# Multi-Agent Reinforcement Learning Based 3D Trajectory Design in Aerial-Terrestrial Wireless Caching Networks

Yu-Jia Chen, *Member, IEEE*, Kai-Min Liao, Meng-Lin Ku, *Senior Member, IEEE*, Fung Po Tso, and Guan-Yi Chen

***Abstract*—This paper investigates a dynamic 3D trajectory design of multiple cache-enabled unmanned aerial vehicles (UAVs) in a wireless device-to-device (D2D) caching network with the goal of maximizing the long-term network throughput. By storing popular content at the nearby mobile user devices, D2D caching is an efficient method to improve network throughput and alleviate backhaul burden. With the attractive features of high mobility and flexible deployment, UAVs have recently attracted significant attention as cache-enabled flying base stations. The use of cache-enabled UAVs opens up the possibility of tracking the mobility pattern of the corresponding users and serving them under limited cache storage capacity. However, it is challenging to determine the optimal UAV trajectory due to the dynamic environment with frequently changing network topology and the coexistence of aerial and terrestrial caching nodes. In response, we propose a novel multi-agent reinforcement learning based framework to determine the optimal 3D trajectory of each UAV in a distributed manner without a central coordinator. In the proposed method, multiple UAVs can cooperatively make flight decisions by sharing the gained experiences within a certain proximity to each other. Simulation results reveal that our algorithm outperforms the traditional single- and multi-agent Q-learning algorithms. This work confirms the feasibility and effectiveness of cache-enabled UAVs which serve as an important complement to terrestrial D2D caching nodes.**

***Index Terms*—Unmanned aerial vehicles (UAVs), trajectory design, wireless caching, multi-agent reinforcement learning.**

## I. Introduction

In RECENT years, the proliferation of data-intensive wireless applications, such as augmented reality and mobile online gaming, leads to an explosion of network traffic. To alleviate the backhaul overload caused by duplicate content transmission, device-to-device (D2D) caching is a promising approach by storing popular content at the users' cache. In that way, the content requests can be served via D2D communications without incurring the cost of using cellular bandwidth [1]. However, designing wireless D2D caching systems face two major challenges. Firstly, the content requests may not be satisfied due to the limited cache storage of D2D users. Secondly, caching at static nodes may not be effective in a mobile environment. Although caching contents at multiple D2D nodes or base stations (BSs) may resolve this challenge, it still suffers from signaling overhead and additional storage cost. Therefore, there is a need for a flexible deployment of cache-enabled BSs that can track the users' movement to effectively transmit the requested contents.

Unmanned aerial vehicles (UAVs) as aerial BSs [2] in which popular contents can be cached provide several benefits to the cellular network. Firstly, UAVs can provide wider wireless coverage due to high line-of-sight (LoS) probabilities at high altitudes [3]. Secondly, cache-enabled UAVs can be dynamically deployed and moved to deliver the requested files to the desired users, thereby improving caching efficiency [4]. However, the operation time of UAVs is constrained by its limited battery capacity. Therefore, how to design an efficient UAV trajectory to achieve a high overall content delivery performance is a critical issue.

In this paper, we consider an aerial-terrestrial wireless caching network in which popular contents can be cached at UAVs and ground mobile users. In this case, to utilize the channel and storage resources efficiently, the trajectory design should consider both the movement of the ground user and the behavior of other UAVs. This makes the problem of finding optimal trajectories more complex and challenging. In short, two key problems are addressed in this paper: 1) how to design a cooperative moving strategy for multiple cache-enabled UAVs taking into account the user mobility; and 2) how to improve network throughput by efficiently allocating the cache storage capacity at the UAVs.

### A. Related Work

Deploying cache-enabled UAVs in the presence of terrestrial networks has been discussed in [5]–[12]. In [5], the authors considered the joint caching and resource allocation of cache-enabled UAVs that can serve ground users over licensed and unlicensed bands. An effective UAV spectrum allocation scheme was proposed to allocate appropriate bandwidth with the objective to maximize the number of stable queue users. The authors of [6] investigated the UAV-assisted content caching and

Manuscript received July 1, 2020; revised October 25, 2020 and March 20, 2021; accepted June 29, 2021. Date of publication July 2, 2021; date of current version August 13, 2021. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2218-E-008 -016 -MY2. The review of this paper was coordinated by Dr. Berk Canberk. *(Corresponding author: Yu-Jia Chen.)*

Yu-Jia Chen, Kai-Min Liao, Meng-Lin Ku, and Guan-Yi Chen are with the Department of Communication Engineering, National Central University Taoyuan City 320, Taiwan (e-mail: yjchen@ce.ncu.edu.tw; lkiamin@gmail.com; mlku@ce.ncu.edu.tw; billchenyi0531@gmail.com).

Fung Po Tso is with the Department of Computer Science, Loughborough University, U.K. (e-mail: p.tso@lboro.ac.uk).

Digital Object Identifier 10.1109/TVT.2021.3094273

transmission problems for the wireless virtual reality networks, whose goal is to find the optimal content and cache storage capacity at the ground BSs. To maximize the quality-of-experience (QoE) of ground users, a machine learning (ML) framework was proposed to determine the UAVs' position and optimal cache contents at the UAV [7]. A blockchain-based approach was proposed to solve the node failure and network connectivity problem for maintaining the reliability requirements of a drone-caching network [8]. Some studies on cache-enabled wireless networks focused on the design of secure transmission schemes by adjusting the UAV trajectories [9] and performing interference management [10]. In addition, there exist works focusing on utilizing UAVs to assist terrestrial D2D networks. In [11], a UAV transmission resource allocation problem was considered in which the UAV acts as a carrier to transfer energy to the D2D pairs. The authors of [12] considered a UAV enabled caching in which contents can be transferred via a terrestrial BS or a UAV. However, to our best knowledge, the design of aerial-terrestrial wireless caching network in which contents can be cached at both UAVs and terrestrial D2D users has not been explored in the existing literature.

Furthermore, the trajectory design or placement problem for optimizing the performance of the UAV-assisted network has received tremendous attention under different setups, such as coordinate multipoint (CoMP) architectures [13], UAV-enabled multicasting systems [14], and UAV sensing systems [15]. To fully exploit the benefit of the UAV, several works have studied the trajectory design taking into account the user mobility [16], [17]. In these works, the UAV trajectory was designed by assuming that the positions of ground users are known [16] or predictable [17] in a given period. However, in practice the users may move randomly and independently, resulting in unpredictable mobility patterns. Moreover, the UAV trajectory design in the existing works was solved offline either in 2D space [16], [17] or by separately designing the altitude and horizontal location [18]. Different from these works, in this paper the 3D movement design for UAVs is adjusted in an online fashion taking into account the time dynamics of user positions.

To realize the highly maneuverable autonomous UAVs, ML-based solutions are desired for the UAV control without human intervention, which has been considered as a use case in the 3 rd Generation Partnership Project (3GPP) [19]. The K-means clustering algorithm was adopted to partition the ground users into K clusters, in which the UAV can be initially placed at the centroid of the cluster [20], [21]. For the online UAV trajectory design with mobile ground users, the Markov decision process (MDP) has been widely applied to model the UAV control decision problem, which can be solved by reinforcement learning (RL) techniques. In RL, an agent can learn the optimal policy by interacting with the unknown environment (e.g., user movements, channel variations). A joint K-means clustering and single RL-based algorithm for the multi-UAV deployment was proposed in [22]. However, the single RL requires a centralized controller with full knowledge of network information from each UAV, which is infeasible for highly dynamic aerial networks [23]. Compared to the single RL, multi-agent reinforcement learning (MARL) has been shown to provide a more effective learning performance, especially when only local information is available [24]. By considering the individual and application-specific information, MARL solves the sequential multi-agent decision making problem in distributed manners. The authors of [25] investigated a cellular network in which multiple UAVs transmit their sensory data to terrestrial nodes. By utilizing MARL, a multi-UAV trajectory design algorithm was developed, whose goal is to optimize the number of successful data transmissions. In [26], an MARL-based approach was proposed to solve the joint trajectory design and power control problem that aims to maximize the instantaneous transmit rate. In the above works, each UAV is regarded as an independent learning agent that conducts the standard single agent RL algorithm with no interaction between other UAVs. However, it is revealed that appropriate cooperation between UAVs can substantially improve the long-term performance [27]. Therefore, the cooperation and learning for the online trajectory design of multiple cache-enabled UAVs still require further investigation.

### B. Contribution

As discussed above, invoking MARL to UAV-assisted wireless networks offers a promising solution for intelligent UAV control and resource allocation. However, the research gap still exists in investigating the aerial-terrestrial wireless caching networks, which is worthy of further study. In this paper, we aim to develop an MARL framework for the continuous operation of multiple cache-enabled UAVs. Specifically, we consider a downlink wireless caching network that allows the coexistence of terrestrial D2D caching and aerial UAV-to-Device (U2D) caching using sub-6 GHz and millimeter wave (mmWave) spectrum bands, respectively. Due to higher data transmission rates in LoS as compared to sub-6 GHz, content delivery over the U2D links is prioritized first. We assume that each UAV can communicate with ground users and other UAVs in the absence of a central controller. Based on the proposed framework, the main contributions of this paper can be summarized as follows:

- We develop a novel framework for 3D UAV trajectory design in which multiple cache-enabled UAVs are deployed to serve ground users with an unpredictable mobility pattern. Meanwhile, we formulate the network throughput maximization problem by optimizing the UAVs' initial positions and their dynamic movements. We show that the formulated problem is NP-hard due to the coupled association constraint.
- We exploit the MARL technique to design the optimal trajectories of multiple UAVs in a distributed manner, in which each UAV acts as an agent and conducts a decision algorithm independently. Utilizing the local state measurements, the proposed online UAV control scheme autonomously adjusts the real-time UAV position without prior knowledge of content request statistics or channel models. Moreover, unlike the existing MARL-based trajectory design based on fully independent agents, in the proposed scheme agents can share the gained experiences within a certain proximity. By optimally selecting the size of cooperative region, the proposed algorithm can strike a balanced tradeoff between independent action selection in

TABLE I
LIST OF NOTATIONS

| Parameter | Description |
|---|---|
| $K$ | Number of UAVs |
| $\lambda_u$ | Density of users |
| $h_{k,t}$ | Altitude of the $k_{\rm th}$ UAV at the $t_{\rm th}$ time slot |
| $w_{k,t}$ | Horizontal position of the $k_{\rm th}$ UAV at the $t_{\rm th}$ time slot |
| $q_{u,t}$ | Horizontal position of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $v_{u,t}$ | Velocity of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $\theta_{u,t}$ | Moving direction of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $P_{\rm UAV}$, $P_{\rm D2D}$, $P_{\rm BS}$ | Transmit power of UAVs/users/ground BS |
| $\chi_{\sigma_{\rm LoS}}$, $\chi_{\sigma_{\rm NLoS}}$ | Shadowing random variable for LoS/NLoS |
| $l^{\rm LoS}_{u,k,t}$, $l^{\rm NLoS}_{u,k,t}$ | LoS/NLoS path loss from the $k_{\rm th}$ UAV to the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $d^u_{u,k,t}$ | Distance between the $k_{\rm th}$ UAV and the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $d^d_{u,m,t}$ | Distance between the $m_{\rm th}$ user and the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $d^c_{u,t}$ | Distance between the ground BS and the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $\Gamma^{\rm UAV}_{u,k,t}$, $\Gamma^{\rm D2D}_{u,m,t}$, $\Gamma^{\rm BS}_{u,t}$ | SNR of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $N$ | Total number of contents |
| $p_i$ | Request probability of the $i_{\rm th}$ content |
| $p^{\rm D2D}_{\rm hit}$, $p^{\rm UAV}_{\rm hit}$ | Cache hit probability of the $i_{\rm th}$ content |
| $\tilde{p}^{\rm UAV}_{u,k,t}$, $\tilde{p}^{\rm D2D}_{u,m,t}$, $\tilde{p}^{\rm BS}_{u,t}$ | Successful transmission probability of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot |
| $T_{\rm UAV}$, $T_{\rm D2D}$, $T_{\rm BS}$ | Total throughput of U2D/D2D/B2D links |
| $c_k$ | Centroid point of cluster $k$ |
| $s_k$ | State of the $k_{\rm th}$ UAV |
| $a_k$ | Action of the $k_{\rm th}$ UAV |
| $r_k$ | Reward of the $k_{\rm th}$ UAV |
| $Q_k(s,a)$ | Q-value function |

the existing MARL and joint action selection in the single RL.

- Simulation results confirm the effectiveness of integrating cache-enabled UAVs into the terrestrial D2D wireless caching network. Also, it is demonstrated that our proposed learning algorithm can significantly improve the network throughput as well as the U2D link utilization over other existing state-of-the-art solutions. Finally, we provide some valuable insights for the design of learning parameters (e.g., learning rate) and network parameters (e.g., cache storage capacity).

### C. Organization

The rest of the paper is organized as follows. In Section II, we introduce the system model. Section III presents the content delivery service and the problem formulation. In Section IV, we provide the RL-based trajectory design with independent agents. We propose the cooperative MARL-based trajectory design in Section V. Simulation results are given in Section VI. We conclude the paper in Section VII. Additionally, the list of notations is given in Table I.

## II. System Model

### A. Network Model

We consider an aerial-terrestrial wireless caching network, where $K$ UAVs equipped with cache storage are deployed to serve ground mobile users, as shown in Fig. 1. Let $\mathbf{U}$ and $\mathbf{K}$ be the sets of all users and UAVs, respectively. Besides, one ground BS is located at the center of the geographical area while all the UAVs are covered by the ground BS. We assume that the locations of mobile users are spatially distributed as a homogeneous Poisson point process (HPPP) $\Phi_u$ with density $\lambda_u$ [28]. The HPPP is widely used in the performance analysis of mobile networks due to its mathematical tractability [29].

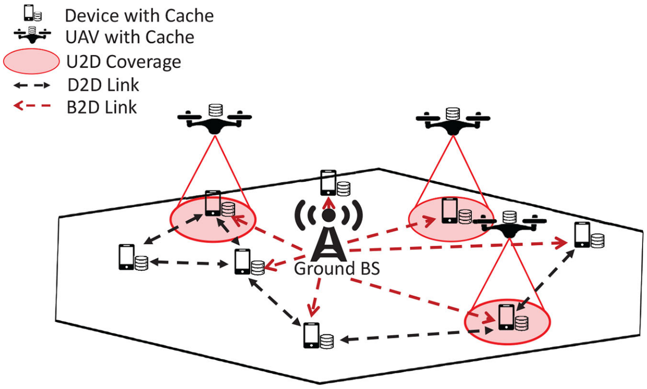

Fig. 1. Illustration of aerial-terrestrial wireless caching networks.

The user content requests are served via one of the following links: U2D links, D2D links, and BS-to-device (B2D) links. The UAVs operate at the mmWave band while the B2D and D2D links operate at the same sub-6 GHz band. We also assume that the UAVs are connected to the ground BS via high-speed backhaul links. We will discuss the link selection policy for the content delivery in Section III.

Let $h_{k,t} \in [h_{\min}, h_{\max}]$ denote the altitude of the $k_{\rm th}$ UAV at the $t_{\rm th}$ time slot, where $h_{\min}$ and $h_{\max}$ are the lowest and the highest altitudes for all the UAVs, respectively. The horizontal coordinate of the $k_{\rm th}$ UAV at the $t_{\rm th}$ time slot is denoted by $w_{k,t} = (x_{k,t}, y_{k,t})$ and the coordinate of the $u_{\rm th}$ user at the $t_{\rm th}$ time slot is denoted by $q_{u,t} = (r_{u,t}, s_{u,t})$. We consider a realistic Gauss Markov mobility model [30] for the movement of ground users. In the Gauss Markov mobility model, both the values of velocity and moving direction at time $t$ are calculated based on the values of velocity and moving direction at time $t-1$. Namely, the velocity and direction for the $u_{\rm th}$ user at the $t_{\rm th}$ time slot can be written as

$$v_{u,t} = \alpha v_{u,t-1} + (1-\alpha)\bar{v} + \sqrt{(1-\alpha^2)}\hat{v}_{t-1} \ , \tag{1}$$

$$\theta_{u,t} = \alpha \theta_{u,t-1} + (1-\alpha)\bar{\theta} + \sqrt{(1-\alpha^2)}\hat{\theta}_{t-1} \ , \tag{2}$$

where $\alpha \in [0, 1]$ reflects the degree of randomness in the mobility pattern [31]. Also, $\bar{v}$ and $\bar{\theta}$ denote the asymptotic mean value of velocity and direction for all users when $t$ approaches infinity, respectively. Parameters $\hat{v}$ and $\hat{\theta}$ are independent stationary Gaussian processes with zero mean and unit variance. In our system, we focus on the case where the mobility model incorporates randomness and memory by setting $0 < \alpha < 1$.

### B. Channel Model

Two different channel models are considered in our system, namely UAV-to-ground channel and ground-to-ground channel.

*1) UAV-to-Ground Channel Model:* Compared to the propagation of terrestrial communications, the UAV-to-ground channel is highly dependent on the altitude and the elevation angle. For the propagation model, we adopt the log-normal shadowing channel [7], in which LoS links and non-line-of-sight (NLoS) links can be modeled with corresponding channel parameters. The LoS and NLoS pathloss (in dB) from the $k_{\mathrm{th}}$ UAV to the $u_{\mathrm{th}}$ user at the $t_{\mathrm{th}}$ time slot are given by

$$l^{\mathrm{LoS}}_{u,k,t} = l_{\mathrm{FS}}(d_0) + 10\mu_{\mathrm{LoS}}\log(d_{u,k,t}) + \chi_{\sigma_{\mathrm{LoS}}} \ , \tag{3}$$

$$l^{\mathrm{NLoS}}_{u,k,t} = l_{\mathrm{FS}}(d_0) + 10\mu_{\mathrm{NLoS}}\log(d_{u,k,t}) + \chi_{\sigma_{\mathrm{NLoS}}} \ , \tag{4}$$

where $l_{\mathrm{FS}}(d_0) = 20\log(\frac{4\pi d_0 f_c}{c})$ is the free-space path loss at reference distance $d_0$ and $d^u_{u,k,t}$ is the distance between the $k_{\mathrm{th}}$ UAV and the $u_{\mathrm{th}}$ user at the $t_{\mathrm{th}}$ time slot, i.e.,

$$d^u_{u,k,t} = \sqrt{h^2_{k,t} + (x_{k,t} - r_{u,t})^2 + (y_{k,t} - s_{u,t})^2} \ . \tag{5}$$

Also, $f_c$ and $c$ are the carrier frequency and the speed of light, respectively. Here, $\mu_{\mathrm{LoS}}$ and $\mu_{\mathrm{NLoS}}$ are the large-scale path loss exponents for the LoS and NLoS links, respectively. $\chi_{\sigma_{\mathrm{LoS}}}$ and $\chi_{\sigma_{\mathrm{NLoS}}}$ represent the Gaussian random variables with zero mean for the LoS and NLoS links, respectively. $\sigma_{\mathrm{LoS}}$ and $\sigma_{\mathrm{NLoS}}$ are, respectively, the standard deviations for the LoS and NLoS links.

The LoS probability can be modeled as a logistic function of the elevation angle $\phi_{u,k,t}$ [32], i.e.,

$$\Pr(l^{\mathrm{LoS}}_{u,k,t}) = \frac{1}{1 + Xe^{-Y(\phi_{u,k,t} - X)}} \ , \tag{6}$$

where $X$ and $Y$ are environment-dependent parameters (e.g., urban or rural). Also, the elevation angle is given by $\phi_{u,k,t} = \sin^{-1}(\frac{h_{k,t}}{d_{u,k,t}})$. Therefore, the average path loss for the U2D links can be expressed as

$$\bar{l}_{u,k,t} = l^{\mathrm{LoS}}_{u,k,t} \times \Pr(l^{\mathrm{LoS}}_{u,k,t}) + l^{\mathrm{NLoS}}_{u,k,t} \times (1 - \Pr(l^{\mathrm{LoS}}_{u,k,t})) \ . \tag{7}$$

According to [33], the interference can be neglected if the distance is large enough in the mmWave UAV networks. The signal-to-noise ratio (SNR) for the U2D link from the $k_{\mathrm{th}}$ UAV to the $u_{\mathrm{th}}$ user is given by

$$\Gamma^{\mathrm{UAV}}_{u,k,t} = \frac{P_{\mathrm{UAV}}|g_{u,k,t}|^2}{10^{\bar{l}_{u,k,t}/10}\sigma^2} \ , \tag{8}$$

where $P_{\mathrm{UAV}}$ denotes the transmission power of a UAV and $\sigma^2$ denotes the power of additive white Gaussian noise (AWGN). In addition, $|g_{u,k,t}|^2$ denotes the small-scale fading gain, which follows a Nakagami-$m$ distribution to characterize a wide range of fading environments.

*2) Ground-to-Ground Channel Model:* For the terrestrial links, we consider the general power-law propagation model and the small-scale Rayleigh fading channel. We denote the channel gain of the D2D link between the $u_{\mathrm{th}}$ user and the $m_{\mathrm{th}}$ user at the $t_{\mathrm{th}}$ time slot as $|g^d_{u,m,t}|^2$. Besides, the channel gain of the B2D link between the $u_{\mathrm{th}}$ user and the ground BS at the $t_{\mathrm{th}}$ time slot is denoted as $|g^c_{u,t}|^2$. It is assumed that $|g^d_{u,m,t}|^2$ and $|g^c_{u,t}|^2$ are independent and identically distributed exponential random variables with mean $\mu_d$ and $\mu_c$ [34]. For the D2D links, the receivers experience both inter-D2D interference and cross-tier interference from the ground BS. Therefore, the signal-to-interference-plus-noise ratio (SINR) of the D2D link achieved by the $u_{\mathrm{th}}$ user from the $m_{\mathrm{th}}$ user at the $t_{\mathrm{th}}$ time slot can be expressed as

$$\Gamma^{\mathrm{D2D}}_{u,m,t} = \frac{P_{\mathrm{D2D}}\left|g^d_{u,m,t}\right|^2 {d^d_{u,m,t}}^{-\beta}}{\sigma^2 + \sum\limits_{j\in\Phi\backslash\{m\}} P_{\mathrm{D2D}}\left|g^d_{u,j,t}\right|^2 {d^d_{u,j,t}}^{-\beta} + P_{\mathrm{BS}}\left|g^c_{u,t}\right|^2 {d^c_{u,t}}^{-\beta}} \ , \tag{9}$$

where $\Phi$ denotes the set of D2D transmitters; $\beta$ represents the path loss exponent; $P_{\mathrm{D2D}}$ and $P_{\mathrm{BS}}$ denote the transmission power of the D2D user and the ground BS, respectively. Besides, $d^d_{u,m,t} = \sqrt{(r_{u,t} - r_{m,t})^2 + (s_{u,t} - s_{m,t})^2}$ is the distance between the $u_{\mathrm{th}}$ user and the $m_{\mathrm{th}}$ user; $d^c_{u,t} = \sqrt{{r_{u,t}}^2 + {s_{u,t}}^2}$ is the distance between the $u_{\mathrm{th}}$ user and the ground BS which is located at the origin.

On the other hand, for the B2D links, we ignore the interference from other D2D links since the transmit power of D2D users is much less than that of the ground BS. Hence, the SNR of the B2D link at the $u_{\mathrm{th}}$ user at the $t_{\mathrm{th}}$ time slot is given by

$$\Gamma^{\mathrm{BS}}_{u,t} = \frac{P_{\mathrm{BS}}\left|g^c_{u,t}\right|^2 {d^c_{u,t}}^{-\beta}}{\sigma^2} \ . \tag{10}$$

### C. Caching Model

Suppose that each ground user and UAV have finite cache storage capacity $M_q$ and $M_w = cM_q$ with integer $c > 1$, respectively. Also, we consider a requested content library $\mathbf{F} = \{f_1, f_2, \ldots, f_N\}$ that consists of $N$ equal-sized content files. It is assumed that the ground BS has the entire requested content library. We denote the content request probability of $f_i$ as $p_i$, which satisfies the Zipf law [35], [36]. The request probability of $f_i$ is given by

$$p_i = \frac{i^{-\kappa}}{\sum\limits_{j=1}^{N} j^{-\kappa}} \ , \tag{11}$$

where $\kappa$ is the popularity factor. A large $\kappa$ implies that the content files are concentrated distribution. On the contrary, a smaller $\kappa$ implies that the content request probability is more evenly distributed.

We adopt the geographic caching strategy [37] for both U2D and D2D links. In the U2D caching, each UAV stores content $f_i$ with probability $q^u_i \in \mathbf{q^u} = [q^u_1, \ldots, q^u_N]$, $i \in [1, N]$. Similarly, each D2D user stores content $f_i$ with probability $q^d_i \in \mathbf{q^d} = [q^d_1, \ldots, q^d_N]$, $i \in [1, N]$. Also, we have $\sum_{i=1}^{N} q^u_i \leq M_w$ and $\sum_{i=1}^{N} q^d_i \leq M_q$ due to the cache storage capacity constraints at UAVs and users, respectively [38].

The following cases are considered for calculating the cache hit probability, which is defined as the probability that a user can receive the requested file.

*1) Self-Request Cache Hit:* In this case, the requested file has been cached in its own local device. The cache hit probability can be written as

$$p_{\text{self}} = \sum_{i=1}^{N} p_i q_i^d \ . \tag{12}$$

*2) D2D Cache Hit:* The probability of having the requested file cached at a D2D user depends on the D2D user density $\lambda_u$ and the area size. According to [39], the cache hit probability of file $f_i$ within distance $R_d$ can be written as

$$p_{\text{hit}}^{\text{D2D}} = \sum_{i=1}^{N} p_i (1 - q_i^d)(1 - e^{-\lambda_u q_i^d \pi R_d^2}) \ . \tag{13}$$

*3) U2D Cache Hit:* Recall that the UAVs can adjust their positions according to the users' positions. Thus, for the infinite time horizon, the UAVs can also be considered to be spatially distributed as an HPPP. In principle, the UAV cache hit probability $p_{\text{hit}}^{\text{UAV}}$ depends on the average number of UAVs in which the user-UAV distance is within the UAV's transmission range $R_u$. Clearly, the actual value of $p_{\text{hit}}^{\text{UAV}}$ is affected by the UAV trajectory design. The upper bound of $p_{\text{hit}}^{\text{UAV}}$ is obtained when the requested user is within the transmission range of all the $K$ UAVs. Hence, we have

$$p_{\text{hit}}^{\text{UAV}} \leq \sum_{i=1}^{N} p_i (1 - q_i^d)(1 - e^{-K q_i^u \pi R_u^2}) \ . \tag{14}$$

## III. Content Delivery in Aerial-Terrestrial Wireless Caching Networks

### A. Content Delivery

Let the requested file of the $u_{\text{th}}$ user at the $t_{\text{th}}$ time slot be denoted as $\tau_t^u$. The sets of cached files at the $k_{\text{th}}$ UAV and the $m_{\text{th}}$ user are denoted by $\mathbf{C}_t^k$ and $\mathbf{D}_t^m$, respectively. The following actions will be performed when receiving a content request from a ground user.

- *Case 1:* The requested file is stored at the $k_{\text{th}}$ UAV located in the U2D transmission range, i.e.,

$$\exists k : (\tau_t^u \in \mathbf{C}_t^k) \wedge (d_{u,k,t}^u \leq R_u) \ , \tag{15}$$

  where $\wedge$ is the logical "and". The $u_{\text{th}}$ user can obtain $\tau_t^u$ from the $k_{\text{th}}$ UAV via the U2D link.
- *Case 2:* There does not exist a UAV caching $\tau_t^u$ within the U2D transmission range, i.e.,

$$\nexists k : (\tau_t^u \in \mathbf{C}_t^k) \wedge (d_{u,k,t}^u \leq R_u) \ . \tag{16}$$

  Then, there exist two possible sub-cases:
  - If the requested file $\tau_t^u$ is stored at the $m_{\text{th}}$ ground user within the D2D transmission range, i.e.,

$$\exists m : (\tau_t^u \in \mathbf{D}_t^m) \wedge (d_{u,m,t}^d \leq R_d) \ , \tag{17}$$

    the $u_{\text{th}}$ user receives $\tau_t^u$ from the $m_{\text{th}}$ user via the D2D link.
  - Otherwise, $\tau_t^u$ is delivered to the $u_{\text{th}}$ user by the ground BS.

Next, we analyze the successful transmission probability under different transmission modes.

*1) U2D Links:* If the required content is stored in one or more UAVs, the user is assumed to associate with the nearest UAV. Let $k_{f_i,u}$ denote the index of the nearest UAV caching the content $f_i$ requested by the $u_{\text{th}}$ user. To simplify the notation, we use $k$ instead of $k_{f_i,u}$ hereafter. Given the U2D SNR threshold $\eta_{\text{UAV}}$, the successful transmission probability is given by

$$\begin{aligned}\tilde{p}_{u,k,t}^{\text{UAV}} &= \Pr\left(\Gamma_{u,k,t}^{\text{UAV}} \geq \eta_{\text{UAV}}\right) \\ &= \Pr\left(|g_{u,k,t}|^2 \geq \frac{10^{\bar{l}_{u,k,t}/10}\sigma^2\eta_{\text{UAV}}}{P_{\text{UAV}}}\right) \\ &= 1 - E_{d_{u,k,t}^u}\left[\frac{\gamma(m, \frac{10^{\bar{l}_{u,k,t}/10}\sigma^2\eta_{\text{UAV}}}{b \cdot P_{\text{UAV}}})}{\Gamma(m)}\right] \ ,\end{aligned} \tag{18}$$

where $E_x[f(x)]$ denotes the expected value of $f(x)$ with respect to a random variable $x$, $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx$ is the Gamma function, and $\gamma(\alpha, \beta) = \int_0^\beta x^{\alpha-1} e^{-x} dx$ is the lower incomplete Gamma function. Also, $m$ and $b$ denote the shape parameter and scale parameter of the Gamma distribution, respectively.

For the Rayleigh fading channel, which can be obtained by setting $m = 1$ in the Nakagami fading channel, the channel gain $|g_{u,k,t}|^2$ follows the exponential distribution with unit mean. Since the UAVs are assumed to be distributed according to HPPP with parameter $Kq_i^u$, the probability density function (PDF) of the distance between the $u_{\text{th}}$ user and the $k_{\text{th}}$ UAV is given by $f_{d_{u,k,t}^u}(r) = 2\pi q_i^u K r \exp(-\pi r^2 q_i^u K)$ [35]. Then, we can rewrite (18) as

$$\begin{aligned}\tilde{p}_{u,k,t}^{\text{UAV}} &= \int_0^\infty f_{d_{u,k,t}^u}(r) q_{u,k,t}(\mathbf{q^u}, r) \mathrm{d}r \\ &= 2\pi q_i^u K \int_0^\infty r \exp\left(-\frac{10^{\bar{l}_{u,k,t}/10}\sigma^2\eta_{\text{UAV}}}{P_{\text{UAV}}} - \pi r^2 q_i^u K\right) \mathrm{d}r \ ,\end{aligned} \tag{19}$$

where $q_{u,k,t}(\mathbf{q^u}, r)$ is the successful transmission probability conditioned on $d_{u,k,t}^u = r$.

*2) D2D Links:* Let $\Phi_i$ and $\Phi_{-i}$ denote the sets of D2D users with and without content $f_i$ requested by the $u_{\text{th}}$ user, respectively. Considering the interference from the other D2D links and the ground BS, we rewrite $\Gamma_{u,m,t}^{\text{D2D}} = |g_{u,m,t}^d|^2 {d_{u,m,t}^d}^{-\beta} / (\sigma^2 + I_i + I_{-i} + I_{\text{BS}})$, where $I_i = \sum_{j \in \Phi_i \setminus \{m\}} P_{\text{D2D}} |g_{u,j,t}^d|^2 {d_{u,j,t}^d}^{-\beta}$ and $I_{-i} = \sum_{j \in \Phi_{-i} \setminus \{m\}} P_{\text{D2D}} |g_{u,j,t}^d|^2 {d_{u,j,t}^d}^{-\beta}$ denote the interference from other D2D users with and without content $f_i$, respectively. Also, $I_{\text{BS}} = P_{\text{BS}} |g_{u,t}^c|^2 {d_{u,t}^c}^{-\beta}$ represents the interference from the ground BS. Then, the successful transmission probability of D2D links conditioned on $d_{u,m,t}^d = r$ is given by

$$\begin{aligned}q_{u,m,t}(\mathbf{q^d}, r) &= \mathcal{L}_{I_i}(s, r)|_{s = r^\beta \eta_{\text{D2D}}} \mathcal{L}_{I_{-i}}(s, r)|_{s = r^\beta \eta_{\text{D2D}}} \\ &\times \exp\left(-\frac{\eta_{\text{D2D}} r^\beta \sigma^2}{P_{\text{D2D}}}\right) \exp\left(-\frac{P_{\text{BS}} \eta_{\text{D2D}} {d_{u,t}^c}^{-\beta} |g_{u,t}^c|^2 r^\beta}{P_{\text{D2D}}}\right),\end{aligned} \tag{20}$$

where $\mathcal{L}_{I_i}(s, r)|_{s = r^\beta \eta_{\text{D2D}}} = E[\exp(-sI_i)]$ denotes the Laplace transform of $I_i$ and $\eta_{\text{D2D}}$ is the SINR threshold for D2D links.

According to [40], we can have $\mathcal{L}_{I_i}(s,r)|_{s=r^{\beta}\eta_{\text{D2D}}}$ given by

$$\mathcal{L}_{I_i}(s,r)|_{s=r^{\beta}\eta_{\text{D2D}}} = \exp\Bigg( -\frac{2\pi}{\beta}\eta_{\text{D2D}}{}^{\frac{2}{\beta}}\lambda_{\text{D2D}} \times q_i^d B'\Big(\frac{2}{\beta}, 1-\frac{2}{\beta}, \frac{1}{\eta_{\text{D2D}}+1}\Big)\Bigg) \ , \quad (21)$$

where $B'(x,y,z) \triangleq \int_z^1 u^{(x-1)}(1-u)^{(y-1)}\mathrm{d}u$ is the complementary incomplete Beta function. Similarly, we have

$$\mathcal{L}_{-I_i}(s,r)|_{s=r^{\beta}\eta_{\text{D2D}}} = \exp\Bigg( -\frac{2\pi}{\beta}\eta_{\text{D2D}}{}^{\frac{2}{\beta}}\lambda_{\text{D2D}} \times (1-q_i^d)B\Big(\frac{2}{\beta}, 1-\frac{2}{\beta}\Big)\Bigg) \ , \quad (22)$$

where $B(x,y) \triangleq \int_0^1 u^{(x-1)}(1-u)^{(y-1)}\mathrm{d}u$ denotes the Beta function. Since the D2D users form an HPPP with parameter $q_i^d\lambda_{\text{D2D}}$, the PDF of $d_{u,m,t}^d$ can be expressed as $f_{d_{u,m,t}^d}(r) = 2\pi q_i^d\lambda_{\text{D2D}} r\exp(-\pi q_i^d \lambda_{\text{D2D}} r^2)$ [35]. Then, the successful transmission probability of the D2D links can be obtained as

$$\begin{aligned}\tilde{p}_{u,m,t}^{\text{D2D}} &= \Pr(\Gamma_{u,m,t}^{\text{D2D}} \geq \eta_{\text{D2D}}) \\ &= 2\pi q_i^d\lambda_{\text{D2D}}\int_0^{\infty} r\cdot\exp\Bigg( -\frac{2\pi}{\beta}\eta_{\text{D2D}}{}^{\frac{2}{\beta}}\lambda_{\text{D2D}} \\ &\times\left(q_i^d B'\left(\frac{2}{\beta}, 1-\frac{2}{\beta}, \frac{1}{\eta_{\text{D2D}}+1}\right) + (1-q_i^d)B\left(\frac{2}{\beta}, 1-\frac{2}{\beta}\right)\right) \\ &-\frac{\eta_{\text{D2D}}r^{\beta}\sigma^2}{P_{\text{D2D}}} - \frac{P_{\text{BS}}\eta_{\text{D2D}}r^{\beta}d_{u,t}^{c}{}^{-\beta}}{P_{\text{D2D}}} - \pi q_i^d\lambda_{\text{D2D}}r^2\Bigg)\mathrm{d}r \ . \quad (23)\end{aligned}$$

*3) B2D Links:* Recall that the ground BS has the entire requested content library. The successful transmission probability of the B2D link can be obtained from (19) by setting $K = 1$ and $q_i^u = 1$, i.e.,

$$\begin{aligned}\tilde{p}_{u,t}^{\text{BS}} &= \Pr(\Gamma_{u,t}^{\text{BS}} \geq \eta_{\text{BS}}) \\ &= 2\pi\int_0^{\infty} r\exp\left( -\frac{\eta_{\text{BS}}\sigma^2 r^{\beta}}{P_{\text{BS}}} - \pi r^2\right)\mathrm{d}r \ , \quad (24)\end{aligned}$$

where $\eta_{\text{BS}}$ is the SNR threshold for B2D links.

With the expressions of the successful transmission probabilities reported above, in the following subsection we define the performance metric for the considered aerial-terrestrial wireless caching network.

### B. Problem Formulation

As discussed previously, we assume that the U2D links have the highest priority in order to best utilize the mmWave spectrum resource. The throughput of the U2D link can be written as

$$T_{\text{UAV}} = p_{\text{hit}}^{\text{UAV}}\sum_{\forall u\in\mathbf{U}}\left(\tilde{p}_{u,k,t}^{\text{UAV}}\cdot\frac{B_{\text{UAV}}}{N_k}\log_2(1+\Gamma_{u,k,t}^{\text{UAV}})\right) \ , \quad (25)$$

where $B_{\text{UAV}}$ is the system bandwidth of UAV transmission and $N_k$ is the number of users receiving data from all the UAVs.

If the content request is fulfilled by the D2D links, the throughput of the D2D links can be written as

$$T_{\text{D2D}} = (1-p_{\text{hit}}^{\text{UAV}})p_{\text{hit}}^{\text{D2D}} \times\sum_{\forall u\in\mathbf{U}}\left(\tilde{p}_{u,m,t}^{\text{D2D}}\cdot\frac{B_{\text{D2D}}}{N_u}\log_2(1+\Gamma_{u,m,t}^{\text{D2D}})\right) \ , \quad (26)$$

where $B_{\text{D2D}}$ is the system bandwidth of D2D transmission and $N_u$ is the number of users receiving data through D2D links.

If both the U2D and the D2D links cannot provide the requested content, the user acquires the content from the ground BS. The throughput of the B2D links can be written as

$$T_{\text{BS}} = (1-p_{\text{hit}}^{\text{UAV}})(1-p_{\text{hit}}^{\text{D2D}}) \times\sum_{\forall u\in\mathbf{U}}\left(\tilde{p}_{u,t}^{\text{BS}}\cdot\frac{B_{\text{BS}}}{N_b}\log_2(1+\Gamma_{u,t}^{\text{BS}})\right) \ , \quad (27)$$

where $B_{\text{BS}}$ is the system bandwidth of cellular transmissions and $N_b$ is the number of users receiving data from the ground BS. Then, the total network throughput can be calculated as the sum of throughput obtained from all the links, i.e.,

$$T_{\text{total}} = T_{\text{UAV}} + T_{\text{D2D}} + T_{\text{BS}} \ . \quad (28)$$

From (25) and (28), it can be observed that the network throughput is substantially affected by the positions of UAVs. More specifically, the positions of UAVs not only affect the U2D successful transmission probability but also the UAV cache hit probability. It is worth noting that (14) only gives the upper bound of the UAV cache hit probability. The real value of the UAV cache hit probability depends on the real-time positions of all the UAVs and all the ground users. Thus, the UAV trajectory optimization problem is formulated as

$$\max_{x_{k,t},y_{k,t},h_{k,t}} T_{\text{total}} \quad (29a)$$

$$s.t. \quad x_{\min} \leq x_{k,t} \leq x_{\max}, \ \forall t, \forall k, \quad (29b)$$

$$y_{\min} \leq y_{k,t} \leq y_{\max}, \ \forall t, \forall k, \quad (29c)$$

$$h_{\min} \leq h_{k,t} \leq h_{\max}, \ \forall t, \forall k, \quad (29d)$$

$$\sqrt{\dot{x}_{k,t}^2 + \dot{y}_{k,t}^2 + \dot{h}_{k,t}^2} \leq v_{\max}, \ \forall t, \forall k, \quad (29e)$$

$$\sum_{k=1}^{K}\rho_{u,k} \leq 1, \ \forall u, \quad (29f)$$

$$\rho_{u,k} \in \{1, 0\}, \ \forall u, \forall k, \quad (29g)$$

where (29b) and (29c) indicate the constraints of the limited area, with $[x_{\min}, x_{\max}]$ and $[y_{\min}, y_{\max}]$ representing the x-axis and y-axis movement range of the UAVs, respectively; (29d) is the altitude constraint for the UAVs. Given the maximum UAV velocity $v_{\max}$, (29e) represents the flight velocity constraint, in which $\dot{x}_{k,t}$, $\dot{y}_{k,t}$, and $\dot{h}_{k,t}$ denote the first derivatives of $x_{k,t}$, $y_{k,t}$, and $h_{k,t}$, with respect to $t$, respectively. Furthermore, (29f) and (29g) are the association constraints to ensure that each user is only served by one UAV at most, with $\rho_{u,k}$ denoting the binary association indicator of the $u_{\text{th}}$ user and the $k_{\text{th}}$ UAV.

Note that problem (29a) is a non-convex mixed integer programming problem because of the complicated mathematical expression of the successful transmission probability and the integer association constraint in (29f). Such problems are generally difficult to obtain the solution in polynomial computational complexity [41]. In the following sections, we propose an MARL-based method for optimizing the UAV trajectory strategy.

## IV. Reinforcement Learning for 3D UAV Trajectory Design With Independent Agents

Generally, the proposed RL framework for UAV trajectory design consists of two phases. In the first phase, initial UAVs positions are calculated by the K-means clustering algorithm. In the second phase, the UAVs positions are dynamically adjusted based on the Q-learning algorithm to achieve the maximum network throughput.

### *A. UAV Trajectory Initialization Based on K-Means Clustering*

From (8), the distance between the UAVs and the users plays a key role in network performance. To minimize the distance between the UAVs and the users, we apply the K-means clustering algorithm to divide all the users into $K$ clusters and find the initial UAVs positions. As an unsupervised learning method, K-means clustering algorithm can be employed to cluster the objective nodes into several groups and find the centroid of each group, which is widely adopted in recent works for UAV trajectory design [22]. We denote the cluster set of all the users as $\mathbf{S} = \{S_1, S_2, \ldots, S_K\}$. The objective of the K-means clustering algorithm is to minimize the overall squared distance between each user and its nearest centroid, which is given by

$$\min_{\mathbf{S}} \sum_{k=1}^{K} \sum_{q_u \in S_j} ||q_u - c_k||^2 \ , \tag{30}$$

where

$$c_k = \frac{1}{|S_k|} \sum_{q_u \in S_k} q_u, \ \ \forall k \tag{31}$$

is the centroid point of cluster $k$. We select $c_k$ from the final results of the K-means clustering algorithm to be the initial UAVs positions.

### *B. Independent Q-Learning Based Trajectory Design*

Although the K-means clustering algorithm can provide the initial UAVs positions, it has the following limitations. First, the K-means clustering algorithm may converge to a local minimum and thus becomes inefficient when the network consists of a large number of mobile nodes [22]. Second, the clustering-based approach fails to take into account other key factors, such as the UAV cache hit probability and the successful transmission probability. To deal with these limitations, we propose to adopt the Q-learning algorithm in the second phase of our proposed method. Hereafter, the "UAV" and "agent" are used interchangeably.

The UAV trajectory design problem can be formulated as a MDP, which is composed of five tuples $(S, A, R, P, \gamma)$, where $S = \{s_1, \ldots, s_I\}$ denotes a finite set of states, $A = \{a_1, \ldots, a_J\}$ denotes a finite set of actions, $R = \{r(s_i, a_j) | s_i \in S, a_j \in A\}$ is the set of immediate reward $r(s_i, a_j)$ when action $a_j$ is selected while in state $s_i$, $P = \{p(s,' s, a_j) | s, s' \in S, a_j \in A\}$ is the transition probability for an agent that moves from the state $s$ to the state $s'$ when performing action $a_j$, and $\gamma \in (0, 1]$ is the discount factor. Next, we show how the MDP can be solved using independent Q-learning. In the conventional Q-learning algorithm, the UAVs can be regarded as independent agents that learn the optimal action independently. Hereafter, we refer to this kind of Q-learning method as *independent Q-learning*.

To learn the optimal action, the first step is to let the agent recognize the current system state through interacting with the environment. In particular, the interaction experience is represented by a Q-value, which is an estimate of the expected reward for performing a certain action at a certain state. The Q-values are usually stored in a look-up table called Q-table for reusing the gained knowledge. In the learning phase, the Q-learning algorithm iteratively updates the Q-values by the following rule:

$$Q_k^*(s, a) \leftarrow (1 - \mu) Q_k(s, a) + \mu [r_k(s, a) + \gamma \max_{a'} Q_k(s,' a')], \tag{32}$$

where $Q_k(s, a)$ and $Q_k^*(s, a)$ are the old and new Q-values, respectively, for the $k_{\text{th}}$ agent performing action $a$ at state $s$. Here, $r_k(s, a)$ denotes the reward for the $k_{\text{th}}$ agent and $\mu$ is the learning rate which determines how much the new observation overwrites the old one. Also, $\gamma$ is the discount factor, which represents the impact degree of the future reward on the current decision.

Next, we describe the MDP design for the proposed trajectory design using the Q-learning method.

*1) State:* We consider the 3D position of the UAV to be the state in our system. Hence, the state of the $k_{\text{th}}$ UAV is denoted by $s_k = (x_k, y_k, h_k)$. To simplify the learning process, we adopt a discrete state-space approximation in which the state space is quantized into discrete regions.

*2) Action:* $a_k = (d_k^a, l_k^a)$ denotes the action of the $k_{\text{th}}$ UAV, which consists of the direction $d_k^a$ and the moving distance $l_k^a$ of the UAV. We consider that, at any time slot, the UAV can fly to an adjacent grid in one of the six directions: up, down, right, left, forward, and backward. Also, the action space for the moving distance at each time slot is discretized into $z$ levels associated with corresponding flight velocities ranging from $[0, v_{\max}]$.

*3) Reward:* In this work, the reward can be interpreted as how the action affects the total network throughput $T_{\text{total}}$. The reward function is designed in a way that encourages the UAV to take actions that maximize its own throughput. At a time slot, if the SNR for the U2D link is larger than a minimum SNR threshold, the UAV receives a reward $r_k$, which is defined as the throughput provided by the $k_{\text{th}}$ UAV. Otherwise, the UAV receives a zero reward. Hence, the reward function $r_k$ can be expressed as

$$r_k = \begin{cases} \sum_{\forall u \in \mathbf{E}_k} \frac{B_{\text{UAV}}}{N_k} \log_2(1+\Gamma_{u,k}^{\text{UAV}}), & \text{if } \Gamma_{u,k}^{\text{UAV}} \geq \eta_{\text{UAV}}, \\ 0, & \text{otherwise}, \end{cases} \tag{33}$$

where $\mathbf{E}_k = \{u_i \in \mathbf{U} | \tau^{u_i} \in \mathbf{C}^k, d_{u_i,k}^{u_i} \leq R_{u_i}\}$ is the set of users who are located within the transmission range of the $k_{\text{th}}$ UAV which caches the requested file $\tau^{u_i}$.

Additionally, we adopt the $\varepsilon$-greedy policy [42] for the action selection in the Q-learning algorithm. More specifically, the UAV takes the action that has the largest Q-value with a probability $1-\varepsilon$ for a given current state while randomly selecting other actions with a probability $\frac{\varepsilon}{|A|-1}$.

## V. Cooperative Multi-Agent Reinforcement Learning for 3D UAV Trajectory Design

### A. Overview

Independent Q-learning methods present inefficiency in terms of learning speed and effectiveness when they are adopted in multi-UAV systems. Since each of the UAVs learns the policy only from its own experience, the learned knowledge of a single UAV (e.g., channel environment and user mobility) cannot be reused by other UAVs. It is time wasted for all the UAVs to explore the unknown environment when some of them are near to each other.

To deal with this problem, we propose a distributed cooperative mechanism for the learning process of Q-learning. In the proposed mechanism, UAVs within the same cooperative region (defined formally later) work together to achieve a common goal by sharing the past experience. More specifically, these coordinated UAVs exchange the obtained reward and update the shared Q-table. Hence, unnecessary learning processes can be reduced by reusing the past experiences learned in the same cooperative region.

Figure 2 shows illustrations of different Q-learning methods for multi-UAV systems. In the single Q-learning, the action selection is performed in a centralized manner in which the ground BS maintains a global Q-table. On the other hand, in the conventional MARL, each UAV distributedly learns its own policy based on its own Q-table. Unlike the above two independent Q-learning methods, the proposed cooperative MARL allows UAVs to select actions in a coordinated fashion by updating the shared Q-table. Hence, the proposed algorithm is executed on both the ground BS and the UAVs.

### B. Algorithm Design

Now we present the proposed cooperative MARL (CMARL) based solution for multi-UAV trajectory design in the aerial-terrestrial wireless caching network. Each UAV learns an optimal policy for controlling and optimizing its trajectory from its own experience as well as from other nearby UAVs within the same region.

The whole observed area is divided into $N_A$ equal-size cooperative regions. We denote the cooperative region number at which the $k_{\text{th}}$ UAV is located as $c_k^r \in \{1,\ldots,N_A\}, \forall k \in \mathbf{K}$. For

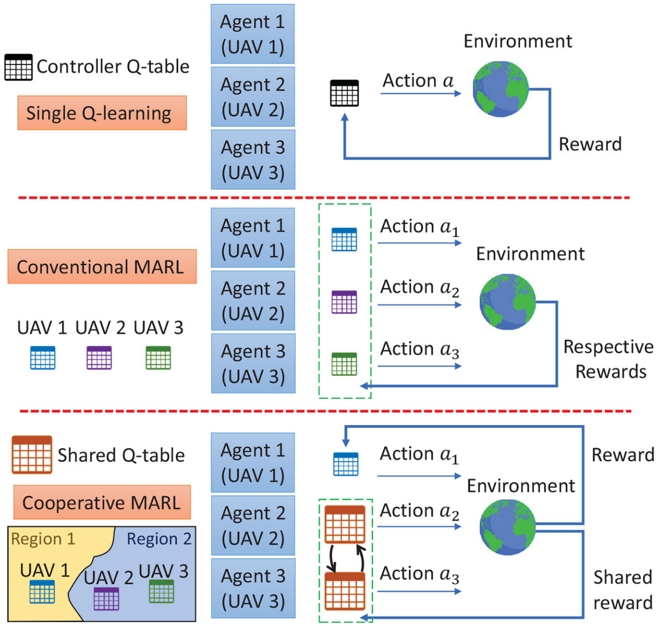

Fig. 2. Illustrations of three Q-learning methods for multi-UAV systems with $K = 3$.

a target UAV, if there are other UAVs located in the same cooperative region as the target UAV, these UAVs are considered to be in the same coordinated state. UAVs in the same coordinated state perform action selection based on a shared Q-table. Otherwise, if no UAV is located in the same cooperative region as the target UAV, the target UAV is said to be in the uncoordinated state and use its own Q-table. Therefore, we have the following cases for the Q-table updating:

- *Uncoordinated state to uncoordinated state:* In this case, each UAV performs flight decisions independently according to its own Q-table. The action $a_k$ of the $k_{\text{th}}$ UAV is only affected by its own reward $r_k$. In fact, this case is equivalent to that in the conventional MARL. Therefore, the update rule is the same as (32).
- *Coordinated state to coordinated state:* When the UAVs move from a coordinated state to another coordinated state, the optimal action is selected based on the shared Q-table. The update rule in this case can be written as

$$\begin{aligned} Q^*(s_k, a_k) &\leftarrow (1-\mu)Q(s_k, a_k) \\ &\quad + \mu[r(s_k, a_k) + \gamma \max_{a'_k} Q(s'_k, a'_k)] \end{aligned}, \tag{34}$$

  where $Q(s_k, a_k)$ and $Q^*(s_k, a_k)$ are the old and new Q-values, respectively. Also, $r(s_k, a_k)$ represents the obtained shared reward when executing the action $a_k$ at the state $s_k$, which is calculated as the average reward over all cooperative UAVs. That is, $r(s_k, a_k) = \sum_{j \in \mathbf{G}_k} \frac{1}{|\mathbf{G}_k|} \cdot r_j(s_k, a_k)$, where $\mathbf{G}_k = \{j \in \mathbf{K} | c_j^r = c_k^r\}$ denotes the set of UAVs within the cooperative region $c_k^r$.
- *Coordinated state to uncoordinated state:* If the UAV moves from a coordinated state to an uncoordinated state, the new Q-value should take into account the policy from its

own Q-table. Specifically, both the immediate reward and the expected discounted reward stored in its own Q-table are incorporated into the update equation, i.e.,

$$Q^*(s_k, a_k) \leftarrow (1-\mu)Q(s_k, a_k) + \mu[r_k(s,a) + \gamma \max_{a'} Q_k(s,' a')] \ . \tag{35}$$

- *Uncoordinated state to coordinated state:* Last, in case that the UAV moves from an uncoordinated state to a coordinated state, the update equation incorporates the immediate shared reward and the expected discounted reward stored in the shared Q-table. Thus, we have

$$Q_k^*(s, a) \leftarrow (1-\mu)Q_k(s, a) + \mu[r(s_k, a_k) + \gamma \max_{a_k'} Q(s_k', a_k')] \ . \tag{36}$$

The details of the proposed CMARL based 3D UAV trajectory design algorithm are summarized in Algorithm 1. Lines 2-6 initialize the positions of UAVs by using the K-means algorithm. In lines 9-19, the Q-table is updated using the aforementioned cooperative mechanism. Lines 23-29 determine the optimal action and the corresponding 3D UAV position.

It is worth pointing out that state discretization is employed by using the grid approach in order to obtain an appropriate size of state space. To deal with the problem induced by large state space, the proposed method can be extended by simply replacing the Q-table with deep neural network [42]. Moreover, in this paper we assume a perfect communication link between the ground BS and the UAVs. However, in realistic environments, UAVs may fly out of the wireless coverage of the ground BS [3]. In this case, more complex Q-value updating mechanisms are required for ensuring the learning stability, which will be our future work.

**Algorithm 1:** Proposed CMARL Based 3D UAV Trajectory Design Algorithm.

```
Input: Number of UAVs $K$; number of cooperative regions $N_A$.
Output: UAV's horizontal position $w_k$, UAV altitude $h_k$.
1:  Initialize $Q_k(s,a) = 0$ and $Q(s_k,a_k) = 0, \forall s \in S$, $\forall s_k \in S, \forall a \in A, \forall a_k \in A$.
2:  Initialize $K$ centroids at random.
3:  Divide all the users into $K$ clusters according to (30).
4:  for each UAV agent $k$ do
5:      Obtain the initial position of the $k_{\mathrm{th}}$ UAV according to (31).
6:  end for
7:  for each step of episode do
8:      for each UAV agent $k$ do
9:          Select action $a_k$ based on $\varepsilon$-greedy policy.
10:         Observe the state $s_k'$ and receive the reward $r_k$.
11:         if Uncoordinated $\rightarrow$ uncoordinated then
12:             Update its own Q-table from (32).
13:         else if Coordinated $\rightarrow$ coordinated then
14:       Update the shared Q-table from (34).
15:     else if Coordinated $\rightarrow$ uncoordinated then
16:       Update the shared Q-table from (35).
17:     else
18:       Update its own Q-table from (36).
19:     end if
20:   end for
21: end for
22: for each UAV agent $k$ do
23:     if In a coordinated state then
24:         Select $\hat{a}_k = \arg\max_{a_k} Q(s_k, a_k)$ as the optimal action for a given state $s_k$.
25:         Obtain $w_k$ and $h_k$ from $\hat{a}_k$.
26:     else
27:         Select $\hat{a} = \arg\max_a Q_k(s,a)$ as the optimal action for a given state $s$.
28:         Obtain $w_k$ and $h_k$ from $\hat{a}$.
29:     end if
30: end for
31: return $w_k$, $h_k$;
```

### *C. Analysis of the Proposed CMARL Based Algorithm*

The feasibility of utilizing on-device RL for UAV flight control has been demonstrated in [23], [25]. Also, reducing power consumption is one of the major challenges in UAV networks. According to [12], the energy cost of the UAV communication link is proportional to the transmission data size. We note that the proposed method requires message exchange between the UAV and the ground BS when updating the shared Q-table. Clearly, the extra transmission overhead for updating the Q-value of a single state-action pair is relatively small compared to the content data transmission and thereby the related energy cost can be neglected.

*1) Convergence Analysis:* The convergence analysis can be divided into two cases based on the state transition.

- *Non-cooperative Case:* In this case, the UAV moves from an uncoordinated state to an uncoordinated state. Recall that the agent in this case only considers its own reward during the update of Q-value. Therefore, this case can be regarded as non-cooperative MARL in which multiple agents execute the single Q-learning independently. For the convergence of the non-cooperative MARL, it has been proved in [43] that the non-cooperative MARL can converge to the optimal policy $\cup_{k=1}^{K} \pi_k^*(s_k)$ under the following conditions.
  (a) $\pi_k^*(s_k) = \hat{\pi}_k(s_k), \forall k$
  (b) $\nexists(\mathbf{a}^\dagger, a_k^*)|a_k^* = \pi_k^*(s_k)$ and $\forall \mathbf{a}^*|a_k^* \in \mathbf{a}^*$, $Q_k^*(s_k, \mathbf{a}^\dagger) > Q_k^*(s_k, \mathbf{a}^*)$
  (c) $\nexists(\mathbf{a^1}, \mathbf{a^2} \neq \mathbf{a^1})|\forall(k, j \in \{1,2\}, \mathbf{a})$, $Q_k^*(s_k, \mathbf{a}^j) \geq Q_k^*(s_k, \mathbf{a})$
  Here, $\pi_k^*(s_k)$ and $\hat{\pi}_k(s_k)$ refer to the optimal policy from the global perspective and local perspective, respectively. If condition (a) is satisfied, the global optimal action $a_k^*$ of the $k_{\mathrm{th}}$ UAV is equal to the local optimal action $\hat{a}_k$ in which each UAV selects the optimal action independently. In addition, let $\mathbf{a}^*$ be the set of actions including $a_k^*$. Condition (b) states that there does not exist another action set $\mathbf{a}^\dagger$ such

that the Q-value with respect to $\mathbf{a}^\dagger$ is larger than that to $\mathbf{a}^*$. Last, from condition (c), we know that there must be at most one global optimal action that maximizes the Q-values of all the UAVs. That is, no coordination between the UAVs is required to achieve an optimal equilibrium.

- *Cooperative Case:* On the other hand, the UAV whose Q-value update involves the shared Q-table is classified as a cooperative case. By regarding UAVs as nodes, the interaction topology of information exchange between agents can be described as a graph, with each edge representing the interaction between two UAVs. Also, each edge is assigned a nonnegative weight indicating the strength of interaction. Let $(i, j)$ denote the edge between node $i$ and node $j$. It has been proved in [44] that the convergence of the cooperative MARL is guaranteed when the weight of the edge $(i, j)$, denoted by $w(i, j)$, is chosen by the Metropolis criterion [45], which is given by

$$w(i,j)= \{1 + \max[d(i), d(j)]\}^{-1}, \ \forall(i,j) \in \mathbf{H} \quad (37)$$

and

$$w(i,i) = 1 - \sum_{j \in N(i)} w(i,j), \ 1 \le i \le K' \ , \quad (38)$$

where $K'$ denotes the number of neighboring agents; $\mathbf{H}$ represents the set of edges; $N(i) = \{1 \le j \le K' : (i, j) \in \mathbf{H}\}$ denotes the set of neighboring agents of the $i_{\text{th}}$ node; $d(i) = |N(i)|$ is the degree of the $i_{\text{th}}$ node of the graph. Recall that, in our proposed CMARL, agents within the same cooperative region can communicate with each other. Hence, the proposed design is a particular case of the Metropolis weights where the communication graph for the agents within a cooperative region is always a fully connected graph with $d(i) = d(j) = K' - 1$ in (37) and (38).

*2) Complexity Analysis:* For the single Q-learning algorithm, the computational complexity is $O(T_Q)$, where $T_Q$ denotes the time required to converge to an optimal solution. Then, for the conventional MARL algorithm, due to having $K$ independent UAVs, the computational complexity is $O(KT_Q)$. In our proposed CMARL algorithm, additional complexity is required for each UAV to handle the shared Q-table. Then, the complexity for each UAV is $O(T_Q + K_c T_Q)$, where $K_c$ denotes the number of UAVs within a cooperative region. Clearly, $K_c$ depends on the actual environment and the total number of cooperative regions $N_A$. In the worst case, we have $K_c = K$ and thereby the total computational complexity of CMARL is $O((K^2 + K)T_Q)$. However, although the complexity for one UAV is linearly related to the UAV number in the network, it is worth noting that the UAVs are usually too far away from a UAV to affect its flight decision. According to our simulation results, which will be presented later, the optimal network performance can be achieved by selecting an appropriate cooperative region size such that $K_c \approx 1$. In this case, the complexity of the proposed CMARL is $O(KT_Q)$, which is the same as the conventional MARL algorithm.

Table II compares the execution time of different RL algorithms using the Gauss-Markov mobility model and the real users' mobility traces from [46]. The execution time is measured by over 500 iterations. It is observed that the execution time of the proposed CMARL is close to that of the conventional MARL, which is consistent with our complexity analysis.

TABLE II
COMPLEXITY AND EXECUTION TIME OF DIFFERENT RL ALGORITHMS

| Algorithm | Single RL | MARL | CMARL |
|---|---|---|---|
| Complexity | $O(T_Q)$ | $O(KT_Q)$ | $O(KT_Q)$ |
| Execution time using real traces [46] | 602.5 s | 1901.6 s | 1524.2 s |
| Execution time using mobility model | 589.2 s | 1887.7 s | 1531.0 s |

TABLE III
SIMULATION PARAMETERS

| Notation | Description | Value |
|---|---|---|
| $K$ | Number of UAVs | 5 |
| $\lambda_u$ | User density | 0.3/m$^2$ |
| $\alpha$ | Randomness of mobility pattern | 0.6 |
| $h_{k,t}$ | UAV operating altitude | 60 m - 80 m [48] |
| $v_{\max}$ | UAV maximum velocity | 20 m/s |
| $f_{\text{UAV}}$ | Carrier frequency for U2D links | 28 GHz [49] |
| $B_{\text{UAV}}$ | System bandwidth of U2D links | 2.16 GHz [49] |
| $B_{\text{BS}}$ | System bandwidth of B2D links | 10 MHz |
| $B_{\text{D2D}}$ | System bandwidth of D2D links | 10 MHz |
| $P_{\text{UAV}}$ | Transmit power of UAV | 30 dBm |
| $P_{\text{BS}}$ | Transmit power of ground BS | 40 dBm [50] |
| $P_{\text{D2D}}$ | Transmit power of D2D user | 23 dBm |
| $\eta_{\text{UAV}}$ | U2D SNR threshold | 0 dB |
| $\chi_{\sigma_{\text{LoS}}}$, $\chi_{\sigma_{\text{NLoS}}}$ | Shadowing standard deviation | 5.3, 5.27 |
| $X$, $Y$ | Environment-dependent constant | 11.9, 0.13 |
| $\sigma^2$ | Noise power | -174 dBm/Hz |
| $\mu$ | Learning rate | 0.9 |
| $\gamma$ | Discount factor | 0.8 |
| $N_A$ | Number of cooperative regions | 4 |
| $z$ | Number of velocity levels | 20 |
| $\kappa$ | Popularity factor | 0.5 |
| $N$ | Total number of contents | 500 files |
| $M_w$ | Cache storage capacity of each UAV | 200 files |
| $M_q$ | Cache storage capacity of each user | 40 files |

## VI. SIMULATION RESULTS

In this section, the network performances of the proposed CMARL based trajectory design algorithm are evaluated. We use MATLAB to implement the proposed algorithm based on the system model described in Section II. In our simulations, we consider that the ground users are uniformly and independently distributed within an area of size 1000 m × 1000 m. The UAVs can dynamically move in the 3D space with adjustable altitude ranging between 60 and 80 m. The maximum velocities of users and UAVs are set to 1 and 20 m/s, respectively. The discount factor is set as 0.8 due to the relatively fast convergence. We adopt the time-decayed exploration rate $\varepsilon = \frac{1}{\Delta}$ [47], where $\Delta$ denotes the number of episodes. The default parameter values for simulation are given in Table III.

We compare the proposed CMARL with another three algorithms:

- *K-means:* In this case, the initial positions of UAVs are determined by the K-means clustering algorithm and then remain unchanged in the rest of the time slots.

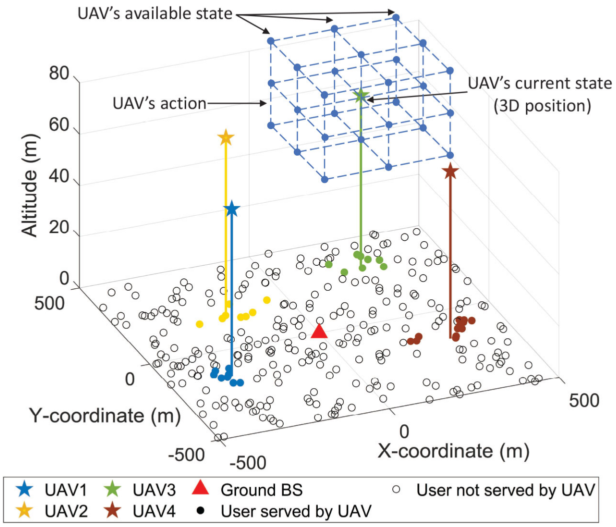

Fig. 3. An example of 3D deployment of multiple UAVs by using the K-means clustering algorithm.

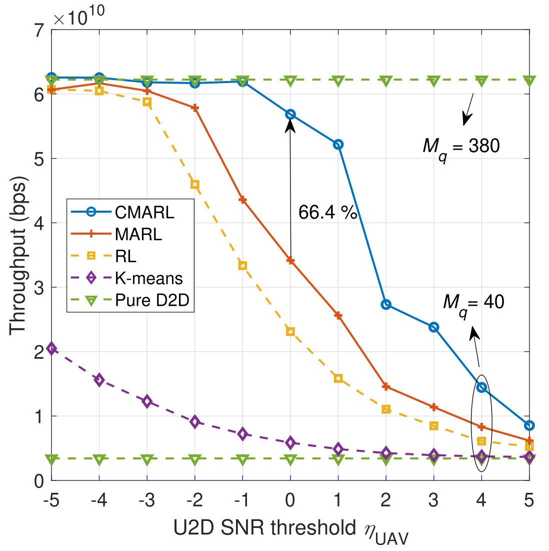

Fig. 4. Network throughput versus U2D SNR threshold.

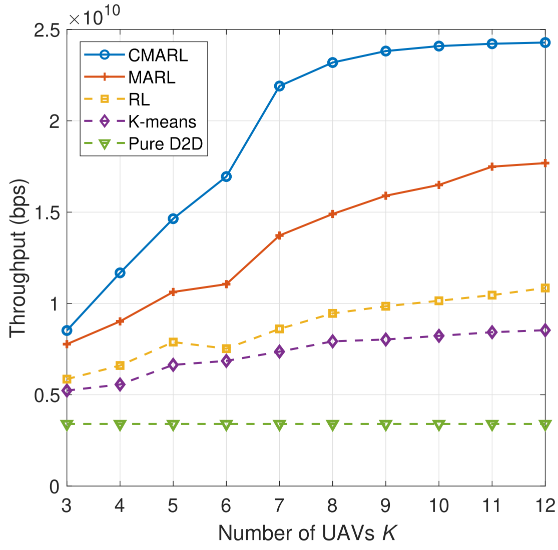

Fig. 5. Network throughput versus number of UAVs.

- *RL:* After the initialization by the K-means clustering algorithm, the UAV trajectory is determined by a centralized controller executing the single Q-learning algorithm.
- *MARL:* The third one is that of using the conventional MARL algorithm in which each UAV learns its optimal policy in a distributed and independent manner.

Fig. 3 plots an example of the 3D deployment of four UAVs in our simulations. The ground users are divided into four clusters, each of which is associated with one UAV. We can observe that some users are not served by any UAV due to the limited U2D transmission range. Each UAV's state can be represented by a 3D grid point as shown in the figure.

Fig. 4 compares the total network throughput for different UAV trajectory design algorithms. In addition to the aforementioned algorithms, we also compare the performance of the pure D2D scheme in which the contents can only be cached at D2D users (without UAV caching). Hence, it is reasonable that the U2D SNR threshold does not affect the throughput performance. For the K-means algorithm, it is observed that there is almost no performance gain when the U2D SNR threshold $\eta_{\mathrm{UAV}}$ is high. The reason is that the static UAV placement using the clustering scheme cannot make adaptations to the real-time location-dependent channel conditions. On the other hand, we can observe that the network throughput can be significantly improved by adopting the RL-based UAV trajectory design algorithms. Furthermore, we observe that different RL-based algorithms achieve a similar performance when the U2D SNR threshold is low. This is due to the fact that the maximum throughput is limited by the available bandwidth of U2D links. On the other hand, when the U2D SNR threshold is high, the performance gain of the RL-based algorithms is not significant since a large portion of users are served by a D2D transmitter or the ground BS. Finally, our proposed CMARL significantly outperforms all the compared algorithms, regardless of the U2D SNR threshold, which demonstrates the benefits of our proposed cooperative UAV trajectory design. For example, when $\eta_{\mathrm{UAV}} = 0\,\mathrm{dB}$, the network throughput can be increased by 66.4% compared with the conventional MARL algorithm. As shown in the green dashed line with downward triangles, the pure D2D scheme is observed to achieve the same throughput performance as CMARL when the cache storage capacity of each user is increased to 380 files, which is 9.5 times larger than that of CMARL.

Fig. 5 shows the network throughput performance with different numbers of UAVs. All of the cases can provide better network performance as the number of UAVs increases except the pure D2D scheme. It is observed that, when the number of UAVs increases, the distributed schemes (MARL and CMARL) can provide significant performance improvement, whereas the performance improvement is not obvious when the centralized schemes (RL and K-means) is applied. Also, as the UAV number increases, the performance improvement of CMARL is higher

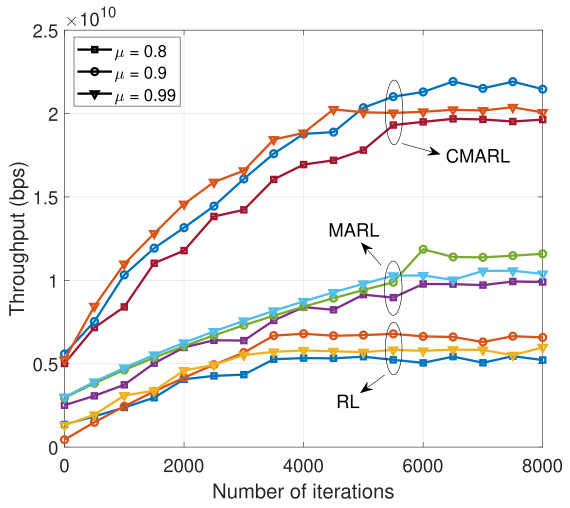

Fig. 6. Network throughput versus number of iterations.

compared to that of MARL. This shows the effectiveness of the proposed cooperative mechanism in exchanging UAVs' reward information. Clearly, the performance gains of using UAVs tend to decrease with the increase of the UAV number. Since the coverage of each UAV is limited, there exists a minimum number of UAVs that can provide ubiquitous coverage to the ground users such that the throughput performance is maximized. It is observed that CMARL converges to a stable performance when $K = 10$. Again, we see that our proposed CMARL can provide the highest throughput among all compared algorithms, regardless of the UAV number.

Fig. 6 shows the throughput performance obtained within a given number of iterations. We can see that, as the number of iterations increases, the throughput performance increases until convergence. Furthermore, compared to the MARL algorithm, the proposed CMARL algorithm presents a similar convergence speed while achieving around 50% throughput improvement. This result is consistent with our complexity analysis presented in Section V. Finally, it can be seen that the learning rate of 0.9 used for all the RL-based algorithms outperforms that of 0.8 and 0.99. This can be explained by the fact that a large learning rate (e.g., $\mu = 0.99$) will hinder convergence while a small learning rate (e.g., $\mu = 0.8$) leads to slow convergence.

Fig. 7 shows the throughput ratio of each UAV with different UAV numbers. The throughput ratio of each UAV is defined as the throughput per UAV provided relative to the total network throughput. From the figure shown, the throughput ratio decreases as the UAV number increases. This is reasonable, since the number of average connected users per UAV decreases with an increased UAV number. Furthermore, we can see that the throughput ratio of CMARL is larger than that of all other algorithms, which confirms the effectiveness of the proposed scheme for improving the U2D link utilization. We note that although the throughput ratio of each UAV decreases when the UAV number increases, the total network throughput can be improved as shown in Fig. 5. Then, Fig. 8 demonstrates a similar phenomenon for U2D SNR thresholds. For the same reason, the throughput ratio decreases with the increase of U2D SNR threshold. Again, we observe that CMARL outperforms all other algorithms in terms of throughput ratio. In particular, when $\eta_{\text{UAV}} = 0$ dB, the throughput ratio can be increased by 36.1% compared with the conventional MARL algorithm.

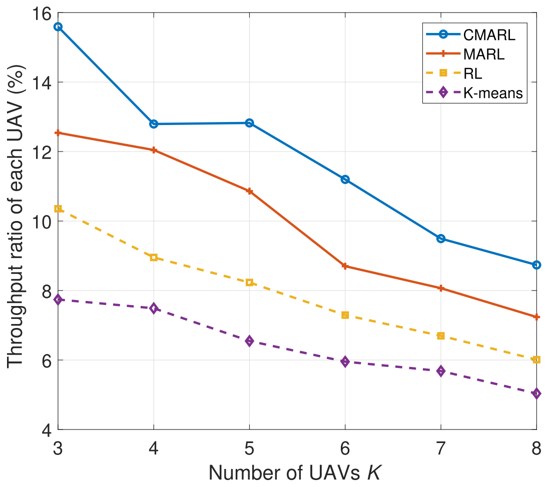

Fig. 7. Throughput ratio of each UAV versus number of UAVs.

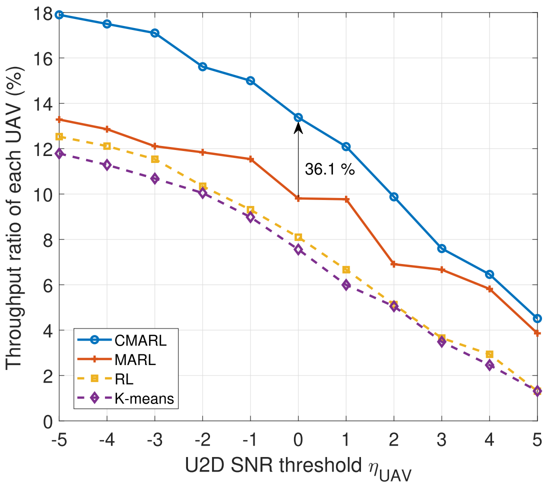

Fig. 8. Throughput ratio of each UAV versus U2D SNR threshold.

We present the impact of cooperative region number $N_A$ on the throughput performance of CMARL as shown in Fig. 9. We can see that $N_A = 4$ leads to the highest throughput when the number of UAVs $K$ is less than 10. However, CMARL setting $N_A = 9$ yields the best performance when $K = 10$. This indicates that $N_A$ needs to be adjusted according to $K$ for achieving the optimal throughput performance. To further demonstrate this, we plot the throughput performance of each UAV with respect to different $N_A$ and $K$, as shown in Fig. 10. The expected number of UAVs within a cooperative region can be expressed by $\bar{K}_c = \frac{K}{N_A}$. In the case of $\bar{K}_c \ll 1$, as shown in the upper left corner of Fig. 10, it is difficult for a UAV to find another UAV to cooperate with, since the total number of cooperative regions is much larger than the UAV number. As a consequence, the UAV will tend to update its own Q-table. Essentially, this case can be regarded as the conventional MARL in which all the agents independently learn their own policy. On

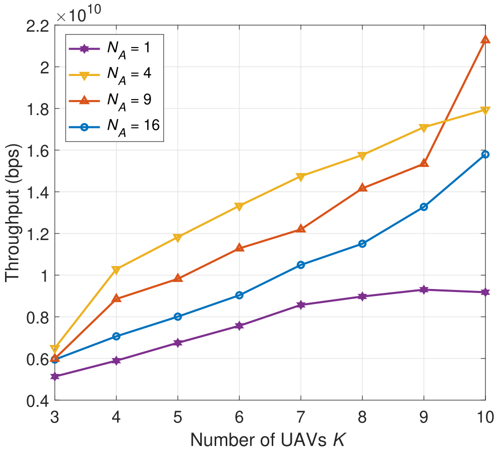

Fig. 9. Network throughput using the proposed CMARL algorithm with different numbers cooperative regions.

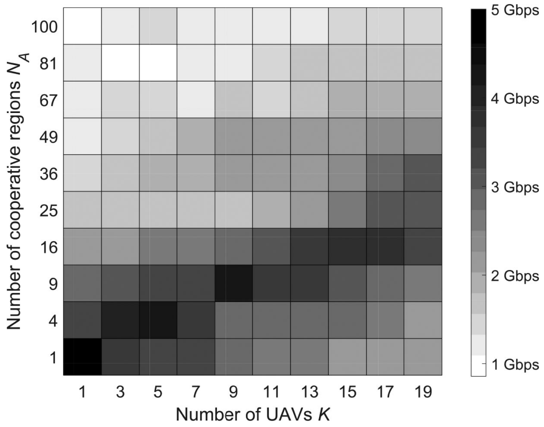

Fig. 10. Network throughput of each UAV using the proposed CMARL algorithm in the case of different UAV numbers and cooperative region numbers.

the other hand, in the case of $\bar{K}_c \approx K$, as shown in the lower right corner of Fig. 10, UAVs are highly likely to cooperate with each other, since the UAV number is much larger than the total number of cooperative regions. Therefore, the UAVs will tend to select actions according to the shared Q-table, which can be regarded as the single Q-learning where a central Q-table is used to achieve the global optimality. Remarkably, the best performance is achieved when $\bar{K}_c \approx 1$, i.e., $N_A \approx K$, which is in line with the results shown in Fig. 9. This result actually indicates that, by setting appropriate cooperative region number $N_A$, the proposed CMARL can achieve the optimal balance between the distributed and centralized RL systems.

Fig. 11 shows the throughput performance of the proposed CMARL with different cache-related parameters. We can see that the throughput increases with the cache storage capacity of UAV. This is because a larger UAV cache storage capacity will result in a higher UAV cache hit probability. Furthermore, the throughput performance increases as the popularity factor $\kappa$ increases, which is consistent with previous results reported in [35]. The reason is that, when the content requests are more concentrated, the U2D cache hit probability increases. Also,

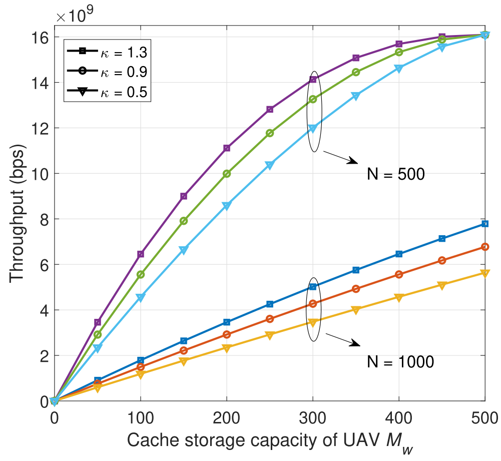

Fig. 11. Network throughput using the proposed CMARL algorithm with different cache storage capacities.

a linear relationship between the throughput performance and the cache storage capacity can be observed when $N = 1000$. However, when $N = 500$, a log-linear relationship between the throughput performance and the cache storage capacity is observed. In particular, as the cache storage capacity increases, the throughput performance for different popularity factors $\kappa$ becomes closer and eventually converges to a constant when $N = 500$. The reason is that when $M_w = N$, all the contents can be stored on the UAVs, resulting in the maximum UAV cache hit probability.

## VII. Conclusion

In this paper, we have proposed a multi-UAV trajectory design algorithm based on reinforcement learning (RL) for cache-enabled UAVs to dynamically learn their optimal 3D positions while maximizing the network throughput. A cooperative multi-agent RL (CMARL) method has been developed to overcome the inefficiency of independent RL in multi-agent systems. Using the proposed CMARL trajectory design algorithm, each UAV can autonomously decide whether or not the flight decisions should be coordinated with other UAVs. Simulation results have shown that the proposed trajectory design algorithm can improve the network throughput and the UAV-to-Device (U2D) link utilization compared to the conventional RL algorithms. Also, our results have revealed that the proposed cooperative method can achieve the optimal balance between the distributed and centralized RL systems with an appropriate number of cooperative regions. These results can provide important design guidelines for high-throughput aerial-terrestrial wireless networks.

## References

[1] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 32–38, Feb. 2018.

[2] G. J. Nunns, Y. J. Chen, D. K. Chang, K. M. Liao, F. P. Tso, and L. Cui, "Autonomous flying WiFi access point," in *Proc. IEEE Symp. Comput. Commun.*, 2019, pp. 278–283.

[3] Y. J. Chen and D. Y. Huang, “Trajectory optimization for cellular-enabled UAV with connectivity outage constraint,” *IEEE Access*, vol. 8, pp. 29205–29218, Feb. 2020.

[4] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 3, pp. 2334–2360, Jul.–Sep. 2019.

[5] M. Chen, W. Saad, and C. Yin, “Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[6] 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 Trans. Commun.*, vol. 67, no. 9, pp. 6386–6400, Sep. 2019.

[7] 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.

[8] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K. R. Choo, “Neural-blockchain-based ultrareliable caching for edge-enabled UAV networks,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 10, pp. 5723–5736, Oct. 2019.

[9] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, “UAV-relaying-assisted secure transmission with caching,” *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3140–3153, May 2019.

[10] N. Zhao *et al.*, “Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment,” *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.

[11] H. Wang, J. Wang, G. Ding, L. Wang, T. A. Tsiftsis, and P. K. Sharma, “Resource allocation for energy harvesting-powered D2D communication underlaying UAV-assisted networks,” *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 14–24, Mar. 2018.

[12] A. Asheralieva and D. Niyato, “Game theory and lyapunov optimization for cloud-based content delivery networks with device-to-device and UAV-enabled caching,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10094–10110, Oct. 2019.

[13] L. Liu, S. Zhang, and R. Zhang, “CoMP in the sky: UAV placement and movement optimization for multi-user communications,” *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5645–5658, Aug. 2019.

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

[15] S. Zhang, H. Zhang, B. Di, and L. Song, “Cellular UAV-to-X communications: Design and optimization for multi-UAV networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[16] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-UAV enabled wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[17] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, “Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3424–3438, Mar. 2020.

[18] P. Yang, X. Cao, X. Xi, Z. Xiao, and D. Wu, “Three-dimensional drone-cell deployment for congestion mitigation in cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9867–9881, Oct. 2018.

[19] *3GPP*, “Enhancement for unmanned aerial vehicles,” Tech. Rep. 22.829 version 17.1.0, Sep. 2019. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/22_series/22.829/

[20] Y. J. Chen, K. M. Liao, M. L. Ku, and F. P. Tso, “Mobility-aware probabilistic caching in UAV-assisted wireless D2D networks,” in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.

[21] L. Wang, Y. Chao, S. Cheng, and Z. Han, “An integrated affinity propagation and machine learning approach for interference management in drone base stations,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 83–94, Mar. 2020.

[22] X. Liu, Y. Liu, and Y. Chen, “Reinforcement learning in multiple-UAV networks: Deployment and movement design,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, Aug. 2019.

[23] J. Cui, Y. Liu, and A. Nallanathan, “Multi-agent reinforcement learning-based resource allocation for UAV networks,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[24] Y. J. Chen, D. K. Chang, and C. Zhang, “Autonomous tracking using a swarm of UAVs: A constrained multi-agent reinforcement learning approach,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13702–13717, Nov. 2020.

[25] F. Y. Wu, H. L. Zhang, J. J. Wu, and L. Y. Song, “Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning,” *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4175–4189, Jul. 2020.

[26] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, “Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.

[27] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” *Studies Syst., Decision Control Handbook RL Control*, 2020.

[28] R. Amer, W. Saad, and N. Marchetti, “Mobility in the sky: Performance and mobility analysis for cellular-connected UAVs,” *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3229–3246, May 2020.

[29] J. G. Andrews, F. Baccelli, and R. K. Ganti, “A tractable approach to coverage and rate in cellular networks,” *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 3122–3134, Nov. 2011.

[30] M. N. Anjum and H. Wang, “Mobility modeling and stochastic property analysis of airborne network,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1282–1294, Jul.–Sep. 2020.

[31] Z. Ma, B. Ai, R. He, G. Wang, Y. Niu, and Z. Zhong, “A wideband non-stationary air-to-air channel model for UAV communications,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1214–1226, Feb. 2020.

[32] Y. Zeng, J. Xu, and R. Zhang, “Energy minimization for wireless communication with rotary-wing UAV,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[33] X. Wang and M. C. Gursoy, “Coverage analysis for energy-harvesting UAV-assisted mmWave cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 12, pp. 2832–2850, Dec. 2019.

[34] J. Dai, J. Liu, Y. Shi, S. Zhang, and J. Ma, “Analytical modeling of resource allocation in D2D overlaying multihop multichannel uplink cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6633–6644, Aug. 2017.

[35] Z. Chen, N. Pappas, and M. Kountouris, “Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal,” *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.

[36] F. Song *et al.*, “Probabilistic caching for small-cell networks with terrestrial and aerial users,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9162–9177, Sep. 2019.

[37] J. Wen, K. Huang, S. Yang, and V. O. K. Li, “Cache-enabled heterogeneous cellular networks: Optimal tier-level content placement,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5939–5952, Sep. 2017.

[38] B. Blaszczyszyn and A. Giovanidis, “Optimal geographic caching in cellular networks,” in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 3358–3363.

[39] J. Rao, H. Feng, C. Yang, Z. Chen, and B. Xia, “Optimal caching placement for D2D assisted wireless caching networks,” in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[40] R. Wang, R. Li, E. Liu, and P. Wang, “Performance analysis and optimization of caching placement in heterogeneous wireless networks,” *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1883–1887, Oct. 2019.

[41] Q. Pham, S. Mirjalili, N. Kumar, M. Alazab, and W. Hwang, “Whale optimization algorithm with applications to resource allocation in wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4285–4297, Apr. 2020.

[42] A. M. Koushik, F. Hu, and S. Kumar, “Deep Q-learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 554–566, Sep. 2019.

[43] N. Fulda and D. Ventura, “Predicting and preventing coordination problems in cooperative Q-learning systems,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 780–785.

[44] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, “Fully decentralized multi-agent reinforcement learning with networked agents,” in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 10–15.

[45] X. Wu and J. Lu, “Fenchel dual gradient methods for distributed convex optimization over time-varying networks,” *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4629–4636, Nov. 2019.

[46] Y. Chon, E. Talipov, H. Shin, and H. Cha, “Mobility prediction-based smartphone energy optimization for everyday location monitoring,” in *Proc. 9th ACM Conf. Embedded Netw. Sensor Syst.*, Seattle, WA, USA, 2011, pp. 82–95.

[47] M. Carrascosa and B. Bellalta, “Decentralized AP selection using multi-armed bandits: Opportunistic $\varepsilon$-greedy with stickiness,” in *Proc. IEEE Symp. Comput. Commun.*, 2019, pp. 1–7.

[48] N. Rupasinghe, Y. Yapc. Güvenç, and Y. Kakishima, "Non-orthogonal multiple access for mmWave drone networks with limited feedback," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 762–777, Jun. 2019.

[49] X. Lu *et al.*, "Integrated use of licensed- and unlicensed-band mmWave radio technology in 5 G and beyond," *IEEE Access*, vol. 7, pp. 24376–24391, Feb. 2019.

[50] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-effective cache deployment in mobile heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11 264–11 276, Dec. 2017.

**Yu-Jia Chen** (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2010 and 2015, respectively. From 2015 to 2018, he was a Postdoctoral Research Fellow with National Chiao Tung University and from 2018 to 2019, he was a Postdoctoral Research Fellow with Harvard University, Cambridge, MA, USA. In 2019, he joined National Central University, Taoyuan City, Taiwan, where he is currently an Assistant Professor with the Department of Communication Engineering. He has authored or coauthored more than 40 articles in peer-reviewed journal and conference papers. He is holding four U.S. patents and four ROC patents. His research interests include low-latency communications, wireless sensing, and network security.

**Kai-Min Liao** received the B.S. degree in communication engineering from National Taipei University, Taipei, Taiwan, in 2018, and the M.S. degree in communication engineering from National Central University, Taoyuan City, Taiwan, in 2021. His current research interests include UAV wireless caching network, low-latency analysis, and machine learning.

**Meng-Lin Ku** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2002, 2003, and 2009, respectively. From 2009 to 2010, he was a Postdoctoral Research Fellow with Prof. Li-Chun Wang at the Department of Electrical and Computer Engineering, National Chiao Tung University, and with Prof. Vahid Tarokh at the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. In 2010, he became a Faculty Member of the Department of Communication Engineering, National Central University, Jung-li, Taiwan, where he is currently a Professor. In 2013, he was a Visiting Scholar with the Signals and Information Group of Prof. K. J. Ray Liu, University of Maryland, College Park, MD, USA. His current research interests include green communications, UAV communications, and optimization and learning for radio access. He was the recipient of the Best Counseling Award in 2012 and the university-level Best Teaching Award in 2014, 2015, and 2016, Research Excellence Award in 2018, 2019, and 2020, all at National Central University. He was also the recipient of the Exploration Research Award of the Pan Wen Yuan Foundation in 2013 and the Chinese Institute of Electrical Engineering Outstanding Young Electrical Engineer Award in 2019. He is currently an Associate Editor for the IEEE ACCESS.

**Fung Po Tso** received the B.Eng., M.Phil., and Ph.D. degrees from the City University of Hong Kong, Hong Kong, in 2006, 2007, and 2011, respectively. He was an SICSA Next Generation Internet Fellow with the School of Computing Science, University of Glasgow, Glasgow, U.K., from 2011 to 2014, and a Lecturer with Liverpool John Moores University, Liverpool, U.K., from 2014 to 2017. He is currently a Lecturer with the Department of Computer Science, Loughborough University, U.K. He has authored or coauthored more than 20 research articles in top venues and outlets. His research interests include network policy management, network measurement and optimization, cloud data center resource management, data center networking, software-defined networking, distributed systems, and mobile computing and system.

**Guan-Yi Chen** received the B.S. degree in communication engineering from National Central University, Taoyuan City, Taiwan, in 2020. His current research interests mainly include wireless caching network and device-to-device communication.