Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8842694, 11 pages
https://doi.org/10.1155/2020/8842694

Research Article

Deep Reinforcement Learning-Based Content Placement and Trajectory Design in Urban Cache-Enabled UAV Networks

Chenyu Wu,[1] **Shuo Shi**,[1,2] **Shushi Gu**,[2,3] **Lingyan Zhang**,[3] **and Xuemai Gu**[1,2]

[1]*School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China*
[2]*Peng Cheng Laboratory, Shenzhen, China 518052*
[3]*Harbin Institute of Technology (Shenzhen), Shenzhen, China 518055.*

Correspondence should be addressed to Shuo Shi; crcss@hit.edu.cn

Received 27 May 2020; Revised 19 June 2020; Accepted 13 July 2020; Published 14 August 2020

Academic Editor: Bingxian Lu

Cache-enabled unmanned aerial vehicles (UAVs) have been envisioned as a promising technology for many applications in future urban wireless communication. However, to utilize UAVs properly is challenging due to limited endurance and storage capacity as well as the continuous roam of the mobile users. To meet the diversity of urban communication services, it is essential to exploit UAVs' potential of mobility and storage resource. Toward this end, we consider an urban cache-enabled communication network where the UAVs serve mobile users with energy and cache capacity constraints. We formulate an optimization problem to maximize the sum achievable throughput in this system. To solve this problem, we propose a deep reinforcement learning-based joint content placement and trajectory design algorithm (DRL-JCT), whose progress can be divided into two stages: offline content placement stage and online user tracking stage. First, we present a link-based scheme to maximize the cache hit rate of all users' file requirements under cache capacity constraint. The NP-hard problem is solved by approximation and convex optimization. Then, we leverage the Double Deep Q-Network (DDQN) to track mobile users online with their instantaneous two-dimensional coordinate under energy constraint. Numerical results show that our algorithm converges well after a small number of iterations. Compared with several benchmark schemes, our algorithm adapts to the dynamic conditions and provides significant performance in terms of sum achievable throughput.

1. Introduction

With the development of wireless communication technology, the future networks will require high-quality multimedia streaming applications and highly diversified traffic demand. Unmanned aerial vehicles (UAVs) have brought promising opportunities to assist conventional cellular communication [1]. In urban environment, wireless communication suffers from severe shadowing due to Non-Line-of-Sight (NLoS) propagation [2, 3]. Compared with conventional terrestrial infrastructures, UAVs can be deployed at flexible altitudes, which leads to high probability of Line-of-Sight (LoS) dominant link. Due to the agility and low cost, UAVs can be easily and quickly deployed in a large number of scenarios including disaster relief. In addition, the maneuverability and mobility offer new opportunities for communication enhancement. The performance of communication is significantly improved by dynamically adjusting the UAV states, including flying direction, speed, transmission scheme, and storage resources allocation. The continuous and proper control of UAVs better suits the varying communication conditions.

Thanks to the advantages of mobility, agility, and high probability of LoS propagation, UAVs can be deployed to serve many practical applications, such as base station (BS) offloading [4], Internet-of-Things (IoT) data collection [5], mobile relays [6], and massive machine type communications [7]. Extensive research efforts focus on the two- or three-dimensional deployment of UAVs which are treated as stationary aerial BSs [8–10]. Another important branch focuses on the trajectory design which fully unleashes the potential of mobility to enhance performance [11]. To take fully advantage of limited resources, there are a lot of research aiming at optimizing the resource allocation such as power control and computing offloading. Moreover, the

deployment of multiple UAVs makes it ideal for meeting Adhoc demands with more flexible network architectures.

Despite all the promising benefits mentioned above, there are still many challenges to be overcome. First of all, though UAV helps to assist wireless communication, reacting as relay nodes limits the effectiveness of UAV networks, which still brings backhaul burden. Secondly, UAVs have limited endurance due to the constraint of load bearing. Thirdly, it is hard to track users with continuous mobility to maintain the high performance of optimization algorithms.

Content caching can be an effective approach to reduce the burden of backhaul link by storing popular files during off-peak periods [12]. To overcome the aforementioned challenges, we consider cache-enabled UAV networks, where multiple UAVs serve mobile users with preloaded content. By leveraging optimization and reinforcement learning method, we give the content placement and trajectory design to maximize the achievable sum throughput.

1.1. Related Work

(1) Cache-enabled UAV networks: in cache-enabled UAV networks, content requested from the users can be obtained via the local cache of UAVs, which bypasses the backhaul bottleneck and enhances the network capacity. The existing literature has studied many problems relating to time delay, throughput, and hit rate. Content placement [13, 14], transmission scheme, and trajectory design are studied in recent work to make performance gain. In [12], the authors focus on content-centric communication system and present a caching scheme which is divided into two stages. By jointly designing the trajectory, communication scheduling, and file caching policy, the authors make trade-off between the file caching and file retrieval cost. In [15], the authors optimize the deployment of UAVs and the content to cache to maximize the quality of experience of wireless devices. The authors in [16] study the content placement and virtual reality network to meet the delay requirement. The problem of trajectory design and content placement and delivery for UAV to vehicle link is studied in [17]. However, the movement of users and the energy-efficient control of UAV are mostly neglected. Moreover, most related work in urban scenario treats UAVs as stationary BS, which ignores the efficient control of UAVs

(2) Machine learning in UAV networks: machine learning, especially reinforcement learning (RL), has become a powerful tool to tackle wireless communication problems [18]. As hardware computing efficiency increases, machine learning can be used to deal with more complex problems including nonconvex optimization and multiagent regression. Interference management, trajectory design, power control, and energy harvesting problems are tackled with machine learning. The authors in [19] proposed a deep reinforcement learning-based algorithm to ensure fair coverage of ground terminals. The proposed deep deterministic policy gradient-based algorithm is extended to a crowd sensing scenario, where UAVs aid to collect data with the limited energy reserved [20]. In [21], Q-learning-based algorithm is used to solve the three-dimensional placement of UAVs to coverage ground users. In [22], the authors take the advantage of content-centric caching. UAV trajectory and content delivery are jointly optimized through actor-critic reinforcement learning-based algorithm to reduce the average request queue. A power transfer and data collection scheme is proposed in [23]. The problem of minimizing the overall packet loss is solved using deep reinforcement learning. However, there is little work on cache-enabled UAV networks by leveraging machine-learning approach. Machine learning can play a role in the wider field of UAV-assisted communication to meet the diversity of urban communication services

1.2. Our Contributions. In summary, the main contributions of the paper are as follows:

(1) We propose a scheme for cache-enabled UAV networks with mobile users in urban scenario. We formulate an optimization problem to maximize the sum achievable throughput under the storage capacity and energy constraints. The target function is multiobjective and nonconvex which is hard to solve using traditional optimization methods. Thus, we propose a deep reinforcement learning-based joint content placement and trajectory design algorithm (DRL-JCT) to solve this problem

(2) At the offline content placement stage, we formulate a link-based caching strategy to maximize the sum hit rate of users' file requirements. The objective function is nonlinear with binary variables, which is NP-hard. Caching scheme is obtained through approximation and convex optimization. The proper caching strategy obtains great gain in file hit rate by making trade-off between file popularity and diversity

(3) Considering the mobility of users and the constraint of UAV endurance, we propose a Double Deep Q-Network- (DDQN-) based online trajectory design algorithm to track real-time users. To the best of our knowledge, the mobility of users and energy-efficient UAV control have not been considered in most current research on cache-enabled UAV networks

(4) We compared our algorithm with several benchmark schemes. We demonstrate that the proposed algorithm shows a fast convergence and great gain. Numerical results shows that our algorithm achieves significant performance gain in terms of achievable throughput.

1.3. Organization. The rest of papers is organized as follows. In Section 2, we describe the system model and formulate the

Table 1: List of notations.

Notations	Descriptions
K	Number of UAVs
T	Total time slots
$Vmax$	Maximum speed of UAVs
$x_i^k(t)y_i^k(t)$	Coordinate of users
F	Total number of files
q_f^k	Caching probability
$P_{\text{LoS}}, P_{\text{NLoS}}$	LoS and NLoS probability
$r_i^k(t)$	Instantaneous rate of user
B	Total bandwidth of each UAV
$\mu_{\text{LoS}}, \mu_{\text{NLoS}}$	Additional path loss for LoS, NLoS
$P_0, P_1, P_2, v_0, U_{\text{Tip}}$	Propulsion power relevant parameters
$\mathcal{K}_u(i)$	UAV set with link to user i
U	Number of users
δ	Time slot length
H	Altitude of UAVs
γ	Zipf exponent
P_f	Content popularity
C^k	Storage capacity
θ_i^k	Elevation angle
N_0	Noise power spectral density
a, b	Environmental parameters (urban)
α	Path loss exponent
$E_{\max}$	Battery capacity
s_t, a_t, r_t	State, action, and reward in RL

problem of content placement and trajectory design. In Section 3, the efficient content placement policy is presented. In Section 4, we propose our RL-based trajectory design with roaming users. The simulation results are illustrated and analyzed in Section 5. The conclusions are presented in Section 6. In addition, the list of notations is shown in Table 1.

2. System Model and Problem Formulation

We consider the downlinks of cache-enabled UAV networks in urban environment as shown in Figure 1. Multiple UAVs are deployed as aerial BSs to cache files and serve the users of our target area. The UAVs connect with the core network by terrestrial BS. After the UAVs complete the content placement, the drones use online tracking algorithm to obtain achievable maximum service rate. There are a set $\mathcal{K}$ of K UAVs serving a set $\mathcal{U}$ of U mobile users. We take the fixed link scheme that the links are predetermined by random or signal-to-noise ratio- (SNR-) based user allocation algorithms. The network topology and the maximum users that a UAV can serve are predetermined. Each user is served by at least one UAV, and we denote $\mathcal{K}_u(i)$ as the set of UAVs with link to user i. We denote U_k as the total amount of users

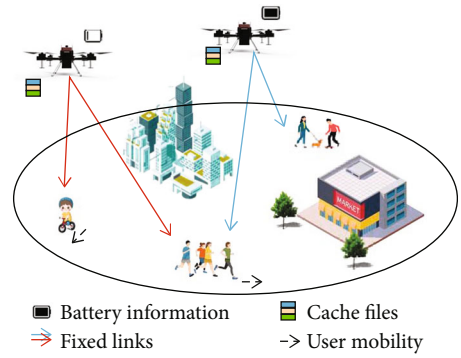

Figure 1: Illustration of cache-enabled UAV networks in urban environment. Multiple UAVs serve mobile users with fixed links. We maximize the achievable sum rate under storage capacity and energy constraints.

served by UAV k and u_i^k as the i-th user. Users affiliated to the same UAV are served simultaneously by frequency division multiple access (FDMA).

2.1. Mobility and Caching Model. We assume that the users are distributed as the Homogeneous Poisson Point Process (HPPP) [24]. The users may move continuously during the task period of UAVs. Mobility models can be random walk models or deterministic models. Considering the high mobility of UAVs, we divide the flying time horizon into T equal time slots with sufficiently small slot length δ, indexed by t. The UAVs are assumed to fly at a fixed altitude H with a speed not exceeding maximum speed $V_{\max}$. The two-dimensional coordinate of user at time slot t can be expressed as $[x_i^k(t), y_i^k(t)]^T$, while the horizontal coordinate of UAV k is $[x^k(t), y^k(t)]^T$. The position set of UAVs during the flying period is denoted as $\boldsymbol{x}$ and $\boldsymbol{y}$. Using time and trajectory discretization, the distance between UAV k to user u_i^k can be assumed to be a constant during one time slot t which is

$$d_i^k(t) = \sqrt{H^2 + \left[x^k(t) - x_i^k(t)\right]^2 + \left[y^k(t) - y_i^k(t)^2\right]}. \quad (1)$$

For the file database system, we assume that there are F contents with the same normalized size. The popularity which indicates the users' interest in a certain file is modeled using the Zipf law [25, 26]. The popularity p_f of content f, $f = \{1, 2, \cdots, F\}$ is modeled as

$$p_f = \frac{1/f^\gamma}{\sum_{j=1}^{F} 1/j^\gamma}, \quad (2)$$

where γ is the Zipf exponent describing the probability of content reuse. The larger the γ is, the more concentrated the files are.

Due to the fact that UAVs have limited storage capacity, we further assume that the capacity of UAV k is C^k and thus that each UAV can proactively cache no more than C^k popular contents. The caching probability of UAV k for

content f is denoted as q_f^k, and then we have the following constraint:

$$\sum_{f=1}^{F} q_f^k \le C^k \quad \forall k \epsilon \mathcal{K}. \tag{3}$$

The caching strategy set with $K * F$ binary variables is denoted as $\boldsymbol{q}$. Considering that the UAVs are not able to cache all required files, an appropriate content placement strategy is of vital importance. Proactive caching and proper content placement save transmission resource and improve the achievable sum coverage throughput.

2.2. Transmission Model. The downlinks between UAVs and users can be regarded as a LoS dominant air-to-ground channel. In urban environment, the LoS link may be occasionally blocked by obstacles like high buildings and towers. We use Probabilistic LoS Channel model by taking account of the shelter [27–29]. The LoS probability can be expressed as

$$P_{\mathrm{LoS}}\left(\theta_i^k\right) = \frac{1}{1 + a\ \exp\left(-b\left(\theta_i^k - a\right)\right)}, \tag{4}$$

where $\theta_i^k = \sin^{-1}(H/d_i^k)$ is the elevation angle UAV k and user u_i^k and a and b are parameters related to the environment. The probability of not having direct link to user u_i^k is thus given by $P_{\mathrm{NLoS}}(\theta_i^k) = 1 - P_{\mathrm{LoS}}(\theta_i^k)$. Intuitively, P_{LoS} increases as the elevation angle increases and approaches 1 as θ_i^k gets sufficiently large.

Then, the channel's power gain between UAV k and user u_i^k can be expressed as

$$g_i^k(t) = \left(\frac{4\pi f_c}{c}\right)^{-2} \left[d_i^k(t)\right]^{-\alpha} \left[P_{\mathrm{LoS}}\left(\theta_i^k\right)\mu_{\mathrm{LoS}} + P_{\mathrm{NLoS}}\left(\theta_i^k\right)\mu_{\mathrm{NLoS}}\right]^{-1}, \tag{5}$$

where f_c is the carrier frequency, c is the speed of light, α is the path loss exponent, and μ_{LoS} and μ_{NLoS} are the attenuation factors of LoS and NLoS links, respectively.

For simplicity, we assume that the bandwidth B for each UAV is equally allocated to the associated users; thus, the licensed spectrum for all the U^k user is $B_i^k = B/U^k$. Also, the maximum transmit power $P_{\max}$ for each UAV k is also uniformly allocated; thus, the transmit power allocated to user u_i^k is $P_i^k = P/U^k$. By calculating the received SINR, the achievable rate of user u_i^k at time slot t can be expressed in bit/s as

$$r_i^k(t) = B_i^k \log_2\left(1 + \frac{p_i^k g_i^k(t)}{\sigma^2}\right), \tag{6}$$

where $\sigma^2 = B_i^k N_0$ is the variance of Addictive White Gaussian Noise (AWGN) and N_0 is the noise power spectral density.

2.3. Energy Consumption Model. The energy consumption of a UAV-aided communication system generally consists of two parts: transmission energy, which is related to communication, and propulsion energy, which aims at supporting the movement of UAVs. Compared to the flight energy consumption, the communication-related power is small enough to be negligible. In this paper, we consider the rotary-wing UAV propulsion energy consumption model which depends on the instantaneous velocity in [30]. The propulsion power of UAV k with scalar velocity v is

$$e^k(v) = P_0\left(1 + \frac{3v^2}{U_{\mathrm{tip}}^2}\right) + P_1\left(\sqrt{1 + \frac{v^4}{4v_0^4}} - \frac{v^2}{2v_0^2}\right)^{1/2} + P_2 v^3, \tag{7}$$

where P_0 and P_1 are two constants representing the blade profile power and induced power when hovering, respectively. U_{tip} denotes the tip speed of the rotor blade, and P_2 and v_0 are parameters related to fuselage. We denote $E_{\max}$ as the onboard energy, and the total energy consumption should not exceed it for the sake of safe return or landing.

2.4. Problem Formulation. To investigate the benefits brought by cache-enabled UAVs, we optimize the following formulated problem to maximize the achievable sum service rate R_{sum}.

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{q}} R_{\mathrm{sum}} = \sum_{f=1}^{F}\sum_{i=1}^{U}\sum_{t=1}^{T} p_f r_{i,f}^k(t) \mathbb{1}\left\{\sum_{k \epsilon K_u(i)} q_f^k \ge 1\right\}, \tag{8a}$$

$$s.t \sum_{f=1}^{F} q_f^k \le \mathrm{C}^k, \tag{8b}$$

$$q_f^k \epsilon \{0, 1\} \quad \forall k, f, \tag{8c}$$

$$\sum_{t=1}^{T} e^k(t) \le E_{\max} \forall k, \tag{8d}$$

$$0 \le v^k(t) \le V_{\max} \forall k, t, \tag{8e}$$

where $\mathbb{1}\{\bullet\}$ is the indicator function and guarantees the file request from user i which can be responded by any of the neighboring UAVs. Equation (8b) represents the probability of content placement is binary variables. $r_{i,f}(t)$ is the service rate of UAVs associated with user i which is positively relevant to $r_i^k, k \in \mathcal{K}_u(i)$. Equation (8d) ensures that the sum propulsion consumption will not exceed the battery capacity. Equation (8e) is the constraint of maximum flying speed for UAV control.

From (8a) we can see that the achievable service rate is related to content placement strategy and instantaneous position of UAVs. A proper caching scheme makes tradeoff between cooperation gain and content diversity gain, thus increasing the hit rate of users' requirements. In real scenario, due to the mobility of users and energy constraint of UAVs, appropriate control policy of UAVs can enhance the endurance and throughput, which both improve the overall system service rate.

To solve the optimizing problem is challenging due to the nonconvex target function and restricted conditions. Any search-based algorithms will be of high computational complexity. To solve this problem, we proposed our algorithm DRL-JCT to jointly optimize content placement and online trajectory.

3. Offline Content Placement

In this section, we present our offline content placement strategy based on file popularity and existed links.

Considering that the users move continuously and randomly, it is hard to predict the real-time positions of users. Hence, it is difficult to predefine which content to cache according to user locations. A link-based algorithm is a good substitute to deal with offline content placement. We assume the channel between UAVs and users are fixed according to frequency spectrum and user-side information. Let l_{ik} denotes whether there is a link between user i and UAV k. Then, the responsible UAV set for serving user i is $\mathscr{K}_u(i) = \{l_{ik} = 1, i \in U\}$. When the initial locations of users are known, the links can be allocated according to SNR or throughput. When there is no user-side information, the links can be allocated randomly according to the numbers of UAV and the maximum users that a UAV can serve.

We use deterministic caching model which is commonly used in cache-enabled networks [31, 32]. To take the full advantages of caching capacity, we transform the constraint F of binary variables q_f^k to equality, which is $\sum_{f=1}^{F} q_f^k = C^k$. We aim to maximize the hit rate of users' file requirements, which is formed as

$$\max H(\boldsymbol{q}) = \sum_{i=1}^{U}\sum_{f=1}^{F} p_f \mathbb{1}\left\{\sum_{k\epsilon\mathscr{K}_u(i)} q_f^k \geq 1\right\}, \tag{9}$$

However, the hit rate is also related to the arrival rates of files which varies with the activity of users. We assume the arrival rates are normalized for simplicity. The indicated function which presents the ability of the adjacent UAVs to provide content f for user i can also be written as $\mathbb{1}\{\sum_{k\epsilon\mathscr{K}_u(i)} q_f^k \geq 1\} = 1 - \prod_{k\epsilon K_u(i)}(1 - q_f^k)$. The problem is a binary optimization problem with nonlinear objective funtion which is proved to be NP-hard. To tackle this problem, we introduce nonnegative slack variables μ_m and reformed the problem as

$$\min H'(\boldsymbol{q}) = -\left\{\sum_{i=1}^{U}\sum_{f=1}^{F} p_f\left[1 - \prod_{k\epsilon\mathscr{K}_u(i)}\left(1 - q_f^k\right)\right] + \mu_m\left(1 - q_f^k\right)q_f^k\right\}. \tag{10}$$

Note that the maximization problem (9) is identical to (10), and the introduced slack variables do not affect the optimal value. We then relax the variables q_f^k to closed interval [0,1] to form $H'(\widehat{\boldsymbol{q}})$ with continuous variables.

Lemma 1. *$H'(\widehat{\boldsymbol{q}})$ is convex [33] when the slack variables satisfies*

$$\mu_m > \frac{1}{2}\sum_{j\neq m}\left|\mathscr{H}'(\widehat{\boldsymbol{q}})_{mj}\right| \; m = 1, 2, \cdots, KF, \tag{11}$$

where $\mathscr{H}'(\widehat{\boldsymbol{q}})_{mj}$ is the mj-th term of the Hessian matrix of $H'(\widehat{\boldsymbol{q}})$.

Proof. We can calculate that

$$\frac{\partial^2 H'(\widehat{q}_m)}{\partial \widehat{q}_m^2} = \frac{\partial^2[-\mu_m(1 - q_m)q_m]}{\partial \widehat{q}_m^2} = 2\mu_m. \tag{12}$$

Then, the diagonal elements of the Hessian matrix is $2\mu_m$. According to the Gershgorin Theorem, the range of eigenvalues of a square matrix **A** satisfies

$$|\lambda - a_{ii}| \leq \sum_{j\neq i} a_{ji}. \tag{13}$$

Then, the lower bound of each eigenvalue of the Hessian is $2\mu_m - \sum_{j\neq m}|\mathscr{H}^i(\widehat{q})_{mj}|$. Given the constraint in (11), all the eigenvalues are nonnegative, which means the Hessian matrix is positive semidefinite. According to the properties of convex function, $H'(\widehat{\boldsymbol{q}})$ is convex.

By properly choosing slack variables, we can efficiently solve the convex problem by tools like CVX. Note that the solution contains the original problem (9) since it searches on a larger scale. Since the optimal solution may be fractional and we need a binary solution, we can approximate the optimal solution by the greedy method. Specifically, we choose one for solution bigger than 1/2 and inversely, zero.

It can be seen that the larger the overlapping part of the users, the more decentralized the content should be to satisfy multiple content requirements. When the number of the users that are served by a single UAV is larger, the optimal solution tends to be the most popular strategy. Ultimately, when all the users are covered by only one UAV, the product terms in (10) vanished, which makes the problem linear.

The link-based caching strategy is independent from user locations when the popularity of each content is identical so that the caching stage can be completed before taking off. However, the optimal achievable throughput relies on the relative position between UAVs and users. In the next section, we propose our online trajectory design to meet the need of mobile users.

4. Online Trajectory Design

In the actual scene, the users tend to move continuously which may lead to throughput reduction. Actually, there are no traditional solutions to track the users efficiently by calculating. Exhaustive search may cause big calculation time and high latency. Thus, the reinforcement learning is invoked to track real-time users. In this section, we first introduce

some preliminaries of Deep Reinforcement learning, and then we present our Double Deep Q-learning-based algorithm to maximize the sum throughput.

4.1. Deep Reinforcement Learning Background. Reinforcement learning contains basic elements including environment, agent, state, action, and reward. In reinforcement learning, an agent interacts with the environment with discrete decision epochs. Our training agent is selected as each UAV. The state can be set as all relevant parameters such as speed, location, and energy, and actions are chosen according to the current state. The process can be modeled as a Markov Decision Process (MDP) with a set, $\mathscr{M} = <\mathscr{S}, \mathscr{A}, \mathscr{R}, \Pr(s_{t+1} \mid s_t, a)$ where $\mathscr{S}, \mathscr{A}$, and $\mathscr{R}$ are the set of state, action, and reward, respectively. $\Pr(s_{t+1} \mid s_t, a_t)$ denotes the transmit probability set from s_t to s_{t+1} when action a_t is taken.

Q-learning is one of the simple algorithms of reinforcement learning used in UAV control. The basic idea for Q-learning is to maintain a table to record and maximize the long-term discounted cumulative reward.

$$\max C = \mathbb{E}^{\pi}\left(\sum_{t=1}^{\infty} \gamma_d^{t-1} r(s_{t+1} \mid s_t, a_t)\right), \tag{14}$$

where $\pi = \arg\max_{a_t \epsilon \mathscr{A}} Q(s_t, a_t)$ is the policy to choose action.

A good tip to choose the best action is to adopt the ϵ -greedy policy in order to explore the environment. γ_d is the discount factor for future state. Following the Bellman equation to find the best decision of MDP process, the Q table which is also known as value function is updated by

$$Q_{t+1}(s_t, a_t) = (1-\alpha) Q_t(s_t, a_t) + \alpha\left(r_t + \gamma_d \max_{a'} Q_t\left(s_{t+1}, a'\right)\right), \tag{15}$$

where α is a small positive fraction indicating learning rate. Q-learning algorithm is proved to be convergent. However, since this algorithm requires a value table for each action and state pair, when the state space get larger, the table gets extremely huge, which causes a curse of dimensionality [34] problem. Also, Q-learning is unable to deal with continuous space problem.

Combined Q-learning with neural network, Deep Q-Network (DQN) [35] can be seen as a "deep" version of Q-learning. DQN uses a neural network to estimate the huge Q table. The DQN is trained by minimizing the loss function:

$$L\left(\theta^Q\right) = \mathbb{E}\left[r_t + \gamma_d Q'\left(s_{t+1}, \pi(s_{t+1}) \mid \theta^{Q'}\right) - Q\left(s_t, a_t \mid \theta^Q\right)\right]^2, \tag{16}$$

where the first part $y_t = r_t + \gamma_d Q'(s_{t+1}, \pi(s_{t+1}) \mid \theta^{Q'})$ is the target value to reach and θ^Q is the weight vector of the DQN. The network updates θ^Q from the derivative $\nabla L(\theta^Q)$ by common methods like gradient decent and back propagation.

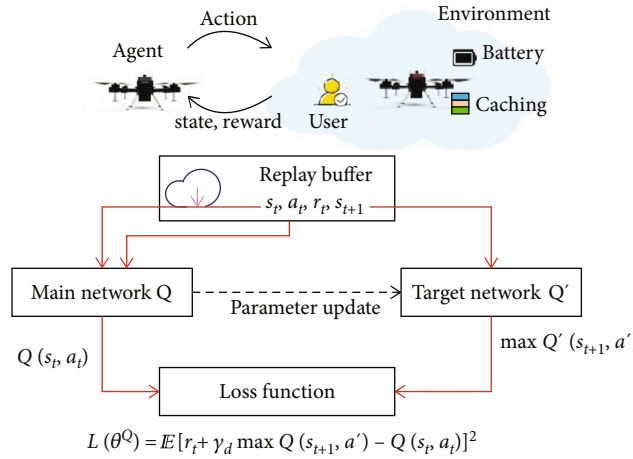

Figure 2: The architecture and basic idea of DQN.

Additionally, DQN adopts two techniques, experience replay and target network, to mitigate the impact of correlations between data. The experience replay buffer stores past training data and packs them in a batch. Experience replay is conducted to randomly choose a minibatch with the size of B_s from the experience replay buffer. Moreover, DQN uses target network with the same structure as the original neural network. The parameters of target network are updated using weights of original network with delay. The architecture and basic idea of DQN are shown in Figure 2.

4.2. DDQN for Trajectory Design. Based on the idea of DQN, our agent, state, action, and reward are defined as follows:

(1) Agent: our training agents are UAV $k, k = 1, 2, \cdots, K$

(2) State: for each training epoch t (the training epoch can be seen as time slot or training step), we define $s_t = [x^k(t), y^k(t), x_1^k(t), x_2^k(t), \cdots, x_{U_k}^k(t), E^k(t)]$. The state for each agent k consists of the dynamic position of UAV and users and also the current energy available. Thus, the agent can take actions according to its battery capacity and the current user-side information

(3) Action: the actions represent the flying velocity and direction of each UAV. The instantaneous speed can be discretized to several options and the upper bound $V_{\max}$. Also, the agent can choose to hover at one point. The UAV can fly to eight directions: forward, backward, left, right, northeast, northwest, southeast, and southwest

(4) Reward: the reward of epoch t is defined as:

$$r(t) = \frac{\sum_{i=1}^{U_k} r_i^k(t)}{e_i^k(t)}, \tag{17}$$

which is the current energy efficiency. Using this, the agent can make a trade-off between sum throughput and energy consumption, which further improves the endurance.

1: Initialize content placement $\boldsymbol{q}$.
2: Randomly initialize value function Q with weight $\boldsymbol{\theta}$.
3: Initialize target value function $Q^{'}$ with weight $\boldsymbol{\theta}^{-}$.
4: Initialize replay memory $\mathcal{D}$ to size N, replay buffer size to B_s.
5: **for** episode $m = 1, 2, \cdots, M$ **do**
6: Initialize environment and state to s_1.
7: **while** available energy > 0 **do**
8: **if**random $\leq \epsilon$ **then**
9: choose action $a_t = \arg\max_a Q(s_t, a; \boldsymbol{\theta})$.
10: **else**
11: randomly choose an action.
12: **end if**
13: Execute a_t and observe s_{t+1}, r_t.
14: store transition (s_j, a_j, r_j, s_{j+1}) in $\mathcal{D}$.
15: sample random minibatch (s_j, a_j, r_j, s_{j+1}) with size B_s from $\mathcal{D}$.
16: Calculate target value: $y_j = r_j + \gamma_d Q^{'}(s_{j+1}, \arg\max_a Q(s_{j+1}, a \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{-})$.
17: Loss function $L(\boldsymbol{\theta}^{Q}) = \sum_{j=1}^{B_s}[y_i - Q(s_j, a_j \mid \boldsymbol{\theta})]^2$
18: update $\boldsymbol{\theta}$ using $\nabla L(\boldsymbol{\theta}^{Q})$ by gradient decent.
19: Every B_{up} steps reset $\boldsymbol{\theta}^{-} = \boldsymbol{\theta}$.
20: **end while**
21: **end for**

Algorithm 1: Deep reinforcement learning-joint caching and trajectory design (DRL-JCT).

Table 2: Simulation parameters.

Notations	Descriptions	Value
δ	Time slot length	1 s
$V_{\max}$	Maximum speed of UAVs	40 m/s
H	Altitude of UAVs	100 m
N_0	Noise power spectral density	-174 dBm
B	Total bandwidth of each UAV	1 MHz
a, b	Environmental parameters(urban)	10, 0.15
μ_{LoS}	Additional path loss for LoS	2
μ_{NLoS}	Additional path loss for NLoS	100
α	Path loss exponent	2
P_0, P_1	Parameters of blade profile	580, 790
U_{Tip}	Tip speed of the rotor blade	200
P_2, v_0	Parameters related to fuselage	0.79, 7.2

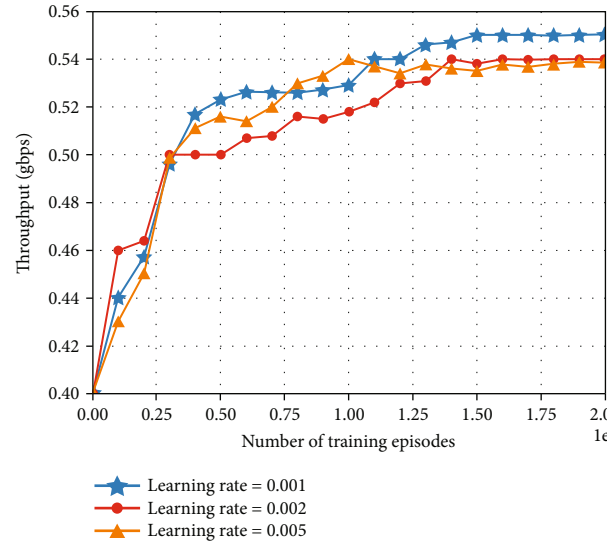

Figure 3: Convergence of DDQN algorithm with different learning rates versus the number of training episodes.

We also use the Double Deep Q-Network (DDQN) [36] which is an improved version of DQN. This architecture makes small changes at the action chosen but brings significant improvement. Traditional Deep Q-learning has the drawback of over estimating the value function. The Double Deep Q-network uses separate architecture for choosing action and estimating the value brought by actions, which eliminates the correlation between the two networks. Thus, the estimated value function is closer to the true value. Instead of choosing action according to the target net, DDQN finds the action corresponding to the maximum Q value according to the current network.

5. Simulation Results

We conducted extensive simulations to evaluate our proposed solution: DRL-JCT. In this section, we introduce our simulation settings at first and then present results and analysis.

5.1. Simulation Settings. Our experiments are performed with TensorFlow 1.0 and Python 3.7. In our simulation, we set the target area to be square with the size of 800 × 800 m. The simulation parameters are summarized in Table 2. We compared DRL-JCT with some commonly used baselines. For caching strategy, we choose the Uniform Distributed Caching (UDC)

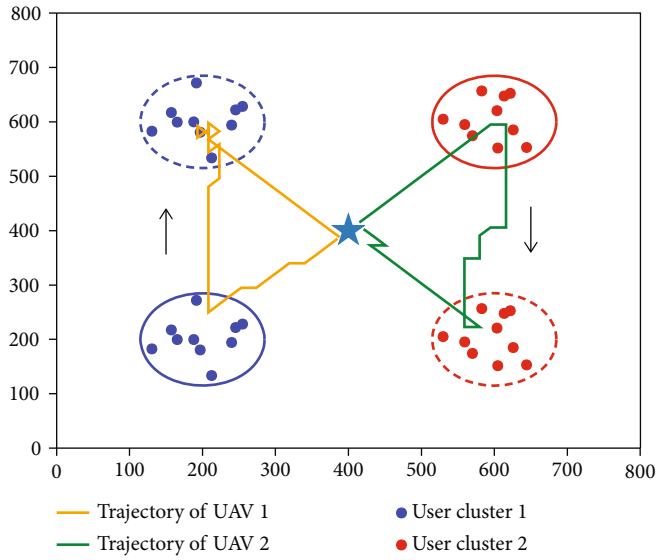

Figure 4: Trajectory of UAVs when users are roaming. The blue star stands for the starting point. The coordinate stands for the testing place.

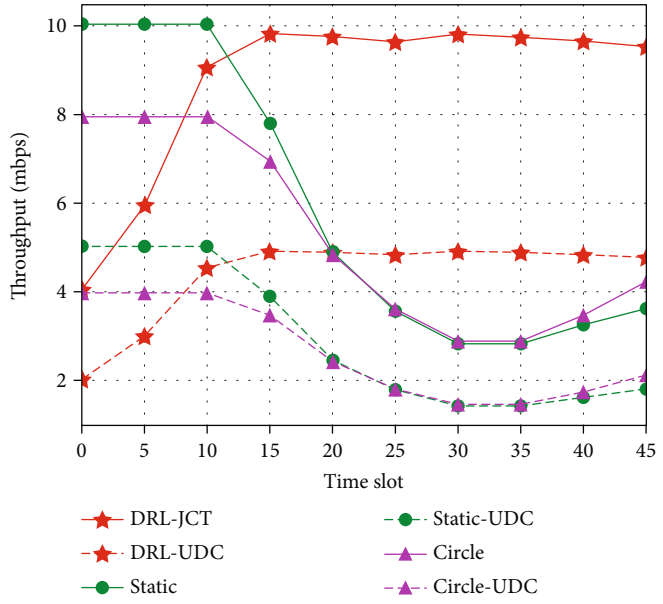

Figure 5: Comparing between DRL-JCT and other baselines over instantaneous throughput in movement scenario.

[37] and Most Popular Contents (MPC) as baselines. Using UDC, each UAV randomly selects C^k different files to store according to the uniform probability $q_f^k = C^k/F$. Using MPC, each UAV dependently caches C^k most popular contents.

For trajectory design, we choose 2 common baselines:

(1) Hovering: to avoid collision, the UAVs hover at the cluster centers of the users

(2) Circular flight trajectory: the UAVs periodically move around the cluster centers with radius of 100 m and constant velocity.

We find the appropriate hyperparameters in neural networks by a great number of experiments. We set the learning rate as 0.001, batch size $B_s = 32$, update iteration $B_{\text{up}} = 200$, memory size $N = 2000$, and discount factor $\gamma_d = 0.9$. We use a two-layer fully connected neural network to serve as the target and evaluate networks, and the number of neural units of the hidden layer is 120.

5.2. Results and Analysis. Figure 3 shows that the sum throughput with the number of training episodes. We set the total energy available as 46 kJ. RL contains many hyperparameters such as learning rate, discount factor, and memory size as mentioned above. Choosing appropriate hyperparameters can improve the performance of DDQN. Among them, we demonstrate the influence of learning rate. Learning rate can be neither too large nor small since large

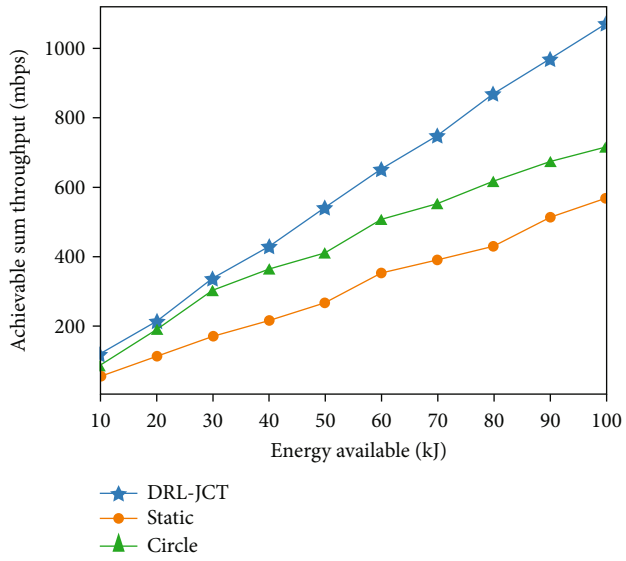

Figure 6: Achievable throughput versus the energy available.

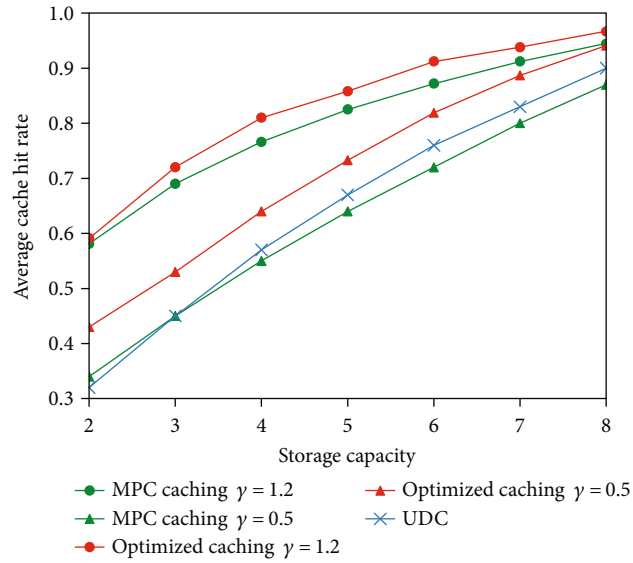

Figure 7: Average cache hit rate versus the storage capacity and the Zipf parameter γ.

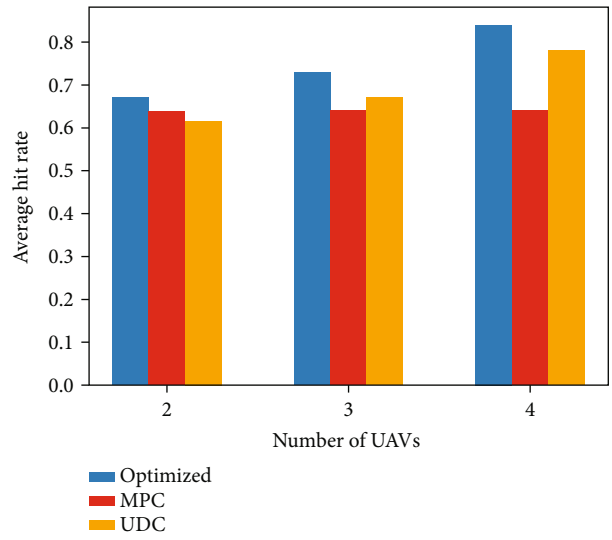

Figure 8: Average cache hit rate versus the number of UAVs.

learning rate causes fluctuation while small one tends to cause slow training speed. We choose different learning rates 0.001, 0.002, and 0.005 to train the neural network. We can see from the figure that 0.001 is the best choice in this scene since the throughput tends to increase smoothly to convergence with the training episodes. The other two learning rates can also reach a relatively small convergence throughput. Figure 4 plots the trajectories of UAVs derived from the proposed approach under the circumstance of moving users. There are totally $F = 10$ files, and the storage capacity $C^k = 5$. For simplification, we assume that there are 20 users which are clustered into 2 groups, demonstrated as red and blue solid circle. Two UAVs take off from the airport displayed as the blue star. The users follow deterministic moving model. The users of the first cluster move from (200, 200) at the 20th time slot to (400, 400) while the second cluster moves from (600, 600) to (600, 200). Since the users tend to at relatively slow speed, the UAVs may hover around to wait the users thus forming polyline.

Figure 5 characterizes the real-time throughput of one UAV derived from different algorithms in the movement scenario shown in Figure 4. We compared DRL-JCT with other baseline trajectory with optimized content caching and UDC separately.

We can observe that the throughput of DRL-JCT keeps increasing and maintains relatively constant since the algorithm can track the mobile users. The throughput of the static UAV is high at the beginning because it is deployed at the cluster center initially, and the throughput approximates the optimal coverage. But, as the users start to roam, neither the static deployment nor circle trajectory can meet the need of tracking mobile users. We can also analysis that DRL-JCT improves the performance of the system in terms of optimal content placement compared with the random caching strategy.

Figure 6 demonstrates the achievable sum throughput versus the available energy over different trajectories. We consider a more general situation where the users move following the random walk model and the UAVs carry optimized cache content. The moving direction and speed of each user are uniformly distributed. It can be observed that the sum throughput increases as the rise of endurance and DRL-JCT achieves the best performance over the benchmarks. Note that hovering at a still position is not a good choice in UAV control since it consumed more energy in unit time slot, and this is confirmed by the result of the figure.

Figure 7 shows the average cache hit rate versus the storage capacity and the Zipf parameter γ. In this scenario, 15 users are served by 3 UAVs. The total files $F = 10$. The links are established according to SNR threshold $\tau = 500$ kbps. Intuitionally, the hit rate increases with larger storage space. For different parameters, the optimized caching scheme outperforms the random caching and most populer content

caching. When γ gets larger, the files are more concentrated, which means the popular files are more often required, so that the MPC approximates the optimized solution.

Figure 8 compares the average cache hit rate with different numbers of serving UAVs. The UAVs are deployed separately using circle packing algorithm [38] to maximize the coverage range. The links are chosen based on throughput threshold $\tau = 500$ kbps. The number of users, the maximum users a UAV can serve, the total files, and the caching capacity are 15, 10, 10, and 5, respectively. It is clear that the hit rate increases with more serving UAVs. The behavior of MPC remains unchanged since it does not exploit the diversity of content. As the number of UAV gets larger, there are more overlapping areas and more stable links. The UDC scheme may perform well but waste the resource of air vehicles.

6. Conclusion

Cache-enabled UAV networks have been an appealing technology in wireless communication. However, the efficient use of UAVs meets greet challenge due to power and storage constraints. In this paper, we propose the DRL-JCT algorithm to jointly optimize the caching strategy and trajectory in cache-enabled UAV networks. We give the optimal caching scheme and trajectory design, respectively, using convex approximation and deep reinforcement learning approaches. Numerical results show that our algorithm has much better performance than the baselines in terms of achievable sum throughput. In our future work, we will consider a more complicated scenario of real-time transmission and content requirements in urban cache-enabled UAV networks.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Sciences Foundation of China, under grant 61701136, and the project "The Verification Platform of Multi-tier Coverage Communication Network for Oceans" (LZC0020).

References

[1] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: potential, challenges, and promising technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2019.

[2] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5379–5388, 2020.

[3] X. Liu, C. Sun, M. Zhou, C. Wu, B. Peng, and P. Li, "Reinforcement learning-based multislot double-threshold spectrum sensing with Bayesian fusion for industrial big spectrum data," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2020.

[4] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3988–4001, 2018.

[5] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2020.

[6] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[7] M. N. Soorki, M. Mozaffari, W. Saad, M. H. Manshaei, and H. Saidi, "Resource allocation for machine-to-machine communications with unmanned aerial vehicles," in *2016 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Washington, DC, USA, December 2016.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, 2016.

[9] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2017.

[10] M. M. Azari, F. Rosas, K. Chen, and S. Pollin, "Optimal UAV positioning for terrestrial-aerial communication in presence of fading," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Washington, DC, USA, December 2016.

[11] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming endurance issue: UAV-enabled communications with proactive caching," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1231–1244, 2018.

[12] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2018.

[13] Y. Zhu, G. Zheng, L. Wang, K.-K. Wong, and L. Zhao, "Content placement in cache-enabled sub-6 gHz and millimeter-wave multi-antenna dense small cell networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 2843–2856, 2018.

[14] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 247–257, 2020.

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

[16] M. Chen, W. Saad, and C. Yin, "Echo-liquid state deep learning for 360° content transmission and caching in wireless vr networks with cellular connected UAVs," *IEEE Transactions on Communications*, vol. 67, no. 9, pp. 6386–6400, 2019.

[17] H. Wu, J. Chen, F. Lyu, L. Wang, and X. Shen, "Joint caching and trajectory design for cache-enabled UAV in vehicular networks," in *2019 11th International Conference on Wireless*

Communications and Signal Processing (WCSP), pp. 1–6, Xi'an, China, China, October 2019.

[18] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[19] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: a deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, 2018.

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

[21] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: deployment and movement design," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8036–8049, 2019.

[22] S. Chai and V. K. N. Lau, "Online trajectory and radio resource optimization of cache-enabled UAV wireless networks with content and energy recharging," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1286–1299, 2020.

[23] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "On-board deep Q-network for UAV-assisted online power transfer and data collection," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12215–12226, 2019.

[24] E. Bastug, M. Bennis, and M. Debbah, "Cache-enabled small cell networks: modeling and tradeoffs," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pp. 649–653, Barcelona, Spain, August 2014.

[25] J. Song, H. Song, and W. Choi, "Optimal content placement for wireless femto-caching network," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4433–4444, 2017.

[26] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, pp. 126–134, New York, NY, USA, USA, March1999.

[27] Q. Feng, E. K. Tameh, A. R. Nix, and J. McGeehan, "Wlcp2-06: modelling the likelihood of line-of-sight for air-to-ground radio propagation in urban environments," in *IEEE Globecom 2006*, pp. 1–5, San Francisco, CA, USA, December 2006.

[28] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *2014 IEEE Global Communications Conference*, pp. 2898–2904, Austin, TX, USA, December 2014.

[29] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[30] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[31] S. Krishnendu, B. N. Bharath, and V. Bhatia, "Cache enabled cellular network: algorithm for cache placement and guarantees," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1550–1554, 2019.

[32] J. Yao and N. Ansari, "Joint content placement and storage allocation in c-rans for IoT sensing service," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1060–1067, 2019.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, MA, USA, 2013.

[34] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, pp. 1329–1338, New York, NY, USA, 2016.

[35] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[36] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI conference on artificial intelligence*, pp. 2094–2100, Phoenix, AZ, USA, 2016.

[37] S. Tamoorulhassan, M. Bennis, P. H. J. Nardelli, and M. Latvaaho, "Modeling and analysis of content caching in wireless small cell networks," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 765–769, Brussels, Belgium, August 2015.

[38] Z. Gáspár and T. Tarnai, "Upper bound of density for packing of equal circles in special domains in the plane," *Periodica Polytechnica Civil Engineering*, vol. 44, no. 1, 2000.