Joint Resource, Deployment and Caching Optimization for AR Applications in Dynamic 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

The cache-enabling unmanned aerial vehicle (UAV) non-orthogonal multiple access (NOMA) networks for mixture of augmented reality (AR) and normal multimedia applications are investigated, which is assisted by UAV base stations. The user association, power allocation of NOMA, deployment of UAVs and caching placement of UAVs are jointly optimized to minimize the content delivery delay. A branch and bound (BaB) based algorithm is proposed to obtain the per-slot optimization. To cope with the dynamic content requests and mobility of users in practical scenarios, the original optimization problem is transformed to a Stackelberg game. Specifically, the game is decomposed into a leader level user association sub-problem and a number of power allocation, UAV deployment and caching placement follower level sub-problems. The long-term minimization was further solved by a deep reinforcement learning (DRL) based algorithm. Simulation result shows that the content delivery delay of the proposed BaB based algorithm is much lower than benchmark algorithms, as the optimal solution in each time slot is achieved. Meanwhile, the proposed DRL based algorithm achieves a relatively low long-term content delivery delay in the dynamic environment with lower computation complexity than BaB based algorithm.

Index terms— deep deterministic policy gradient, edge caching, non-orthogonal multiple access, Stackelberg game, unmanned aerial vehicle

This work was supported by National Natural Science Foundation of China under Grants 61971060. Part of this work has been presented at the 2020 IEEE Global Communications Conference (GLOBECOM) [1].

Tiankui Zhang, Ziduan Wang and Wenjun Xu are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 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, U.K. (e-mail: {yuanwei.liu, a.nallanathan}@qmul.ac.uk).

I. Introduction

Due to the quick deployment, flexibility and stable transmission channel of unmanned aerial vehicles (UAVs), deploying the UAVs to assist wireless communication has attracted much attention. Especially for the hotspot areas with temporarily high user density, where increasing fixed ground nodes is cost-ineffective, UAV-assisted communications show great advantages of high flexibility and easy deployment [2]. Caching the contents at UAVs has shown its effectiveness of alleviating the backhaul traffic of the existing terrestrial communication systems. Because of the ability to enhance both the access capability and spectral efficiency of networks [3], non-orthogonal multiple access (NOMA) has been used in UAV networks with the scenarios not limited to hotspot areas coverage.

The resource management in the UAV networks with normal multimedia service, where NOMA is deployed to enhance the communication ability, has received remarkable attention [4, 5]. However, for the scenario with dense mixture of augmented reality (AR) and normal multimedia requests, the complex data flow, which is different from that in the networks with unified service, necessitates a significant rethinking of resource allocation approaches. In the mixed scenario, the requested contents are sent to users requesting normal multimedia application. Meanwhile, both of the computation result package and requested contents are sent to the users requesting AR application. The UAVs employ NOMA to send the requested content to users based on the same wireless resource, which improves the spectrum efficiency. The conflict between complex data flow and the request for low content delivery delay puts forward a high demand for the access capability of wireless networks. Compared with normal access techniques, deploying NOMA is a promising method to provide efficient access for users, as the content transmitted by the same frequency band could be utilized to realize mixture AR and normal multimedia applications delivery. The objective of this article is to realize the potential of the UAV NOMA networks in the new scenario through intelligent resource allocation and UAV deployment.

A. Related Works

There have been research contributions related to cache-enabling UAV networks [6–11]. The authors in [6] jointly optimized wireless channels allocation and UAV's activity in cloud-enabled cellular networks with device-to-device to maximize the long-term reward. The authors in [7] optimized caching threshold between different subsets of content files to maximize the performance of cache-enabling UAV networks. The work in [8] was an early outstanding study on

online UAV control and content scheduling of cache-enabling UAV networks, whcih proposed an online UAV-assisted wireless caching design via jointly optimizing UAV trajectory, transmission power and caching content scheduling. Different from [8], we propose a framework of UAV NOAM networks for mixture of AR and normal multimedia applications delivery, and investigate the joint optimization of UAV deployment, caching placement and radio resource allocation with varying content requests and movement of users in this paper. The authors in [9] studied the user-centric information in cache-enabling UAV-assisted cellular networks, where the trajectory and caching placement were jointly optimized. The authors in [10] studied the joint optimization of UAV deployment and caching placement on IoT devices to maximize throughput among IoT devices. The authors in [11] studied the secure transmission for scalable videos in hyper-dense networks via cache-enabling UAVs.

The works in [12–18] have studied the problems related to UAV NOMA cellular networks. The authors in [12] proposed three study cases of UAV NOMA networks, including stochastic geometry based UAVs and ground users position model, joint trajectory planning and power allocation, and machine learning aided adaptive UAV placement. A single-antenna UAV was used to serve dense users by employing NOMA in [13], where a max-min rate optimization problem was formulated, with the constraints of power consumption, bandwidth allocation, UAV altitude, and antenna beamwidth. The authors in [15] focused on the uplink transmission of NOMA links in UAV-assisted cellular networks, where the information bits were offloaded to ground base station (BS). The authors in [16] studied cooperative NOMA to avoid interference of the links from users to UAV. The authors in [17] utilized NOMA to improve the spectrum scarcity of UAV-assisted cellular networks and investigated the viability of NOMA UAV network operating in realistic operating environments. The authors in [18] investigated the deployment of NOMA in UAV-assisted cellular networks to guarantee security.

Moreover, there have been research contributions about caching in virtual reality/augmented reality/mixed reality systems [19–21]. The authors in [19] studied the caching and transmission joint optimization for UAV networks, where users' reliability in virtual reality (VR) systems was improved by deploying UAVs to capture videos and transmit to small BSs. The authors in [20] jointly optimized caching placement and virtual viewport of video to maximize the overall quality of the 360^o videos delivered to the end-users. The authors in [21] considered both of edge caching and edge computing in the fog radio access networks based mobile AR delivery framework, where radio communication, caching policy and computing offloading were

jointly optimized to maximize the tolerant latency.

B. Motivation and Contribution

In UAV networks, the movement of UAVs should be considered in real scenarios. Besides, the real time movement and dynamic request of users are new challenges to the resource allocation problems, including time varying parameters of the formulated problems. To fill in this gap, this article studies a dynamic environment with ground users' and UAV's mobility as well as varying requests for contents, where UAVs serve users with the AR and normal multimedia content requesting by NOMA. Jointly considering resource allocation and UAV placement is very promising as UAVs can be placed strategically so that NOMA can achieve the best content delivery performance for ground users. Besides, jointly considering the resource allocation and UAVs caching placement improves the content delivery performance through optimizing user association, as serving users with the cache of UAV could obviously offload wireless backhaul traffic. Driven by it, we optimize the deployment of UAVs, the caching placement of UAVs and resource allocation, which includes user association and power allocation of NOMA. The main contributions of this article are summarized as follows.

- We propose a cache-enabling UAV NOMA framework for AR application, where both of the users requesting for AR application and users requesting for normal multimedia application are served by the UAV-assisted cellular networks. We assume that the contents are required for AR applications and normal multimedia applications. We define the long-term content delivery delay to express the sum downlink transmission delay of users in the dynamic environment with time varying request and moving users.
- We formulate an optimization problem to minimize the content delivery delay by dynamically optimizing the user association, power allocation of NOMA, deployment of UAVs and caching placement of UAVs. We use a branch and bound (BaB) based algorithm to achieve the per-slot optimization and obtain a local optimal solution, which provides a benchmark for the proposed long-term optimization problem.
- We formulate the original proposed problem as a Stackelberg game, which consists of a leader sub-problem and several follower sub-problems corresponding to UAVs. To achieve the long-term optimization in large-scale dynamic scenarios, we further propose a deep reinforcement learning (DRL) based algorithm for the formulated game. In particularly, we add a correction mechanism in DRL to optimize the users association in leader level. To

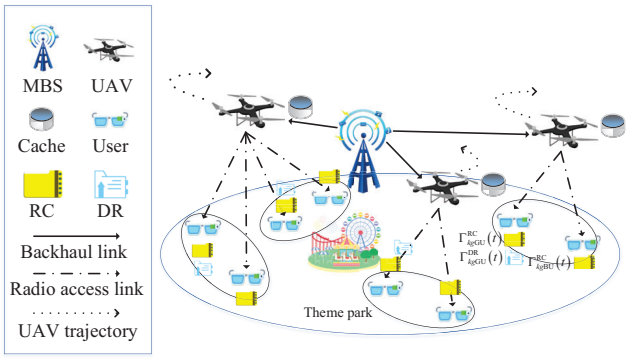

Fig. 1: Cache-enabling UAV NOMA networks for AR application.

mitigate the unobservable interference from the decision of other followers, we propose a meta actor network in DRL to jointly optimize the power allocation of NOMA, deployment of UAVs and caching placement of UAVs in follower level. The proposed DRL based algorithm is lower-complexity alternative solution of the BaB based algorithm.

- We demonstrate the performance of the proposed resource allocation, UAV deployment and caching placement algorithms by comparing them with the benchmark algorithms. The proposed BaB based algorithm achieves the optimal solution in each time slot, and performs better than benchmark algorithms. Meanwhile, the proposed DRL based algorithm obtains a good long-term networks performance in dynamic scenario with relatively lower complexity.

C. Organization

The rest of this article is organized as follows. Section II presents the system model. In Section III, we introduce the optimization problem for long-term content delivery delay minimization and propose a BaB based algorithm to solve the joint optimization problem. In Section IV, we formulate a Stackelberg game and further propose a DRL based algorithm to solve the long-term optimization in dynamic environment. Simulation results are presented in Section V. We present the conclusion in Section VI.

II. System Model

Considering the UAV-assisted cellular network with one macro BS (MBS), K UAVs, N ground users, as shown in Fig. 1. The UAVs connect to a macro BS (MBS) via wireless backhaul

links, and transmit the contents to corresponding users via radio access links. In our model, a dynamic scenario, which includes moving UAVs, users' time varying locations and users' real-time requsting for M contents is considered. According to the procedure of the AR applications described in [22], we assume that the contents are required by the AR application and the normal multimedia application both. Several additional computation steps are needed for AR application service, mainly including tracker, mapper and object detection, which are offloaded from users to UAVs. Detection result (DR) packages of computing result and the requested content (RC) m, $1 \leq m \leq M$, are sent to users requesting the AR application. For the users requesting normal multimedia application, only the RC m is transmitted. Since the users' compressed raw image required by AR computation is relatively small, we ignore the delivery delay in the uplinks from the AR requesting users to the UAVs. Two users requesting the AR application and normal multimedia application of the same content are grouped together and served by NOMA. Table. I is the summary of the notations.

TABLE I: Notation

Notation	Description	Notation	Description
N	Number of users	K	Number of UAVs
M	Number of contents	η	Zipf distribution parameter
$L_{\mathrm{uav}k}(t)$	Location of UAV k in time slot t	$L_n(t)$	Location of user n in time slot t
h	Height of UAV	$e_{nm}(t)$	Requesting index of AR application
$c_{km}(t)$	Proactive cache of UAV k	p_{Bk}^{LoS}	LoS channel probability
$D_{cn}(t)$	Computing delay	$D_{Bn}(t)$	Delay of backhaul link
N_0	Noise spectral density	$D_n(t)$	Sum delay of user n
C_1, C_2	Size of content and result package	J	Computing resource of each UAV
ω	Number of clock requested per bit calculation	ψ	CPU clock frequency of UAV
H	Size of data to be processed for AR	$r_{nm}(t)$	Requesting index of normal multimedia application
P_B, P_A	Power of MBS and UAVs	B_B, B_A	Bandwidth of MBS and UAVs
$D_{An}(t)$	Delay of radio access link of user n in time slot t	$R_{Bk}(t)$	Data rate of backhaul link of UAV k in time slot t
$\Gamma_{kg\mathrm{GU}}^{\mathrm{RC}}(t)$, $\Gamma_{kg\mathrm{GU}}^{\mathrm{DR}}(t)$, $\Gamma_{kg\mathrm{BU}}^{\mathrm{RC}}(t)$	SINR of three kinds of radio access link in time slot t	$R_{kg\mathrm{GU}}^{\mathrm{RC}}(t)$, $R_{kg\mathrm{GU}}^{\mathrm{DR}}(t)$, $R_{kg\mathrm{BU}}^{\mathrm{RC}}(t)$	Delay of three kinds of radio access link in time slot t
$q_{kn}(t)$	Indicator of user n served by UAV k in time slot t	$o_m(t)$	Content ranking in time slot t
$f_{ng1}(t)$	GU indicator of user n in group g in time slot t	$f_{ng2}(t)$	BU indicator of user n in group g in time slot t

A. UAVs and Users Mobility Model

We define the fly period of K UAVs is divided into certain time slots with the equal length δ to conveniently describe the movement of UAVs. The time slot index is t. We consider a three-dimensional deployment of UAVs, where the coordinate of the k-th UAV in time slot t is $L_{\text{uav}k}(t) = [x_{\text{uav}k}(t), y_{\text{uav}k}(t)]$. Referring to [23], we focus on the positions of UAVs at each time slot. The maximum of flying speed is $v_{\max}$.

The time varying locations of N users are modeled as a finite state Markov sequence [24]. The positions of MBS and user n are $L_B = [x_B, y_B]$ and $L_n(t) = [x_n(t), y_n(t)]$. The distance from UAV k to MBS in time slot t is given by

$$d_{Bk}(t) = \sqrt{h^2 + (x_{\text{uav}k}(t) - x_B)^2 + (y_{\text{uav}k}(t) - y_B)^2}. \tag{1}$$

The distance from UAV k to user n in time slot t is calculated in the same way of (1), which is expressed as $d_{Akn}(t)$. Because of the time varying locations of the UAVs and users, the distances between the UAVs and the MBS/users are time varying.

B. Caching and Computing Model

We assume the M contents are requested by users through the AR and normal multimedia applications with the same probability. We define $e_{nm}(t) = 1$ to indicate that user n requests RC m for the AR application, otherwise $e_{nm}(t) = 0$. The m-th RC with size C_1 and DR packages with size C_2 are sent to the users requesting for AR application. We define the probability of user n requesting RC m for the AR application follows Zipf distribution $\mathbb{P}(e_m(t)) = \frac{1/o_m(t)^\eta}{2\sum_{j_1=1}^{M} 1/{j_1}^\eta}$, where η is Zipf factor, $o_m(t)$ is the content ranking in time slot t, which is modeled as a finite state Markov sequence. We define $r_{nm}(t) = 1$ to indicate that user n makes normal multimedia request for RC m, with the same probability of $\mathbb{P}(e_m(t))$. The cache capacity of each UAV is Z_1. We define $c_{km}(t) = 1$ to indicate that RC m is proactive cached by UAV k in time slot t, otherwise $c_{km}(t) = 0$.

Considering the AR computation steps offloaded to UAVs, the requested content of user n is processed based on tracking information of users. We assume that the computation resource of each UAV is J. As indicated in [19], the time needed for processing one bit data is defined as $1/J = \omega/\psi$, where ψ represents central processing unit (CPU) frequency, ω is CPU cycles requirement for the computation of per bit data at the UAV. The computation resource of virtual server k is equally allocated to computation tasks of AR application connected to UAV k. We de-

$$\mu_{Bk}^{\mathrm{LoS/NLoS}}(t)=\begin{cases}30.9+(22.25-0.5\log_{10}h)\log_{10}d_{Bk}+20\log_{10}f_c, \text{ if LoS link}\\ \max\left\{\mu_{Bk}^{\mathrm{LoS}}(t),32.4+(43.2-7.6\log_{10}h)\log_{10}d_{Bk}+20\log_{10}f_c\right\}, \text{ if NLoS link}\end{cases} \tag{3}$$

$$p_{Bk}^{\mathrm{LoS}}(t)=\begin{cases}1,\ if\sqrt{d_{Bk}(t)^2-h^2}\le d_o,\\ \frac{d_o}{\sqrt{d_{Bk}(t)^2-h^2}}+\exp\left\{\left(\frac{-\sqrt{d_{Bk}(t)^2-h^2}}{p1}\right)\left(1-\frac{d_o}{\sqrt{d_{Bk}(t)^2-h^2}}\right)\right\},\ if\sqrt{d_{Bk}(t)^2-h^2}>d_o.\end{cases} \tag{4}$$

fine $q_{kn}(t)=1$ to indicate that user n associates with UAV k in time slot t, otherwise $q_{kn}(t)=0$. We define $q^c=\sum_{n=1}^{N}\left(q_{kn}(t)\sum_{m=1}^{M}e_{nm}(t)\right)$ to express the number of tasks connected to UAV k. The computation delay of user n in time slot t is given by

$$D_{Cn}(t)=\sum\nolimits_{k=1}^{K}q_{kn}(t)\,Hq^c/J, \tag{2}$$

where H is size of the data to be processed for AR application.

C. Channel Model

We model the path loss of wireless backhaul link and radio access link according to 3GPP specifications [25], where we consider quasi-static channels. This means that the channel condition remains constant within a time slot. We assume that different frequency bands are allocated to the wireless backhaul links and radio access links.

Considering the transmission of the wireless backhaul links, the path loss between UAV k and the MBS is stochastically determined by line-of-sight (LoS) and non-line-of-sight (NLoS) link states in (3), where f_c represents the carrier frequency. The LoS/NLoS link states are stochastically determined by the LoS probability defined in (4), where $d_o=\max\left[294.05\log_{10}h-432.94,18\right]$, and $p_1=233.98\log_{10}h-0.95$, and the NLoS probability, which is $p_{Bk}^{NLoS}=1-p_{Bk}^{LoS}$. The channel gain is given by

$$g_{Bk}(t)=\left[p_{Bk}^{LoS}10^{\mu_{Bk}^{LoS}/10}+p_{Bk}^{NLoS}10^{\mu_{Bk}^{NLoS}/10}\right]^{-1}. \tag{5}$$

The transmission rate from the MBS to UAV k in time slot t is

$$R_{Bk}(t)=\frac{B_B}{K}\log_2\left(1+\frac{P_Bg_{Bk}(t)}{B_BN_0}\right), \tag{6}$$

where P_B represents the power of the MBS, N_0 is noise spectral density, and B_B represents available backhaul bandwidth of the MBS.

We define $r_{Ukm}(t) = \sum_{n=1}^{N} q_{kn}(t)\left(r_{nm}(t) + e_{nm}(t)\right)$ as the request for RC m served by UAV k, and $w_{km}(t) = (1 - c_{km}(t-1))\min\left[c_{km}(t) + r_{Ukm},\ 1\right] = 1$ to indicate that RC m is transmitted from the MBS to UAV k when UAV k serves the users requesting RC m but does not cache RC m. The backhaul link transmission delay of user n in time slot t is given by

$$D_{Bn}(t) = \sum\nolimits_{k=1}^{K} q_{kn}(t)\frac{C_1 \sum_{m=1}^{M} w_{km}(t)}{R_{Bk}(t)}. \tag{7}$$

Considering the radio access links, two users that request for the AR application and normal multimedia application of the same content are grouped for NOMA transmission. The user requesting AR application is recognized as the good user (GU), and the user requesting normal application is recognized as the bad user (BU). The data of RC are transmitted to GU and BU by multicast, while the data of DR packages are transmitted to GU by unicast. As modeled in [26], the superposition signal of RC and DR packages are transmitted to the users based on superposition coding, then the GU could get the content of the AR application with RC and DR packages. The channel gain of NOMA links in time slot t are $\mu_{kg\mathrm{GU}}(t)$ and $\mu_{kg\mathrm{BU}}(t)$, which are calculated in the similar way of (3). The power allocation coefficient among the RC and DR signals in group g served by UAV k in time slot t is $\left[h_{kg}^{\mathrm{RC}}(t), h_{kg}^{\mathrm{DR}}(t)\right]$.

For the radio access link from UAV k to the GU of group g based on NOMA in time slot t, the interference from other UAVs is $I_{kg\mathrm{GU}}(t) = \sum_{i=1, i\neq k}^{K} P_A \mu_{ig\mathrm{GU}}(t)$, where P_A is total power of UAV. The interference from DR signal to RC signal at the GU of group g is $I_{kg\mathrm{GU}}^{\mathrm{DR}}(t) = P_A h_{kg}^{\mathrm{DR}}(t)\,\mu_{kg\mathrm{GU}}(t)$. The SINR for RC signal from UAV k to the GU of group g in time slot t is

$$\Gamma_{kg\mathrm{GU}}^{\mathrm{RC}}(t) = \frac{P_A h_{kg}^{\mathrm{RC}}(t)\,\mu_{kg\mathrm{GU}}(t)}{I_{kg\mathrm{GU}}^{\mathrm{DR}}(t) + I_{kg\mathrm{GU}}(t) + B_A N_0}, \tag{8}$$

where B_A is total bandwidth of UAV, which is allocated to the user groups equally. Based on serial interference cancelation (SIC), the GU first removes the signal of RC and then decodes the signal of DR. As we assume that the SIC deployed in this article has a certain probability of failure, the interference of RC signal received by GU is defined as $I_{kg\mathrm{GU}}^{\mathrm{RC}}(t) = P_A h_{kg}^{\mathrm{RC}}(t)\,\mu_{kg\mathrm{GU}}(t)$. The

SINR for DR signal from UAV k to the GU of group g in time slot t is

$$\Gamma_{kg\mathrm{GU}}^{\mathrm{DR}}(t)=\frac{P_A h_{kg}^{\mathrm{DR}}(t)\,\mu_{kg\mathrm{GU}}(t)}{\alpha I_{kg\mathrm{GU}}^{\mathrm{RC}}(t)+I_{kg\mathrm{GU}}(t)+B_A N_0}, \tag{9}$$

where α represents the imperfect SIC factor ranged from 0 to 1, i.e., $0\leq\alpha\leq 1$ [27].

Remark 1. *It is observed that the SINR of radio access link is influenced by SIC success probability. As a result, we know that the content delivery delay in this network is related to the imperfect SIC factor* α.

Considering the BU of group g, the interference from the DR signal to the RC signal at the BU of group g is $I_{kg\mathrm{BU}}^{\mathrm{DR}}(t)=P_A h_{kg}^{\mathrm{DR}}(t)\,\mu_{kg\mathrm{BU}}(t)$, the interference from other UAVs is $I_{kg\mathrm{BU}}(t)=\sum_{i=1,i\neq k}^{K}P_A\mu_{ig\mathrm{BU}}(t)$. The SINR for RC signal from UAV k to the BU of group g in time slot t is

$$\Gamma_{kg\mathrm{BU}}^{\mathrm{RC}}(t)=\frac{P_A h_{kg}^{\mathrm{RC}}(t)\,\mu_{kg\mathrm{BU}}(t)}{I_{kg\mathrm{BU}}^{\mathrm{DR}}(t)+B_A N_0+I_{kg\mathrm{BU}}(t)}. \tag{10}$$

The transmission rate from UAV k to the GU of group g for RC in time slot t is expressed as

$$R_{kg\mathrm{GU}}^{\mathrm{RC}}(t)=\frac{2B_A}{\sum_{n=1}^{N}q_{kn}(t)}\log_2\left(1+\Gamma_{kg\mathrm{GU}}^{\mathrm{RC}}(t)\right), \tag{11}$$

where 2 means that the same frequency band is shared among the GU and the BU in the same group. The transmission rate from UAV k to the GU of group g for DR $R_{kg\mathrm{GU}}^{\mathrm{DR}}(t)$ and the BU of group g for RC $R_{kg\mathrm{BU}}^{\mathrm{RC}}(t)$ are calculated in the similar way of (11).

For the convenience of expression, we utilize Boolean variable to express the request condition for AR application and normal multimedia application. The radio access link transmission delay of user n in time slot t is expressed with a single symbol, which is given by

$$D_{An}(t)=f_{ng1}(t)\left(\frac{C_1}{R_{kg\mathrm{GU}}^{\mathrm{RC}}(t)}+\frac{C_2}{R_{kg\mathrm{GU}}^{\mathrm{DR}}(t)}\right)+f_{ng2}(t)\,\frac{C_1}{R_{kg\mathrm{BU}}^{\mathrm{RC}}(t)}, \tag{12}$$

where $f_{ng1}(t)=1$ indicates that user n is the GU of group g in time slot t, $f_{ng2}(t)=1$ indicates that user n is the BU of group g in time slot t, and $f_{ng1}(t)+f_{ng2}(t)=1$.

Remark 2. *In the considered scenario, the users requesting for AR and normal multimedia application of the same content are served by NOMA. For the radio access link with limit radio resource, the deployment of NOMA improves the spectrum efficiency.*

III. Problem Formulation and Per-slot Optimization

In this section, we jointly optimize the user association, power allocation of NOMA, deployment of UAVs and caching placement of UAVs to minimize the long-term content delivery delay.

A. Problem Formulation

Considering the backhaul link transmission delay, the radio access link transmission delay and the computation delay, the content delivery delay of user n in time slot t is

$$D_n(t) = D_{Bn}(t) + D_{An}(t) + D_{Cn}(t). \tag{13}$$

The computation delay $D_{Cn}(t) = 0$ when user n requests for the normal multimedia application. As modeled above, the form of radio access link transmission delay $D_{An}(t)$ of user n is different between AR application request and normal multimedia request. For the AR application, the radio access link latency corresponds to that of GU, and the radio access link latency of normal multimedia application corresponds to that of BU.

We minimize the long-term average content delivery delay by jointly optimizing the user association, power allocation of NOMA, deployment of UAVs and caching placement of UAVs. The long-term optimization problem is

$$\min_{q,h^{\mathrm{RC}},L_{\mathrm{uav}},c} \quad \frac{1}{T}\sum\nolimits_{t=1}^{T}\sum\nolimits_{n=1}^{N} D_n(t) \tag{14a}$$

$$\text{s.t.} \quad q_{kn}(t) \in \{0,1\}, \forall k,n,t, \tag{14b}$$

$$c_{km}(t) \in \{0,1\}, \forall k,m,t, \tag{14c}$$

$$th_L \le h_{kg}^{\mathrm{RC}}(t) \le th_H, \forall g,t, \tag{14d}$$

$$\sum\nolimits_{k=1}^{K} q_{kn}(t) = 1, \forall n,t, \tag{14e}$$

$$\sum\nolimits_{m=1}^{M} c_{km}(t) \le Z_1, \forall k,t, \tag{14f}$$

$$\sqrt{\left(L_{\mathrm{uav}k}(t+1) - L_{\mathrm{uav}k}(t)\right)^2} \le \delta v_{\max}, \forall k,t, \tag{14g}$$

where $q = \{q_{kn} | 1 \le k \le K, 1 \le n \le N\}$ represents the association indicator between users and UAVs, $h^{\mathrm{RC}} = \left\{h_{kg}^{\mathrm{RC}} | 1 \le k \le K, 1 \le g \le G\right\}$ represents the power allocation coefficient of NOMA users, $L_{\mathrm{uav}} = \{L_{\mathrm{uav}k} | 1 \le k \le K\}$ represents the deployment of UAVs, and $c = \{c_{km} | 1 \le k \le K, 1 \le m \le M\}$ represents the caching placement. Constraint (14b) shows that

the user association between UAV k and user n, $q_{kn}(t)$, should be 0 or 1. In constraint (14c), the caching placement for RC m in UAV k, $c_{km}(t)$, should be 0 or 1. Constraint (14d) shows that power allocation coefficient $h_{kg}^{\mathrm{RC}}(t)$ is set between th_L and th_H, $th_L + th_H = 1$. As shown in constraint (14e), user n should associate to one UAV at each time slot. In constraint (14f), the sum of caching placement index of each time slot should be no more than cache capacity of each UAV Z_1. The index of hover horizontal coordinate is given by $L_{\mathrm{uav}k}(t) = [i_{\mathrm{uav}k}^{x}(t), i_{\mathrm{uav}k}^{y}(t)]$. In constraint (14g), the movement of UAV during each time slot should not exceed its maximum speed. It is shown that the formulated problem has three features: nonlinear mixed integer, dynamic objective function and long-term optimization.

B. Solution based on Branch and Bound

As the dimension of the variables in the long-term optimization problem (14) is relatively large, it is hard to be solved efficiently. Therefore, the long-term problem is decomposed into sub-problems focusing on per-slot optimization. Since the sub-problem in each time slot is a nonlinear mixed-integer programming, which can be solved by branch and bound method [28]. We propose a BaB based resource allocation, UAV deployment and caching placement algorithm to solve the formulated problem (14), which achieves the per-slot optimization and obtain a local optimal solution. There exist four basic components: objective result function, nodes, lower bound and historically best solution. In the following, we define $\mathbf{SU} = \left(q, h^{\mathrm{RC}}, L_{\mathrm{uav}}, c\right)$ as the solution of per-slot optimization sub-problem in a given time slot t for simplicity.

Utility Function: As the dimension of the variables in the long-term optimization problem is too huge to be solved with the variable-wise recursive algorithm, we decompose the long-term problem into subproblems focusing on per-slot optimization [29], and then measure the instantaneous content delivery delay of the networks in a given time slot t as the cost of solution $\mathbf{SU} = \left(q, h^{\mathrm{RC}}, L_{\mathrm{uav}}, c\right)$, which is given by

$$R(\mathbf{SU}) = \sum\nolimits_{n=1}^{N} D_n(t). \tag{15}$$

Nodes: A certain number of nodes represent the corresponding solutions and integer constraints. The root node represents the initial solution $\mathbf{SU}_0$ without considering the integer constraints. In particular, two son nodes are generated by replacing the first non-integer variable $\mathbf{SU}_{\mathrm{non-in}}$ in the corresponding solution with the floor integer and ceil integer of that variable. These integer solutions are achieved by adding the corresponding linear integer constraints

$\mathbf{SU}_{\text{non-in}} \geq v_{c\mathbf{SU}_{\text{non-in}}}$ and $\mathbf{SU}_{\text{non-in}} \leq v_{f\mathbf{SU}_{\text{non-in}}}$, where $v_{c\mathbf{SU}_{\text{non-in}}}$ represents the ceil integer of the variable value and $v_{f\mathbf{SU}_{\text{non-in}}}$ represents the floor integer of the variable value.

Lower Bound: As the aim of optimization is to minimize the content delivery delay, we define the lower bound as the optimization utility value of the nonlinear programming with only the constraint of current node. As it is more likely to get the better solution with less constraints, which has lower content delivery delay, we define the optimization utility value as 'Lower Bound'.

Historically Best Solution: We define the integer solution with the lowest utility value until current iteration as the historically best solution $\mathbf{SU}_{\text{H}}^{*}$. Moreover, we express the historically best utility value as $R(\mathbf{SU}_{\text{H}}^{*})$.

For the nodes storage stage, each node forms two son nodes with the corresponding constraints. The solutions and constraints are stored in a first in last out (FILO) queue, as we want a depth-first-search. During the branch reduction stage, a node is took out from the FILO queue. First, the nonlinear programming is solved by function 'fmincon' in MATLAB with only the corresponding constraints of that node, where we express the solution as $\mathbf{SU}_{\text{candi}}$. Then we get the lower bound $R(\mathbf{SU}_{\text{candi}})$. When comparing the lower bound with the historically best utility value $R(\mathbf{SU}_{\text{H}}^{*})$, if $R(\mathbf{SU}_{\text{H}}^{*}) \leq R(\mathbf{SU}_{\text{candi}})$, this branch would be cut. Then we consider the update of the historically best solution. If the candidate solution $\mathbf{SU}_{\text{candi}}$ happens to meet all the constraints of the sub-problem and the candidate utility value is lower than the historically best utility value, we update the historically best solution $\mathbf{SU}_{\text{H}}^{*}$ and historically best utility value $R(\mathbf{SU}_{\text{H}}^{*})$ with the candidate solution $\mathbf{SU}_{\text{candi}}$ and candidate utility value $R(\mathbf{SU}_{\text{candi}})$. We make the historically best solution as the optimal solution of the per-slot optimization when the queue is empty.

According to the above components definition, we propose a BaB based algorithm, which is summarized in **Algorithm 1**.

Remark 3. *In the formulated problem (14), the constraint (14g) shows that the potential deployment of UAVs in current time slot is related to the positions of UAVs. Therefore, the long-term optimal solution may be unavailable, because the optimization of UAV deployment in each time slot does not consider the long-term influence of UAV deployment to next time slot. The proposed BaB based algorithm can be seen as a heuristic algorithm for the long-term optimization problem and provides a benchmark for performance comparison.*

Algorithm 1 BaB based resource allocation, UAV deployment and caching placement algorithm

Initialization:

Set total time slot.

Main Loop:

for each time slot **do**
  Update users' locations and users' preference
  Get an initial solution $\mathbf{SU}_0$ without integer constraints for the sub-problem in time slot t
  Set initial historically best solutions of all players randomly and historically best utility value of all players as positive infinity
  while Leader queue is not empty **do**
    Get a node from the queue according to FILO
    if $R(\mathbf{SU}_{0\mathrm{H}}^{*}) \leq R(\mathbf{SU}_{0\mathrm{candi}})$ **then**
      Cut this branch and continue
    else
      if $\mathbf{SU}_{\mathrm{candi}}$ satisfies all the integer constraints **then**
        Update historically best solution $\mathbf{SU}_{\mathrm{H}}^{*}$ with the candidate solution $\mathbf{SU}_{\mathrm{candi}}$
        Update the historically best utility value $R(\mathbf{SU}_{\mathrm{H}}^{*})$ with the candidate value $R(\mathbf{SU}_{\mathrm{candi}})$
      else
        Generate two son nodes by adding integer constraints and store them in the queue according to FILO
      end if
    end if
  end while
  Get the solution of the sub-problem $\mathbf{SU}_{\mathrm{H}}^{*}$ in current time slot t
end for

IV. Stackelberg Game and Long-term Optimization

Since the proposed **Algorithm 1** does not consider the long-term optimization, we proposed a DRL based algorithm in this section to solve long-term optimization problem (14). We firstly present the formulation of a Stackelberg game with one leader and multiple followers, where the leader optimizes the user association and followers optimize power allocation of NOMA, deployment of UAVs and caching placement of UAVs. Such a game model is suitable for distributed implementation as the algorithm efficiency would be improved by the distributed optimization of variables. Then, the formulated game is solved by the proposed DRL based algorithm.

A. Stackelberg Game Formulation

The optimization problem is formulated as a Stackelberg game. Among the variables in the proposed optimization objective function, user association q is suitable to be optimized centrally to avoid the conflict between agents. More specifically, distributed optimization for that variable set can not avoid the condition that more than one UAV provides service to the same user

group. Variables c, L_{uav} and h^{RC} are suitable for distributed optimization of K players to split the dimensions of variables, where h^{RC} of UAV k only decides the power allocation of NOMA user groups connected to this UAV. As shown in (14a), the joint optimization of c, L_{uav} and h^{RC} is closely related to the user association q of the current time slot, which directly decides the division of those three high dimensional variables.

Leader level: For the leader level, user association q of users is optimized with the aim of minimizing the long-term average content delivery delay in the networks, which is given by

$$U_L(t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N} D_n(t) \tag{16}$$

Then, the leader level optimization problem is given by

$$\min_{q} \quad U_L(t), \tag{17a}$$

$$\text{s.t.} \quad q_{kn}(t) \in \{0,1\}, \forall k,n,t, \tag{17b}$$

$$\sum_{k=1}^{K} q_{kn}(t) = 1, \forall n,t, \tag{17c}$$

where $q = \{q_{kn} | 1 \le k \le K, 1 \le n \le N\}$ represents user association. The UAVs deployment, caching placement and power allocation of NOMA follow the followers' decision of the previous iteration, which are expressed as c', ${L_{\text{uav}}}'$, and ${h^{\text{RC}}}'$. In constraint (17b), $q_{kn}(t) = 1$ implies that user n is served by UAV k, otherwise $q_{kn}(t) = 0$. Constraint (17c) shows one user should connect to only one UAV in the same time slot. It is obvious that the problem (17) is an nonlinear integer programming, which is also not convex. As a result, it is hard to get the optimal solution.

Follower level: For the follower level, there are K selfish players aiming to minimize their own content delivery delay, which means the sum content delivery delay for the users connected to the corresponding UAVs. The long-term average delay is given by

$$U_F(t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N} q_{kn}(t) D_n(t). \tag{18}$$

With the user association message from the leader agent, each follower agent jointly optimize power allocation of NOMA user groups connected to the corresponding UAV, UAV deployment

and caching placement. The follower level problem is formulated as:

$$\min_{h_k^{\mathrm{RC}}, L_{\mathrm{uav}k}, c_k} \quad U_F(t), \tag{19a}$$

$$\text{s.t.} \quad c_{km}(t) \in \{0,1\}, \forall k, m, t, \tag{19b}$$

$$\sum\nolimits_{m=1}^{M} c_{km}(t) \leq Z_1, \forall k, t, \tag{19c}$$

$$th_L \leq h_{kg}^{\mathrm{RC}}(t) \leq th_H, \forall g, t, \tag{19d}$$

$$\sqrt{\left(L_{uavk}(t+1) - L_{uavk}(t)\right)^2} \leq \delta v_{\max}, \forall k, t, \tag{19e}$$

where h_k^{RC} is power allocation coefficient for user groups connected to UAV k. $L_{\mathrm{uav}k}$ is deployment of UAV k. c_k is caching placement of UAV k. The user association q follows the leader level decision. In constraint (19b), $c_{km}(t) = 1$ implies that UAV k proactively caches RC m, otherwise $c_{km}(t) = 0$. Constraint (19c) limits the cache capacity of UAV k. The range of power allocation is in (19d). Constraint (19e) limits movement of UAVs during each time slot. It is obvious that optimization problem (19) is a nonlinear mixed integer programming.

Next, an optimal solution of the Stackelberg game is described by Nash equilibrium [30], which is a collection of solutions. For the leader player and follower players, each player gets the optimal solution of its corresponding sub-problem, which is the best response to the solution of other players. We define q^* and $\left\{c_k^*, L_{\mathrm{uav}k}^*, h_k^{\mathrm{RC}*}\right\}$ to respectively express the optimal solution for the leader level sub-problem and the follower level sun-problems. That is if a solution $\left\{q^*, c_1^*, L_{\mathrm{uav}1}^*, h_1^{RC*}, \cdots, c_K^*, L_{\mathrm{uav}K}^*, h_K^{\mathrm{RC}*}\right\}$ is a Nash equilibrium, then for the leader player, the optimal sub-problem solution satisfies

$$U_L\left(q^*, c_k^*, L_{\mathrm{uav}k}^*, h_k^{\mathrm{RC}*} | 1 \leq k \leq K\right) \leq U_L\left(q, c_k^*, L_{\mathrm{uav}k}^*, h_k^{\mathrm{RC}*} | 1 \leq k \leq K\right). \tag{20}$$

Meanwhile, for the follower player k, the optimal sub-problem solution satisfies

$$U_F\left(q^*, c_k^*, L_{\mathrm{uav}k}^*, h^{\mathrm{RC}*}\right) \leq U_F\left(q, c_k, L_{\mathrm{uav}k}, h^{\mathrm{RC}}, c_{-k}^*, L_{\mathrm{uav}-k}^*, h{-}k^{\mathrm{RC}*}\right), \tag{21}$$

where $-k$ represents the follower players other than k.

Although it is hard for the algorithms with pure strategies to get the mixed-strategy equilibrium, the algorithms try to approximate them. Each agent alternately updates its action selection policy according to the current stable policies of other agents. Until the current learning agent no longer changes its own action selection policy, then the subsequent agents have no motivation to change

their own action selection policy, the entire algorithm converges.

B. Solution based on DRL

In the proposed BaB based algorithm, the number of branches increases sharply as the scale of the networks, which causes a low efficiency of the algorithm in the large-scale scenario. In this subsection, we propose a DRL based algorithm to solve the proposed game for large-scale networks in practical scenarios. The framework of the proposed DRL based algorithm is given in Fig. 2. In the proposed DRL based algorithm, there exists no direct interaction between leader agent and follower agents. These agents select actions according to the observed state of environment, where the actions of other agents are also considered as the variables in environment. For example, the user association, which is decided by the action of leader agent, is a part of the followers' state.

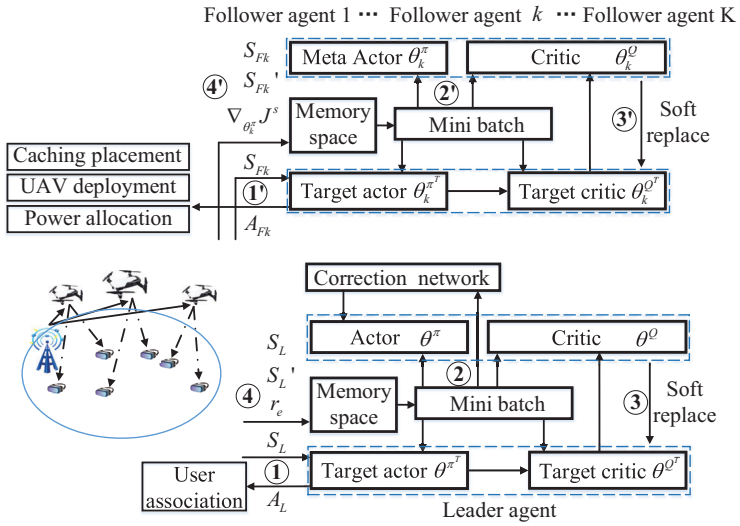

Fig. 2: Framework of the proposed DRL based algorithm.

For the leader sub-problem, we applied deep deterministic policy gradient (DDPG) algorithm to solve the formulated problem (17). There exist four basic elements in the DDPG algorithm for leader agent, including leader state, leader action, leader reward and correction mechanism. We define leader state to describe the observed variables in the environment. The leader action is the user association. In the process of DDPG for leader agent, the decision of follower agents in current episode is still unknown. To release the instability caused by that, we add a correction mechanism to assist DDPG, which is inspired by QMix [31]. In the proposed DRL based algorithm for leader sub-problem, the leader agent observes the leader state of environment

and selects leader action according to the output of the target actor network. Then the mini-batch sampled from memory space is used to train the actor network, and critic network. Besides, the correction network is also trained based on the bias stored in the memory space. Specificially, the output of correction network is utilized to train the actor network. Then, the target actor network and target critic network are trained through soft replace. finally, the state of current slot, action, reward and state of next slot are stored in the memory space.

Leader State: $S_L(t)$ is the environment state of the leader agent in time slot t, which is characterized by the SINR matrix $\Gamma_{kg\mathrm{GU}}^{\mathrm{RC}}(t)$, $\Gamma_{kg\mathrm{GU}}^{\mathrm{DR}}(t)$, $\Gamma_{kg\mathrm{BU}}^{\mathrm{RC}}(t)$ between UAVs and users, and satisfaction matrix of users to the cache of UAVs $W_{\mathrm{s}}(t)$. The elements of the satisfaction matrix is calculated by $w_{skn}(t) = q_{kn}(t)\sum_{m=1}^{M}(e_{nm}(t) + r_{nm}(t))c_{km}(t-1)$. $w_{skn}(t) = 1$ means that the content request of user n is in the cache of UAV k in time slot t. The state vector of the leader agent is $S_L(t) = \left\{\Gamma_{kg\mathrm{GD}}^{\mathrm{RC}}(t), \Gamma_{kg\mathrm{GD}}^{\mathrm{DR}}(t), \Gamma_{kg\mathrm{BD}}^{\mathrm{RC}}(t), W_{\mathrm{s}}(t)\right\}$, where UAVs' positions and power allocation coefficient follow the follower actions of the previous time slot.

Leader Action: $A_L(t)$ is the action of the leader agent in time slot t. As the action space is too large to process if we directly use the matrix of user association, $[q_{kn}(t)]$, we consider a continuous action space $A_L(t) = [a_{L1}(t), \cdots, a_{LG}(t)]$, where the approximate discrete value of $a_{Lg}(t)$ represents the UAV serving the user group g.

Leader Reward: The instantaneous cost is given by

$$R_L(S_L(t), A_L(t)) = \sum_{n=1}^{N} D_n(t). \tag{22}$$

Since c, L_{uav}, and h^{RC} are decided by the actions of the follower agents in the previous time slot, which causes instability to the training of the critic network. So the leader reward is only used to train the critic network, rather than evaluate the proposed algorithm. We define $R_G(S_L(t), A_L(t), S_F(t), A_F(t))$ to express global reward, where $S_F(t)$ is the state set of follower agents, $A_F(t)$ is the action set of follower agents.

Correction Mechanism: It is undesirable for the leader agent to forecast the follower agents' actions in this algorithm, since the complex state increases the difficulty of neural network fitting. We add a correction mechanism to approximate the global reward with leader reward. The bias between global reward and leader reward is defined as $r_c(t) = R_G(S_L(t), A_L(t), S_F(t), A_F(t)) - R_L(S_L(t), A_L(t))$. To release the instability caused by $S_F(t)$ and $A_F(t)$, the expected value

of reward bias is given by

$$r_e(t) = \mathbb{E}_{S_F, A_F}(r_c(t)), \tag{23}$$

which is approximated by a two hidden layers fully connected neural networks. $r_c(t)$ is supervisor and the input value is $\{S_L(t), A_L(t)\}$.

The actor network is used to obtain the optimal action, with state $S_L(t)$ as input value. A noise with variance var_L is added to the selected action for exploration. The critic network is used to obtain the Q value, with $\{S_L(t), A_L(t)\}$ as input value. Memory reply is deployed to train the networks, where each item contains the information of $\{S_L(t), A_L(t), R_L(t), S_L(t+1), r_c(t)\}$. The size of leader memory space M_{BL} is δ_{MS}, from which a mini-batch is sampled. Soft replace with factor ς is used to train the target networks. The actor network is trained by policy gradient [32]. The critic network is trained by minimizing the corrected temporal difference (TD) error, which is

$$\begin{aligned}\mathbb{E}_{M_{\mathrm{BL}}}\Big[R_L(t) + \gamma_1 r_e(t) - Q\left(s, a \left| \theta^Q \right.\right)\Big|_{s=S_L(t), a=A_L(t)} \\ +\gamma Q^T\left(s, a \left| \theta^{Q^T} \right.\right)\Big|_{s=S_L(t+1), a=\pi^T(S_L(t+1))}\Big]^2,\end{aligned} \tag{24}$$

where $\mathbb{E}_{M_{\mathrm{BL}}}$ represents the expected value among M_{BL}, γ_1 is the soft correction factor, γ is the discount factor, $r_e(t)$ is the expected reward bias, Q^T is the output of the target critic, θ^Q is the critic parameter, π^T is the output of the target actor, and θ^{Q^T} is the target critic parameter.

For the follower sub-problem, we employ the multi-agent DDPG to optimize power allocation of NOMA, deployment of UAVs and caching placement of UAVs. In the proposed DRL based algorithm for follower agents, the basic structure is similar to that of the leader agent. However, to cope with the dynamic interference between UAVs, we proposed a meta actor network, that is, each follower agent observes the action selection of other agents and store the data in the local memory space. During the training process, the mini-batch sampled from memory space is used to train the meta actor network. Therefore, generalized training result makes the meta actor network suitable for the complex interference environment.

Follower State: We define $S_{Fk}(t)$ to denote the environment state of the follower agent k in time slot t. $S_{Fk}(t)$ is characterized by the leader action $A_L(t)$, cache condition of UAV k, $C_k(t-1) = [c_{k1}(t-1), \cdots, c_{kM}(t-1)]$, and the distance from UAV k to all the users $D_{Ak}(t) = [d_{Ak1}(t), \cdots, d_{AkN}(t)]$.

Follower Action: $A_{Fk}(t)$ denotes the action of follower agent k in time slot t, which is characterized by power allocation $h_k^{\mathrm{RC}}(t) = \left[h_{k1}^{\mathrm{RC}}(t), \cdots, h_{kG}^{\mathrm{RC}}(t)\right]$, proactive caching index $C(t) = [c_{k1}(t), \cdots, c_{kM}(t)]$, and index of hover horizontal coordinate $[ix_{\mathrm{uav}k}(t), iy_{\mathrm{uav}k}(t)]$, where we randomly sample ten hover points at dimensions x and y in the given UAV deployment area. We assume that the height of UAVs h is a constant. We denote L_p power allocation levels as $h_{kg}^{\mathrm{RC}}(t) \in \left\{h^1, \cdots, h^{L_p}\right\}$.

Follower Reward: The instantaneous cost is given by

$$R_{Fk}\left(S_{Fk}(t), A_{Fk}(t)\right) = \sum\nolimits_{n=1}^{N} q_{kn}(t) D_n(t), \tag{25}$$

where the user association $q_{kn}(t)$ follows the leader action. With action of the leader agent and K follower agents, the global reward $R_G(t)$ is calculated in the way of (22).

Meta Actor: Considering the interference from other follower agents, we describe the action selection in different interference condition as different tasks. We add the gradients of other follower agents to the gradient of follower agent $\mathrm{k1}$. The gradients of follower agent $\mathrm{k2}$ cirtic network to the actor of $\mathrm{k1}$ is

$$\nabla_{\theta_{\mathrm{k1}}^{\pi}} J\left(s, \theta_{\mathrm{k2}}^{Q}\right) = \nabla_{\theta_{\mathrm{k1}}^{\pi}} \pi\left(s \middle| \theta_{\mathrm{k1}}^{\pi}\right)\big|_{s=S_L(t)} \times \nabla_a Q\left(s, a \middle| \theta_{\mathrm{k2}}^{Q}\right)\Big|_{s=S_L(t), a=\pi(S_L(t))}, \tag{26}$$

where $\theta_{\mathrm{k1}}^{\pi}$ represents the actor parameter of follower agent $\mathrm{k1}$, θ_{k2}^{Q} represents the critic parameter of follower agent $\mathrm{k2}$. We trained the actor network of follower agent $\mathrm{k1}$ with the composite gradient of

$$\nabla_{\theta_{\mathrm{k1}}^{\pi}} J_{\mathrm{k1}}^{Me} = \mathbb{E}_{M_{BF}}\left[\nabla_{\theta_{\mathrm{k1}}^{\pi}} J\left(s, \theta_{\mathrm{k1}}^{Q}\right) + \kappa \sum\nolimits_{\mathrm{k2} \neq \mathrm{k1}}^{K} \nabla_{\theta_{\mathrm{k1}}^{\pi}} J\left(s, \theta_{\mathrm{k2}}^{Q}\right)\right], \tag{27}$$

where $\mathbb{E}_{M_{\mathrm{BF}}}$ represents the expected value among M_{BF}, κ is the meta factor representing the influence from the gradients of other follower agents to the training of the meta actor network of follower agent $\mathrm{k1}$.

Remark 4. *We are able to choose meta factor κ for getting the trade off between the adaptability of the follower agent* $\mathrm{k1}$ *actor network to its influenced occasion and other follower agents. For the algorithm with $\kappa = 0$, the trained actor network is unstable because of the varying interference condition.*

The gradients of other follower agent are provided through memory reply. The item in the

Algorithm 2 DRL based resource allocation, UAV deployment and caching placement algorithm

Initialization:

Set total time slot, meta factor κ. Initialize parameters of all the networks.

Main Loop:

for each step **do**
  Update users' locations and users' preference
  Observe leader state $S_L(t)$
  Select $A_L(t)$ and update user access arrangement
  Get leader reward $R_L(S_L(t), A_L(t))$
  for $k = 1$ to K **do**
    Observe follower state $S_{Fk}(t)$
    Select follower action $A_{Fk}(t)$ and update deployment of UAV, caching placement of UAV and power allocation of NOMA
    Get follower reward $R_{Fk}(S_{Fk}(t), A_{Fk}(t))$
    Calculate the gradient with other agents' follower states and actor parameters
    Store $\{S_{Fk}(t), A_{Fk}(t), R_{Fk}(t), S_{Fk}(t+1), \nabla_{\theta_k^\pi} J^s\}$ in the memory space of follower agent k
    Train the DRL networks of follower agent k with (27), policy gradient, supervised learning and soft replace
  end for
  Get global reward R_G, and bias $r_c(t)$
  Store $\{S_L(t), A_L(t), R_L(t), S_L(t+1), r_c(t)\}$ in the memory space of leader agent
  Train the DRL networks of leader agent with (24), policy gradient, and soft replace
end for

memory space of follower agent k is $\{S_{Fk}(t), A_{Fk}(t), R_{Fk}(t), S_{Fk}(t+1), \nabla_{\theta_k^\pi} J^s\}$, where $\nabla_{\theta_k^\pi} J^s$ is the sum of the gradients of other follower agents to k. A mini-batch M_{BFk} is sampled from the memory space of follower agent k with δ_{MS} items. A noise with variance var_F is added to the selected action for exploration. We train the critic network by minimizing the TD error [32]. Soft replace factor of follower agents is ς, which is used to train the follower and target critic networks.

The DRL based resource allocation, UAV deployment and caching placement algorithm is given in **Algorithm 2**. The proposed DRL based algorithm is implemented in the Macro BS serving multiple UAVs and users in the networks. The control information, such as the location of users, is reported from users to MBS via the control channel in the cellular networks, e.g., the physical uplink control channel (PUCCH) defined in 5G new radio (NR) by 3GPP.

Compared with the BaB based algorithm, the DRL based algorithm focuses on the long-term optimization. To solve the instability and dynamic interference in the formulated problem, we employ a correction network and a meta actor network in the DRL based algorithm, which provides insightful method for solving such kind of problem.

C. Analysis of the proposed algorithm

1. Complexity: First we analyze the temporal computational complexity of the proposed BaB based algorithm and the proposed DRL based algorithm. As mentioned in the system model, the number of UAVs is K, the number of users is N, and the number of contents is M. The BaB based algorithm has the temporal computational complexity of $O\left(KN2^{M+N/2+2}\right)$. As the number of variables optimized by follower agent is much higher than that of leader agent, the tree structure of the follower agent has much more branches, which are related to computational complexity. Therefore, we focus on the complexity of follower agent when analyzing the temporal computational complexity of the proposed DRL based algorithm,which includes the actor network, critic network and correction network. We define δ_{MB} as the size of mini-batch and T_1 as the upper bound of training step. The temporal computational complexity of actor network in training process is $O\left(\gamma_a = \delta_{MB}T_1\left(Z_{A0}Z_{Al} + \sum_{l=1}^{L_1-2} Z_{Al}Z_{Al+1} + Z_{Al}Z_{AL_1}\right)\right)$, where$L_1$ is the number of laryers of actor network, the size of input layer is $Z_{A0} = M + N + 6K$, and the size of out layer is $Z_{AL_1} = 6 + M + 2$. The temporal computational complexity of critic network in training process is $O\left(\gamma_b = \delta_{MB}T_1\left(Z_{C0}Z_{Cl} + \sum_{l=1}^{L_2-2} Z_{Cl}Z_{Cl+1} + Z_{Cl}Z_{CL_2}\right)\right)$, where$L_2$ is the number of laryers of critic network, the size of input layer is $Z_{C0} = 6 + N + 2M + 6K + 2$, and the size of out layer is $Z_{CL_2} = 1$. The training process of the correction network has the same computational complexity funciton as that of critic network, which is denoted as $O\left(\gamma_c\right)$ with L_3 laryers. The temporal computational complexity of the proposed DRL based algorithm is $O\left(\max\left(\gamma_a, \gamma_b, \gamma_c\right)\right)$.

2. Convergence: Then we discuss convergence of the proposed DRL based algorithm in this section. During the interaction process of the proposed algorithm, the leader agent makes the best response about user association according to the current policies of follower agents, which will influence the action selection of follower agents in the subsequent phases. Besides, the power allocation of NOMA, deployment of UAVs and caching placement of UAVs decided by each follower agent also influence the policies of other follower agents. All agents alternately promote updates to their policies by cooperating to perceive each other's action selection. We assume that during the learning process, each agent sequentially updates policies according to the proposed DRL based algorithm. As the learning process progresses, the best action value of the action selection policy is a non-decreasing Cauchy sequence [33].

V. Performance Evaluation

We evaluate the performance of the proposed BaB based algorithm and DRL based algorithm. There are one MBS, K UAVs and N users in the system. We define the size of M unified contents as $C_1 = 16$ MB. The size of DR packages for AR application is $C_2 = 4$ MB. The content ranking and the mobility of users are modeled as two finite state Markov sequences. The number of states of each sequence is 5, where the transition probability is $Pr\left(s_{x+1}\left(t+1\right)|s_x\left(t\right)\right) = 0.7$, and the probabilities to other states are equal. Moreover, $f_c = 1$ GHz, $\omega = 10$ Hz, $\psi = 1000$ MHz, $H = 0.2$ MB, η= 0.9. Considering the fairness of power allocation among users, we set the power allocation coefficient th_L to 0.1 and th_H to 0.9, rather than 0 and 1. The length of time slot is assumed as $\delta = 0.5s$. The maximum flying speed of UAVs is given by $v_{\max} = 40m/s$. The main settings are summarized in Table. II.

The DRL networks have the same structure of 2 fully connected hidden layers, containing 50 neurons in each layer. The activation functions of hidden layers and output layer are Relu and Sigmoid [34]. We deploy batch normalization before the mini-batches are used to train the networks. The learning rate of leader agent and follower agents are 0.0001 and 0.0001 [35]. We set $\varsigma = 0.2$, $\gamma = 0.7$,$\gamma_1 = 0.25$, $\mathrm{var}_L = 3$, $\mathrm{var}_F = 0.3$, and $\delta_{MS} = 5000$. The size of mini-batch δ_{MB} is 32. The training of TD error and policy gradient is based on the Adam optimization algorithm [31].

Firstly, we verify the convergence of the proposed DRL based resource allocation, UAV deployment and caching placement algorithm, named as RUCDRL for short. We use **multi-agent deep deterministic policy gradient (MADDPG) based algorithm** as a benchmark, in which, the agents select the optimal actions by conventional MADDPG. In this simulation, we

TABLE II: Simulation Settings

Parameter	**Value**
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 N_0	-174 dBm/Hz
UAV flight altitude h	100 m
UAV deployment area side length	200 m
Long-term period T	100 s

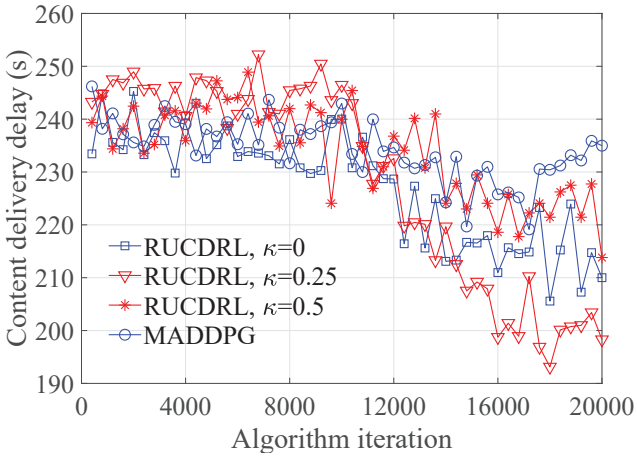

Fig. 3: Convergence of content delivery delay.

set $M = 6$, $N = 16$, and $Z_1 = 2$. As shown in Fig. 3, the content delivery delay decreases as the training continues, which demonstrates the effectiveness of RUCDR. Since the bigger meta actor coefficient κ corresponds to larger influence of the meta actor networks from other agents, the proposed RUCDRL with κ=0.5 achieves the worst performance. This provides a proof for **Remark 4**. However, RUCDRL with κ=0 performs better than MADDPG, which proves the effectiveness of employing the correction mechanism in the leader gent of the proposed RUCDRL. Because RUCDRL with κ=0.25 achieves a tradeoff between the attention to the environment of other agents and the attention to the environment of the target agent, this algorithm performs better than RUCDRL with κ=0 and κ=0.5. So we use $\kappa = 0.25$ in the following simulations.

Then we demonstrate the performance of the proposed BaB based algorithm and RUCDRL in Fig. 4, Fig. 5, and Fig. 6. We further consider two benchmarks:

- **single-agent deep deterministic policy gradient (SADDPG) based algorithm**, in which, the UAV selects the optimal actions by a single agent.
- **Fixed algorithm**, in which, each UAV caches the most popular contents, the user groups are associated with the nearest UAV, the UAV have the fixed trajectories with random center and radius of 50 m, and the NOMA power coefficient of RC signal is 0.7.

We show the performance of these algorithms with Zipf distribution parameters η= $\{0.9, 1.4\}$. Fig. 4 shows that the content delivery delay increases with the increasing of the user numbers in the network, where $M = 6$ and $Z_1 = 2$. Since Fig. 4 shows the content delivery delay of users in

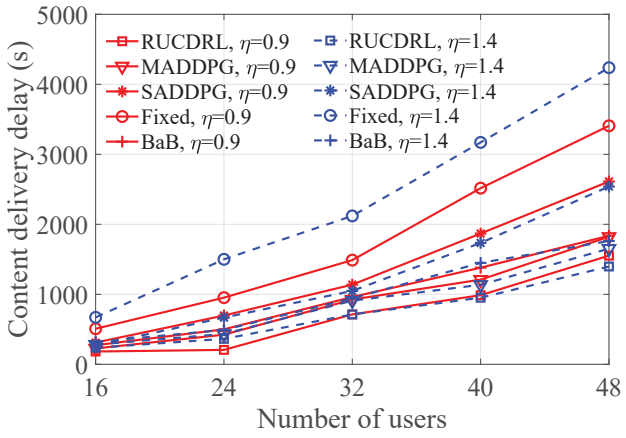

Fig. 4: Average content delivery delay versus the number of users.

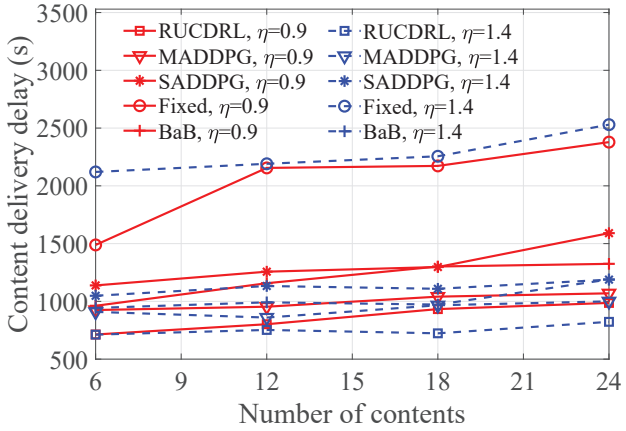

Fig. 5: Average content delivery delay versus the number of contents.

the whole network rather than single user, the average content delivery delay for each user is not quite high. Although the quality of AR application would be influenced, there exists no influence for the achievement of AR application. Although the training step of the proposed RUCDRL shows a high content delivery delay, the algorithm achieves a relative low content delivery delay after it converges. As shown in Fig. 5, the content delivery delay of BaB and RUCDRL also increases with the numbers of contents. As we observe from Fig. 4 and Fig. 5, the performance of BaB based algorithm is not the best. This is because that the proposed BaB based algorithm focuses on the optimization of per slot delay rather than long-term content delivery delay, which proves the importance of considering the long-term optimization in a dynamic networks, as stated in **Remark. 3**. In Fig. 4 and Fig. 5, SADDPG in small networks with $N = 16$ or $M = 6$

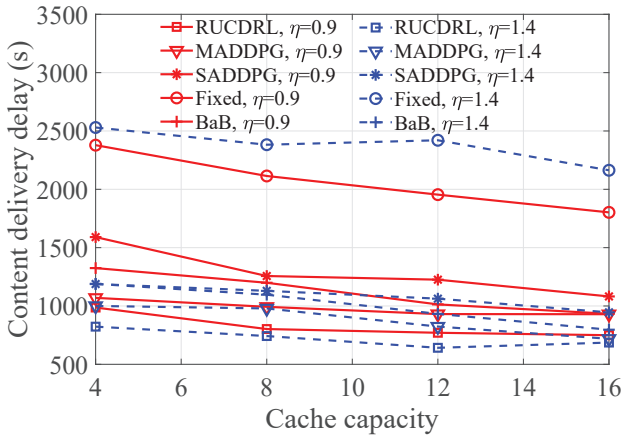

Fig. 6: Average content delivery delay versus the cache capacity.

performs much worse than RUCDRL and MADDPG as the numbers of users and contents increase. This is because that the considerable amount of users and contents causes the shortage of radio resource and leads to large action space and state space. Fig. 4 and Fig. 5 show that, the Fixed algorithm achieves the highest average content delivery delay compared with others algorithm, which means that our proposed algorithms chould improve the network performance by the joint optimiziation of the user association, power allocation of NOMA, deployment of UAVs and caching placement of UAVs.

We demonstrate the content delivery delay of the proposed algorithms and benchmark algorithms with varying cache capacity in Fig. 6. We set $N = 32$ and $M = 24$. The performance of these algorithms with different Zipf parameters $\eta = \{0.9, 1.4\}$ is shown. From Fig. 6, we can see that the content delivery delay of these algorithms decreases with the number of cache capacity. This is due to the fact that the large cache capacity increases the probability of cache hits. However, the tendency of the decreasing is not monotonous and there is slight increase from $Z_1 = 4$ to $Z_1 = 8$. This is because that although the dynamic proactive cache releases the backhaul traffic for the latter several time slots. As we observe from Fig. 6, the algorithms with $\eta = 1.4$ achieve lower content delivery delay compared to those with $\eta = 0.9$. This is because that a concentrated distribution of users' interests is conducive to increase the profit of proactive cache, as the cached content may be used to serve the requests from more than one user group. Moreover, the performance of the BaB based algorithm is affected more significantly than that of RUCDRL.

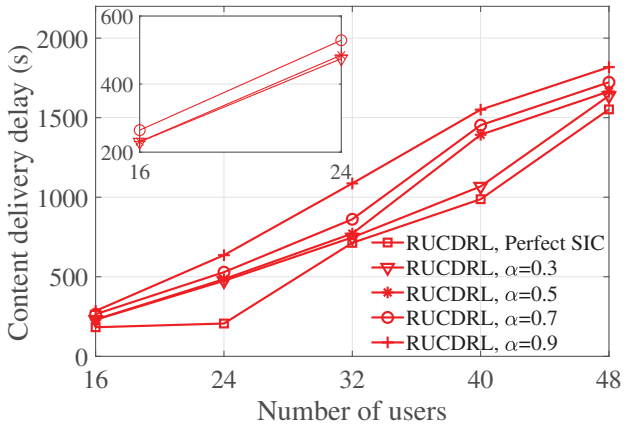

Fig. 7: Average content delivery delay versus the number of users with imperfect SIC.

Fig. 7 shows the influence of imperfect SIC by comparing the performance of RUCDRL with different imperfect SIC factors $\alpha = \{0.3, 0.5, 0.7, 0.9\}$, which influences the NOMA transmission capacity. As we can observe from Fig. 7, the algorithm with a smaller imperfect SIC factor performs better than the algorithm with a bigger factor. In particularly, the algorithm with perfect SIC, which can be considered as a special condition of imperfect SIC with $\alpha = 0$, achieves the lowest content delivery delay among these algorithms. This is because that the deployment of SIC improves the transmission capacity of the radio access links with NOMA. Moreover, the influence of the imperfect SIC factor to content delivery delay increases with the number of users. As the content delivery delay of each user group is related to the SIC quality, the effect of the imperfect SIC factor to content delivery delay of the whole network also increases with the number of users. This provides a proof for **Remark. 1**.

In Fig. 8, we verify the influence of AR application and normal multimedia application by comparing the performance of RUCDRL with different request repetition rate $\Phi = \{0.3, 0.5, 0.7, 1\}$, which represents the rate that two users request for the AR and normal multimedia application of the same content. We set $\kappa = 0.25$. As we can observe from Fig. 8, the performance of the algorithm with larger request repetition rate is better than that of the algorithm with a smaller rate, which means that the algorithm with larger request repetition rate has more NOMA user groups requesting for the AR and normal multimedia application of the same content. Therefore, the effectiveness of deploying NOMA to serve users requesting for the different applications of the same content is proved.

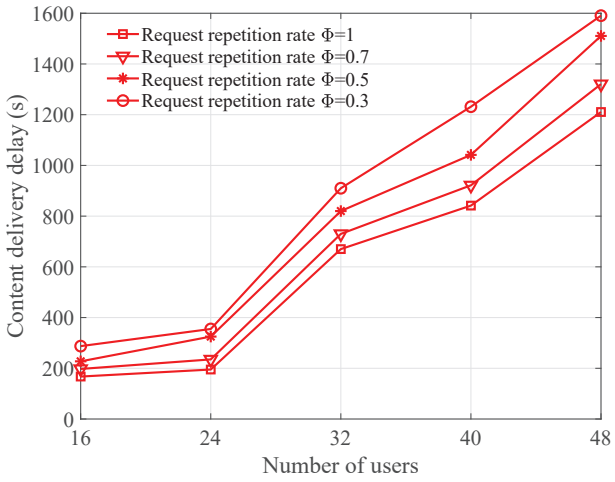

Fig. 8: Average content delivery delay versus the number of users in the scenario with different request repetition rate.

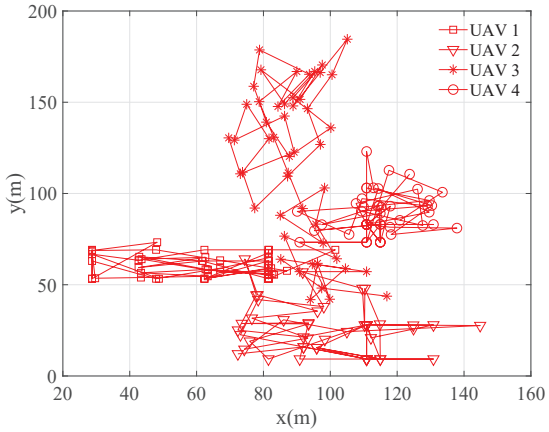

Fig. 9: The trajectories of UAVs among 50 slots of RUCDRL to minimize the content delivery delay.

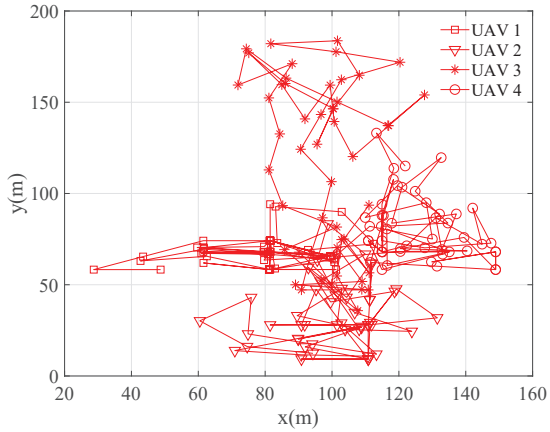

Fig. 10: The trajectories of UAVs among 50 slots of RUCDRL to maximize the cache hit ratio.

We demonstrate the trajectories of the optimized UAV deployment of the proposed RUCDRL with optimization objective of content delivery delay and cache hit ratio in Fig. 9 and Fig. 10, respectively. As we can see from Fig. 9, the UAV hovers over a specific area to serve the same group of users for the content delivery delay minimization, since the UAV trends to cache the contents that requested by the users in specific areas according to the proposed RUCDRL. For comparison, the UAV hovers over more scattered areas for cache hit ratio maximization as shown in Fig. 10. The reason is that UAVs focus on their cache without considering their radio access

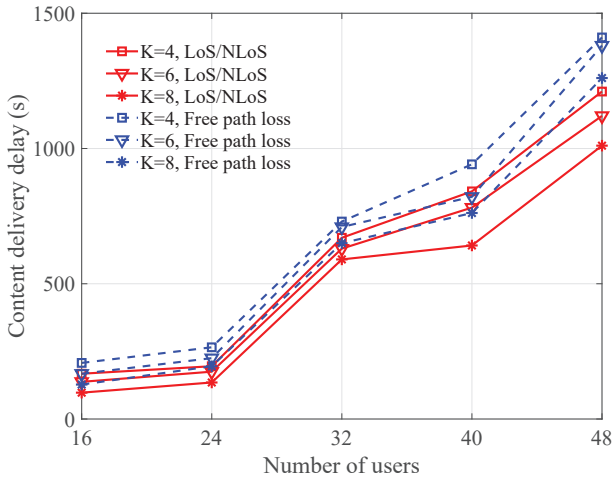

Fig. 11: Average content delivery delay comparison against system parameters and setups.

links with users. Fig. 9 and Fig. 10 demonstrate that the UAVs move around to serve the users with dynamic locations. Meanwhile, because of the limited flight speed, the UAV may not arrive at the optimal locations.

Fig. 11 shows the robustness of the proposed RUCDRL against different UAV number and channel models. The content delivery delay of the proposed RUCDRL increases with the number of users, since the growth of the number of users increases the probability of cache hit failure, which leads to the wireless backhaul delay increasing consequentially. Besides, the limited radio access resource makes the radio access delay increase with the number of users. As we can observe from Fig. 11, the proposed RUCDRL in the LoS/NLoS scenario achieves a lower content delivery delay than that in Free path loss scenario. This proves the superiority of the transmission channel from UAV to ground, since LoS/NLoS is a typical channel model for UAV communications.

VI. Conclusion

The cache-enabling mobile UAVs served the mixed AR and normal multimedia applications based on NOMA, cached limited contents to provide traffic offloading for backhaul links, and provided computation resource to users. The long-term problem was decomposed to subproblems in each time slot, which were solved by BaB based algorithm to achieve the optimization in each time slot. We transformed the original problem to a Stackelberg game and proposed DRL based algorithm to solve the game. We demonstrated that the considerable gains are achieved by

the proposed algorithms. We believe that the problem of energy consumption in the considered UAV-assisted networks needs to be further explored. The AR application has a high demand for energy consumption, which can be jointly considered with energy consumption of UAV in our future work. For algorithm, we could consider the deployment of federated learning in the reinforcement learning to further enhance the performance in solving the distributed problem.

REFERENCES

[1] Z. Wang, T. Zhang, Y. Liu, and W. Xu, "Caching placement and resource allocation for AR application in UAV NOMA networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec 2020, pp. 1–6.

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

[3] T. Hou, Y. Liu, Z. Song, X. Sun, Y. Chen, and L. Hanzo, "Reconfigurable intelligent surface aided NOMA networks," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2020.

[4] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7025–7037, 2019.

[5] 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, 2019.

[6] 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. 10 094–10 110, 2019.

[7] F. Zhou, N. Wang, G. Luo, L. Fan, and W. Chen, "Edge caching in multi-UAV-enabled radio access networks: 3D modeling and spectral efficiency optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 329–341, 2020.

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

[9] 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.

[10] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in internet-of-things system based on optimization of cache-enabled UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3525–3532, April 2019.

[11] 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.

[12] 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, February 2019.

[13] A. A. Nasir, H. D. Tuan, T. Q. Duong, and H. V. Poor, "UAV-enabled communication using NOMA," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5126–5138, 2019.

[14] M. Liu, G. Gui, N. Zhao, J. Sun, H. Gacanin, and H. Sari, "UAV-aided air-to-ground cooperative nonorthogonal multiple access," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2704–2715, 2020.

[15] X. Mu, Y. Liu, L. Guo, and J. Lin, "Non-orthogonal multiple access for air-to-ground communication," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2934–2949, May 2020.

[16] W. Mei and R. Zhang, "Uplink cooperative NOMA for cellular-connected UAV," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 644–656, June 2019.

[17] T. Z. H. Ernest, A. S. Madhukumar, R. P. Sirigina, and A. K. Krishna, "NOMA-aided UAV communications over correlated rician shadowed fading channels," *IEEE Trans. Signal Process.*, vol. 68, pp. 3103–3116, 2020.

[18] X. Chen, Z. Yang, N. Zhao, Y. Chen, J. Wang, Z. Ding, and F. R. Yu, "Secure transmission via power allocation in NOMA-UAV networks with circular trajectory," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10 033–10 045, Sep. 2020.

[19] 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, 2019.

[20] P. Maniotis and N. Thomos, "Viewport-aware deep reinforcement learning approach for 360° video caching," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[21] T. Dang and M. Peng, "Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, 2019.

[22] P. Chiu, P. Tseng, and K. Feng, "Interactive mobile augmented reality system for image and hand motion tracking," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9995–10 009, 2018.

[23] 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, 2019.

[24] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, 2019.

[25] 3GPP, "3GPP TR 36.777," in *Study on Enhanced LTE Support for Aerial Vehicles(Release 15)*, Dec. 2017.

[26] J. Zhao, Y. Liu, T. Mahmoodi, K. K. Chai, Y. Chen, and Z. Han, "Resource allocation in cache-enabled CRAN with non-orthogonal multiple access," in *2018 ICC*, 2018, pp. 1–6.

[27] A. S. de Sena, F. R. M. Lima, D. B. da Costa, Z. Ding, P. H. J. Nardelli, U. S. Dias, and C. B. Papadias, "Massive MIMO-NOMA networks with imperfect SIC: Design and fairness enhancement," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6100–6115, Sep. 2020.

[28] A. Li, F. Liu, C. Masouros, Y. Li, and B. Vucetic, "Interference exploitation 1-bit massive MIMO precoding: A partial branch-and-bound solution with near-optimal performance," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3474–3489, 2020.

[29] Z. Zhang and M. Tao, "Deep learning for wireless coded caching with unknown and time-variant content popularity," *IEEE Trans. on Wireless Commun.*, pp. 1–1, 2020.

[30] 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, 2020.

[31] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *ICML*, 2018.

[32] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1262–1276, 2019.

[33] X. Fang, T. Zhang, Y. Liu, and Z. Zeng, "Multi-agent cooperative alternating Q-learning caching in D2D-enabled cellular networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec 2019, pp. 1–6.

[34] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan 2021.

[35] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, March 2020.