Delay Minimization in Multi-UAV Assisted Wireless Networks: A Reinforcement Learning Approach

Chenyu Wu[1(✉)], Xuemai Gu[1,3], and Shuo Shi[1,2]

[1] School of Electronic and Information Engineering, Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
{wuchenyu, guxuemai, crcss}@hit.edu.cn
[2] Network Communication Research Centre, Peng Cheng Laboratory, Shenzhen 518052, Guangdong, China
[3] International Innovation Institute of HIT in Huizhou, Huizhou 516000, Guangdong, China

Abstract. Unmanned Aerial Vehicles (UAVs) assisted communications are promising technology for meeting the demand of unprecedented demands for wireless services. In this paper, we propose a novel framework for delay minimization driven deployment of multiple UAVs. The problem of joint non-convex three dimensional (3D) deployment for minimizing average delay is formulated and solved by Deep Q network (DQN), which is a reinforcement learning based algorithm. Firstly, we obtain the cell partition by K-means algorithm. Then, we find the optimal 3D position for each UAV in each cluster to provide low delay service. Finally, when users are roaming, the UAVs are still able to track the real-time users. Numerical results show that the proposed DQN-based delay algorithm shows a fast convergence after a small number of iterations. Additionally, the proposed deployment algorithm outperforms several benchmarks in terms of average delay.

Keywords: Unmanned Aerial Vehicles · Delay minimization · Deployment · Reinforcement learning

1 Introduction

With the pullulating and landing deployment of wireless skills, as well as the birth of killer apps, users' pursuit of service quality is higher, and the existing skills cannot meet the needs of tomorrow communication. People are looking for ever-increasing turnkey solutions, including exploration on higher airways, better encoding and transmission skills, and a large-scale connection that incorporates multiple networks. UAVs are thought to be killers of auxiliary communication [1]. Rather than the orthodox ground wireless-skills, UAV assisted communication has the preponderances of high movability, low expenditure, especially better LOS positioning ability. Therefore, the employment of UAVs to acquire high rate is expected to play a pivotal role.

S. Shi et al. (Eds.): AICON 2020, LNICST 356, pp. 249–259, 2021.
https://doi.org/10.1007/978-3-030-69066-3_22

The deployment of UAVs as locomotive BS to assist surficial infrastructure has been deemed as an prominent technology for handling cellular network discharging and offloading in hot spots, such as prompt renew after infrastructure damage, important recreational gathering, high level meeting and natural disasters. Under the premise of dependency and adjustability, these criticisms can be solved universally by UAVs. In this paper, UAV, as a relay node, not only improves the total throughput of the system, but also provides reliable connection for remote users without perfect direct link [2, 3]. In addition, UAVs can also be used to assist the Internet of things network to ensure large-scale connectivity and low latency [4, 5].

In reference [6], the air ground model is given and the altitude problem of UAV is well solved. In this paper, we can seek out the emblematic parameters of the air-ground model and bring inspiration to the deployment of UAV. Recent strategy is not only about maximizing coverage, but also on algorithms that try to cover the largest number of users. In order to improve the system coverage, the deployment layout of single UAV and multi UAV has been studied [7, 8]. The layout algorithm can be synchronous or asynchronous [9]. However, due to the high computational complicacy, especially in dealing with dynamic circumstances such as roaming users and ever-changing channel conditions, the three-dimensional layout of multiple UAVs is defiant. RL reduces the complexity of convex optimization by means of iteration and interaction, and has great effect in shaping planning and multi-objective and constraint problems. In reference [10], the author proposes a deep reinforcement learning algorithm for UAV control, which considers fairness, energy consumption and connectivity. The object is to seek a tactic to control the movement mode of each UAV. However, the three-dimensional layout of multiple UAVs is ignored.

The indicators of user relationship are various, such as delay, flux, the number of users meeting the threshold, file hit rate and so on. However, they can not be separated from each other. It can be summarized by the quality of user service, which is nothing more than choosing the best service target and service mode according to the user's needs, location, channel information, etc. In this paper, we consider a scenario that multiple UAVs serve ground users for delay minimization. Firstly, we obtain the user association to reduce the impact of user interleaving by K-means algorithm. Then, we find the optimal 3D placement bourn for each UAV to minimize the sum delay of the users. Finally, when users are moving, the UAVs are still able to track the real-time users and provide low latency service.

2 System Model and Problem Formulation

We consider the downlink of UAV assisted ground users in urban as shown in Fig. 1. Multiple UAVs act as BSs in the air to carry files of users' interest and serve the users in the target region. There exists K UAVs serving users set $\mathcal{U}$ with the total number of U. Users are separated in K clusters. We assume that there are U^k users in the k-th cluster and the specific user u_i^k is the i-th user in class k, $i \in \{1, 2, \ldots, U^k\}$.Each user

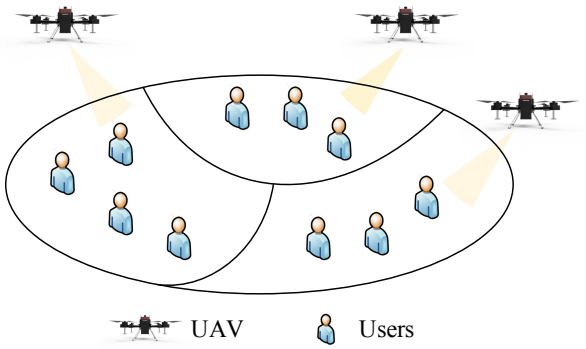

Fig. 1. System model for delay minimization. Each UAV serves one cluster.

belongs to the coverage of only one UAV. The users in the city are very dense, so the delay will be greatly increased if the time division method is used to serve the users in turn. So that we assume at the same time, UAV can serve multiple users and adopt multiple access based on frequency.

2.1 Transmission Model

The user's location is random. In some literatures, the user is modeled as a PPP or a uniform distribution around the center of a circle by using statistical methods. This is not the most important because our algorithm is scalable and can be applied to different user distributions without changing the model. Users can move continuously during the service period of UAVs. Due to the high mobility, It is difficult to study the control of UAV in large time scale. Thus, we use technique called time discretization, which is to divide the time T into equal slots with length δ and index t. The UAV flies at an appropriate fixed altitude H and the maximum speed is $V_{\max}$. The 2D position of a specific user u_i^k at each time slot is $[x_i^k(t), y_i^k(t)]^T$, and the 3D coordinate of the k-th UAV is $[x^k(t), y^k(t), h^k(t)]^T$. Compared with the whole mission cycle, the moving distance of UAV in a short time is relatively small, which can be approximately static in the initial or terminal position. Making use of time discretization, the instantaneous distance between UAV k and the specific user u_i^k can be fixed in a small time:

$$d_i^k(t) = \sqrt{[h^k(t)]^2 + [x^k(t) - x_i^k(t)]^2 + [y^k(t) - y_i^k(t)]^2} \tag{1}$$

The downlink between UAV and user can be regarded as the line of sight dominated air-to-ground channel. Occasionally, in the environment of high-rise buildings and high-rise buildings in the city, it may be connected by high-rise buildings and high-rise buildings. We adopt the probabilistic Los channel model and consider occlusion. The probability of LoS can be expressed as

$$P_{LOS}(\theta_i^k) = \frac{1}{1 + a\exp(-b(\theta_i^k - a))} \tag{2}$$

where $\theta_i^k = \sin^{-1}\frac{H}{d_i^k}$ is the elevation angle, a and b are parameters according to the change of environment conditions. The probability of NLOS with user u_i^k's feedback is given by $P_{NLOS}(\theta_i^k) = 1 - P_{LOS}(\theta_i^k)$. Intuitively, P_{LOS} increases as the UAVs fly directly on the target and approximate one when θ_i^k becomes large enough.

Then, the path loss for user u_i^k is

$$PL_{LOS} = (\frac{4\pi f_c}{c})^{-2}(d)^{-\alpha}10^{\eta_{LOS}} \tag{3}$$

$$PL_{NLOS} = (\frac{4\pi f_c}{c})^{-2}(d)^{-\alpha}10^{\eta_{NLOS}} \tag{4}$$

$$PL{=}P_{LOS} \times PL_{LOS} + P_{NLOS} \times PL_{NLOS} \tag{5}$$

where f_c stands for the carrier frequency.c is speed constant of light. α is the exponent indicating loss, η_{LOS} and η_{NLOS} are the attenuation factors according to the existence of LoS and NLoS.

Many assume that the number of spectrum is variable and can be continuously allocated. This assumption has certain truth, but it is very difficult to practice.

We discuss simple scheme of FDMA and assume the bandwidth B is allocated to users belonging to the same sphere in an equal manner, thus the spectrum for U^k user is $B_i^k = B/U^k$. The maximum power carried is equally distributed similarly with each user u_i^k having $P_i^k = P/U^k$. By estimating from the receiver along with the SNR, the service rate for user u_i^k with bit/s in unit of measurement:

$$r_i^k(t) = B_i^k \log_2(1 + \frac{p_i^k}{PL_{d_i^k(t)}\sigma^2}) \tag{6}$$

where $\sigma^2 = B_i^k N_0$ is the AWGN var, N_0 is power spectral density for general noise.

2.2 Problem Formulation

We consider the UAV hovering over the user with variable altitude when the user is stationary or continuously moving. The bandwidth and transmission power of each UAV are uniformly allocated to each user. Therefore, the optimization problem is simplified as a region segmentation problem, and its formula is as follows

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{h}} \quad \delta_{\text{sum}} = \sum_{k=1}^{K} \sum_{i=1}^{U_k} \sum_{t=1}^{T} s/r_i^k(t) \tag{7}$$

Where s is the standardized file size of content to transfer. It can be seen from Eq. (7) that the altitude and horizontal coordinates of UAV have influence on the delay of users. This is because both the distance and the Los probability are related to the altitude of the UAV. Increasing the flight altitude of UAV will lead to greater path loss, but also will obtain higher Los probability.

Due to the combination of user association and optimal location search, exhaustive search algorithm is a direct method to obtain the optimal result. However, this is computationally complex. Therefore, a low complexity 3D deployment algorithm based on DQN is proposed. In addition, when the optimal position of UAV is fixed, the acquisition of dynamic tracking is also very important due to the nonconvex problem of sum delay.

3 Deployment and Movement of UAVs Using DQN

In the actual scene, the user roams continuously, which leads to the increase of delay. Traditional methods tend to predict with high complexity solutions. Therefore, RL is employed to tail after users.

Reinforcement learning (RL) is a forceful tool to solve decision-making problems. In recent years, reinforcement learning has reached the limit of human cognition in many aspects in the field of game, and can be used as an auxiliary means to solve optimization problems. In this part, we first introduce some basic knowledges of RL, and then we propose an algorithm to minimize average delay based on Deep Q-Network (DQN).

Reinforcement learning contains basic elements including: environment which is preset and can not be changed, agent which is trained, state which stands for the status of robots that are being trained, action that the robots take using their habits, and reward gained after each step. In RL, agents interact with the atmosphere in a way of action and reward. The process is a MDP $\mathcal{M} = <\mathcal{S}, \mathcal{A}, \mathcal{R}, \Pr(s_{t+1}|s_t, a) >$, where $\mathcal{S}$ is state set, $\mathcal{A}$ is action set, $\mathcal{R}$ is the set of reward. When taking action a_t, there is a transition probability of $\Pr(s_{t+1}|s_t, a_t)$ from state s_t to s_{t+1}.

The aim of RL is to conceive a policy that maximizes the total rewards observed during the episodes. Value is a common term in RL which stand for the set of policies that evaluate the long-term reward of the policy. Q-learning is a basic value-based algorithm of RL, which maintains a Q-table to record and minimize the discounted cumulative reward which is

$$\min C = \mathbb{E}^{\pi}(\sum_{t=1}^{\infty} \gamma_d^{t-1} r(s_{t+1}|s_t, a_t)) \tag{8}$$

The integral is from the present moment to the infinite future which is the final state of other restrictions, where $\pi = \arg\max_{a_t \in \mathbb{A}} Q(s_t, a_t)$ is the policy to choose action, γ_d is the discount factor. Allowing agents to choose actions according to the maximum value cannot achieve good results, because it will destroy the balance between exploration and optimization. An excellent tutorial tip to explore the environment is the ε-greedy policy. The Q table which is also known as value function is updated by

$$Q_{t+1}(s_t, a_t) = (1 - \alpha) Q_t(s_t, a_t) + \alpha(r_t + \gamma_d \min_{a'} Q_t(s_{t+1}, a')) \tag{9}$$

where α is learning rate. However, since this algorithm holds a big form for each action-state pair, it is intolerable for large scale problem. For example, when we play chess, the state is the current chess piece, the actions set is to drop a piece randomly in the blank position of the current chessboard. Considering the size of the board, the total action space is equal to the length times the width, which still does not include some actions that can and cannot be done according to the rules of the game. So we can see that maintaining a table consumes huge resources and sometimes can't solve problems. Neural network is a good substitute, because large-scale network can approximate any nonlinear function to meet our needs (Table 1).

Table 1. Simulation Parameters.

Parameter	Description	Value
U	Number of users	80
K	Number of clusters and UAVs	4
P	Total transmit power	0.1 W
δ	Time slot length	1 s
N_0	Noise power spectral density	−174 dBm/Hz
f_c	Carrier frequency	2 GHz
B	Total bandwidth of each UAV	1 MHz
a, b	Environmental parameters	10.39,0.05(urban)
η_{LOS},η_{NLOS}	Additional path loss for LOS, NLOS	1,20 (dB)

Moreover, the control of UAVs is a continuous control problem. Many works regard the UAV as a static base station, which plays the same role as the small base station and studies the optimal solution in statistical sense. I don't think this assumption is very reasonable because the UAV is a mobile agent, so it is necessary to give full play to its mobility advantages to carry out path planning. DQN take example by neural networks to reckon the value. The NN target is minimizing the loss:

$$L(\boldsymbol{\theta}^{Q}) = \mathbb{E}\left[r_t + \gamma_d Q'(s_{t+1}, \pi(s_{t+1})|\boldsymbol{\theta}^{Q'}) - Q(s_t, a_t|\boldsymbol{\theta}^{Q})\right]^2 \tag{10}$$

where the first part $y_t = r_t + \gamma_d Q'(s_{t+1}, \pi(s_{t+1})|\boldsymbol{\theta}^{Q'})$ is the target value to reach, $\boldsymbol{\theta}^{Q}$ is the weight of NN. The network back propagates and updates $\boldsymbol{\theta}^{Q}$ using gradient decent with derivative $\nabla L(\boldsymbol{\theta}^{Q})$.

In addition, DQN adopts two kinds of technologies: experience playback and target network to reduce the influence of data correlation. The correlation between data can not make neural network learn useful knowledge well. With the introduction of stochastic gradient, this problem is solved well. Experience playback is selecting batch size B_s experience from buffer in a random manner. In addition, DQN tries identical target network Q' as the NN of the original one. The weight of the original NN is to update the parameters in a delay manner of the target network.

We explain the important elements:

1) Agent: Agent is one of the core of RL. At present, the mainstream research direction has been extended to multi-agent learning. It considers the multi-objective cooperation or competition game, which itself is a difficult problem to see the optimal solution, because there are still many challenges. In contrast, single agent has been proved to be a good solution to some simple decision-making problems, and the distributed single-agent solution is also a choice. Because the interference is not considered, there is no cooperation and competition between UAVs. The training agent: each UAV
2) State: During each training step t (also time index for epochs of the whole progress), $\boldsymbol{s}_t = [x^k(t), y^k(t), h^k(t), x_1^k(t), x_2^k(t), \ldots, x_{U_k}^k(t)]$. The state is the 3D site of UAV and 2D coordinates for ground customers.
3) Action: In order to provide continuous control of the UAVs, we denote the operating direction as the action. Also, the agent can suspend in a still manner. There are 6 directions available: left, forward, up, backward, right, as well as down.
4) Reward: Reward is a common term, which is suitable for our goal related. In the actual scene, users can't give us immediate feedback because the user's experience is delayed, but in the simulation and training, we can choose experience data according to the parameters. Data generation is one of the benefits of RL, which does not rely on training data sets, but through experience. However, it also brings about the problem of data utilization. The reward of epoch t is defined as:

$$r(t) = \sum_{i=1}^{U_k} s / r_i^k(t) \tag{11}$$

which is the current sum delay.

Using DQN, the UAVs can quickly and efficiently find the location and moving direction to obtain the minimum delay. The progress of the whole algorithm is shown in Algorithm 1.

Algorithm 1 Deep Reinforcement Learning for Delay Minimization

Initialize value Q with random weights $\boldsymbol{\theta}$
Initialize target value Q' with same parameters $\boldsymbol{\theta}^{-}=\boldsymbol{\theta}$
Initialize N capacity memory $\mathcal{D}$, and buffer size is set as B_s.
for episode $m = 1, 2, ..., M$ **do**:
  Initialize initial state s_1 and prepare training environment
  if random $< \varepsilon$:
    choose action $a_t = \arg\min_a Q(s_t, a; \boldsymbol{\theta})$
  else:
    randomly choose an action
  execute a_t and observe s_{t+1}, r_t
  store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  sample random mini-batch (s_j, a_j, r_j, s_{j+1}) with size B_s from $\mathcal{D}$
  Set value for target: $y_j = r_j + \gamma_d \min Q'(s_{j+1}, a \mid \boldsymbol{\theta}^{-})$
  Loss function $L(\boldsymbol{\theta}^{Q}) = \sum_{j=1}^{B_s} [y_i - Q(s_j, a_j \mid \boldsymbol{\theta})]^2$
  update $\boldsymbol{\theta}$ using $\nabla L(\boldsymbol{\theta}^{Q})$ using GD
  Set $\boldsymbol{\theta}^{-}=\boldsymbol{\theta}$ in a repetitive manner for every B_{up} steps
end for

4 Results and Analysis

First of all, we introduce the simulation platform and the specific super parameters in machine learning. We conduct our experiments in Tensorflow with version 1.0. It is a time-consuming and laborious process to find the suitable hyperparameters. In order to simplify, we only give the best hyperparameters which represent the best performance of the system, but we don't talk about testing and selecting the parameters

The main hyperparameters are as follows: rate for learning α is 0.001, memory size $\mathcal{D}$ is 5000, factor of discount as 0.9, repetitive update B_{up} = 300 steps. The neural network adopts two-layer fully connected architecture, because in lots of experiments, the single-layer network can not fit the model well, and the three-layer network also has the problem of over fitting and slow training speed. Our algorithm is also compared with the traditional exhaustive-based algorithm and random deployment algorithm in terms of convergence and system performance.

Figure 2 depicts the instantaneous delay for each ground user. We draw the three-dimensional equipotential diagram of all users' delay. Intuitively, it is a concave surface. The cluster has 20 users and is served by one UAV. It can be observed that with the increase of the distance between UAV and ground user, the delay of the ground user also increases.

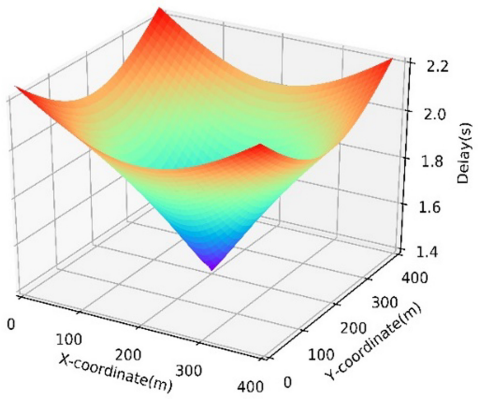

Fig. 2. Minimum delay versus user location

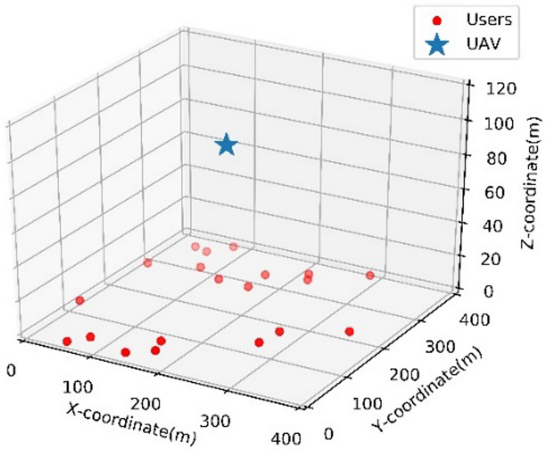

Fig. 3. Optimal UAV location versus user distribution

Figure 3 draws the optimal 3D map of UAV from the position of the first fleet and ground user. The blue star represents the best location for the UAV. The horizontal coordinates and height of UAVs are determined by the user's position, because they affect the line of sight probability and path loss.

Figure 4 depicts the relationship between total delay and training times. It can be seen that the UAV can perform its actions in an iterative manner and learn from the mistakes, thus improving the and delay. It can be seen that the algorithm converges after a certain number of iterations. Despite the initial position of the UAV, it was integrated after about 5000 sets. The process of convergence is not a straight line or has been declining, but a fluctuating decline, which is one of the basic common sense of RL, because RL constantly carries out trial and error and iteration to complete learning

from experience. At every moment, it is possible to learn new knowledge to optimize the objective function, so the loss value of neural network will be increased. Finally, the method to judge the convergence is that the overall performance tends to be stable, and the variance is small.

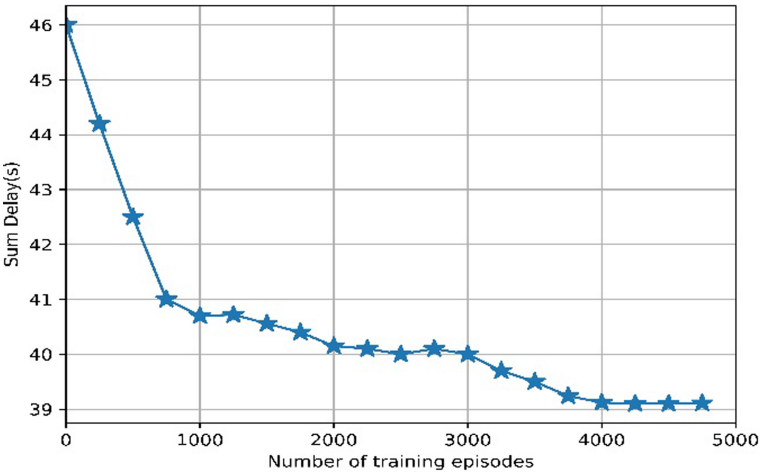

Fig. 4. Convergence of DQN

Figure 5 shows the total latency compared to random deployment. When the user remains static, the optimal location keeps an optimal sum delay. The green line represents the delay optimal solution when the user does not move. It is obtained by brute force exhaustion. The calculation amount of this exhaustion is very large. When the user moves, we list the best position at all times to carry out path planning, and a more

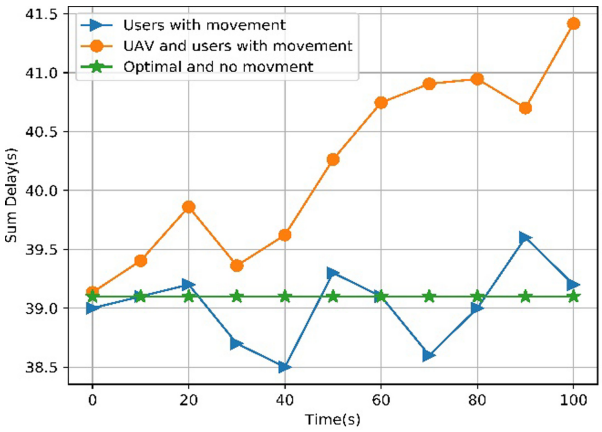

Fig. 5. Sum delay when users are moving and static

intelligent algorithm is needed to give the optimal decision in real time. The blue line represents the real-time total flux under the path planning given by our algorithm when the user moves. It can be seen that the value fluctuates around the initial value. This is because the user's movement is random and sometimes tends to gather near the UAV, so the total flux must be relatively large. When the ground user moves according to the random walk model, the UAV should move along the user's direction. Otherwise, as the user leaves the initial point, and the delay increases. As can be seen from the figure, our algorithm is suitable for dynamic environment.

References

1. Zeng, Y., Zhang, R., Lim, T.J.: Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Commun. Mag. **54**(5), 36–42 (2016)
2. Zeng, Y., et al.: Throughput maximization for UAV-enabled mobile relaying systems. IEEE Trans. Commun. **64**(12), 4983–4996 (2016)
3. Zhang, S., Zhang, H., He, Q., Bian, K., Song, L.: Joint trajectory and power optimization for UAV relay networks. IEEE Commun. Lett. **22**(1), 161–164 (2018)
4. Qin, Z., Fan, J., Liu, Y., Gao, Y., Li, G.Y.: Sparse representation for wireless communications: a compressive sensing approach. IEEE Signal Process. Mag. **35**(3), 40–58 (2018)
5. Qin, Z., Li, F.Y., Li, G.Y., McCann, J.A., Ni, Q.: Low-power wide-area networks for sustainable IoT. IEEE Wireless Commun. **26**(3), 140–145 (2019)
6. Al-Hourani, A., Kandeepan, S., Lardner, S.: Optimal LAP altitude for maximum coverage. IEEE Wireless Commun. Lett. **3**(6), 569–572 (2014)
7. Lyu, J., Zeng, Y., Zhang, R., Lim, T.J.: Placement optimization of UAV-mounted mobile base stations. IEEE Commun. Lett. **21**(3), 604–607 (2017)
8. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. IEEE Commun. Lett. **20**(8), 1647–1650 (2016)
9. Sun, J., Masouros, C.: Deployment strategies of multiple aerial BSs for user coverage and power efficiency maximization. IEEE Commun. Lett. **67**(4), 2981–2994 (2019)
10. Liu, C.H., Chen, Z., Tang, J., Xu, J., Piao, C.: Energy-efficient UAV control for effective and fair communication coverage: a deep reinforcement learning approach. IEEE J. Sel. Areas Commun. **36**(9), 2059–2070 (2018)