenue for a short period of time. As unmanned aerial vehicles (UAVs) have a unique rapid response opportunity and reduced vulnerability to natural disasters, a potential solution to mitigate the above situations is to resort to the drone-cells-assisted-coverage, where drone-cells (i.e., low-altitude UAVs acting as aerial base stations or flying relays [1], [2]) are deployed to deliver radio access services to those crowded users. Furthermore, there is substantial interest in the research community towards the utilization of drone-cells as radio access platforms in future wireless networks [3]–[7].

To achieve the promising drone-cells-assisted-coverage scheme, an efficient deployment problem of drone-cells under specific communication constraints (e.g., quality-of-service (QoS) requirements of users and average network delay) should be firstly investigated [1], [8], [9].

A. Prior Works

Recently, much attention from the academia has been paid to the under-studied deployment problem of one/multiple drone-cell(s) in both two-dimensional (2-D) [8]–[16] and three-dimensional (3-D) [17]–[19] airspace. In these works, the deployment problem is first formulated as a mathematical programming problem and then many approaches such as mathematical-programming-based [9]–[12] and dimension-reduction-based [1], [18], [19] approaches are developed to find the optimal/suboptimal location(s) of the drone-cell(s). For example, Mozaffari *et al.* partitioned the UAV deployment problem into two sub-problems. The goal of the first sub-problem was to optimize cell partitions by assuming that positions of UAVs were fixed, where the cell was the coverage region of a UAV. Given the cell partitions, the second sub-problem was responsible for finding the optimal UAV positions. By optimizing these two sub-problems iteratively, the optimal UAV horizontal positions were obtained [9]. Alzenad *et al.* obtained the UAV position by optimizing the altitude and the horizontal location of a UAV independently. Specifically, they optimized first the UAV deployment altitude. Given this optimal altitude, the UAV horizontal location was then achieved by solving a 2-D optimal deployment problem using optimization tools [18]. These studies, however, focus on discussing time-independent deployment problems and do not consider the inherent time-varying nature

Manuscript received December 15, 2018; revised March 14, 2019; accepted April 24, 2019. Date of publication April 29, 2019; date of current version July 16, 2019. This work was supported by the National Natural Science Foundation of China under Grants 61425014, 91738301, and 61827901. The review of this paper was coordinated by Dr. Z. Fadlullah. *(Corresponding author: Xianbin Cao.)*

P. Yang, X. Cao, X. Xi, W. Du, and Z. Xiao are with the School of Electronic and Information Engineering, Beihang University, Beijing 100083, China, and also with the Key Laboratory of Advanced Technology, Near Space Information System (Beihang University), Ministry of Industry and Information Technology of China, Beijing 100083, China (e-mail: yangp09@buaa.edu.cn; xbcao@buaa.edu.cn; xixing@buaa.edu.cn; wenbodu@buaa.edu.cn; xiaozy@buaa.edu.cn).

D. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: dpwu@ieee.org).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TVT.2019.2913988

of an environment. The time-varying environment may result in significant degradation of deployment performance of these studies [17].

To mitigate the impact of the time-varying environment, dynamic deployment problems of drone-cells have been researched in recent years [17], [20]. For example, to compensate the QoS loss owing to the movement of terrestrial users, Ghanavi *et al.* proposed to assist ground infrastructure with a drone-cell and developed a Q-learning-based algorithm to identify its location in 3-D airspace efficiently [17]. Cheng *et al.* focused on optimizing a UAV trajectory for maximizing the sum rate of ground users, while ensuring that users' rate requirements could be satisfied [21]. Although these works explore time-dependent deployment problems, they do not take the energy consumption optimization of drone-cell networks into consideration. Energy-saving operation fashion is crucial for drone-cell networks (especially, small drones) as it can prolong the lifetime of drone-cell networks [3].

Some research efforts have investigated energy efficiency for the control of drone-cells. For instance, Zeng *et al.* explored a throughput maximization problem in UAV relaying systems by optimizing the UAV transmit power as well as the UAV trajectory, subject to UAV mobility constraints [22]. Chen *et al.* proposed to leverage user-centric information to deploy cache-enabled UAVs while maximizing users' Quality-of-Experience (QoE) with the minimum UAV transmit power [14]. Besides, Wu *et al.* studied a multi-UAV enabled wireless communication system that jointly optimized multiuser communication scheduling and energy-efficient UAV trajectory for maximizing the minimum rate of ground users [10].

B. Motivation and Contributions

Different from previous research efforts, this paper focuses on optimizing energy consumption for drone-cell movements (with consideration for constraints of network connectivity, moving airspace, and communication requirements), rather than energy consumed by controller units, data processing and transmission [23]. Particularly, this paper studies the continuous movement control of multi-drone-cells with the minimum movement energy consumption to maximize the total user throughput and achieve fair communication coverage, while satisfying users' QoS and drone-cell networks' connectivity requirements. The continuous movement here indicates that drone-cells have a continuous (real-valued) action space.

This task is quite challenging because drone-cells have a significant energy constraint and limited communication coverage ranges. First, due to the significant energy constraint, drone-cells should be operated in an energy-saving fashion (e.g., hovering at current locations instead of flying around) so that the lifetime of drone-cell networks can be prolonged. Second, owing to limited coverage ranges and relatively high acquisition and maintenance costs, it is impossible to have sufficient drones to serve a large target region all the time. To provide fair communication services for the target region, drones need to fly around to ensure that each small area is served for a reasonable amount of time, which will inevitably consume a large amount of energy, and thus shorten the network lifetime. Therefore, the objective of minimizing the energy consumption for drone-cell movements and the goal of obtaining fair communication services are contradictory, and it is considerably difficult to strike a good trade-off between the coverage fairness and the level of energy consumption. Besides, the network connectivity is crucial for drone-cell networks. This is because the interconnected drone-cell networks cannot only save the transmission energy due to the shorter air-to-air (AtA) transmission links but also help enhance the reliability of the networks [24].

This paper explores an emerging deep reinforcement learning (DRL) method [25] to mitigate the above challenging issues. The main benefits of the DRL are: 1) DRL can well deal with control problems with sophisticated state space and time-varying environments using limited to even zero domain knowledge [25], [26]; 2) DRL can use powerful deep neural networks to develop strategies; 3) DRL has achieved significant success on a few game-playing tasks (e.g., Torcs, CartPole) [25]. Thus, invoking DRL to drone-cell networks provides a promising proposal for intelligent mobile communication services. The exploration of the DRL method in communication networks, however, is still in its infancy, and it remains unknown whether the DRL method can succeed in control tasks in complex communication networks or not. In addition, the control problem in this paper is much more complicated than control problems in games as it involves multiple considerably different objectives including the energy consumption, the sum of users' QoS requirements, and the coverage fairness as well as a constraint on the network connectivity. Therefore, it is highly challenging to design DRL-based drone-cell networks to achieve energy-efficient communication coverage.

Motivated by advantages of DRL, this paper aims to develop a 3-D continuous movement control framework of multiple drone-cells for energy-efficient communication coverage. Specifically, a multiple drone-cells enabled downlink wireless communication network is considered, in which multiple drone-cells attempt to communicate with terrestrial users independently. Each drone-cell flies in a pre-defined 3-D airspace in terms of deterministic and continuous policies made by powerful deep neural networks. Based on the framework, the major contributions of this paper fall into four aspects:

- This paper explores a problem of continuous movement control of multiple drone-cells for providing efficient communication coverage to terrestrial users by simultaneously considering communication requirements and constraints of both drone-cells and users. Specifically, an optimization problem with a goal of maximizing the energy-efficient communication coverage of drone-cell networks is formulated. Besides, this problem is constrained by QoS requirements of users, connectivity of drone-cell networks and flying airspace of each drone-cell. Owing to a time-dependent characteristic, a high-dimensional variable space and complex constraints, the formulated optimization problem is non-trivial. To mitigate this challenging problem, this paper proposes a DRL-based drone-cell movement control algorithm.
- This paper performs fundamental analysis on optimal deployment altitudes of drone-cells. Particularly, the relationship between a deployment altitude of a drone-cell, which is used to cover terrestrial users with different QoS

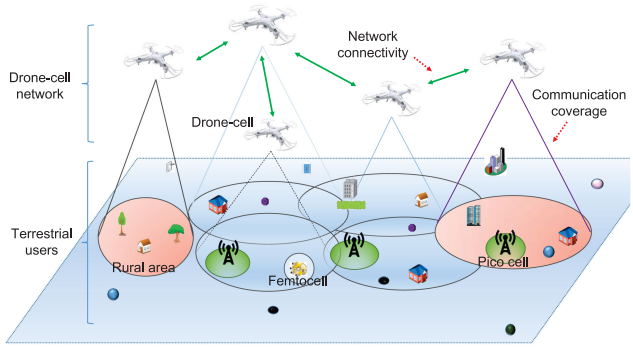


Fig. 1. A 3-D drone-cell deployment scenario.

requirements, and the corresponding maximum coverage radius of the drone-cell is discussed. Next, the upper bound of the optimal drone-cell deployment altitude is derived in terms of the altitude-radius correspondence.

- Based on an emerging DRL method and the derived theoretical results, this paper develops an energy-efficiency and continuous-movement-control algorithm (E^2CMC) to alleviate the formulated optimization problem efficiently. In E^2CMC, a novel reward function considering the energy consumption for drone-cell movements, the sum of users' QoS requirements, and coverage fairness of drone-cell networks is designed. Reward penalty mechanisms are also independently developed to alleviate the violation of both network connectivity and airspace boundary constraints.
- This paper conducts a simulation verification of the proposed learning algorithm on several communication indexes. Simulation results demonstrate the superiority of the learning algorithm over two benchmark algorithms.

C. Organization

The remainder of this paper is organized as follows: Section II builds system models. Based on the models, a 3-D continuous movement control problem is formulated in Section III. In Section IV, a DRL-based learning algorithm is developed to mitigate the formulated problem. Section V presents the simulation results, and this paper is concluded in Section VI.

II. System Models

This paper considers a downlink communication coverage scenario as depicted in Fig. 1. In this figure, a set of terrestrial heterogeneous users (denoted by $\mathcal{U}$) is uniformly distributed in a given geographical region of $L \times L$ m^2. Heterogeneous users are defined as users with different QoS requirements, which are measured by users' required data transfer rates. The user set $\mathcal{U}$ is further partitioned into $|\mathcal{K}|$ groups (or subsets) with $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_k, \ldots, \mathcal{U}_{|\mathcal{K}|}\}$ according to users' QoS requirements, where $\mathcal{U}_k = \{u_1^{(k)}, \ldots, u_{i_k}^{(k)}, \ldots u_{|\mathcal{U}_k|}^{(k)}\}$, $k \in \mathcal{K}$, and $|\cdot|$ represents the cardinality of a set. All users in $\mathcal{U}_k$ have the same QoS requirement. For the symbol $u_{i_k}^{(k)}$, the subscript i_k denotes the i_k-th user, and the superscript (k) indicates that the user is in the subset $\mathcal{U}_k$. For a terrestrial user $u_{i_k}^{(k)}$, $i_k = 1, \ldots, |\mathcal{U}_k|$, $k \in \mathcal{K}$, this paper denotes its location by $\boldsymbol{\omega}_{i_k}^{(k)} = [x_{i_k}^{(k)}, y_{i_k}^{(k)}]$.

Besides, a group of drone-cells that can be aware of their locations is deployed to deliver telecommunication services to ground users. The time domain is discretized, and a communication task of drone-cell networks will continue for a sequence of time-steps, i.e., $t = \{1, \ldots, T\}$. Owing to the limited number of drone-cells and restricted coverage ranges, drone-cells need to adjust their altitudes and horizontal locations during executing the communication task such that more users can receive some services such as the voice and the video from them. This paper also discusses the continuous movement control of drone-cells in discrete time-steps, and the terms "time-step" and "decision epoch" are interchangeable. The following subsections present the detailed system models including the drone-cell mobility model and the channel propagation model.

A. Drone-Cell Mobility Model

In this paper, drone-cells take off from random initial locations at the beginning of the communication task and then fly or hover in a restricted 3-D airspace. The time-varying location of the j-th drone-cell, $j \in \mathcal{J}$, at the time-step t is denoted by $\boldsymbol{\omega}_{t,j} = [x_{t,j}, y_{t,j}, h_{t,j}]$, and the distance between the j-th and j'-th drone-cells is defined as $d_{t,jj'}(\boldsymbol{\omega}_{t,j}, \boldsymbol{\omega}_{t,j'}) = \|\boldsymbol{\omega}_{t,j} - \boldsymbol{\omega}_{t,j'}\|_2$.

This paper involves a widely used drone-cell mobility model, which can take the following form [27]

$$x_{t+1,j} = x_{t,j} + m_{t,j} sin(\theta_{t,j}) cos(\phi_{t,j}) \quad (1)$$

$$y_{t+1,j} = y_{t,j} + m_{t,j} sin(\theta_{t,j}) sin(\phi_{t,j}) \quad (2)$$

$$h_{t+1,j} = h_{t,j} + m_{t,j} cos(\theta_{t,j}) \quad (3)$$

where $m_{t,j} \in [0, m_{max}]$ represents the moving distance of the j-th drone-cell at time-step t with m_{max} denoting the maximum moving distance in a time-step, $\theta_{t,j} \in [0, \pi]$ denotes the pitch angle of the j-th drone-cell, and $\phi_{t,j} \in (0, 2\pi]$ represents the yaw angle of the drone-cell.

In addition, this paper considers the following scheduling strategies for both flying and data transmission of drone-cells: each drone-cell is assigned a time-step to fly and transmit data. A drone-cell makes a decision on movements at the very beginning of a time-step; once arriving at a desired location, it will hover there and start transmitting data to ground users.

B. Propagation Model

In this subsection, both an AtA propagation model and an air-to-ground (AtG) propagation model will be introduced.

1) Air-to-air Propagation Model: Without considering the blockage of airplane bodies, drone-cells may have a line-of-sight (LoS) view towards each other. This paper thus leverages the free-space path loss (FSPL) in decibels to calculate the AtA link propagation loss, which can take the following form [28]

$$L_{t,jj'}(\boldsymbol{\omega}_{t,j}, \boldsymbol{\omega}_{t,j'}) = 20 \log_{10} \left(\frac{4\pi f_c}{c} d_{t,jj'}(\boldsymbol{\omega}_{t,j}, \boldsymbol{\omega}_{t,j'}) \right) \quad (4)$$

where f_c (in Hz) is the carrier frequency, c (in m/s) is the speed of light.

Besides, the j-th and j'-th drone-cells are deemed to be connected only if the link propagation loss $L_{t,jj'}(\boldsymbol{\omega}_{t,j}, \boldsymbol{\omega}_{t,j'})$ is lower than a pre-defined propagation loss threshold γ_1.

2) Air-to-ground Propagation Model: For AtG communications, each user will typically have a LoS view towards a drone-cell with a specific probability. The LoS probability relies on the environment, density and height of buildings, locations of both the user and the drone-cell as well as the elevation angle between the user and the drone-cell. An expression of the LoS probability can be calculated as [28]

$$P_r\left(L_{t,i_kj}^{LoS}\right) = \frac{1}{1+\lambda_1 exp(-\lambda_2(\theta_{t,i_kj}^{(k)}-\lambda_1))} \tag{5}$$

where the constants λ_1 and λ_2 depend on the type of the environment (e.g., rural, urban, and dense urban), and $\theta_{t,i_kj}^{(k)} = \frac{180}{\pi}\arctan(\frac{h_{t,j}}{r_{t,i_kj}^{(k)}})$ is the elevation angle between the user $u_{i_k}^{(k)}$ and the j-th drone-cell at time-step t with $r_{t,i_kj}^{(k)} = \sqrt{\left(x_{i_k}^{(k)}-x_{t,j}\right)^2+\left(y_{i_k}^{(k)}-y_{t,j}\right)^2}$ denoting a horizontal distance between them.

Without considering the heights of users as well as the antenna heights of both users and drone-cells, the AtG link propagation loss can be expressed as [28]

$$L_{t,i_kj}^{(k)}(h_{t,j}, r_{t,i_kj}^{(k)}) = 20\log_{10}\left(\sqrt{h_{t,j}^2+\left(r_{t,i_kj}^{(k)}\right)^2}\right) + EP_r\left(L_{t,i_kj}^{LoS}\right) + F \tag{6}$$

where $E = \eta_{LoS} - \eta_{NLoS}$ and $F = 20\log_{10}(\frac{4\pi f_c}{c}) + \eta_{NLoS}$ are two constants with η_{LoS} (in dB) and η_{NLoS} (in dB) representing propagation losses corresponding to the LoS and NLoS connections, respectively.

III. Problem Formulation and Analysis

A. Problem Formulation

For a drone-cell j, $j \in \mathcal{J}$, if a user $u_{i_k}^{(k)}$, $i_k = 1, \ldots, |\mathcal{U}_k|$, $k \in \mathcal{K}$, can be served/covered by it, the user's receiving data rate should not be less than its QoS requirement. Formally, the condition that $u_{i_k}^{(k)}$ is served can be written as

$$C_{t,i_kj}^{(k)} \geq C_k^{th} \tag{7}$$

where $C_{t,i_kj}^{(k)}$ denotes the receiving data rate from the j-th drone-cell of $u_{i_k}^{(k)}$ at time-step t.

It is noteworthy that $u_{i_k}^{(k)}$ may be served by multiple drone-cells. This paper, however, does not identify the number and the index of drone-cells that can serve $u_{i_k}^{(k)}$. $u_{i_k}^{(k)}$ is said to be served, if $\exists j \in \mathcal{J}$ such that the condition (7) is satisfied. Further, this paper lets $b_{t,i_k,j}^{(k)} \in \{0,1\}$ indicate whether $u_{i_k}^{(k)}$ can be served by the j-th drone-cell at time-step t or not, and then, $b_{t,i_k}^{(k)} = \max_{j\in\mathcal{J}}\{b_{t,i_kj}^{(k)}\}$ indicates whether $u_{i_k}^{(k)}$ can be served or not. $b_{t,i_k}^{(k)} = 1$, if $u_{i_k}^{(k)}$ is served; otherwise, $b_{t,i_k}^{(k)} = 0$. Then, (7) can be rewritten as

$$C_{t,i_kj}^{(k)} \geq C_k^{th} + M(b_{t,i_kj}^{(k)} - 1) \tag{8}$$

where M is a constant that is slightly larger than the largest C_k^{th}.

Next, according to the Shannon equation, the receiving data rate of the user $u_{i_k}^{(k)}$ is calculated by

$$C_{t,i_kj}^{(k)} = B_w\log_2(1+SNR_{t,i_kj}^{(k)}) \tag{9}$$

where the constant B_w denotes the transmission bandwidth allocated to the user-drone-cell association $(u_{i_k}^{(k)}, j)$. $SNR_{t,i_kj}^{(k)}$ denotes the received signal-to-noise ratio of $u_{i_k}^{(k)}$ that can take this form

$$SNR_{t,i_kj}^{(k)} = 10^{\frac{P_D - L_{t,i_kj}^{(k)}(h_{t,j}, r_{t,i_kj}^{(k)}) - P_N}{10}} \tag{10}$$

where P_D (in dBm) is a drone-cell transmission power, P_N (in dBm) represents the noise power. The interference scenario is not investigated here.

As the network connectivity is crucial for drone-cell networks, this paper considers the network connectivity as an essential constraint of the formulated problem and constructs a graph G for the networks. The graph is composed of a vertex set $V(G) = \mathcal{J}$ and an edge set $E(G)$ consisting of connected AtA links. In addition, G is connected if it has a j, j'-path whenever $j, j' \in V(G)$ (otherwise, G is disconnected) (Definition 1.2.6 [29]).

Next, the objective function of the movement control problem will be identified. Intuitively, the goal of deploying drone-cell networks is to maximize the sum of required date rates of users. It, however, may result in unfair coverage of users. Owing to limited coverage ranges achievable by drone-cell networks, a subset of users (e.g., users with high required data rates) may be covered during most of or even all the time-steps, while others are left uncovered. Therefore, the fair communication coverage is needed. This paper leverages the Jain's fairness index, denoted by f_t, to measure the coverage fairness of drone-cell networks within the first t time-steps

$$f_t = \frac{\left(\sum_{k\in\mathcal{K}}\sum_{i_k=1}^{|\mathcal{U}_k|}\bar{b}_{t,i_k}^{(k)}\right)^2}{|\mathcal{U}|\sum_{k\in\mathcal{K}}\sum_{i_k=1}^{|\mathcal{U}_k|}\left(\bar{b}_{t,i_k}^{(k)}\right)^2} \tag{11}$$

where $\bar{b}_{t,i_k}^{(k)}$ represents the coverage ratio of the user $u_{i_k}^{(k)}$ with

$$\bar{b}_{t,i_k}^{(k)} = \frac{\sum_{\varepsilon=1}^{t} b_{\varepsilon,i_k}^{(k)}}{T} \tag{12}$$

It can be quickly known that $f_t \in [0,1]$, and the larger the fairness index is, the fairer the coverage will be.

Controlling drone-cells to fly around will help improve the coverage fairness of drone-cell networks. Flying around, however, may consume more energy compared with hovering at current locations. There is thus a trade-off between the level of energy consumption and the coverage fairness of drone-cell networks. Besides, this paper uses (13) to calculate the energy consumption of the j-th drone-cell in a time-step

$$e_{t,j}(m_{t,j}) = (m_{t,j}/m_{max})(e_r - 1)e_h + e_h \tag{13}$$

where $m_{t,j}$ represents the moving distance of the j-th drone-cell during time-step t, m_{max} is the maximum moving distance in a time-step, e_r represents the drone-cells' energy consumption ratio of flying with the maximum distance to hovering at current locations, e_h is defined as the energy consumption during hovering, which includes energy consumed by controller units and data processing [23]. Moreover, this paper lets $m_{0,j} = 0$, and then $e_{0,j}(m_{0,j}) = e_h$ denotes the initial energy consumption of the j-th drone-cell. Although a specific energy consumption model is invoked, this paper is not restricted to this one. The model is used to reflect that flying around will consume more energy than hovering at the current location for a drone-cell.

In short, this paper aims at maximizing the energy efficiency of the communication coverage of multiple drone-cells. The energy-efficient coverage here indicates fair and efficient communication coverage with less energy consumption. Using (5)–(13), the problem of the 3-D continuous movement control of multiple drone-cells can be formulated as

$$\underset{\substack{x_{t,j},y_{t,j},h_{t,j},\\ m_{t-1,j},\{b_{t,i_k}^{(k)}\}}}{\text{maximize}} \quad \frac{\sum_{t=1}^{T}\sum_{k\in\mathcal{K}}\sum_{i_k=1}^{|\mathcal{U}_k|} f_t b_{t,i_k}^{(k)} C_k^{th}}{\sum_{t=1}^{T}\sum_{j\in J} e_{t-1,j}(m_{t-1,j})} \tag{14}$$

$$\text{s.t.} \quad B_w \log_2\left(1+10^{\frac{P_D-L_{t,i_kj}^{(k)}\left(h_{t,j},r_{t,i_kj}^{(k)}\right)-P_N}{10}}\right) \geq C_k^{th}+$$

$$M_1(b_{t,i_kj}^{(k)}-1), \forall j\in\mathcal{J}, i_k=1,\ldots,|\mathcal{U}_k|, t=1,\ldots,T \tag{15}$$

$$b_{t,i_k}^{(k)} = \max_{j\in\mathcal{J}}\{b_{t,i_kj}^{(k)}\} \tag{16}$$

$$The\ graph\ G\ is\ connected \tag{17}$$

$$x_l \leq x_{t,j} \leq x_u, \quad \forall j\in\mathcal{J}, t=1,\ldots,T \tag{18}$$

$$y_l \leq y_{t,j} \leq y_u, \quad \forall j\in\mathcal{J}, t=1,\ldots,T \tag{19}$$

$$h_l \leq h_{t,j} \leq h_u, \quad \forall j\in\mathcal{J}, t=1,\ldots,T \tag{20}$$

$$0 \leq m_{t-1,j} \leq m_{max}, \quad \forall j\in\mathcal{J}, t=1,\ldots,T \tag{21}$$

$$b_{t,i_kj}^{(k)} \in \{0,1\},\ \forall k\in\mathcal{K}, i_k=1,\ldots,|\mathcal{U}_k|, t=1,\ldots,T \tag{22}$$

where subscripts l and u denote the minimum and maximum values of $x_{t,j}$, $y_{t,j}$, and $h_{t,j}$. The constraint (15) means that at least a drone-cell should serve the user $u_{i_k}^{(k)}$. (17) is used to preserve the connectivity of drone-cell networks, (18)–(20) limit the moving range of drone-cells to a particular 3-D airspace and (21) limits moving distances of drone-cells during a time-step.

It is noteworthy that there are logarithmic-exponential-terms and continuous-binary-variables in the formulated problem; thus, it is a mixed-integer-nonlinear programming problem that is well-known NP-hard [30]. Moreover, during a period of T time-steps, the goal of this problem is to control multiple drone-cells to move to: 1) maximize the sum of users' QoS requirements; 2) maximize the fairness index of coverage; 3) minimize the energy consumption of drone-cell networks; 4) maintain the network connectivity in each time-step; 5) ensure that drone-cells are flying in the bounded 3-D airspace; 6) strike a trade-off between the coverage fairness and the level of energy consumption. Therefore, it will be highly challenging to achieve the optimal solution of such a control problem.

In theory, some heuristic approaches may be able to relieve this problem, it would be, however, impractical to explore them. As the control time of drone-cells can be long enough (i.e., a large T) in real situations, using heuristic approaches entails unbearable high computational complexity. Further, heuristic approaches need repeated runnings under different types of traffic distribution. Resorting to DRL, which has been demonstrated as an efficient way of handling complex control problems in continuous and high dimensional state spaces [26], to mitigate this challenging control problem may be a promising proposal. This paper therefore proposes to design a DRL-based learning algorithm elaborately to alleviate the problem. Before presenting the procedure of the proposed algorithm, this paper performs fundamental analysis on the optimal drone-cell deployment altitude.

B. Optimal Deployment Altitude

Owing to the objective of maximizing the sum of users' QoS requirements, a large drone-cell coverage radius may bring great benefits whatever the type of a user distribution is. Motivated by this characteristic, the subsection studies the largest coverage radius of a drone-cell and the corresponding optimal drone-cell deployment altitude.

To this aim, this paper assumes that there is only one group of users $\mathcal{U}_k$ in the considered region and introduces an auxiliary variable $D_k > 0$. D_k represents the coverage radius of the j-th drone-cell, $j\in\mathcal{J}$. Meanwhile, this paper lets h_k denote the drone-cell deployment altitude corresponding to D_k. Then, the condition that a user $u_{i_k}^{(k)}$ can be covered by the j-th drone-cell can take another form

$$D_k \geq r_{t,i_kj}^{(k)} \tag{23}$$

Combining (7), (9), (10) with (23), the functional relationship between h_k and D_k can be expressed as

$$h_k^2 + D_k^2 \leq g(h_k, D_k) \tag{24}$$

where $g(h_k,D_k) = 10^{\frac{P_D-P_N-EP_r(h_k,D_k)-F-10\log_{10}[2^{C_k^{th}/B_w}-1]}{10}}$, and $P_r(h_k,D_k)$ is computed by (5).

Additionally, the maximum D_k corresponding to a given h_k satisfies the following condition.

Theorem 1: Given a drone-cell deployment altitude h_k, the inequality $h_k^2 + D_k^2 \leq g(h_k,D_k)$ has infinite solutions, if the following condition is satisfied [31]

$$h_k^2 \leq g(h_k, 0) \tag{25}$$

where $g(h_k,0)$ is determined by C_k^{th} and system parameters listed in [31, Table 2].

Moreover, the solution is the maxima that can be achieved by solving (26).

$$h_k{}^2 + D_k^2 - g(h_k, D_k) = 0 \tag{26}$$

This paper next introduces a critical variable $\alpha_k \in [0,\infty)$, which is defined as the ratio of the altitude h_k to D_k, i.e.,

$$\alpha_k = \frac{h_k}{D_k} \tag{27}$$

With the definition of α_k, (26) can be rewritten as [31]

$$D_k^2 = \Gamma(\alpha_k) \tag{28}$$

where,

$$\Gamma(\alpha_k) = \frac{10^{\frac{P_D-P_N-EP_r(\alpha_k)-F-10\log_{10}\left[2^{C_k^{th}/B_w}-1\right]}{10}}}{1+\alpha_k^2} \tag{29}$$

and $P_r(\alpha_k)$ is calculated by (5).

Meanwhile, based on both (27) and (28), the following conjecture presents the lower and upper bounds of the optimal deployment altitude of a drone-cell.

Conjecture 1: For a set of users $\mathcal{U} = \{\mathcal{U}_1,\ldots,\mathcal{U}_k,\ldots,\mathcal{U}_{|\mathcal{K}|}\}$, $k\in\mathcal{K}$, the optimal deployment altitude (denoted by h_{opt}) of a drone-cell that maximizes the coverage region for $\mathcal{U}$ meets

the following inequality for the propagation environment whose parameters are listed in Subsection V-A,

$$h_l \leq h_{opt} \leq \min\left\{\max_{k\in\mathcal{K}}\left\{\sqrt{g(h_k,0)}\right\}, \max_{k\in\mathcal{K}}\{h_k^\star\}, h_u\right\} \tag{30}$$

where $h_k^\star$ represents the optimal deployment altitude of the drone-cell for covering $\mathcal{U}_k$ with

$$h_k^\star = \alpha_k^\star\sqrt{\Gamma(\alpha_k^\star)} \tag{31}$$

Proof: See Appendix A. ■

Furthermore, as $\Gamma(\alpha_k)$ has only one maxima at $\alpha_k^\star$ [1], this paper adopts the hill-climbing method [31], [32] to yield $\alpha_k^\star$, and thus $h_k^\star$ can be obtained using (31).

IV. Proposed Algorithm: E^{2}CMC

This section is aimed at presenting the proposed E^{2}CMC algorithm, which is based on both the DRL method and the above theoretical results. Since this paper investigates a continuous movement control problem, an emerging deep deterministic policy gradient (DDPG) method [25] that can find continuous policies with competitive performance is selected as the starting point of the design. In E^{2}CMC, instead of making a decision based on the collected local state information such as the location of the drone-cell, the energy consumption, and the coverage situation, each drone-cell will transmit the local information to a nearby cloud for centralized computing. Besides, an agent is responsible for learning the actions of drone-cells by interacting with a stochastic environment in discrete time-steps and disseminating the learned actions to each drone-cell to perform.

A. Energy-Efficiency and Continuous-Movement-Control Algorithm

This paper considers a standard reinforcement learning setup that is composed of an agent interacting with an environment E in discrete time epochs. At each time epoch t the agent observes an observation $\boldsymbol{s}_t \in \mathcal{S}$, executes a real-valued action $\boldsymbol{a}_t \in \mathcal{A}$, and receives a scalar reward r_t as well as results in the state changes in the stochastic environment, where $\mathcal{S}$ and $\mathcal{A}$ represent the observation space and the action space, respectively. The transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ is used to record the history of changes in the environment. The stochastic environment assumed to satisfy the Markov property is modelled as a finite Markov decision process with the transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ as a Markov state. In addition, this paper defines the agent's behavior as a deterministic policy, μ, which maps the received observation $\boldsymbol{s}_t$ to an action $\boldsymbol{a}_t$, i.e., $\mu : \mathcal{S} \rightarrow \mathcal{A}$.

This paper next starts to describe the procedure of the E^{2}CMC from the elaborate design of the observation, the action, and the reward.

1) Observation $\boldsymbol{s}_t$: At each time-step t, $t = 1, 2, \ldots, T$, as the agent will formulate the control policy according to the collected information of current coverage situation and energy consumption, this paper utilizes the following three components to construct the observation $\boldsymbol{s}_t$.

- $b_{t,i_k}^{(k)} \in \{0, 1\}$ The coverage indicator of the user $u_{i_k}^{(k)}$ at the epoch t. $b_{t,i_k}^{(k)} = 1$, if $u_{i_k}^{(k)}$ is served; otherwise, $b_{t,i_k}^{(k)} = 0$.
- $e_{t-1,j}(m_{t-1,j})$ The energy consumption of the j-th drone-cell at the epoch $t-1$. It can be computed by (13).
- $f_t \in [0, 1]$ The coverage fairness of the drone-cell networks at the epoch t that can be calculated by (11).

Specifically, $\boldsymbol{s}_t$ can be expressed as $\boldsymbol{s}_t = [b_{t,1}^{(1)}, \ldots, b_{t,|\mathcal{U}_{|\mathcal{K}|}|}^{(|\mathcal{K}|)}, e_{t-1,1}(m_{t-1,1}), \ldots, e_{t-1,|\mathcal{J}|}(m_{t-1,|\mathcal{J}|}), f_t]$, where $|\mathcal{U}_{|\mathcal{K}|}|$ denotes the number of users in the $|\mathcal{K}|$-th group. The cardinality of the $\boldsymbol{s}_t$ is $|\mathcal{U}| + |\mathcal{J}| + 1$. Moreover, in an $\boldsymbol{s}_t$, both $b_{t,i_k}^{(k)}$ and f_t reflect the coverage situation of terrestrial users; and the set $\{e_{t-1,1}(m_{t-1,1}), \ldots, e_{t-1,|\mathcal{J}|}(m_{t-1,|\mathcal{J}|})\}$ characterizes the total energy consumption of the drone-cell networks during a time-step.

2) Action $\boldsymbol{a}_t$: From the perspective of the operating mechanism of the DRL method, an action characterized by the movement of drone-cells in this paper will result in the state change in the environment. Exactly, the update of locations of drone-cells changes the coverage situation, and the moving distances of drone-cells determine the level of the energy consumption of drone-cells. Therefore, considering the drone-cell mobility model, this paper uses the Euler angles and moving distances of drone-cells to represent the action $\boldsymbol{a}_t$, $t = 1, 2, \ldots, T$, as described below.

- $m_{t,j} \in [0, m_{max}]$ The moving distance of the j-th drone-cell, $j \in \mathcal{J}$. $m_{t,j} = 0$, if the j-th drone-cell is commanded to hover at the current location at the very beginning of the decision epoch t.
- $\theta_{t,j} \in [-90^\circ, 90^\circ]$ The pitch angle received by the j-th drone-cell from the agent at the very beginning of the decision epoch t.
- $\phi_{t,j} \in (-180^\circ, 180^\circ]$ The yaw angle of the j-th drone-cell at the decision epoch t.

Specifically, $\boldsymbol{a}_t$ can be expressed as $\boldsymbol{a}_t = [m_{t,1}, \ldots, m_{t,|\mathcal{J}|}, \theta_{t,1}, \ldots, \theta_{t,|\mathcal{J}|}, \phi_{t,1}, \ldots, \phi_{t,|\mathcal{J}|}]$. The cardinality of $\boldsymbol{a}_t$ is $3 \times |\mathcal{J}|$. After receiving an $\boldsymbol{a}_t$ from the agent, the j-th drone-cell, $j \in \mathcal{J}$, will hover at the current location or fly to a new location. Since the $m_{t,j}$ and drone-cells' Euler angles are continuous, the agent's behavior is continuous in the system. Thus, this paper studies the problem of the continuous movement control of multiple drone-cells.

3) Reward r_t*:* The design of the reward function is crucial for any reinforcement learning method. Besides, it is challenging to define the reward function in this paper because there are three considerably different objectives (i.e., energy consumption, the sum of users' QoS requirements, and coverage fairness) and two distinct constraints (i.e., network connectivity and airspace boundary) in the formulated control problem. To relieve this issue, this paper discusses the objectives independently from the constraints and designs a novel reward function considering the three objectives jointly. On the other hand, two reward penalty mechanisms are designed to criticize the actions resulting in the violations of the boundary and the connectivity constraints, respectively. Specifically, this paper designs the reward function r_t as the following form

$$r_t = \frac{f_t \sum_{k\in\mathcal{K}} \sum_{i_k=1}^{|\mathcal{U}_k|} b_{t,i_k}^{(k)} C_k^{th}}{\sum_{j\in\mathcal{J}} e_{t-1,j}(m_{t-1,j})} \tag{32}$$

where $\sum_{k\in\mathcal{K}} \sum_{i_k=1}^{|\mathcal{U}_k|} b_{t,i_k}^{(k)} C_k^{th}$ represents the achieved sum of users' QoS requirements at the time-step t. The fairness f_t is

leveraged to discount the achieved gain such that the issue of the unfair coverage may be relieved. Meanwhile, the denominator $\sum_{j\in\mathcal{J}} e_{t-1,j}(m_{t-1,j})$ denotes the energy consumption of the drone-cell networks during time-step $t-1$. This reward function can be referred to as the energy efficiency per time-step (the achieved gain per unit of energy consumption during a time-step). Through maximizing the reward function r_t in a sequence of decision epochs, E^{2}CMC may help the drone-cell networks achieve the goal of the high energy-efficient communication coverage.

Next, this paper presents the two reward penalty mechanisms, boundary-margin mechanism, and disconnection-penalty mechanism.

a. Boundary-margin mechanism: During executing the communication task, some drone-cells may fly outside of the pre-determined boundary of the 3-D airspace. Meanwhile, the drone-cell resource may be wasted if they are deployed close to the boundary of the considered region. To alleviate these issues, this paper designs the boundary-margin mechanism and defines the 'boundary margin' as a narrowed 3-D airspace. This mechanism will penalize the achieved reward r_t if some drone-cells fly out of the 'boundary margin'. Formally, the mechanism can be expressed as

$$r_t = r_t - \left(\alpha\left(\left(l_x^+\right)^2 + \left(l_y^+\right)^2 + \left(l_h^+\right)^2\right) + \beta\right)|r_t| \tag{33}$$

where $\alpha = 1/(12500|\mathcal{J}|)$ and $\beta = 3/|\mathcal{J}| - 9/(25|\mathcal{J}|)$ are two penalty coefficients that adjust the strength of the penalty, and

$$l_x^+ = \max\left(\left|x_{t,j} - \frac{x_l + x_u}{2}\right| - \frac{\upsilon_1}{2}(x_u - x_l), 0\right) \tag{34}$$

$$l_y^+ = \max\left(\left|y_{t,j} - \frac{y_l + y_u}{2}\right| - \frac{\upsilon_1}{2}(y_u - y_l), 0\right) \tag{35}$$

$$l_h^+ = \max\left(\left|h_{t,j} - \frac{h_l + h_{opt}}{2}\right| - \frac{\upsilon_1}{2}(h_{opt} - h_l), 0\right) \tag{36}$$

with υ_1 being a penalty coefficient.

b. Disconnection-penalty mechanism: Drone-cell networks may become disconnected owing to the movement of drone-cells. To relieve this issue, this paper develops the disconnection-penalty mechanism to penalize the obtained reward if the drone-cell networks is disconnected. Specifically, the mechanism can take the following form

$$r_t = r_t - \upsilon_2|r_t| \tag{37}$$

where υ_2 is a large penalty coefficient.

Additionally, the Dijkstra algorithm [33] is used to measure the connectivity of the drone-cell networks. Specifically, if there is not a transmission path between the j-th and j'-th drone-cells, $j, j' \in \mathcal{J}$, at the decision epoch t, the drone-cell networks are considered to be disconnected at this epoch.

Combing the starting point of the design (i.e., the DDPG method) with the above analysis, the algorithm representation of implementing the continuous movement control of multiple drone-cells can be depicted in Algorithm I. Besides, the following subsection describes how this algorithm works.

Algorithm 1: Energy-Efficiency and Continuous-Movement-Control, E^{2}CMC.

Randomly initialize critic evaluation network $Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}^Q)$ and actor evaluation network $\mu(\boldsymbol{s}|\boldsymbol{\theta}^\mu)$ with weights $\boldsymbol{\theta}^Q$ and $\boldsymbol{\theta}^\mu$
Initialize critic target network $Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}^{Q'})$ and actor target network $\mu(\boldsymbol{s}|\boldsymbol{\theta}^{\mu'})$ with weights $\boldsymbol{\theta}^{Q'} = \boldsymbol{\theta}^Q$, $\boldsymbol{\theta}^{\mu'} = \boldsymbol{\theta}^\mu$
Initialize replay buffer R and drone-cell network topology graph G
for each episode in $\{1, 2, \ldots, H\}$ **do**
  Initialize the environment and a random process $\mathcal{N}$ for action-exploration, receive an initial observation $\boldsymbol{s}_1$
  for each decision epoch $t = 1, 2, \ldots, T$ **do**
    Select an action $\boldsymbol{a}_t = \mu(\boldsymbol{s}_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and the exploration noise
    Execute the action $\boldsymbol{a}_t$, observe a reward r_t and a new observation $\boldsymbol{s}_{t+1}$
    for each drone-cell $j = 1, 2, \ldots, |\mathcal{J}|$ **do**
      if boundary-margin mechanism is activated **then**
        Penalize r_t using (33)
        if the j-th drone-cell flies outside of the pre-determined airspace **then**
          Cancel the movement of the j-th drone-cell and update $\boldsymbol{s}_{t+1}$ accordingly
        end if
      end if
      if $L_{t,jj'}(\boldsymbol{\omega}_{t,j}, \boldsymbol{\omega}_{t,j'}) \leq \gamma_1$, $j' \in \mathcal{J}\backslash j$ **then**
        Add the AtA link (j, j') to G
      end if
    end for
    Run the Dijkstra algorithm on G to check whether the drone-cell networks are disconnected or not
    if the drone-cell networks are disconnected **then**
      Penalize r_t using (37)
      Cancel the movement of all drone-cells and update $\boldsymbol{s}_{t+1}$ accordingly
    end if
    Store the transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ in R
    Sample a random minibatch of M transitions $(\boldsymbol{s}_m, \boldsymbol{a}_m, r_m, \boldsymbol{s}_{m+1})$ from R
    Set $y_m = r_m + \gamma Q'(\boldsymbol{s}_{m+1}, \mu'(\boldsymbol{s}_{m+1}|\boldsymbol{\theta}^{\mu'})|\boldsymbol{\theta}^{Q'})$
    Update $\boldsymbol{\theta}^Q$ by minimizing the loss:
    $L(\boldsymbol{\theta}^Q) = \frac{1}{M}\sum_{m=1}^{M}\left(y_m - Q(\boldsymbol{s}_m, \boldsymbol{a}_m)|\boldsymbol{\theta}^Q\right)^2$
    Update $\boldsymbol{\theta}^\mu$ using the sampled gradient:
    $\nabla_{\boldsymbol{\theta}^\mu}\mu|_{\boldsymbol{s}_m} = \frac{1}{M}\sum_{m=1}^{M}(\nabla_{\boldsymbol{a}}Q(\boldsymbol{s}, \boldsymbol{a}|\boldsymbol{\theta}^Q)|_{\boldsymbol{s}=\boldsymbol{s}_m, \boldsymbol{a}=\mu(\boldsymbol{s}_m|\boldsymbol{\theta}^\mu)} \times \nabla_{\boldsymbol{\theta}^\mu}\mu(\boldsymbol{s}|\boldsymbol{\theta}^\mu)|_{\boldsymbol{s}=\boldsymbol{s}_m})$
    Update the target networks:
    $\boldsymbol{\theta}^{Q'} = \tau\boldsymbol{\theta}^Q + (1-\tau)\boldsymbol{\theta}^{Q'}$
    $\boldsymbol{\theta}^{\mu'} = \tau\boldsymbol{\theta}^\mu + (1-\tau)\boldsymbol{\theta}^{\mu'}$
  end for
end for

B. Flow of the Proposed Algorithm

Lines 1–3 initialize the critic and actor networks, a replay buffer and a network topology graph. Explicitly, four neural networks including the critic evaluation and target networks as well as the actor evaluation and target networks are constructed. All the actor and critic networks are two-layer fully-connected feed-forward neural networks, and network parameters are initialized by a Xavier initialization scheme. The network parameters are, however, updated in entirely different ways. For instance, the parameters of the critic evaluation network are updated using line 29, while the critic target network's parameters are updated using line 33. Such design style is developed to stabilize the learning of neural networks [25]. In this paper, there are 400 and 300 neurons in the first and the second hidden layers of the constructed neural networks, respectively. These hidden layers are all activated by the ReLU function [34], and L_2 weight delay [35] is introduced to relieve the issue of overfitting. Furthermore, this paper utilizes the $tanh(\cdot)$ activation function in the output layer of the actor network to bound the outputted action.

Lines 7–24 describe the process of the action-exploration. At each epoch t, an action $\boldsymbol{a}_t$ is generated by adding an Ornstein-Uhlenbeck noise process to a current actor policy $\mu(\boldsymbol{s}_t|\boldsymbol{\theta}^{\mu})$ to enhance the exploration efficiency. In this paper, the mean and standard deviation of the noise process are zero and 0.3, respectively. Besides, the problem (14)–(22) is constrained by both the airspace boundary and the network connectivity. The obtained action $\boldsymbol{a}_t$, however, may lead to the violation(s) of airspace boundary and/or network connectivity constraints. To mitigate this issue, this paper develops two penalty mechanisms, i.e., boundary-margin mechanism (lines 9–15) and disconnection-penalty mechanism (lines 16–24).

Lines 25–34 depict the learning procedure of neural networks. In this procedure, the experience replay technique (refer to lines 25–31) is adopted to break the correlation among samples and decrease the variance of parameters of neural networks. Meanwhile, a target Q-network is designed to break the correlation between the Q-network and its target and help avoid the oscillation (lines 27–29). The ADAM is used as the optimization method [36], where the learning rates of the actor and the critic networks are set to be 0.0001 and 0.001, respectively. Besides, a 'soft' target update technique is applied to slow down the change of the target value to further improve the stability of the learning (lines 33–34) [25].

V. Simulation Results

Extensive simulations are conducted in this section to validate the effectiveness of the proposed learning algorithm.

A. Comparison Algorithms and Parameter Setting

To our best knowledge, there are no existing works to be compared; thus, this paper implements two benchmark algorithms, Random algorithm, and Greedy algorithm, for comparison on Python.

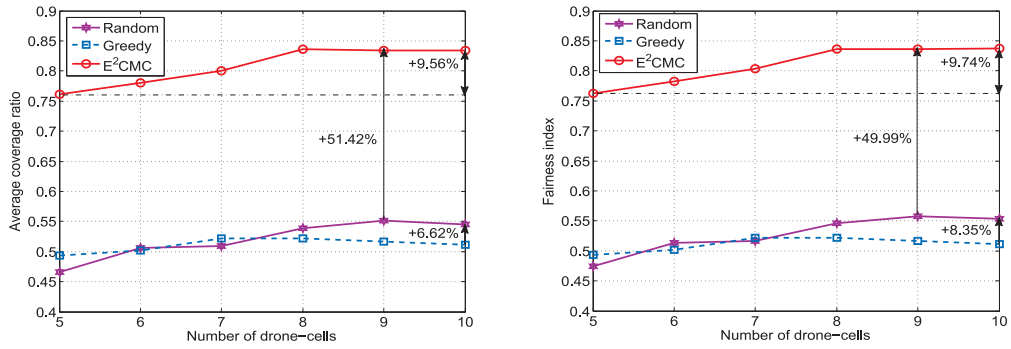
- *Random algorithm*: At each decision epoch t, it generates the action a_t for each drone-cell through selecting randomly a pitch angle from $[-90^{\circ}, 90^{\circ}]$ and a yaw angle from $(-180^{\circ}, 180^{\circ}]$ as well as a flight distance from $[0, m_{max}]$. Meanwhile, if one/multiple drone-cell(s) fly out of the 'boundary-margin' or the drone-cell networks become disconnected after taking this action, all drone-cells then refuse this action and hover at the current locations.
- *Greedy algorithm*: At each decision epoch t, it chooses sequentially a pitch angle from $\{-90^{\circ}, -45^{\circ}, \ldots, 90^{\circ}\}$, a yaw angle from $\{-135^{\circ}, -90^{\circ}, \ldots, 180^{\circ}\}$ and a flight distance from $\{0, m_{max}/2, m_{max}\}$ that can maximize the reward r_t for each drone-cell. Meanwhile, just like the Random algorithm, the selected action resulting in the violation of boundary or connectivity constraints will be declined.
- *E^{2}CMC*: The proposed E^{2}CMC is implemented on Tensorflow 1.4 and Python, and a server configured with the Ubuntu 16.04.3 LTS and four NVIDIA TITAN Xp GPUs is used.

The parameter setting of the simulation is summarized as the following: Set the size of the considered region be 2500×2500 m^2, i.e., $L = 2500$ m, $(x_l, y_l) = (0, 0)$, and $(x_u, y_u) = (2500, 2500)$. A total of 100 stationary terrestrial users are uniformly distributed in this given region. These users are classified into three categories according to their required data rates, i.e., $|\mathcal{K}| = 3$. The required data rates C_k^{th}, $k = \{1, 2, 3\}$, are set to be 1Mb/s, 2Mb/s, and 4Mb/s, respectively. For each user, its required data rate is selected from the set $\{C_1^{th}, C_2^{th}, C_3^{th}\}$ with a specific probability. The probability that a user is in $\mathcal{U}_k$ is denoted by ρ_k with $\rho_1 = 0.6$, $\rho_2 = 0.3$, and $\rho_3 = 0.1$. Specifically, this paper generates the required rate for each user by playing a turntable game [31]. In this game, each user is only allowed to turn the turntable for once. If the needle on the turntable stays in the interval $(\sum_{t=1}^{k-1}\rho_t, \sum_{t=1}^{k}\rho_t]$, let the required rate of the user be C_k^{th} and add it to the subset $\mathcal{U}_k$. Besides, let $M_1 = 10$, $T = 1000$, $H = 500$, penalty coefficients $\upsilon_1 = 0.8$, and $\upsilon_2 = 50$, altitudes $h_l = 100$ m, and $h_u = 800$ m, $m_{max} = 5$ m, $e_h = 1$ unit, $M = 512$, $\tau = 0.001$, and $\gamma = 0.99$. More radio frequency propagation parameters are listed as follows: environment parameters $\lambda_1 = 4.88$, $\lambda_2 = 0.43$, $\eta_{LoS} = 0.1$, and $\eta_{NLoS} = 21$; drone-cell transmission power $P_D = 24$ dBm, noise power $P_N = -75$ dBm, transmission bandwidth $B_w = 2.5$ MHz, frequency $f_c = 2.5$ GHz, speed of light $c = 3 \times 10^8$ m/s, and propagation loss threshold $\gamma_1 = 102$.

B. Performance Evaluation

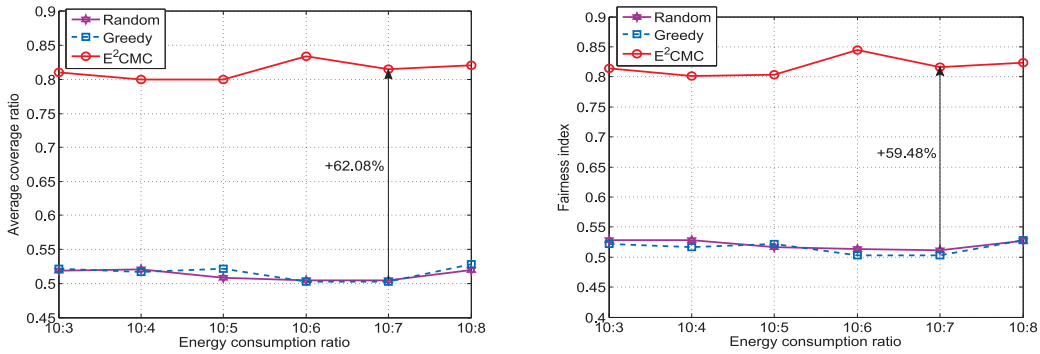
To verify that the proposed algorithm is not only effective for specific traffic distribution, this paper generates twenty-six different types of traffic distribution. Further, the first one is considered as the training data set, and the remaining twenty-five types of traffic distribution are taken as test data sets. In each type of traffic distribution, all one hundred users are distributed uniformly in the considered geographical region, and the QoS requirement of each user is determined by playing the above turntable game.

E^{2}CMC is trained on the training data set for 500 episodes, each of which includes 1000 decision epochs. After training, E^{2}CMC is tested on each test data set for T epochs. The two benchmark algorithms are also performed on all the test data sets.



(a) Average coverage ratio vs. number of drone-cells (b) Fairness index vs. number of drone-cells

Fig. 2. Both average coverage ratio and fairness index vs. number of drone-cells with $e_r = 10:5$.



(a) Average coverage ratio vs. energy consumption ratio (b) Fairness index vs. energy consumption ratio

Fig. 3. Both average coverage ratio and fairness index vs. energy consumption ratio with $|\mathcal{J}| = 7$.

For each comparison algorithm, twenty-five simulation results can be obtained, and the final result is their average value.

Besides, the following metrics are leveraged for the algorithm performance evaluation.

- *Average coverage ratio ($\hat{B}_T$):* The average user coverage ratio at the end of the test epochs. It can take the following form.

$$\hat{B}_T = \frac{\sum_{k\in\mathcal{K}}\sum_{i_k=1}^{|\mathcal{U}_k|}\bar{b}_{T,i_k}^{(k)}}{|\mathcal{U}|} \tag{38}$$

- *Fairness index (f_T):* The Jain's fairness index at the end of the test epochs, which can be calculated by (11).
- *Normalized average energy consumption ($\bar{E}_T$):* The average energy consumption of a drone-cell after T test epochs, which is normalized by E_{max}

$$\bar{E}_T = \frac{\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\sum_{t=1}^{T}e_{t-1,j}(m_{t-1,j})}{E_{max}} \tag{39}$$

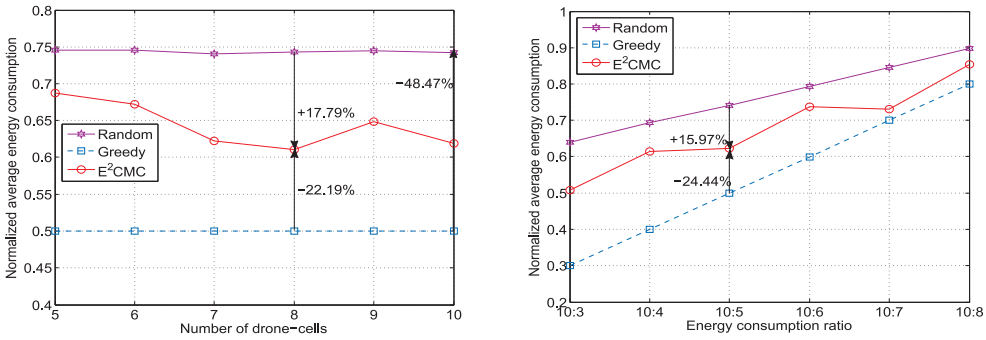
  where, $E_{max} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\sum_{t=1}^{T}e_{t-1,j}(m_{max})$ represents the total maximum possible energy consumption of a drone-cell during T test epochs.
- *Energy efficiency (G_T):* The ratio of the total achieved gain to the total energy consumption of the drone-cell networks during T test epochs, which is computed by (14).

For simplicity, this paper utilizes the term 'system revenue' to represent both the average coverage ratio and the fairness index.

This paper plots first the impact of both the number of drone-cells and their energy consumption ratio e_r on the system revenue. Figs. 2, 3 illustrate the system revenue obtained by all comparison algorithms.

From these figures, the following observations may be obtained:

- The proposed E^{2}CMC algorithm can outperform consistently both benchmark algorithms on the obtained system revenue regardless of the parameter setting of both the number of drone-cells and the energy consumption ratio. For instance, E^{2}CMC improves the average coverage ratio by 51.42% compared with the best benchmark algorithm, when the number of drone-cells is nine and the energy consumption ratio is $10:5$ (refer to Fig. 2(a)). Besides, when $|\mathcal{J}|$ is fixed at seven and the energy consumption ratio is $10:7$ (in Fig. 3(b)), the fairness index obtained by the Random algorithm is only 0.5114. Under the same parameter setting, the fairness index of the E^{2}CMC reaches 0.8157, which indicates a 59.48% improvement. In contrast to the Greedy algorithm, however, the Random algorithm improves the average coverage ratio and the fairness index by 6.62% and 8.35%, respectively, when the number of drone-cells is ten.
- As is shown in Fig. 2, more drone-cells can bring greater system revenue. This may be because more drone-cells can enlarge coverage ranges of the drone-cell networks. For E^{2}CMC, however, when the number of drone-cells is higher than eight, the system cannot obtain greater revenue. This situation may be incurred by the *boundary-margin mechanism*. This mechanism attempts to restrict



(a) Normalized average energy consumption vs. number of drone-cells with $e_r = 10 : 5$

(b) Normalized average energy consumption vs. energy consumption ratio with $|\mathcal{J}| = 7$

Fig. 4. Achieved normalized average energy consumption of all comparison algorithms.

drone-cells to a narrowed 3-D airspace, and thus some users close to the boundary of the considered region cannot be well served. Besides, when the number of drone-cells is doubled, i.e., increased from five to ten, the system revenue gained by E^{2}CMC is improved by less than 10%. This situation may lead to the suspicion of improving the system revenue by increasing the number of drone-cells because the CAPEX and the OPEX[1] of ten drone-cells are more expensive than those of five drone-cells.

- In addition, given a fixed number of drone-cells, it is interesting to find that varying the value of the energy consumption ratio does not change the system revenue of all comparison algorithms significantly. Although a small ratio may encourage drone-cells to move more frequently, it does not lead to a great system revenue improvement. Specifically, the improvement in the system revenue achieved by E^{2}CMC is less than 5%.

This paper further plots the impact of the number of drone-cells and the energy consumption ratio on the energy consumed by drone-cells. The obtained normalized average energy consumption of all algorithms is collected in Fig. 4.

This paper achieves the following observations from Fig. 4:

- The Greedy algorithm consumes the least energy, and the Random algorithm consumes the most energy. Moreover, the Random algorithm needs to consume almost half (48.47%, accurately) as much energy as the Greedy algorithm. Due to the frequent movement of drone-cells, E^{2}CMC consumes more energy than the Greedy algorithm. For example, in Fig. 4(a), the normalized average energy consumption of the E^{2}CMC increases by 22.19% in comparison to the Greedy algorithm when the number of drone-cells is eight. E^{2}CMC, however, obtains a 17.79% energy-consumption reduction compared with the 0.7431 obtained by the Random algorithm.
- In addition, when the number of drone-cells is no more than eight, the energy consumed by the E^{2}CMC decreases with the increasing number of drone-cells. The reason is that more drone-cells may enlarge coverage ranges of the drone-cell networks and then shorten the flight distances. The energy consumption, however, cannot be further reduced when there are more drone-cells (e.g., nine and ten).
- The normalized average energy consumption obtained by both the Random and the Greedy algorithms increases linearly with the decrease of the energy consumption ratio. For E^{2}CMC, a small energy consumption ratio will also result in a large $\bar{E}_T$. This is because a small energy consumption ratio encourages drone-cells to fly around more frequently.
- Furthermore, it is interesting to find that the Greedy algorithm tends to suggest drone-cells to hover at current locations owing to the goal of maximizing the reward function of the drone-cell networks at each time-step.

[1]Capital expenditure (CAPEX), for example, buying machinery and other equipment. Operational expenditure (OPEX), for example, maintenance and repair of machinery.

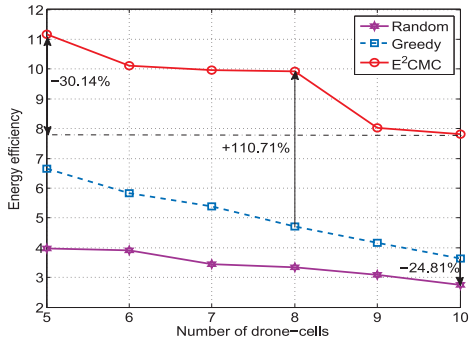


Fig. 5. Energy efficiency vs. number of drone-cells with $e_r = 10 : 5$.

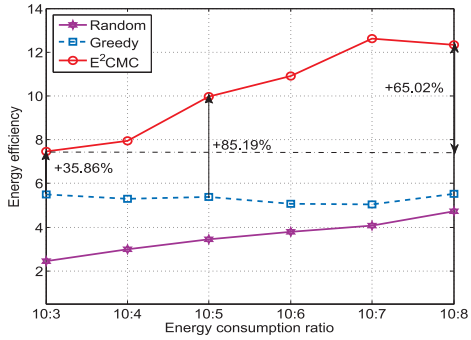


Fig. 6. Energy efficiency vs. energy consumption ratio with $|\mathcal{J}| = 7$.

Moreover, Figs. 5, 6 plot the relationship of the energy efficiency achieved by the comparison algorithms and depict the effect of the number of drone-cells and the energy consumption ratio on the energy efficiency, respectively.

From these figures, the following observations may be achieved:

- The proposed E^{2}CMC outperforms both benchmark algorithms concerning the energy efficiency. Specifically, E^{2}CMC achieves a significant improvement of 110.71% over the Greedy algorithm with the parameter setting of $|\mathcal{J}| = 8$ and $e_r = 10:5$. It further improves the energy efficiency by 196.17% in comparison to the 3.348 obtained by the Random algorithm. Meanwhile, due to the most energy consumption yet the unremarkable revenue improvement, the Random algorithm is defeated by the Greedy algorithm regarding the energy efficiency. For instance, when the number of drone-cells is ten (refer to Fig. 5), the Random algorithm obtains a 24.81% reduction on the energy efficiency compared with the Greedy algorithm.
- Since more drone-cells will consume more energy while a significant gain cannot be achieved, the energy efficiency of all comparison algorithms decreases with the increase of the number of drone-cells. Moreover, for E^{2}CMC, the achieved energy efficiency will reduce by 30.14%, if the number of drone-cells is doubled.
- Besides, given a fixed number of drone-cells, all comparison algorithms except for the Greedy algorithm may obtain greater energy efficiency over a smaller energy consumption ratio. For example, in Fig. 6, when $e_r = 10:8$, E^{2}CMC achieves an energy efficiency improvement of 65.02% in compare with a $10:3$ energy consumption ratio setting. The main reason is that a small e_r reduces the level of energy consumption and thus magnifies the achieved energy efficiency.
- Moreover, as is shown in Fig. 6, E^{2}CMC improves the energy efficiency by 85.19% and 190.39% over the Greedy algorithm and the Random algorithm, respectively, with $e_r = 10:5$ and $|\mathcal{J}| = 7$.

Furthermore, according to Figs. 2–6, this paper may obtain the following observations:

- Given a fixed energy consumption ratio, the Random algorithm consumes more energy yet achieves a system revenue improvement of less than 7% in comparison to the Greedy algorithm. Besides, as expected, the Random algorithm is defeated by the Greedy algorithm concerning the energy efficiency. Therefore, drone-cells would rather hover at current locations than fly around unintentionally.
- For the proposed E^{2}CMC, it improves the achieved system revenue by less than 10% through doubling the number of drone-cells, which requires higher acquisition and maintenance cost and needs to consume more energy. There is thus a trade-off between improving system performance and reducing operational cost for the drone-cell networks. It is out of the scope of this paper to identify the number of drone-cells that can maximize the revenue as well as minimize the operational cost. This paper, however, may provide suggestions for telecom operators when determining the appropriate number of drone-cells according to specific performance requirements (e.g., coverage ratio and energy efficiency).

Summarily, a carefully designed DRL scheme may help decrease both the CAPEX and the OPEX of telecom operators and may be leveraged to improve the utilization efficiency of the telecommunications infrastructure. This paper, however, has only begun to explore the utilization of DRL technology in communications, and much attention is worthy of being paid to interdisciplinary approaches from communication networks and the AI/ML research community.

VI. Conclusion

This paper formulated a problem of 3-D continuous movement control of multiple drone-cells for providing communication coverage for terrestrial heterogeneous users. The goal of the problem was to optimize the energy-efficient communication coverage of the drone-cell networks while maintaining the network connectivity. A DRL-based energy-efficiency and continuous-movement-control algorithm (E^{2}CMC) was developed to alleviate this problem. First, E^{2}CMC designed an energy efficiency reward function with the joint consideration of the energy consumption, the sum of QoS requirements of users, and the coverage fairness. Second, E^{2}CMC learned to identify the locations of drone-cells by interacting with an environment in discrete time-steps. Meanwhile, disconnected drone-cell networks would be penalized by reducing the value of the reward function drastically. Simulation results showed that E^{2}CMC could achieve at least 51.42%, 49.99%, and 35.86% improvements concerning the average coverage ratio, the fairness, and the energy efficiency compared to two benchmark algorithms. This paper studied the continuous movement control of drone-cells in a centralized fashion. Extending this work to decentralized control will be a topic of future study.

Acknowledgment

The authors would like to thank Prof. Chi Harold Liu who helped them a lot on the construction of neural networks and the mitigation of many other questions.

References

[1] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[2] Y. Gu, M. Zhou, F. Shengli, and Y. Wan, "Airborne WiFi networks through directional antennae: An experimental study," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2015, pp. 1314–1319.

[3] X. B. Cao, P. Yang, M. Alzenad, X. Xi, D. P. Wu, and H. Yanikomeroglu, "Airborne communication networks: A survey," *IEEE J. Select. Areas Commun.*, vol. 36, no. 9, pp. 1907–1926, Sep. 2018.

[4] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anticoordination game based partially overlapping channels assignment in combined UAV and D2D-based networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.

[5] D. Takaishi, Y. Kawamoto, H. Nishiyama, N. Kato, F. Ono, and R. Miura, "Virtual cell based resource allocation for efficient frequency utilization in unmanned aircraft systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3495–3504, Apr. 2018.

[6] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 2714–2741, Oct.–Dec. 2018.

[7] X. Xi, X. Cao, P. Yang, Z. Xiao, and O. D. Wu, "Efficient and fair network selection for integrated cellular and drone-cell networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 923–937, Jan. 2019.

[8] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, and D. Wu, "Proactive drone-cell deployment: Overload relief for a cellular network under flash crowd traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2877–2892, Oct. 2017.

[9] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[10] 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. 2017.

[11] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2233–2246, Apr. 2018.

[12] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[13] J. Chen, O. Esrafilian, D. Gesbert, and U. Mitra, "Efficient algorithms for air-to-ground channel reconstruction in UAV-aided communications," in *Proc. Int. Workshop Wireless Netw. Control Unmanned Auton. Veh.*, 2017, pp. 1–6.

[14] 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 J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[15] U. Challita, W. Saady, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellularconnected UAVs," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.

[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 Trans. Ind. Inform.*, vol. 14, no. 4, pp. 1666–1676, Apr. 2018.

[17] R. Ghanavi, E. Kalantariy, M. Sabbaghian, H. Yanikomerogluz, and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2018, pp. 1–6.

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

[19] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 38–41, Feb. 2018.

[20] Y. Zeng, X. Xu, and R. Zhang, "Trajectory optimization for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2233–2246, Apr. 2018.

[21] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[22] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.

[23] B. Uragun, "Energy efficiency for unmanned aerial vehicles," in *Proc. Int. Conf. Mach. Learn. Appl. Workshops*, 2011, pp. 316–320.

[24] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, 2013.

[25] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *Comput. Sci.*, vol. 8, no. 6, 2015, Art. no. A187.

[26] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–541, 2015.

[27] B. Etkin, *Dynamics of Atmospheric Flight*. New York, NY, USA: Wiley, 2005.

[28] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[29] D. B. West, *Introduction to Graph Theory.* Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[30] J. T. Linderoth and M. W. P. Savelsbergh, *A Computational Study of Search Strategies for Mixed Integer Programming*. Catonsville, MD, USA: INFORMS, 1999.

[31] P. Yang, X. B. Cao, X. Xi, Z. Y. Xiao, and D. P. Wu, "3-D drone-cell deployment for congestion mitigation in cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9867–9881, Oct. 2018.

[32] S. J. Russell and P. Norvig, "Artificial intelligence: A modern approach," *Appl. Mech. Mater.*, vol. 263, no. 5, pp. 2829–2833, 2010.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[35] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–41.

Authors' photographs and biographies not available at the time of publication.