Caching Placement and Resource Allocation for Cache-Enabling UAV NOMA Networks

Tiankui Zhang, *Senior Member, IEEE,* Ziduan Wang, Yuanwei Liu, *Senior Member, IEEE,* Wenjun Xu, *Senior Member, IEEE* and Arumugam Nallanathan, *Fellow, IEEE*

Abstract—This article investigates the cache-enabling unmanned aerial vehicle (UAV) cellular networks with massive access capability supported by non-orthogonal multiple access (NOMA). The delivery of a large volume of multimedia contents for ground users is assisted by a mobile UAV base station, which caches some popular contents for wireless backhaul link traffic offloading. In cache-enabling UAV NOMA networks, the caching placement of content caching phase and radio resource allocation of content delivery phase are crucial for network performance. To cope with the dynamic UAV locations and content requests in practical scenarios, we formulate the long-term caching placement and resource allocation optimization problem for content delivery delay minimization as a Markov decision process (MDP). The UAV acts as an agent to take actions for caching placement and resource allocation, which includes the user scheduling of content requests and the power allocation of NOMA users. In order to tackle the MDP, we propose a Q-learning based caching placement and resource allocation algorithm, where the UAV learns and selects action with *soft ε-greedy* strategy to search for the optimal match between actions and states. Since the action-state table size of Q-learning grows with the number of states in the dynamic networks, we propose a function approximation based algorithm with combination of stochastic gradient descent and deep neural networks, which is suitable for large-scale networks. Finally, the numerical results show that the proposed algorithms provide considerable performance compared to benchmark algorithms, and obtain a trade-off between network performance and calculation complexity.

Index terms— dynamic resource allocation, non-orthogonal multiple access, reinforcement learning, unmanned aerial vehicle

I. INTRODUCTION

With the explosion of massive multimedia applications and the continuous growth of mobile data traffic, wireless communication faces the problem of limited resources. In order to effectively meet the increasing user demand for high data rate and low access delay, many works [1–5] have paid attention to wireless connectivity from the sky with unmanned aerial vehicles (UAVs). UAVs, also known as remotely piloted aircraft systems (RPAS) or drones, are small pilotless aircrafts

Tiankui Zhang, Ziduan Wang and Wenjun Xu are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {zhangtiankui, wangziduan, wjxu}@bupt.edu.cn).

Yuanwei Liu and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: {yuanwei.liu, a.nallanathan}@qmul.ac.uk).

that are rapidly deployable for complementing terrestrial communications [1]. Promising scenarios for UAV communications can be as follows: establishing temporal communication infrastructure during natural disasters, offloading traffic for dense cellular networks, data collection for supporting Internet of Things (IoT) [2], and mobile edge computing server for supporting IoT [4].

With the rapid growth of UAV-assisted cellular networks, UAVs perform diverse roles, including radio access nodes, base stations (BSs) [5–10] and relays [11]. In UAV-assisted cellular networks, the data rate is limited by both the radio access links and the wireless backhaul links. For radio access links, multi-users can be served with the same time region and frequency band based on non-orthogonal multiple access (NOMA), which has received remarkable attention [12–14]. In [12], the performance of NOMA in large-scale networks has been investigated with stochastic geometry theory. The resource allocation of NOMA heterogeneous networks (HetNets) has been studied in [13]. In [14], the pair-wise error probability (PEP) performance of different detectors in multiple-input-multiple-output (MIMO) NOMA system has been analyzed, which has also been minimized by the proposed two kinds of user selection methods. Recently, NOMA has been exploited as an effective method to enhance the access capability of UAV-assisted cellular networks [6–10]. To mitigate the traffic load of the backhaul links, edge caching has been studied [15, 16]. In [15], the bandwidth allocation and caching placement have been jointly optimized in HetNets. In [16], multi-tier collaborative caching framework in HetNets has been studied to maximize the network capacity. Caching popular contents at UAVs has been regarded as an effective emerging method to alleviate the backhaul congestion and reduce latency in UAV-assisted cellular networks [17–21]. The cache-enabling UAV NOMA network is a promising framework for high data rate and low transmission latency in numerous multimedia contents distribution scenario.

A. Related Works

Recently, there has been some researches on UAV NOMA cellular networks [6–10]. Based on stochastic geometry theory, the 3-Dimension UAV framework for providing wireless service to randomly roaming NOMA users has been studies in [6]. In the UAV NOMA cellular networks, a cooperation mechanism between a UAV and a macro base station has been proposed in [7], which maximizes the sum rate of users by the UAV trajectory and NOMA precoding joint optimization.

In [8], a hybrid transmission strategy based on VP and NOMA (VP-NOMA) has been proposed, which minimizes total transmit power for certain quality of service (QoS) requirements by designing a beamforming matrix with the power allocation strategy. The UAV trajectory and resource allocation have been jointly optimized for maximizing the minimum data rate of ground users in [9]. In [10], the trajectory, task data, and computing resource allocation have been joinly optimized to minimize the largest energy consumption among UAVs.

Moreover, caching at UAV [17–21] has attracted increasing attention recently to relieve the pressure on the wireless backhaul links. The resource allocation in cache-enabling UAV-assisted cellular networks has been considered in [17], where a joint optimization problem of UAV deployment, caching placement and user association has been solved to maximize the quality of experience (QoE) of users. In [18], user association, spectrum allocation, and content caching have been jointly optimized by a liquid state machine (LSM) based algorithm. In [19], user association, UAV location, and caching placement have been jointly optimized to maximize the users' QoE while minimizing the transmit power used by the UAVs. The cache-enabling UAV assisted secure transmission for scalable videos in hyper-dense networks has been studied in [20], and a distributed algorithm has been proposed to manage the interference by cooperatively generating the precoding matrices of cache-enabling UAVs. In [21], an optimization problem has been formulated to maximize the security of UAV-relayed wireless networks with caching by jointly adjusting the UAV trajectory and time scheduling.

The problem of resource allocation with dynamic networks has been studied in [22, 23]. In [22], a dynamic programming approach for heterogeneous networks (HetNets) has been designed, where communicating nodes have been efficiently matched and radio resources have been assigned in an interference-aware manner. The energy harvesting downlink MIMO systems have been studied in [23], where an online resource allocation algorithm has been proposed to maximized the sum rate. Some works on the resource allocation of UAV-assisted cellular networks have been studied [24, 25]. However, only a few existing works have concentrated on the dynamic resource allocation of UAV-assisted cellular networks [26, 27]. A real-time access points provision algorithm has been developed in [26], where UAV-mounted cloudlets are assumed to carry out adaptive recommendation in a distributed manner so as to reduce computing and traffic load. In [27], sensing and transmission protocol, UAV trajectory design, and radio resource management in U2X communication have been jointly optimized to maximize the average number of valid data transmissions.

B. Motivation and Contribution

As mentioned above, the caching placement and resource allocation optimization problems have been considered in cache-enabling UAV NOMA networks. However, the optimization scenarios of the current studies are most likely to be static, and rarely consider the dynamic environment including the UAV movement and content request varying. Due to the moving characteristic of UAVs [28], the efficiency of caching placement and resource allocation may be improved by considering a long-term optimization problem. To fulfill this gap, this article studies the caching placement and resource allocation in cache-enabling UAV NOMA networks with dynamic UAV locations and content requests. The dynamic optimization problem for caching placement of a UAV, user scheduling of content requests, as well as power allocation of NOMA users is formulated. The caching placement and resource allocation process is modeled as a Markov decision process (MDP), which is solved by reinforcement learning. Moreover, a function approximation based algorithm is proposed to deal with the dynamic optimization problem in large-scale networks. Our contributions are summarized as follows:

- We propose a framework of cache-enabling UAV NOMA cellular networks for content delivery of ground users in a hotspot area. We define the long-term sum delay of users as the content delivery cost of downlink UAV NOMA cellular networks. We formulate an optimization problem to minimize content delivery delay by jointly optimizing the caching placement of a UAV, user scheduling of content requests, and the power allocation of NOMA users.

- We transform the original proposed optimization problem to a MDP based problem and define the dynamic states of UAV movement and content request varying, in which the UAV performs as the agent. Since the instantaneous content delivery delay depends on the current state and action according to the property of Markov chain, we develop a Q-learning based content placement and resource allocation algorithm for solving the MDP based problem. Furthermore, to deal with the high complexity of an action-state table (Q-table) in large-scale networks, we propose a function approximation based caching placement and resource allocation algorithm, which can obtain the near-optimal solution according to the output of the function rather than searching a huge action space.

- We provide simulation results to validate the performance of the proposed caching placement and resource allocation algorithms compared with the benchmark algorithms. The simulation results demonstrate that the proposed Q-learning based algorithm gets a trade-off between network performance and computation complexity. Meanwhile, the proposed function approximation based algorithm obtains a considerable network performance without the complexity ergodic search in the action space.

C. Organization

The rest of this article is organized as follows. In Section II, we present the system model and formulate the optimization problem for long-term content delivery delay minimization. In Section III, we propose reinforcement learning based algorithms for caching placement and resource allocation. Simulation results are presented in Section IV, which is followed by conclusion in Section V.
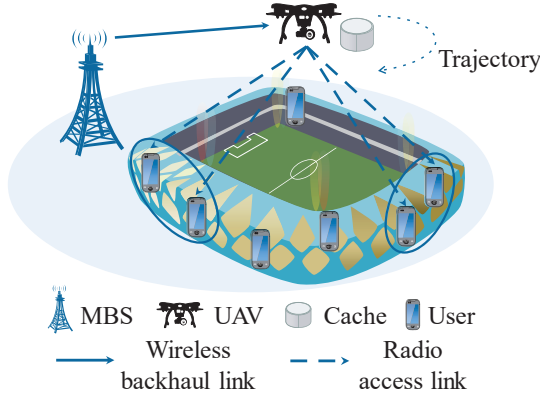
Fig. 1: Cache-enabling UAV NOMA networks.

TABLE I: Notation

Notation	Description
$\mathcal{N}$	Set of users
T_v	Fly period of UAV
$\mathcal{M}$	Set of contents
C_1	Size of content
Z	Cache capacity of UAV
L_n, L_{ma}	Location of user n and MBS
$D_{Bm}(t)$	Backhaul link delay of content m
q_m	Probability of users request for content m
$i_m(t)$	Proactive cache policy of content m
$d_{ma}(t)$	Distance between MBS and UAV
$\Pr(g_{maLoS}(t))$	LoS probability of backhaul link
$h_g(t)$	Coefficient of power allocation of group g
$\mathcal{G}$	Set of user groups
δ	Length of time slot
p_{ma}, p_{uav}	Power of MBS and UAV
η	Zipf distribution Parameter
$b_n(t)$	Response to user n at time slot t
$L_{uav}(t)$	Location of UAV at time slot t
$D_{An}(t)$	Radio access link delay of user n
$c_m(t)$	Cache situation of content m in the beginning of time slot t
$d_{NUg}(t)$, $d_{FUg}(t)$	Distance between UAV and users of group g at time slot t
$\bar{g}_{ma}(t)$	Average path loss of backhaul link
r_{nm}	Request of user n to content m
$\mu_m(t)$	Content virtual queues in time slot t
$R_B(t)$, $R_{NUg}(t)$, $R_{FUg}(t)$	Data rate of backhaul link, NU's and FU's radio access links at time slot t
$\Gamma_{NUg}(t)$, $\Gamma_{FUg}(t)$	SINR of radio access link of NU and FU at time slot t
$\bar{g}_{NUg}(t)$, $\bar{g}_{FUg}(t)$	Average path loss of radio access link of users in group g at time slot t
$g_{maLoS}(t)$, $g_{maNLoS}(t)$	LoS and NLoS path loss of backhaul link at time slot t

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider the downlink transmission for N users in a hotspot area covered by a ground macro base station (MBS) assisting by a mobile UAV base station. The UAV is connected to the MBS via wireless backhaul links, as shown in Fig. 1. We consider a dynamic scenario, which consists of a periodically-moving UAV and time-varying request for contents. We assume that the flight trajectory of the UAV is predetermined as in [29] and the flight time is T_v.

We use the NOMA technique, including superposition coding (SC) technique at the UAV and serial interference cancel (SIC) technique at users. Two users form a group and we denote group set as $\mathcal{G} = \{1, 2, \cdots, G\}$. The UAV could serve more than one group at the same time slot. In each group, the user closer to the UAV is called as the near user (NU), and the other user is called as the far user (FU). Following the nearest near user and nearest far user (NNNF) NOMA users selection scheme proposed in [30], we assume that the nearest NU and nearest FU are assigned to a group. We define the mapping from users to group g at time slot t as $(\varphi_{NUg}(t), \varphi_{FUg}(t))$. Table I provides a summary of the notations used hereinafter.

A. UAV Mobility Model

As indicated above, the UAV flies on the trajectory with finite time period T_v. For the convenience of description, the flying duration T_v is discretized into T equal time slots, i.e., $T_v = T\delta$, where δ is the length of each time slot. We assume moving speed of the UAV is v, and the maximum moving distance of the UAV in each time slot is δv. Therefore, we can assume that the distance between the MBS and the UAV does not change during each time slot, so is the distance between the UAV and users. To simplify the problem, we focus on the stable flight process of the UAV at height h, ignoring the UAV's take-off and landing phases. We use $L_{uav}(t) : (x_{uav}(t), y_{uav}(t), h)$ to express the observed location of the UAV at time slot t, and the location of user n is defined as $L_n : (x_n, y_n)$, which is subject to random distribution. We use $L_{ma} : (x_{ma}, y_{ma})$ to represent the location

of the MBS. We define the distance between the MBS and the UAV at time slot t as $d_{ma}(t)$, which is calculated as

$$d_{ma}(t) = \sqrt{(x_{ma} - x_{uav}(t))^2 + (y_{ma} - y_{uav}(t))^2 + h^2}. \tag{1}$$

Similarly, we obtain the distance from the NU and the FU of group g to the UAV at time slot t, $d_{NUg}(t)$ and $d_{FUg}(t)$. Since the location of the UAV is dynamic in the flying duration, the distances between the UAV and the MBS/users are time varying with time slot t.

B. Content Request and Cache Model

We denote the multimedia contents as a set $\mathcal{M} = \{1, \cdots, M\}$. At time slot t, we assume the probability that user n requests content m follows Zipf distribution, which is a conditional probability with user n generating request at time slot t as the condition, i.e. $r_n(t) = 1$, and can be calculated as

$$P(r_{nm}(t) = 1 | r_n(t) = 1) = \frac{\frac{1}{m^\eta}}{\sum_{j_1=1}^{M} \frac{1}{j_1^\eta}}, \tag{2}$$

where η is the exponent of the Zipf distribution. Let $r_{nm}(t) = 1$ if user n requests content m at time slot t, otherwise $r_{nm}(t) = 0$. The user n requests for at most one content

at time slot t. We assume that the preference of user to content is constant over time, which can be extended to time varying user preference occasion. In this model, the probability of user n's request for content m follows random average distribution over the time slots. The user n requests for contents at time slot t with the probability $P\left(r_n\left(t\right)=1\right)=\frac{R_g}{N}$, where the request generating coefficient R_g is constant over time. Then the probability that user n requests content m at time slot t is given by $P\left(r_{nm}\left(t\right)=1\right)=P\left(r_{nm}\left(t\right)=1|r_n\left(t\right)=1\right)P\left(r_n\left(t\right)=1\right)$.

Remark 1. *From (2), we notice that the value of the contents number affects the diversity of users' interest. It is hard to design a robust caching placement algorithm, which only depends on the statistical characteristics of content requests, to perform well in the varying contents number M. We tend to decide caching placement based on the real-time content request characteristics of networks.*

The users' requests may not be responded immediately and would be scheduled among time slots. In time slot t, $b_n\left(t\right)=1$ if the requested contents by user n is scheduled, otherwise $b_n\left(t\right)=0$. The scheduled users at time slot t will be transmitted for the requested content. Moreover, we assume that users will not request for new contents before their previous requests are responded. It means that we have $r_{nm}\left(t+1\right)=1$ if $r_{nm}\left(t\right)-b_n>0$.

We assume the cache-enabling UAV could cache at most Z contents and $Z\leq M$. We assume the UAV proactively caches contents at each time slot. If the requested content is cached by the UAV, it will be transmitted to users via radio access links directly, otherwise the content will be transmitted from the MBS to the UAV via wireless backhaul links. The UAV only need to fetch the content from the MBS once if it is not cached, even though there are several users requesting for the same content at the same time slot. Besides, contents may be transmitted from the MBS to the UAV because of proactive caching. We define proactive caching index $i_m\left(t\right)=1$ to indicate content m be cached at time slot t for later time slots. At the end of each time slot t, cache status is updated according to the proactive caching index of current time slot $i_m\left(t\right)$. If content m has been cached in the UAV at the beginning of time slot t, $c_m\left(t\right)=1$, otherwise $c_m\left(t\right)=0$. Obviously, the proactive caching index at time slot t decides the cached contents at time slot $t+1$, i.e. $i_m\left(t\right)=c_m\left(t+1\right)$.

C. Channel Model

We assume the downlink transmission consists of two parts, the wireless backhaul links from the ground MBS to the UAV, and radio access links from the UAV to the ground users. As indicated in [31], the UAV-to-ground links can be modeled by a probabilistic path loss model.

Referring to the 3GPP specifications in [32], the path-loss of the wireless backhaul link is denoted as $\bar{g}_{\mathrm{ma}}\left(t\right)$, which is randomly determined by line-of-sight (LoS) and non-line-of-sight (NLoS) link states

$$
\begin{aligned}
g_{\mathrm{maLoS}} = {} & 22.25\log_{10}d_{\mathrm{ma}}\left(t\right) \\
& -0.5\log_{10}h\log_{10}d_{\mathrm{ma}}\left(t\right)+20\log_{10}f+30.9,
\end{aligned}
\tag{3}
$$

and

$$
\begin{aligned}
g_{\mathrm{maNLoS}} = {} & 43.2\log_{10}d_{\mathrm{ma}}\left(t\right) \\
& -7.6\log_{10}h\log_{10}d_{\mathrm{ma}}\left(t\right)+20\log_{10}f+32.4,
\end{aligned}
\tag{4}
$$

where $d_{\mathrm{ma}}\left(t\right)$ represents the distance between MBS and UAV at time slot t, and f represents the frequency of carrier. The probability of LoS is given in (5) at the bottom of this page, where $d_o=\max\left[294.05\log_{10}h-432.94,18\right]$, and $p_1=233.98\log_{10}h-0.95$. Then, the probability of NLoS could be calculated by $\mathrm{Pr}\left(g_{\mathrm{maNLoS}}\left(t\right)\right)=1-\mathrm{Pr}\left(g_{\mathrm{maLoS}}\left(t\right)\right)$. Hence, the average path-loss can be expressed as

$$
\begin{aligned}
\bar{g}_{\mathrm{ma}}\left(t\right) = {} & \mathrm{Pr}\left(g_{\mathrm{maLoS}}\left(t\right)\right)\times g_{\mathrm{maLoS}}\left(t\right) \\
& +\mathrm{Pr}\left(g_{\mathrm{maNLoS}}\left(t\right)\right)\times\max\left\{g_{\mathrm{maLoS}}\left(t\right),g_{\mathrm{maNLoS}}\left(t\right)\right\}.
\end{aligned}
\tag{6}
$$

We assume the wireless backhaul links and the radio access links are allocated non-overlap frequency channels, and therefore have no co-frequency interference between them. The signal-to-interference-plus-noise ratio (SINR) of the wireless backhaul link at time slot t is

$$
\Gamma_{\mathrm{B}}\left(t\right)=\frac{p_{\mathrm{ma}}10^{-\bar{g}_{\mathrm{ma}}\left(t\right)/10}}{\sigma^2+\sum p_{\mathrm{ma}'}10^{-\bar{g}_{\mathrm{ma}'}\left(t\right)/10}},
\tag{7}
$$

where σ^2 represents the variance of additive Gaussian noise (AWGN), p_{ma} is the transmission power of the MBS, $p_{\mathrm{ma}'}$ is the transmission power of the neighboring MBSs in the networks, and $\bar{g}_{\mathrm{ma}'}\left(t\right)$ is the path-loss from the neighboring MBS to the UAV.

We can obtain the average path-loss between the UAV and the NU/FU of group g at time slot t, $\bar{g}_{\mathrm{NU}g}\left(t\right)$ and $\bar{g}_{\mathrm{FU}g}\left(t\right)$, which are modeled following (6). The transmission power of the UAV is p_{uav}, which is a constant during the flying duration, and is evenly allocated to user groups. The power of the UAV allocated to user group g is expressed as p_g. At time slot t, the power allocation coefficient of the UAV to the NU in group g is denoted as $h_g\left(t\right)$, which follows the proportion of power allocation. The power allocation coefficient of the UAV to the FU in group g could be expressed as $1-h_g\left(t\right)$. The radio access links are based on NOMA, as shown in Fig. 2. The received signal at the NU is given by

$$
\begin{aligned}
y_{\mathrm{NU}g}\left(t\right) = {} & \sqrt{p_g h_g\left(t\right)}x_{\mathrm{NU}g}\left(t\right)10^{-\bar{g}_{\mathrm{NU}g}\left(t\right)/10}+ \\
& \sqrt{p_g\left(1-h_g\left(t\right)\right)}x_{\mathrm{FU}g}\left(t\right)10^{-\bar{g}_{\mathrm{NU}g}\left(t\right)/10}+\zeta_{\mathrm{NU}g}\left(t\right),
\end{aligned}
\tag{8}
$$

$$
\mathrm{Pr}\left(g_{\mathrm{maLoS}}\left(t\right)\right)=\begin{cases}1,\ if\ \sqrt{d_{\mathrm{ma}}(t)^2-h^2}\leq d_o,\\[2mm]\frac{d_o}{\sqrt{d_{\mathrm{ma}}(t)^2-h^2}}+\exp\left\{\left(\frac{-\sqrt{d_{\mathrm{ma}}(t)^2-h^2}}{p_1}\right)\left(1-\frac{d_o}{\sqrt{d_{\mathrm{ma}}(t)^2-h^2}}\right)\right\},\ if\ \sqrt{d_{\mathrm{ma}}(t)^2-h^2}>d_o.\end{cases}
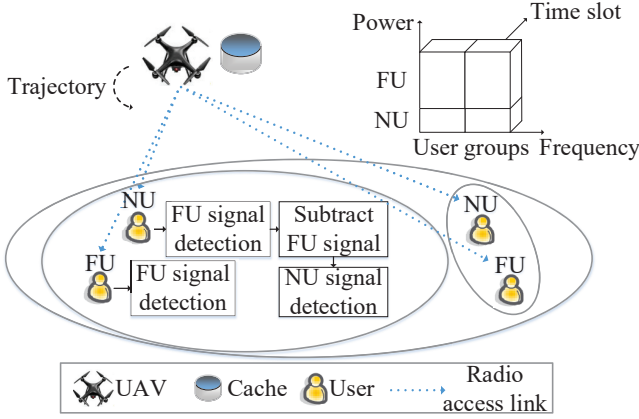\tag{5}
$$

Fig. 2: NOMA based radio access links from UAV to ground users.

where $\sqrt{p_g h_g(t)} x_{NUg}(t) + \sqrt{p_g(1-h_g(t))} x_{FUg}(t)$ represents the composite signal transmitted to the users in group g. $\bar{g}_{\mathrm{NU}g}(t)$ represents path loss of the radio access link to the NU of group g.

As shown in Fig. 2, there exists interference between the NU and the FU in each group. The NU desires to decode and remove the interference from the FU's superposition signal based on SIC. The interference cancellation is successful if the NU's received SINR for the FU's signal is larger or equal to the received SINR of the FU for its own signal [33, 34]. We define the interference from superposition signal of the NU to the FU in receiver of the FU as $I_{\mathrm{NF}} = p_g h_g(t) 10^{-\bar{g}_{\mathrm{FU}g}(t)/10}$. The received SINR of the NU of group g is

$$\Gamma_{\mathrm{NU}g}(t) = \frac{p_g h_g(t) 10^{-\bar{g}_{\mathrm{NU}g}(t)/10}}{\sigma^2}. \qquad (9)$$

The SINR received at the FU of group g is

$$\Gamma_{\mathrm{FU}g}(t) = \frac{p_g(1-h_g(t)) 10^{-\bar{g}_{\mathrm{FU}g}(t)/10}}{I_{\mathrm{NF}} + \sigma^2}. \qquad (10)$$

D. Transmission Model

In order to capture the dynamic of contents transmission and corresponding backhaul link delay from the MBS to the UAV, M content virtual queues are defined to represent the contents waiting to be transmitted from the MBS to the UAV. We define a backhaul transmission index of requested content m as $\rho_m(t)$, which is calculated as $\rho_m(t) = (1 - c_m(t)) \sum_{n=1}^{N} r_{nm}(t)$. It means $c_m(t) = 1$ if the requested content m has been cached, then content m would not be transmitted via the backhaul link, i.e., $\rho_m(t) = 0$. Otherwise, the UAV would fetch the content m from the MBS if the content has not been cached, $\rho_m(t) = 1$. Besides, when content m is proactively cached at time slot t, the UAV will fetch the content from the MBS if the content has not been cached, which is represented as $i_m(t)(1 - c_m(t))$. Hence, the virtual queue backlog for content m, denoted as

$\mu_m(t)$, evolves over time slot as

$$\mu_m(t) = \min[\rho_m(t) + i_m(t)(1 - c_m(t)), 1], \qquad (11)$$

where $\min[*, 1]$ means that for the same content m, the UAV only needs to request from the MBS once in a time slot.

The total bandwidth of the backhaul link is B_{B}. The backhaul transmission rate from the MBS to the UAV at time slot t is

$$R_{\mathrm{B}}(t) = B_{\mathrm{B}} \log_2(1 + \Gamma_{\mathrm{B}}(t)). \qquad (12)$$

As the bandwidth is equally allocated among content virtual queues, backhaul transmission delay of the requested content m at time slot t is expressed as

$$D_{\mathrm{B}m}(t) = \begin{cases} 0, & if \ \sum_{m=1}^{M} \mu_m(t) = 0, \\ \frac{C_1 \mu_m(t)}{\frac{1}{\sum_{m=1}^{M} \mu_m(t)} R_{\mathrm{B}}(t)}, & if \ \sum_{m=1}^{M} \mu_m(t) \neq 0, \end{cases} \qquad (13)$$

where C_1 is the size of each content.

Consider the transmission of the radio access links from the UAV to the users at time slot t, as we define above, $b_n(t) = 1$ represents that the requested content of user n is scheduled at time slot t. Thus, the frequency band of the radio access link will be fairly allocated to the NOMA user groups. We define the total radio access bandwidth of the UAV as B_A. According to (9) and (10), the data rates of the NU in group g is

$$R_{\mathrm{NU}g}(t) = \frac{2B_A}{\sum_{n=1}^{N} b_n(t)} \log_2(1 + \Gamma_{\mathrm{NU}g}(t)), \qquad (14)$$

and that of the FU, $R_{\mathrm{FU}g}(t)$, can be calculated by the similar way. The radio access transmission delay of user n is

$$D_{\mathrm{A}n}(t) = \varphi_{\mathrm{NU}g}(t) \frac{C_1}{R_{\mathrm{NU}g}(t)} + \varphi_{\mathrm{FU}g}(t) \frac{C_1}{R_{\mathrm{FU}g}(t)}, \qquad (15)$$

where $\varphi_{\mathrm{NU}g}(t) = 1$ if user n corresponds to the NU of group g, otherwise $\varphi_{\mathrm{NU}g}(t) = 0$. Similarly, $\varphi_{\mathrm{FU}g}(t) = 1$ if the user n corresponds to the FU in group g, otherwise $\varphi_{\mathrm{FU}g}(t) = 0$.

In this model, the users' requests may not be responded immediately and would be scheduled among time slots. Therefore, the scheduling delay of content requesting users could be expressed as $\left(\sum_{m=1}^{M} r_{nm}(t) - b_n(t)\right) \delta$, where δ is the length of time slot. Considering the quality of experience (QoE) of users, we assume an upper limit of scheduling delay, denoted as β.

E. Problem Formulation

Given the above models, our goal is to minimize the long-term content delivery delay in dynamic networks. To achieve this goal, we formulate a problem by jointly optimizing the caching placement of the UAV, the user scheduling of content requests, and the power allocation of NOMA users.

The considered sum content delivery delay of users at each time slot consists of transmission delay of backhaul link, transmission delay of downlink radio access link, and scheduling delay of content requesting users, and therefore

can be expressed as

$$u(t) = \underbrace{\sum_{m=1}^{M} D_{\mathrm{B}m}(t)}_{backhaul\ delay} + \underbrace{\sum_{n=1}^{N} D_{\mathrm{A}n}(t)}_{radio\ access\ delay} + \underbrace{\sum_{n=1}^{N} \left(\sum_{m=1}^{M} r_{nm}(t) - b_n(t) \right) \delta}_{scheduling\ delay}, \qquad (16)$$

where we ignore the uplink transmission delay and the processing delay. Obviously, those three parts can be zero. For example, backhaul delay will be zero if the requested content is cached by the UAV, and scheduling delay will be zero if request of users is responded immediately.

Remark 2. *From (16), we notice that the user number N affects the content delivery delay of networks. The larger N causes an increase in radio access delay and scheduling delay.*

According to (16), the long-term content delivery delay minimization problem can be expressed as

$$\min_{b,i,h} \sum_{t=1}^{T} u(t) \qquad (17a)$$

$$\text{s.t. } b_n(t) \in \{0,1\}, \forall n, t, \qquad (17b)$$

$$i_m(t) \in \{0,1\}, \forall m, t, \qquad (17c)$$

$$0 \le h_g(t) \le 1, \forall t, g, \qquad (17d)$$

$$b_n(t) \le \sum_{m=1}^{M} r_{nm}(t), \forall t, \qquad (17e)$$

$$\sum_{t-\beta+1}^{t} \left(\sum_{m=1}^{M} r_{nm}(t) - b_n(t) \right) < \beta, \forall n, t, \qquad (17f)$$

$$\sum_{m=1}^{M} i_m(t) \le Z, \forall t, \qquad (17g)$$

where constraints (17b) and (17c) show that the values of $b_n(t)$ and $i_m(t)$ should be either 0 or 1. Constraint (17d) shows the range of power allocation coefficient $h_g(t)$ for the NOMA. Constraint (17e) guarantees that only the users waiting for response could be responded. Considering QoE, constraint (17f) is assigned to limit the scheduling delay. Constraint (17g) guarantees the sum of the proactive cache scheme at time slot t should be no more than cache capacity of UAV. It is obvious that formulated problem (17) is NP-hard, which is demonstrated in appendix A.

III. REINFORCEMENT LEARNING BASED ALGORITHM

We convert problem (17) into a MDP to cope with the dynamic UAV locations and the content requests. Machine learning (ML) has emerged as a powerful artificial intelligence (AI) technique to make the UAV wireless communication highly efficient [35]. Then the Q-learning based caching placement and resource allocation algorithm is used to tackle the MDP. However, the efficiency of the Q-learning based algorithm is limited in the scenario with huge state/action space. Therefore, we use the function approximation based

caching placement and resource allocation algorithm to solve the proposed problem for large-scale networks.

A. Problem Conversion

As the optimize objective of formulated problem is to minimize the content delivery delay in dynamic networks, we convert the formulated problem (17) into a MDP, where the UAV acts as an agent. The proposed MDP consists of four components: a) set of finite states, b) set of finite actions, c) dynamic change of states, which describes how current state and action influence the future state, and d) the cost function defined in (16).

Since we consider the networks are dynamic with the requests of users for M contents in cache-enabling UAV networks, the states of the MDP is characterized by users waiting for response $W(t)$ and cache situation of UAV $C(t)$ as follows. $W(t) = [w_1(t), \cdots, w_n(t), \cdots, w_N(t)]$ denotes the users waiting for response at time slot t, which consists of postponed users of previous time slots and users request for contents at time slot t. As mentioned above, $r_{nm}(t)$ is defined to represent request of user n to content m at time slot t. Thus, $w_n(t)$ could be calculated as $w_n(t) = \sum_{m=1}^{M} r_{nm}(t)$. $C(t) = [c_1(t), \cdots, c_m(t), \cdots, c_M(t)]$ denotes the cache situation of M contents at time slot t. The state vector of the proposed MDP at time slot t is defined as $s(t) = [C(t), W(t)]$. The state space S is equivalent to all possible combination of cache situation and users waiting for response. The content request model can be extended to time varying user preference occasion, where the preference of users can be included in state variables to cope with the complex dynamic environment.

For the action of the MDP, we denote $I(t) = [i_1(t), \cdots, i_m(t), \cdots, i_M(t)]$ as the proactive caching index of M content at time slot t. $B(t) = [b_1(t), \cdots, b_n(t), \cdots, b_N(t)]$ represents the scheduling to N users at time slot t. $h_g(t)$ is the power allocation of NU of group g at time slot t. In order to form a limited action space, we describe the power allocation coefficient of NU with O discrete power levels, i.e., $h_g(t) \in \{h^1, \cdots, h^O\}$. The power allocation coefficient of FU could be calculated as $1 - h_g(t)$. Then power allocation of NOMA users could be represented by power allocation coefficient of NU. The discrete power levels are reasonable since the transmission power is discrete in practical communication equipments. The action vector executed by UAV at time slot t could be expressed as $a(t) = (I(t), B(t), H(t))$, which consists of caching placement of UAVs, user scheduling of content requests, as well as power allocation of NOMA users at time slot t. We define action space as a set of all possible combinations of these three factors, which could be express as $A = I \otimes B \otimes H$, with $\otimes$ denoting the Cartesian product.

Then, the characteristic of dynamic state is introduced. At time slot $t+1$, the environment would switch to a new state $s(t+1)$, which is determined by previous state $s(t)$ and action selection $a(t)$. First, cache condition of content m at time slot $t+1$ could be updated according to caching placement of time slot t, and acts as the element

of $C(t+1)$. Then, users waiting for response could be calculated $w_n(t+1) = \sum_{m=1}^{M} r_{nm}(t+1)$. For convenience, rule of state switch is summarized as $\xi(s(t), a(t), s(t+1))$. When the UAV takes an action $a(t)$ at time slot t, we could get the instantaneous cost considering current state $s(t)$. The instantaneous cost is defined as the instantaneous sum delay $u(t)$, which is calculated according to (16).

The MDP could be described as tuple $\Gamma = (S, A, \xi, u)$. The optimization objective of formulated problem is to minimize the long-term sum delay, which is solved by Q-learning.

B. Q-Learning

Q-learning is a RL method for solving the problem modeled after MDPs, where a learning agent operates in an unknown environment [36]. At each time slot t, the UAV acts as an agent to confirm current state $s(t) \in S$, select action $a(t) \in A$, and get a cost $u(t)$, according to the current state and the action selection. Then, the environment switches to the next state $s(t+1)$, according to the current state and the selected action.

Two fundamental concepts of the algorithm for solving the above MDP are state-value function and action-value function (Q-function) [37]. The state-value function V is defined to measure the importance of states, which is $V(s, \pi) = E\left\{ \sum_{\tau=t}^{+\infty} \gamma^{\tau-t} u(\tau) | s(t) = s \right\}$, where $0 \leq \gamma < 1$ denotes the discount factor. The action-value function (Q-function) is defined to measure the importance of action, which can be expressed as $Q(s, a, \pi) = E\left\{ \sum_{\tau=t}^{+\infty} \gamma^{\tau-t} u(\tau) | s(t) = s, a(t) = a \right\}$. We can get the relationship between the state-value function and the action-value function [38]

$$V(s, \pi) = \sum_{a \in A} \pi(s, a) Q(s, a, \pi), \tag{18}$$

where $\pi(a|s)$ represents the probability that the agent with state s select action a at time slot t, i.e. $\pi(a|s) = P(a_t = a | s_t = s)$ [39].

As we introduced before, the target of the proposed MDP is to minimize the long-term cost by selecting the most suitable strategy. The optimal problem can be written as

$$\pi^* = \arg\min_\pi V(s), \tag{19}$$

where we define π^* to express the optimal action selection scheme. Combining (18) and (19), the optimal problem can be reformulated as

$$V^*(s) = \min_a Q^*(s, a). \tag{20}$$

In other words, we can get the optimal state-value by selecting the action with the least Q-value.

The update rule of the action value function [40] can be denoted as

$$Q_{t+1}(s, a) = (1 - \alpha(t)) Q_t(s, a) + \\ \alpha(t) \left\{ u(t) + \gamma \min_{a \in A} Q_t(s', a') \right\}, \tag{21}$$

where s and a represent the state and the action of time slot t respectively, and s' and a' represent the state and the action of time slot $t+1$ respectively. According to [40], we represent the learning rate as

$$\alpha(t) = \frac{1}{(t + c_\alpha)^{\varphi_\alpha}}, \tag{22}$$

where $c_\alpha > 0$ and $\varphi_\alpha \in (1/2, 1]$. The learning rate represents the impact of learning result to the Q-table.

Action selection mechanism is an important part of Q-learning. In this paper, we use the *soft ε-greedy* method to guide the action selection of the UAV. In the proposed method, instead of keeping a fixed probability of exploration in the action space, we define a gradually decreasing probability to randomly select actions,

$$\begin{cases} 1, & if \ t \leq \varepsilon, \\ \frac{\varepsilon}{t}, & if \ t > \varepsilon, \end{cases} \tag{23}$$

where ε is a positive number. According to (23), for the time slot smaller than ε, the UAV selects actions randomly. Besides, for the time slot bigger than ε, the UAV explores action space with a decreasing probability. Larger ε means that there would be more time slots to select an action randomly at the beginning of the iterations, which leads to a fast exploration of action space. When the action space has been explored in some extent, the Q-table would be more stable with a small probability of randomly select actions, as the random selection would influence the Q-value of actions of former time slots.

Remark 3. *From (23), we notice that the value of ε in the Q-learning affects the trade-off between exploration of action space and exploitation of the explored result. Larger ε achieves a sufficient exploration, but the high proportion of randomly selecting action affects the stability of long-term Q value calculation. Smaller ε achieves a rapidly decreasing content delivery delay, beacause of the generous exploitation for the explored result. However, the insufficient exploration for the action space limits the converged content delivery delay. We should point that the objective of this paper is not to investigate the optimal trade-off between exploration and exploitation by ε, which jointly decide convergence speed and converged value of content delivery delay.*

Moreover, legal action is defined in this paper. We could confirm that only a subset of the action set could be accessed for the specific states, due to the restrictions of the proposed problem. The actions that the agent could choose when the environment is in state $s(t)$ is defined as the legal action of that state. Instead of listing all possible states and actions, state list and action list are initially empty, and then gradually increase. When a new state s is experienced by the UAV, it will be saved in the state list. Besides, when a new action a is selected by the UAV, it will be saved in the action list. As mentioned above, the legitimacy of the action varies with different states, so the legitimacy is judged before the action is taken. In particular, the actions in the list are judged in the descend order of Q-value until a legal action is found. The UAV will randomly take a legal action if there is no legal action in the list.

Based on Q-learning, iterative caching placement and re-

source allocation algorithm is summarized in Algorithm 1.

Algorithm 1 Q-learning based caching placement and re-source allocation algorithm

---

Initialization:
 1: Set parameters
 2: Initialize state, Q-table, action list and state list.
Main Loop:
 3: **while** $t < T$ **do**
 4:    Update state list.
 5:    **if** $t \leq \varepsilon$ or $random[0,1] \leq \varepsilon/t$ **then**
 6:      Select a legal action randomly.
 7:    **else**
 8:      Sort the listed actions according to Q-table.
 9:      **if** the action with current optimal Q-value is legal **then**
 10:        Select the action.
 11:      **else**
 12:        Judge the action with sub-optimal Q-value and back to step (9).
 13:        **if** the action with sub-optimal Q-value is None **then**
 14:          Select a legal action randomly.
 15:        **end if**
 16:      **end if**
 17:    **end if**
 18:    Update action list, cost, Q-table, state.
 19: **end while**

C. Function Approximation Based Algorithm for Large-Scale Networks

In the proposed Q-learning based caching placement and resource allocation algorithm, the Q-table is too large to search and save in the large-scale networks since the size of the action space and the state space are all mainly related to the number of contents and users. In this subsection, we use a function approximation based caching placement and resource allocation algorithm to solve the proposed problem for large-scale networks in practical scenarios. The framework of the function approximation based algorithm is given in Fig. 3, where the stochastic gradient descent method (SGD) is used to search the action space efficiently and the deep neural networks (DNN) is adopted to overcome the limitations of Q-table storage.

In the traditional reinforcement learning algorithm, ε-greedy is deployed as an action selection scheme [1, 38], which is not efficient enough to search the large-scale action space in the formulated MDP of this paper. The insufficient search result will lead to a limitation for DNN training. Thus the large action space in our problem is searched with SGD, which is an efficient algorithm with low complexity.

In the proposed algorithm, we use the SGD to search the action space, whose result is stored in the memory matrix. The mappings between states and actions stored in the matrix act
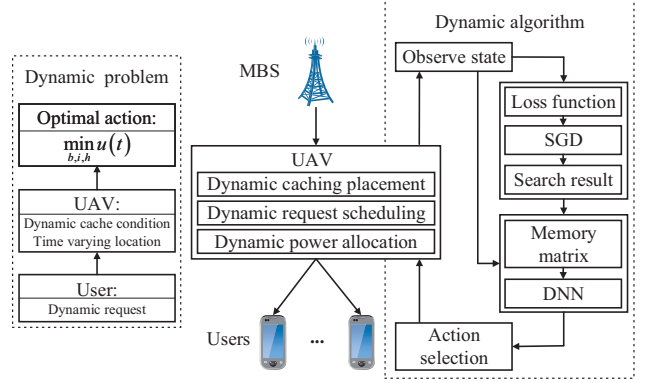

Fig. 3: Framework of the function approximation based algorithm.

as the supervisors for training of the DNN. Loss function is defined as

$$u(I, B, H) = \sum_{m=1}^{M} \frac{w_m(b_n, i_m)}{R_{\mathrm{B}}(b_n, i_m)} + \sum_{n=1}^{N} \left(\sum_{m=1}^{M} r_{nm} - b_n \right) \delta + \sum_{n=1}^{N} \left(\varphi_{\mathrm{NU}g} \frac{C_1}{R_{\mathrm{NU}g}(b_n, h_g)} + \varphi_{\mathrm{FU}g} \frac{C_1}{R_{\mathrm{FU}g}(b_n, h_g)} \right),$$

(24)

where $I = \{i_1, \cdots, i_M\}$, $B = \{b_1, \cdots, b_N\}$, and $H = \{h_1, \cdots, h_G\}$. Besides, discrete variables in vector $\{I, B, H\}$ are relaxed to continuous variables. Variables i_m, b_m, and h_g are updated according to the SGD method in [41] to minimize the loss function.

There are constraints to solution $\{I, B, H\}$ of the optimization problem. Before storing the mappings in the memorize matrix, we transform the output continuous values to the form that meets the constraints. According to (17c) and (17g), we sort all the elements of $I = \{i_1, \cdots, i_M\}$ in the descending order and the j_2th element is $i_{s_{j_2}}$, which corresponds to the s_{j_2}th content before sorting for $j_2 = 1, 2, \cdots, M$. Then, the largest Z elements are set as 1, i.e. $i_{s_{j_2}} = 1$ for $j_2 = 1, \cdots, Z$, where Z is the cache capacity of the UAV. Besides, we convert b_n to Boolean variable with 0.5 as the borderline, according to (17b). According to (17d), continuous variables H are discretized. Then, mapping between the current state and action selection is stored in the memory matrix, where the capacity of the matrix is limited. In particular, if the memory matrix is filled, the longest remembered mapping will be replaced by the mapping of time slot t.

In the traditional reinforcement learning algorithm, Q-table is a common method to store the mappings between states and actions. The large state space and action space in the formulated MDP of this paper make the Q-table difficult to store and search. Thus we use a DNN to approximate the relationship between states and actions.

The action selection is treated as a black box and the DNN is deployed to learn relation between states and actions because of DNN's universal approximation ability [42]. The

TABLE II: Time complexity and execution time of algorithms

Algorithm	Time complexity	Execution time
Q-learning based algorithm	$O\left(\sum_{\tau_1=1}^{T_1} A_e\left(M, N, Z, \tau_1\right)\right)$	593.774 s
Function approximation based algorithm	$O\left(T_1 \max\left(T_2\left(1.5N + M\right), T_3(1.5N + M)^2\right)\right)$	312.727 s
Greedy based exhaustive search algorithm	$O\left(T_1 A\left(M, N, Z\right) \max\left(M, N\right)\right)$	104482.649 s
Fixed algorithm	$O\left(T_1 \max\left(M, N\right)\right)$	99.313 s
Random algorithm	$O\left(T_1 \max\left(M, N\right)\right)$	104.915 s

proposed algorithm contains two stages, training process and testing process. The formulated DNN is trained by memory replay. During the training process of the DNN based function approximation model, samples in the memory matrix act as the supervisor. During the testing process, we could select the optimal action for state s according to the output of the model.

In the DNN based function approximation model, the mappings in memory matrix are replied to train the neural network, where the state and action samples act as the supervisor. Moreover, optimal action selection could be decided by DNN, with state vector s as input. The agent in our DNN is the UAV. The DNN uses an input vector $X\left(t\right) = \left[C\left(t\right), W\left(t\right)\right]$ to represent the state at time slot t. Moreover, the output vector of the DNN is $Y\left(t\right) = \left[y_1\left(t\right), \cdots, y_K\left(t\right)\right]$, which represents the user scheduling of content requests, caching placement of UAV, as well as power allocation of NOMA users for the current state, and $K = M + N + N/2$. We respectively set the input layer and the output layer as $Net_X \in R^{(M+N)\times 1}$, and $Net_Y \in R^{K\times 1}$. The hidden layers consist of feature maps $Net_a \in \mathbb{R}^{4\times 4}$, $Net_b \in \mathbb{R}^{2\times 2}$ and fully connected layer $Net_c \in \mathbb{R}^{4\times 1}$.

The DNN model consists of the input weight matrix $W_a \in \mathbb{R}^{(M+N)\times 16}$, convolution kernel matrix $W_b \in \mathbb{R}^{3\times 3}$, weight matrix W_c, and W_d. Besides, activation function can be rectified linear unit, $f_{relu}\left(x\right) = \max\left(0, x\right)$, and batch normalization is deployed to maintain the network stability. We employ gradient truncation to prevent gradient explosion. The bias matrices have the same size of the corresponding weight matrices.

In order to build the relationship between the input Net_X and the output Net_Y, weight matrices and bias matrices need to be trained. During the traning stage, the model is trained to minimize the distance between the DNN's output and the action in memory, which can be measured by

$$L_1\left(t\right) = \frac{1}{K}\sum_{k=1}^{K}\left(y_k\left(t\right) - y_k\left(t\right)'\right)^2, \qquad (25)$$

where $Y' = \left[y_1', \cdots, y_k'\right]$ represents the action in the memory matrix. Then weight matrices and bias matrices could be respectively updated according to SGD. When the agent is faced with a state, the proposed DNN model can output a vector, which contains user scheduling of content requests, caching placement of UAV, as well as power allocation of NOMA users.

Based on the function approximation method, the proposed dynamic caching placement and resource allocation algorithm is summarized in Algorithm 2.

Algorithm 2 Function approximation based caching placement and resource allocation algorithm

---

Initialization:

1: Randomly initialize starting state including cache situation $W\left(t\right)$, and users waiting for response $C\left(t\right)$. Set time slot $t = 1$, The period for the reset of DNN is T;

Main Loop:

2: **for** time slot t **do**

3:     Select an action randomly.

4:     **for** iteration τ_1 **do**

5:         **SGD**: Update the action according to SGD.

6:     **end for**

7:     Store current state and optimal action selection according to SGD in the memory matrix.

8:     **if** t is a multiple of T **then**

9:         **DNN Reset**: Randomly initialize networks nodes Net_X, Net_a, Net_b, Net_c, Net_Y, weight matrices W_a, W_b, W_c, W_d, and bias matrices.

10:         **for** iteration τ_2 **do**

11:             **DNN Training**: Update Net_X, Net_a, Net_b, Net_c, Net_Y, W_a, W_b, W_c, W_d, and bias matrices according to the mechanism introduced above.

12:         **end for**

13:     **end if**

14:     **Make decision**: Recursively calculate Net_Y. Convert the value of $Net_Y\left(1 : M + N, 1\right)$ to a Boolean variable, and convert the value of $Net_Y\left(M + N + 1 : M + N + N/2, 1\right)$ to discrete values.

15: **end for**

D. Analysis of the Proposed Algorithms

1. Complexity : The time complexity of the Q-learning based algorithm, function approximation based algorithm, greedy based exhaustive search algorithm, fixed algorithm, and random algorithm are listed in Table II. The number of considered time slot is T_1, we defined $A_e\left(M, N, Z, \tau_1\right)$ to express the size of the explored action space, which is jointly determined by the size of the complete action space and current time slot τ_1. We define $A\left(M, N, Z\right)$ to express the size of the complete action space, which is jointly determined by the number of users, the number of contents and the cache capacity of UAV. Considering the function approximation based algorithm, T_2 in $T_2\left(1.5N + M\right)$ represents the converged iteration of the SGD algorithm, $1.5N + M$ represents the dimension of the variables to optimize. T_3 in $T_3(1.5N + M)^2$ represents the size of the mini batch for DNN training, where

$(1.5N + M)^2$ represents the largest dimension of DNN weight matrices. We define $\max()$ to express that the bigger complexity is selected for the calculation of the total time complexity, because of the cascade relationship between SGD algorithm and the training of the DNN. Considering the greedy based exhaustive search algorithm, we define $A(M, N, Z)$ to express the size of the complete action space, which is determined by the number of users N, the number of contents M, and the cache capacity of UAV Z. As the cost of every action is calculated in the greedy based exhaustive search algorithm, the complexity of every calculation is defined as $\max(M, N)$. We also compare the execution time of algorithms with iteration $T_1 = 10^5$ in the matlab simulation software, where $M = 8$, $N = 4$, $Z = 2$, $T_2 = 200$ and $T_3 = 32$. The type of central processing unit of the laptop is Intel(R) Core(TM) i5-7300HQ, with the calculation frequency of 2.50 GHz. The length of time slot is 0.05s, which is assumed according to the movement of the UAV. However, the execution time of the greedy based exhaustive search algorithm is much bigger than the sum length of time slots. This is because that the exhaustive search algorithm puts forward high requirement on the performance of the computer, which proves the importance of proposing algorithms to get the trade-off between performance and complexity.

2. Compatibility of the proposed solution to the networks with multi-MBSs : The solution proposed in this paper is also suitable for the networks with multi-MBSs. For the multi-cell networks with orthogonal frequency resource, the solution proposed in this paper can be applied directly. For the multi-cell networks with frequency reuse, the SINR of transmission links is influenced by the interference from neighbor cells' MBSs and UAVs. The optimization problem for the cellular networks with multiple MBSs and UAVs is relatively complicated, which can be splitted into multiple optimization problems focused on resource allocation optimization of a single UAV [44]. The proposed algorithm and the formulated problem can be applied to the cellular networks with multi-MBSs.

IV. PERFORMANCE EVALUATION

The simulation results are provided in this section to verify the proposed algorithms. We consider a cellular network with multiple MBSs and multiple users. One MBS with heavy traffic load is selected as the target MBS, which is aided by a mobile UAV for traffic offloading and content caching.

In the simulation, we consider one target cell with six neighbor cells, where the side length is 100 m. We assume that the MBSs are loacted in the central of each cell. Users are ramdomly distributed in the cell of the target MBS. During the considered period, the users' location remains the same. We assume that the flight trajectory of the UAV is predetermined in a circle [29] with 200 m diameter and random central point in the cell of the target MBS. We assume that the speed of the UAV is 20m/s. The length of time slot considered in this paper is $\delta = 0.05s$. The upper limit of scheduling slot is assumed to be $\beta = 2$. We assume the request generate coefficient $R_g = 2$, which can be extended to various request generate coefficients

TABLE III: System Parameters

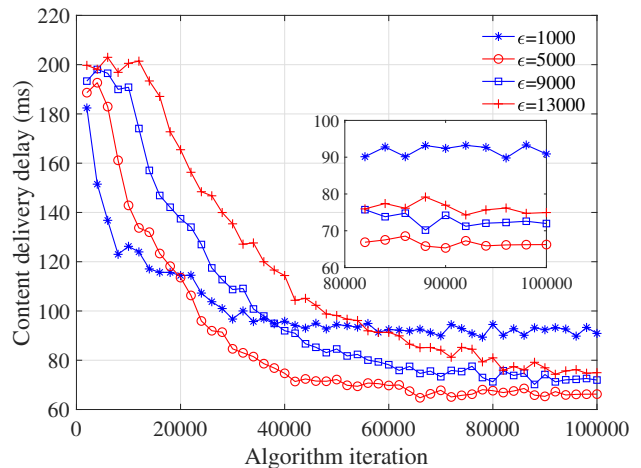| Power of MBS p_{ma}	46 dBm
Power of UAV p_{uav}	30 dBm
Bandwidth of backhaul link B_B	20 MHz
Bandwidth of radio access link B_A	20 MHz
Noise power σ	-174 dBm/Hz
UAV flight altitude h	100 m
Long-term period T	100

Fig. 4: Convergence of content delivery delay.

with the limitation of $R_g \leq \frac{N}{\beta}$. The scheduling method for the users with requests in the same time slot is based on frequency division multiplexing. There is a content library with M unified contents with the same data size. The size of each unified content is $C_1 = 2$ MB. The cache space of the UAV is $Z = 2$. The system can be extended to various content size case easily. The different content files with various data size can be reshaped into unified contents by UAV caching. The number of power allocation levels to NOMA users is 5, which is expressed as $h_g(t) \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The main simulation parameters follow the 3GPP specifications [32], which are summarized in Table III.

We compare the proposed Q-learning (QL) based algorithm and function approximation (FA) based algorithm with the benchmark algorithms to evaluate their effectiveness. The benchmark algorithms are defined as follows.

- Greedy algorithm: the UAV selects the optimal action in current state by exhaustive search to get the optimal instantaneous content delivery delay of current state;
- Fixed algorithm: the UAV caches the most popular contents in previous states, schedules the requesting users with round robin method, and allocates fixed power level for NOMA users;
- Random algorithm: the UAV selects actions randomly for content caching and radio resource allocation.

First, we verify the convergence of the proposed QL based algorithm by Fig. 4 and Fig. 5. In this simulation, we set $M = 4$ and $N = 8$. Fig. 4 depicts the content delivery delay convergence of the proposed QL based algorithm with differ-
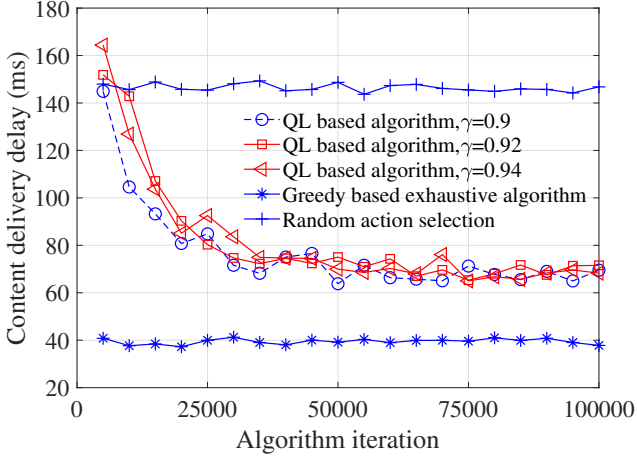
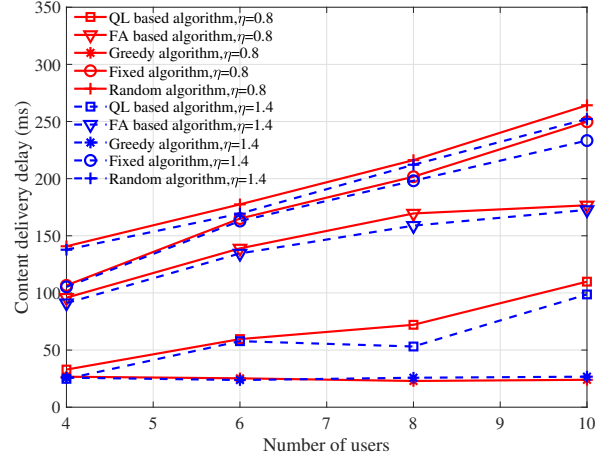Fig. 5: Content delivery delay comparison with iteration numbers.

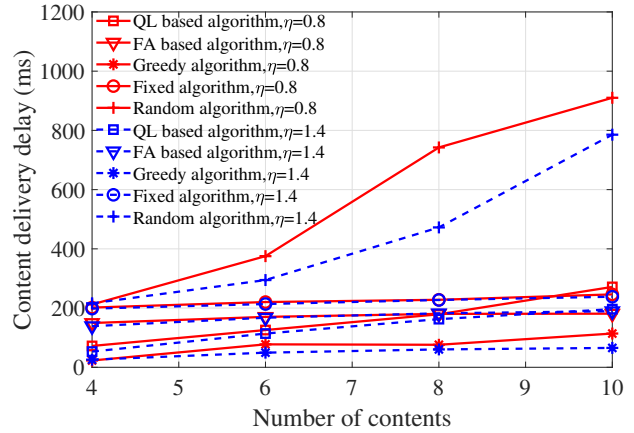Fig. 6: Content delivery delay comparison with varying user numbers in small-scale scenario.

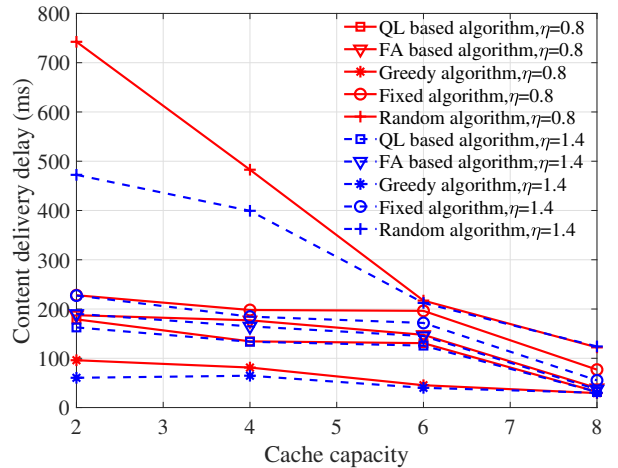Fig. 7: Content delivery delay comparison with varying content numbers in small-scale scenario.

Fig. 8: Content delivery delay comparison with varying cache capacity in small-scale scenario.

ent action selection parameters ε with the discount factor γ as 0.9. In the proposed algorithm, the agent selects the action with the smallest Q-value in addition to exploration. In Fig. 4, the content delivery delay converges gradually with the number of iterations increasing, ignoring the shaking caused by the state transition. Moreover, Fig. 4 indicates that the content delivery delay with different ε has different convergence speed and converged value. It is observed that content delivery delay of the algorithm with $\varepsilon = 13000$ decreases slowly at the beginning of iterations. This is because that although the action space is generously explored, but content delivery delay decreases slowly with insufficient exploitation of the explored actions. Because of the high proportion of selecting action randomly, the Q value calculating of the algorithm with $\varepsilon = 13000$ is unstable. As a result, the converged content delivery delay of the algorithm is relatively bigger. As we observe from Fig. 4, the algorithm with smaller $\varepsilon = 1000$ achieves a rapid decrease of content delivery delay at the beginning of iterations, as the exploration result is efficiently exploited. However, converged content delivery delay of the algorithm is relatively bigger because of the insufficient exploration of the action space. In the simulation, the content delivery delay with $\varepsilon = 5000$ achieves a good trade-off between exploration of action space and exploitation of the result. As a result, the algorithm with $\varepsilon = 5000$ achieves lower converged content delivery delay, which also achieves a better convergence speed than the algorithm with $\varepsilon = 9000$ and $\varepsilon = 13000$. This is a good proof of **Remark 3**. Fig. 5 shows the content delivery delay comparison with iteration numbers. The performance of the QL based algorithm is evaluated with different discount factors $\gamma = \{0.9, 0.92, 0.94\}$. It is observed from Fig. 5 that the content delivery delay of the QL based algorithm decreases and converges gradually with the number of iterations. This is because the probability that the UAV choosing the optimal actions increases with iterations. Compared with the benchmark algorithms, the converged delay of the proposed algorithm is much smaller than that of the random algorithm, and approaches the greedy based exhaustive algorithm.

Fig. 9: Cache hit radio comparison with varying content numbers in small-scale scenario.
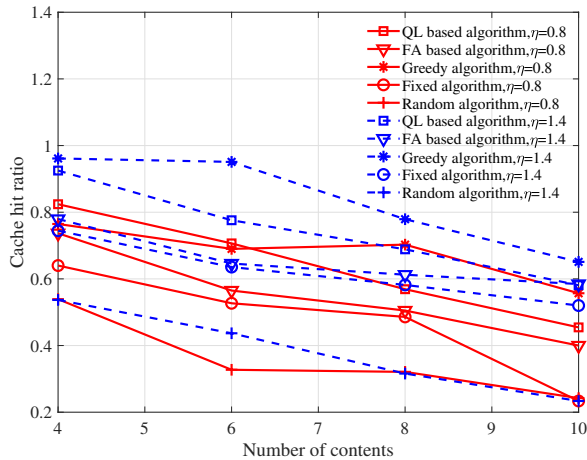
Fig. 10: Content delivery delay comparison with varying user numbers in large-scale scenario.

Then, we evaluate the proposed QL based algorithm and the proposed FA based algorithm in a small scale network with varying number of users, number of contents and cache capacity in Fig. 6, Fig. 7, and Fig. 8, where $\varepsilon = 5000$ and $\gamma = 0.9$. The performance of the proposed algorithms and the benchmark algorithms are evaluated with different parameters of Zipf distribution $\eta = \{0.8, 1.4\}$. Fig. 6 shows the content delivery delay versus different numbers of users in the network with $M = 4$, where the content delivery delay increases monotonically with the number of users in the network. Fig. 7 demonstrates the content delivery delay versus different numbers of contents in the network with $N = 8$, where the content delivery delay of the proposed algorithms increases with the number of contents. This is because that increase of number of the contents leads to bigger state space and legal action space, which reduces the probability that the optimal legal action is selected. Fig. 8 demonstrates the content delivery delay versus different cache capacity in the network with $N = 8$ and $M = 8$, where the content delivery delay decreases with the cache capacity of the UAV. As we can observe from both Fig. 6, Fig. 7, and Fig. 8, the proposed algorithms achieve much smaller content delivery delay than the fixed algorithm and the random algorithm. The performance gap between the proposed algorithms and the greedy algorithm is relative small in the simulation. However, the complexity of the proposed algorithms is much lower than that of the greedy algorithm especially when the size of the action space increases sharply with the number of users and contents increasing. Though there is a certain loss in the performance of the FA based algorithm compared to the QL based algorithm, the FA based algorithm is not limited by the action space and the state space. Thus the FA based algorithm could be deployed in large-scale networks. Fig. 6, Fig. 7, and Fig. 8 also show that the parameter of the Zipf distribution slightly affects the algorithms when the numbers of contents and users are small.

Besides that, Fig. 9 demonstrates the cache hit ratio versus different numbers of contents in the network with $N = 8$, where the content delivery delay of the proposed algorithms
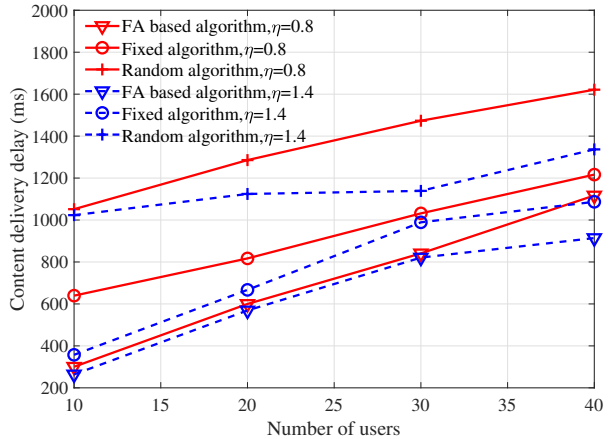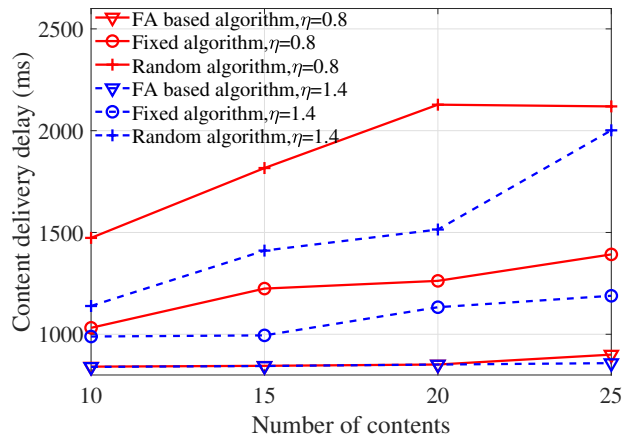

Fig. 11: Content delivery delay comparison with varying content numbers in large-scale scenario.

decreases with the number of contents. As we can observe from Fig. 9, the proposed algorithms achieve much higher cache hit radio than the fixed algorithm and the random algorithm. In some cases, the cache hit radio of QL based algorithm is better than greedy algorithm, this is because that the optimization objective is content delivery delay, which is not only determined by caching placement. Besides, the complexity of the proposed algorithms is much lower than that of the greedy algorithm.

Since the size of action space increases greatly in large-scale networks, it is inefficient for the traditional Q-learning to search the action space. To deal with this, we use the FA based algorithm, which is compared with the benchmark algorithms in a large-scale network in Fig. 10, Fig. 11, and Fig. 12. The popularity of contents is generated according to Zipf distribution with different parameters $\eta = \{0.8, 1.4\}$. As the complexity of the greedy algorithm increases greatly in large-scale networks as well, we use the fixed algorithm and the random algorithm as the benchmark algorithms in the following simulation. Fig. 10 reveals the performance of
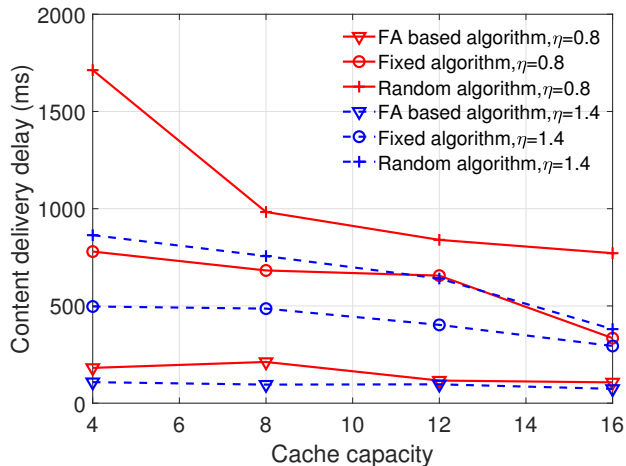
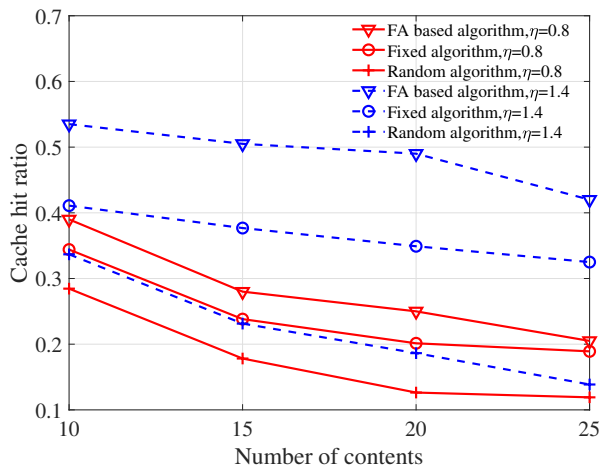Fig. 12: Content delivery delay comparison with varying cache capacity in large-scale scenario.

Fig. 13: Cache hit radio comparison with varying content numbers in large-scale scenario.

the FA based algorithm versus different numbers of users in the network with $M = 10$, where the content delivery delay increases monotonically with the number of users in the network. The increase of user number has a certain impact on the performance of the algorithm, which verifies our obtained insights in **Remark 2**. Fig. 11 reveals the performance of the FA based algorithm versus different numbers of contents with $N = 30$, where the content delivery delay of the FA based algorithm increases with the number of contents. The increase in user number will enhance the performance advantages of the FA based algorithm relative to the fixed algorithm and the random algorithm, which verifies the insights from **Remark 1**. Fig. 12 reveals the performance of the FA based algorithm versus different cache capacity in the network with $N = 30$ and $M = 20$, where the content delivery delay decreases monotonically with the cache capacity in the network. It is observed from Fig. 10, Fig. 11, and Fig. 12 that the FA based algorithm achieves much smaller content delivery delay than the fixed algorithm and the random algorithm. The reason is

that the FA based algorithm can make more reliable decisions with efficient search mechanism. Then, Fig. 13 demonstrates the cache hit ratio versus different numbers of contents in the network with $N = 30$, where the cache hit ratio of the proposed algorithm decreases with the number of contents. As we can observe from Fig. 13, the proposed algorithm achieves much higher cache hit ratio than the fixed algorithm and the random algorithm.

Besides, compared with the content delivery delay and cache hit ratio of small-scale networks, parameters of Zipf distribution play a considerable role in the performance of the FA based algorithm in large-scale networks. This is because the concentrated users' interest distribution reduces the stress on DNN, which no longer needs to approximate users' unusual requests.

V. CONCLUSION

This article has investigated the cache-enabling UAV NOMA networks. The cache-enabling mobile UAV serves the user groups by NOMA and caches limited popular contents for wireless backhaul link traffic offloading. To model the uncertainty of dynamic environment, we have formulated the long-term caching placement and resource allocation optimization problem as a MDP. We have defined the long-term sum delay of users as the content delivery cost, where the UAV acts as an agent. The actions taken by UAV correspond to caching placement, user scheduling and the power allocation of NOMA users. We have used the QL-based algorithm and the FA-based algorithm to solve the dynamic optimization problem. Finally, numerical results show that the proposed algorithms yield significant performance gains compared to the fixed algorithm and the random algorithm, and have acceptable calculation complexity. Moreover, the results also show that the FA-based algorithm is not limited by the scale of networks.

APPENDIX A
PROOF FOR NP-HARD OF (17)

Let $b_n(t) = r_{nm}(t)$ and $h_{NUg}(t) = h_{NU}^1$ to focus on the optimization on caching placement, then the proposed problem (17) could be conversed as

$$\min_{b,i,h} \sum_{t=1}^{T} u(t) \tag{26a}$$

$$\text{s.t.} \quad i_m(t) \in \{0,1\}, \forall m,t, \tag{26b}$$

$$\sum_{m=1}^{M} i_m(t) \le Z, \forall t, \tag{26c}$$

It is obvious that calculation complexity of problem (26) is not bigger than that of the formulated problem (17). Since the problem (26) could be reduced to a 0-1 package problem with in polynomial-time [43], problem (26) is NP-hard. Hence, problem (17) is NP-hard.

REFERENCES

[1] 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.

[2] Y. Liu, Z. Qin, Y. Cai, Y. Gao, G. Y. Li, and A. Nallanathan, "UAV communications based on non-orthogonal multiple access," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 52–57, Feb. 2019.

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

[4] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020.

[5] Z. Xiao, H. Dong, L. Bai, D. O. Wu, and X. Xia, "Unmanned aerial vehicle base station (UAV-BS) deployment with millimeter-wave beamforming," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1336–1349, Feb. 2020.

[6] T. Hou, Y. Liu, Z. Song, X. Sun, and Y. Chen, "Multiple antenna aided NOMA in UAV networks: A stochastic geometry approach," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1031–1044, Feb. 2019.

[7] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M. Alouini, "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[8] L. Bai, L. Zhu, Q. Yu, J. Choi, and W. Zhuang, "Transmit power minimization for vector-perturbation based NOMA systems: A sub-optimal beamforming approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 5, pp. 2679–2692, May 2019.

[9] F. Cui, Y. Cai, Z. Qin, M. Zhao, and G. Y. Li, "Multiple access for mobile-UAV enabled networks: Joint trajectory design and resource allocation," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4980–4994, Jul. 2019.

[10] X. Diao, J. Zheng, Y. Wu, Y. Cai, and A. Anpalagan, "Joint trajectory design, task data, and computing resource allocations for NOMA-based and UAV-assisted mobile edge computing," *IEEE Access*, vol. 7, pp. 117 448–117 459, 2019, doi:10.1109/ACCESS.2019.2936437.

[11] Y. Chen, N. Zhao, Z. Ding, and M. Alouini, "Multiple UAVs as relays: Multi-hop single link versus multiple dual-hop links," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6348–6359, Sep. 2018.

[12] Y. Liu, Z. Qin, M. Elkashlan, Y. Gao, and L. Hanzo, "Enhancing the physical layer security of non-orthogonal multiple access in large-scale networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1656–1672, Mar. 2017.

[13] Y. Liu, Z. Qin, M. Elkashlan, A. Nallanathan, and J. A. McCann, "Non-orthogonal multiple access in large-scale heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2667–2680, Dec. 2017.

[14] Q. Yu, C. Han, L. Bai, J. Wang, J. Choi, and X. Shen, "Multiuser Selection Criteria for MIMO-NOMA Systems With Different Detectors," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1777–1791, Feb. 2020.

[15] Z. Yang, C. Pan, Y. Pan, Y. Wu, W. Xu, M. Shikh-Bahaei, and M. Chen, "Cache placement in two-tier HetNets with limited storage capacity: Cache or buffer?" *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5415–5429, Nov. 2018.

[16] X. Li, X. Wang, K. Li, Z. Han, and V. C. M. Leung, "Collaborative multi-tier caching in heterogeneous networks: Modeling, analysis, and design," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6926–6939, Oct. 2017.

[17] T. Zhang, Y. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Cache-enabling UAV communications: Network deployment and resource allocation," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2020, doi:10.1109/TWC.2020.3011881.

[18] 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.

[19] 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.

[20] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, "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.

[21] 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.

[22] F. Malandrino, Z. Limani, C. Casetti, and C. Chiasserini, "Interference-aware downlink and uplink resource allocation in HetNets with D2D support," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2729–2741, May 2015.

[23] W. Zeng, Y. R. Zheng, and R. Schober, "Online resource allocation for energy harvesting downlink multiuser systems: Precoding with modulation, coding rate, and subchannel selection," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5780–5794, Oct. 2015.

[24] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "AC-POCA: Anticoordination Game Based Partially Overlapping Channels Assignment in Combined UAV and D2D-Based Networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.

[25] Q. Wu and R. Zhang, "Common throughput maximization in UAV-enabled OFDMA systems with delay consideration," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6614–6627, Dec. 2018.

[26] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On a novel adaptive UAV-mounted cloudlet-aided recommendation system for LBSNs," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 4, pp. 565–577, Oct. 2019.

[27] S. Zhang, H. Zhang, and L. Song, "Beyond D2D: Full dimension UAV-to-everything communications in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6592–6602, Jun. 2020.

[28] Y. Cai, F. R. Yu, J. Li, Y. Zhou, and L. Lamont, "Medium access control for unmanned aerial vehicle (UAV) ad-hoc networks with full-duplex radios and multipacket reception capability," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 390–394, Jan. 2013.

[29] 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.

[30] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, "Cooperative non-orthogonal multiple access with simultaneous wireless information and power transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 938–953, Apr. 2016.

[31] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3417–3442, Fourthquarter 2019.

[32] 3GPP, "3GPP TR 36.777," in *Study on Enhanced LTE Support for Aerial Vehicles(Release 15)*, Dec. 2017.

[33] Y. Liu, M. Elkashlan, Z. Ding, and G. K. Karagiannidis, "Fairness of user clustering in MIMO non-orthogonal multiple access systems," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1465–1468, Jul. 2016.

[34] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

[35] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[36] 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.

[37] N. Zhao, Y. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.

[38] G. Tesauro, "Extending Q-learning to general adaptive multi-agent systems," in *NIPS*, 2004, pp. 871–878.

[39] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[40] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, no. 6, pp. 1185–1201, Jun. 1994.

[41] X. Yang, X. Zheng, and H. Gao, "SGD-based adaptive NN control design for uncertain nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5071–5083, Oct. 2018.

[42] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[43] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 95–107, Jan. 2016.
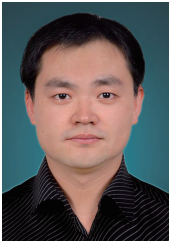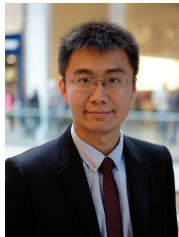
Tiankui Zhang (M'10-SM'15) received the Ph.D. degree in Information and Communication Engineering and B.S. degree in Communication Engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2008 and 2003, respectively. Currently, he is a Professor in School of Information and Communication Engineering at BUPT. His research interests include wireless communication networks, mobile edge computing and caching, signal processing for wireless communications, content centric wireless networks. He had published more than 100 papers including journal papers on IEEE Journal on Selected Areas in Communications, IEEE Transaction on Communications, etc., and conference papers, such as IEEE GLOBECOM and IEEE ICC.
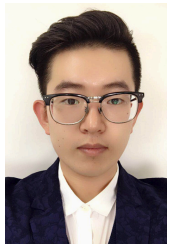
Ziduan Wang received the B.S. degree in Communication Engineering from University of Electronic Science and Technology of China (UESTC) in 2018. He is currently working toward the M.S. degree in Information and Communication Engineering from Beijing University of Posts and Telecommunications (BUPT), China. His current research focuses on caching placement and resource allocation in cache-enabling UAV NOMA networks.

Yuanwei Liu (S'13-M'16-SM'19) received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from the Queen Mary University of London, U.K., in 2016. He was with the Department of Informatics, King's College London, from 2016 to 2017, where he was a Post-Doctoral Research Fellow. He has been a Lecturer (Assistant Professor) with the School of Electronic Engineering and Computer Science, Queen Mary University of London, since 2017.

His research interests include 5G and beyond wireless networks, the Internet of Things, machine learning, and stochastic geometry. He has served as a TPC Member for many IEEE conferences, such as GLOBECOM and ICC. He received the Exemplary Reviewer Certificate of IEEE WIRELESS COMMUNICATIONS LETTERS in 2015, IEEE TRANSACTIONS ON COMMUNICATIONS in 2016 and 2017, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS in 2017 and 2018. He has served as the Publicity Co-Chair for VTC 2019-Fall. He is currently an Editor on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE COMMUNICATIONS LETTERS, and IEEE ACCESS. He also serves as a Guest Editor for IEEE JSTSP special issue on Signal Processing Advances for Non-Orthogonal Multiple Access in Next Generation Wireless Networks.

Wenjun Xu is a professor and Ph.D. supervisor in School of Information and Communication Engineering at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He received his B.S. and Ph.D. degrees from BUPT, in 2003 and 2008, respectively. He currently serves as a center director of the Key Laboratory of Universal Wireless Communications, Ministry of Education, P. R. China. He is a senior member of IEEE, and is now an Editor for China Communications. His research interests include AI-driven networks, UAV communications and networks, green communications and networking, and cognitive radio networks.

Arumugam Nallanathan (S'97-M'00-SM'05-F'17) is Professor of Wireless Communications and Head of the Communication Systems Research (CSR) group in the School of Electronic Engineering and Computer Science at Queen Mary University of London since September 2017. He was with the Department of Informatics at Kings College London from December 2007 to August 2017, where he was Professor of Wireless Communications from April 2013 to August 2017 and a Visiting Professor from September 2017. He was an Assistant Professor in the Department of Electrical and Computer Engineering, National University of Singapore from August 2000 to December 2007. His research interests include Artificial Intelligence for Wireless Systems, Beyond 5G Wireless Networks, Internet of Things (IoT) and Molecular Communications. He published nearly 500 technical papers in scientific journals and international conferences. He is a co-recipient of the Best Paper Awards presented at the IEEE International Conference on Communications 2016 (ICC'2016) , IEEE Global Communications Conference 2017 (GLOBECOM'2017) and IEEE Vehicular Technology Conference 2018 (VTC'2018). He is an IEEE Distinguished Lecturer. He has been selected as a Web of Science Highly Cited Researcher in 2016 and an AI 2000 Internet of Things Most Influential Scholar in 2020.

He is an Editor-at-Large for IEEE Transactions on Communications and Senior Editor for IEEE Wireless Communications Letters. He was an Editor for IEEE Transactions on Wireless Communications (2006-2011), IEEE Transactions on Vehicular Technology (2006-2017) and IEEE Signal Processing Letters. He served as the Chair for the Signal Processing and Communication Electronics Technical Committee of IEEE Communications Society and Technical Program Chair and member of Technical Program Committees in numerous IEEE conferences. He received the IEEE Communications Society SPCE outstanding service award 2012 and IEEE Communications Society RCC outstanding service award 2014.