UAV-Assisted Edge Computing and Streaming for Wireless Virtual Reality: Analysis, Algorithm Design, and Performance Guarantees

Liang Zhang ⬤ , *Member, IEEE*, and Jacob Chakareski, *Senior Member, IEEE*

Abstract—Emerging virtual reality (*VR*) applications require high data rate transmission and low end-to-end latency, which has become one of the main challenges for future wireless networks. Unmanned aerial vehicle (*UAV*) mounted base stations and computing facilities can be used to provide better wireless connectivity and computing services to edge VR users to meet their computing needs and reduce the end-to-end latency. We propose a novel UAV assisted mobile edge computing (*MEC*) network to enable high-quality mobile 360-degree video VR applications by leveraging UAVs to provide the required communication and computing needs. Then, we formulate the joint <u>UAV</u> placement, <u>MEC</u> and radio resource allocation, and 360-degree <u>v</u>ideo content layer assignment (*UAV-MV*) problem, which aims to select the allocation of computing and communications resources and the location of the UAVs such that the delivered quality of experience (*QoE*) is maximized across the mobile VR users, given various system constraints. We show that the problem is NP-hard, and decompose it into three lower-complexity subproblems that we solve sequentially. We design an approximation algorithm with performance guarantees that solves the UAV-MV problem based on the solutions to the three subproblems. Our simulation results show that the average QoE enabled by the proposed algorithm is 15% and 90% greater relative to two competitive reference methods.

Index Terms—Unmanned aerial vehicles (*UAV*), mobile edge computing (*MEC*), Internet of Things (*IoT*), virtual reality, 360-degree video, joint resource allocation, wireless 360-degree video streaming.

I. INTRODUCTION

ONE primary application of future wireless networks is virtual reality (*VR*) [1]. The VR market is expected to produce a $62 billion annual revenue by 2027 [2]. 360-degree video streaming to VR headsets is gaining popularity in diverse areas, e.g., gaming and entertainment, education, healthcare, and remote monitoring. Since 360-degree video streaming has high data rate and low latency requirements, how to deliver such content to mobile devices with high quality-of-experience (*QoE*) is a key challenge for future wireless networks [1].

Unmanned aerial vehicles (*UAV*s), received much attention from both academia and industry, has been proposed to mount base stations to provide better wireless connectivity and enlarge the coverage area [3]. Mobile edge computing (*MEC*) has been proposed and deployed at the mobile edge to provide cloud computing and IT capabilities close to mobile users to improve the user experience such as the latency [4], [5]. UAV-mounted BS/computing facilities have been studied to enhance the implementation flexibility of MEC and improve the wireless connectivity between mobile devices and the ground base station [5]. Wang *et al.* [6] proposed to utilize UAV-mounted server and ground servers to provide computing and communication services to on-line mobile applications, and the mobility of users and edge servers are considered to maximize the number of served mobile applications. Yu *et al.* [7] studied the UAV-aided MEC system for the Internet of Things (*IoT*) devices, and the objective is to maximize the energy efficiency of a UAV and minimize the latency of IoT devices. Guo *et al.* [8] proposed a VR framework with millimeter wave and MEC for wireless VR applications to achieve high QoE, the UE association, caching and offloading are jointly optimized. Du *et al.* [9] investigated the high quality immersive VR video service provisioning problem, and they proposed a method based on the deep reinforcement learning to optimize the view port rendering, offloading and power with the target to minimize the average energy consumption with satisfying the QoE constraints. Dang *et al.* [10] proposed a mobile VR delivery framework based on the fog radio access networks, and the target is to maximize the average tolerant delay with satisfying the fixed transmission rate constraints.

Resource allocation, transmission scheduling, and scalable coding for efficient multi-UAV 360° video capture and streaming has been studied in [1]. Multi-user mobile edge 360° video streaming via joint video tile scheduling, resource allocation and caching has been explored in [11]. The study in [12] proposed to tile the 360° video content using the H.265 standard and stream the tiles in view with the best quality. Gupta *et al.* [13] studied multi-user dual-connectivity scalable 360° video streaming, where the base layer is sent via a WiFi link and the enhancement layers are sent via a millimeter wave link.

To our knowledge, we are the first to study mobile VR scalable 360° video streaming by leveraging UAV-aided MEC
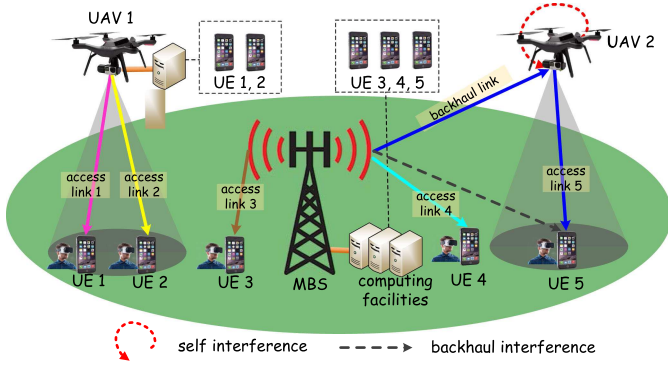
Fig. 1. A framework for wireless 360-degree video virtual reality via UAV-assisted edge computing/streaming.

to provide the required computing and communication needs. We formulate a joint UAV placement, MEC and radio resource allocation, and 360-degree video content layer assignment (*UAV-MV*) problem aiming to maximize the QoE across all mobile VR users, given various system constraints.

Our main contributions are: 1) We propose a novel UAV-aided MEC network for next generation 360-degree video mobile VR streaming; 2) We formulate the UAV-MV problem, prove it is NP-hard, and decompose it into three subproblems; 3) We propose approximation algorithms to solve the sub-problems, and propose a $(1 + \epsilon)$-approximation algorithm to solve the UAV-MV problem with guaranteed performance; 4) Simulation results show that the proposed algorithm is superior to baseline algorithms and can better balance the user workload. Our framework and system setting that we investigate are illustrated in Fig. 1.

The rest of the paper is organized as follows. In Section II, we present our system models of the content, communications, and computing aspects of the scenario we investigate. In Section III, we carry out the formulation of the optimization problem of interest. Analysis of the problem, design of approximation algorithms to solve it, and verification of their performance guarantees is carried out in Section IV. Performance evaluation via simulation experiments is carried out in Section V. Finally, we conclude in Section VI.

II. SYSTEM MODEL

Let $\mathscr{U}$ be the set of users, $\mathscr{B}$ be the set of BSs with computing facilities, including the macro base station (MBS) and UAVs with computing facilities, and $\mathscr{B}'$ be the set of UAVs acting as relays (no computing facilities). Each 360° video in the system comprises multiple embedded content layers that progressively improve the QoE, i.e., a base layer and several enhancement layers. We consider that each layer has the same data size and processing computing requirement. The number of content layers and the QoE delivered to a user depend on its down link data rate. Let $q_i^k = q^k \in \mathcal{Q}$ denote the normalized QoE of user i when the first k layers of its scalable 360° video are delivered, where $\mathcal{Q} = \{q^1, q^2, \ldots q^L\}$, $q^{k-1} > q^k$, and $\sum_k q^k = 1$ [14], [15]. Concretely, if Q^k denotes the absolute QoE of the content

(e.g., video quality PSNR) when reconstructed from its first k scalable layers, then $q^k = (Q^k - Q^{k-1})/\sum_k (Q^k - Q^{k-1})$, for $Q^0 = 0$. Finally, a scalable 360° video streamed to user i requires layer processing from the server and transmission, $A_i = \{r_i, e_i, \tau_i\}$, where r_i, e_i, τ_i denote its data size, the required computing resource (CPU cycles), and the maximum tolerable latency.

A novel framework for wireless VR applications is shown in Fig. 1; UAV 1 and the MBS are equipped with computing resources; UAV 2 is full duplex enabled and works as a relay BS; each user is connected to a VR device to execute VR applications; user 1 and user 2 are served by UAV 1 directly to receive the computing and communication resources, while user 3 and user 4 are served by the MBS directly to receive the computing and communication services; user 5 is served by the MBS via UAV 2 and its computing demand is served by the MBS. Each user requires both communication and computing resources. If a user is served by the MBS or a UAV directly, the MBS/UAV needs to assign computing and communications resources to this user. If a user is served by the MBS via a relay UAV, the MBS needs to assign computing and communications resources to this user, and the relay UAV also needs to assign communication resource to this user. Different users utilize different frequency spectra for communications to avoid interference [16].

A. Communications Model

Let $\beta_{i,j}$ be the data rate towards user i from BS j, $\beta_{i,j}^1$ be the data rate of the direct BS to user link, and $\beta_{i,j}^2$ be the data rate of the indirect MBS to user link via a relay UAV. It holds that $\beta_{i,j} = \max(\beta_{i,j}^1, \beta_{i,j}^2)$, for $j = 1$ (MBS), and $\beta_{i,j} = \beta_{i,j}^1$, for $j > 1$. Moreover, $\beta_{i,j}^1(\omega_{i,j}) = \omega_{i,j}\beta_0 \log_2(1 + s_{i,j})$, where $\omega_{i,j}$ is the allotted frequency spectra (LTE resource blocks (RBs)) to user i by BS j; β_0 is the bandwidth of one RB; $s_{i,j}$ is the SINR between BS j and user i. We formulate it as: $s_{i,j} = p_{i,j}\phi'_{i,1}/\sigma_{i,j}^2$, for $j = 1$, and $s_{i,j} = p_{i,j}\phi_{i,j}/\sigma_{i,j}^2$, otherwise. Here, $\phi'_{i,1}$ is the path loss between user i and the MBS; $\sigma_{i,j}^2 = \omega_{i,j}\beta_0\xi_j$ is the thermal noise power, and ξ_j is its power spectral density; $\phi_{i,j}$ is the path loss between user i and BS j that includes line-of-sight excessive path loss $\phi_{i,j}^L$, non-line-of-sight excessive path loss $\phi_{i,j}^N$, and free space path loss $\phi_{i,j}^F$ [17].

$$\begin{cases} \phi_{i,j} = \phi_{i,j}^L + \phi_{i,j}^N + \phi_{i,j}^F, \\ \phi_{i,j}^L = \psi_{i,j}\eta^L, \\ \phi_{i,j}^N = (1 - \psi_{i,j})\eta^N, \\ \phi_{i,j}^F = 20\log(4\pi f_1 d_{i,j}/f_0), \\ \psi_{i,j} = [1 + a_1 \cdot e^{-a_2(\frac{180\theta_{i,j}}{\pi} - a_1)}]^{-1} \end{cases} \quad (1)$$

Here, a mean path loss is used to calculate the path loss between user i and BS j because of the absence of terrain knowledge [16], [18]; $\psi_{i,j}$ is the probability of a link between user i and BS j (in the air) experiencing line-of-sight excessive path loss; a_1 and a_2 are environment parameters; η^L and η^N are the additional line-of-sight path loss and none-line-of-sight path loss; $d_{i,j}$ is the 3-D distance between user i and BS j; f_1 is the carrier frequency

and f_0 is the speed of light. Then, we have

$$\phi_{i,j} = \psi_{i,j}\eta^L + (1 - \psi_{i,j})\eta^N + 20\log(4\pi f_1 d_{i,j}/f_0). \quad (2)$$

Let $\beta_{i,j'}^{BL}$ be the data rate of a backhaul link from the MBS to UAV j' UAV, $\beta_{i,j}^{AL}$ be the data rate of an access link from UAV j' to user i, and $s_{i,j}^{BL}$ and $s_{i,j}^{AL}$ be the SINR of the backhaul and access links. Then, we can formulate the data rate from the MBS to user i via relay UAV j' as:

$$\begin{cases} \beta_{i,1}^2 = \min(\beta_{j',1}^{BL}, \beta_{i,j'}^{AL}), & \forall i \in \mathscr{U}, \\ \beta_{j',1}^{BL} = \omega_{i,1}\beta_0 \log_2(1 + s_{j',1}^{BL}), & \forall i \in \mathscr{U}, j' \in \mathscr{B}', \\ \beta_{i,j'}^{AL} = \omega_{i,1}\beta_0 \log_2(1 + s_{i,j'}^{AL}), & \forall i \in \mathscr{U}, j' \in \mathscr{B}'. \end{cases} \quad (3)$$

The SINR of the backhaul link and the access link are:

$$\begin{cases} s_{j',1}^{BL} = \dfrac{p_{i,1}\phi'_{j',1}}{p'_{i,j'}/c_0 + \sigma_{j',1}^2}, & \forall i \in \mathcal{U}, j' \in \mathscr{B}', \\ s_{i,j'}^{AL} = \dfrac{p'_{i,j'}\phi_{i,j'}}{p_{i,1}\phi_{i,1} + \sigma_{i,j'}^2}, & \forall i \in \mathcal{U}, j' \in \mathscr{B}'. \end{cases} \quad (4)$$

Here, $\phi'_{j',1}$ is the path loss between the MBS and the relay UAV j'; $p'_{i,j'}$ is the assigned power by the relay UAV j' to the user i; $p'_{i,j'}/c_0$ is the self interference (SI); c_0 is the SI cancellation capacity [19]; $p_{i,1}\phi_{i,1}$ is the backhaul interference.

B. Computing Model

In this article, we consider computing and downlink communications for VR applications [13]. The MBS ($j = 1$) and UAVs ($j > 1, j \in \mathscr{B}$) are equipped with computing facilities to serve users with VR applications, etc., providing computing, caching and storage services to users [5], [20].

Let C_j be the computing capacity (CPU cycles per second) of BS j and $y_{i,j}$ be the assigned share to user i. Then, we can formulate the computing delay associated with streaming 360° content to user i as $t_{i,j}^C = e_i/y_{i,j}$ [21], [22]. Similarly, we can formulate the transmission delay of streaming the content as $t_{i,j}^T = r_i/\beta_{i,j}$. Thus, the total provisioning delay for user i is $t_{i,j}^{total} = t_{i,j}^T + t_{i,j}^C$.

III. PROBLEM FORMULATION

We formulate our optimization problem of interest as shown below, where the key notation and variables we have used in the formulation are summarized in Table I.

$$\mathscr{P}_0 : \max_{x_{i,j}^k, y_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k, v_j} \quad \sum_i \sum_j \sum_k x_{i,j}^k q_i^k,$$

s.t. $C1 : \sum_j x_{i,j}^k \leq 1, \quad \forall i \in \mathscr{U}, \forall j \in \mathscr{B},$

$C2 : x_{i,j}^{k+1} \leq x_{i,j}^k, \quad \forall i, j, k, k < q^{max},$

$C3 : \sum_i \sum_k x_{i,j}^k \omega_{i,j}^k \leq \omega_j^{max}, \quad \forall j \in \mathscr{B},$

$C4 : \sum_i \sum_k x_{i,j}^k p_{i,j}^k \leq P^M, \quad j = 1,$

$C5 : \sum_i \sum_k x_{i,j}^k p_{i,j}^k \leq P^U, \quad \forall j \in \mathscr{B}, j > 1,$

TABLE I
KEY NOTATION AND VARIABLES

Symbol	Definition
ω^{max}	the total bandwidth of the network in terms of RBs.
ω_j^{max}	the total bandwidth the jth BS in terms of RBs.
P^M	the maximum transmission power of the MBS.
P^U	the maximum transmission power of an UAV.
$x_{i,j}$	the user-BS assignment indicator; it is 1 if the ith user is provisioned by the jth BS; otherwise, it is 0.
$x_{i,j}^k$	binary variable, used to indicate the achieved QoE from the jth BS to the kth layer of the ith user.
$y_{i,j}^k$	the assigned computing resource to the kth layer of the ith user by the jth BS.
$\omega_{i,j}^k$	the assigned bandwidth from the jth BS to the kth layer of the ith user.
$p_{i,j}^k$	the assigned power from the jth BS to the kth layer of the ith user.
Θ	the set of candidate positions for UAVs in the horizontal plane.
v_j	the position of the jth BS in the horizontal plane.

$C6 : \sum_i \sum_k x_{i,j}^k y_{i,j}^k \leq C_j, \quad \forall j \in \mathscr{B},$

$C7 : \sum_k x_{i,j}^k t_{i,j,k}^T + \sum_k x_{i,j}^k t_{i,j,k}^C \leq \tau_i, \forall i, j,$

$C8 : v_j \in \Theta, \quad \forall j \in \mathscr{B}, j > 1,$

$C9 : x_{i,j}^k \in \{0, 1\}, \quad \forall i, j, k,$

$C10 : 0 \leq y_{i,j}^k \leq C_j, \quad \forall i, j, k,$

$C11 : 0 \leq \omega_{i,j}^k \leq \omega_j^{max}, \forall i, j, k,$

$C12 : 0 \leq p_{i,j}^k \leq P^M, \quad \forall i, k, j = 1,$

$C13 : 0 \leq p_{i,j}^k \leq P^U, \quad \forall i, k, j \in \mathscr{B}, j > 1. \quad (5)$

Our objective is to maximize the QoE across all users. C1 is the resource allocation constraint to ensure that a content layer is served to a user by one BS at most. C2 is a scalable 360-degree video content layer provisioning constraint to ensure that a lower layer is served before a higher layer. C3 is a bandwidth capacity constraint for a BS. C4 and C5 are power capacity constraints for the MBS and a UAV. C6 is a computing capacity constraint for a BS. C7 is a latency threshold constraint to ensure that a user is served within a given latency threshold. C8 is a UAV placement constraint in the horizontal plane. C9, C10, C11, C12 and C13 are bounds for all variables. In this work, we divide a coverage area into $|\Theta|$ equal sub-areas, and the center of each sub-area is a potential position to place a UAV. Then, these sub-areas formed set Θ and they are labeled as $1, 2, \ldots, |\Theta|$. Note that v_j is the position of UAV j, it can be any position in the set of these candidate positions, $v_j \in \Theta$.

IV. PROBLEM ANALYSIS

We can show that any instance of the Max-Generalized Assignment Problem (Max-GAP) can be reduced to the problem in (5) by considering $|\mathscr{Q}| = 1$ (the BSs, the user computing and

communication requirements, and the user QoE can be mapped to the bin, cost, and profit factors in *Max-GAP*). Thus, (5) is also NP-hard, as the Max-GAP problem is a well known NP-hard problem *Max-GAP*) [23].

To reduce the complexity of (5), we consider its power assignment to be proportional to the bandwidth assignment as $p_{i,j}^k = \varsigma_j \omega_{i,j}^k$, where ς_j is the assigned power per Hz for BS j. Then, constraints C4 and C5 can be omitted. We also consider the computing resource assignment to be proportional to the radio resource assignment as $y_{i,j}^k = \kappa_j \omega_{i,j}^k$, where κ_j is a factor of proportionality for BS j. Then, constraint C6 can be omitted, too. Hence, we can reformulate $\mathscr{P}_0$ as problem $\mathscr{P}_1$:

$$\mathscr{P}_1 : \max_{x_{i,j}^k, \omega_{i,j}^k, v_j} \quad \sum_i \sum_j \sum_k x_{i,j}^k q_i^k \ ,$$

$$\text{s.t.} \quad C1, C2, C3, C7, C8, C9, \quad \text{in} \quad \mathscr{P}_0. \tag{6}$$

To solve $\mathscr{P}_1$, we decompose it into three subproblems that we solve sequentially. These are: (i) joint computing and communication resource assignment and user assignment (*Joint-CUE*), (ii) joint computing and communication resource assignment and scalable 360-degree video layer assignment (*Joint-CAL*), and (iii) UAV placement.

Concretely, we first consider the UAV positions as given, and solve a simpler instance of $\mathscr{P}_1$, where the user and video layer assignments coincide (*Joint-CUE*). We then introduce the concept of a sub-user, associated with a video layer served to a given user, and reformulate $\mathscr{P}_1$ to solve the assignment of sub-users and computing/communication resources, again for given UAV positions. We build upon the solution to *Joint-CUE*, to design an algorithm that solves the above problem (*Joint-CAL*). Finally, we leverage the latter algorithm to formulate an optimization method that identifies the best UAV positions via an exhaustive search.

A. Joint-CUE Problem

For given UAV positions, we reformulate $\mathscr{P}_1$ by linking the video layer assignment to the user assignment. Concretely, if user i is served by BS j, then all layers of its 360° content are served by this BS. Then, constraint C2 can be omitted. Hence, we can reformulate (6) as problem $\mathscr{P}_2$, where $x_{i,j} = \sum_k x_{i,j}$, $\omega_{i,j} = \sum_k \omega_{i,j}$, $q_i = \sum_k q_i^k$, and $q_i = \sum_k q_i^k$.

$$[\textit{Joint-CUE}] \ \mathscr{P}_2 : \max_{x_{i,j}, \omega_{i,j}} \quad \sum_i \sum_j x_{i,j} q_i,$$

$$\text{s.t.} \quad C1 : \sum_j x_{i,j} \le 1, \quad \forall i \in \mathscr{U},$$

$$C2 : \sum_i x_{i,j} \omega_{i,j} \le \omega_j^{max}, \quad \forall j \in \mathscr{B},$$

$$C3 : x_{i,j} t_{i,j}^T + x_{i,j} t_{i,j}^C \le \tau_i, \forall i \in \mathscr{U}, j \in \mathscr{B},$$

$$C4 : x_{i,j} \in \{0, 1\}, \quad \forall i, j. \tag{7}$$

Note that $\omega_{i,j}$ is determined by $x_{i,j}$. Concretely, for every user-BS assignment ($x_{i,j} = 1$), constraint C3 in problem $\mathscr{P}_2$ has only one variable $\omega_{i,j}$, and we can compute it as

$\arg\min_{\omega_{i,j}}(t_{i,j}^T + t_{i,j}^C \ge \tau_i)$, which identifies the smallest $\omega_{i,j}$ that meets the above inequality. Otherwise ($x_{i,j} = 0$), $\omega_{i,j}$ will be set to zero. Thus, we can reformulate $\mathscr{P}_2$ as $\mathscr{P}_3$ by omitting C3, and compute $\omega_{i,j}$ as above, from the solution of $\mathscr{P}_3$.

$$\mathscr{P}_3 : \max_{x_{i,j}} \sum_i \sum_j x_{i,j} q_i \ ,$$

$$\text{s.t.} \quad C1, C2, C4, \quad \text{in} \quad \mathscr{P}_2. \tag{8}$$

To carry out these two tasks, we formulate Algorithm 1. In its design, we consider that there are more users than BSs and the computing/communications requirements of any user are smaller than the respective capabilities of every BS.

In Algorithm 1, two independent solutions are obtained first. Then, the objective value for the two solutions is computed, and the solution with a bigger value is returned. The first solution is obtained by *Steps* 1–19 based on a weight factor defined as a ratio between the maximum QoE for user i and the respective communications resource assignment by BS j ($q_i/w_{i,j}$). The second solution is obtained by *Steps* 20–24 based on identifying the $|\mathscr{B}|$ users with biggest maximum QoE values. Finally, the bigger objective value for the two solutions is set as the final output in *Steps* 25–30.

Theorem 1: Algorithm 1 is a $\frac{1}{2}$-approximation algorithm for the joint-CUE problem. Moreover, it produces the optimal result when all users are served, i.e., $\sum_i \sum_j x_{i,j} = |\mathscr{U}|$.

Proof: Since $\mathscr{P}_3$ is an integer linear program problem, it is difficult to obtain the optimal solution. Thus, we transform $\mathscr{P}_3$ into problem $\mathscr{P}_4$ by relaxing the discrete variables $x_{i,j}$ to be continuous, i.e., $0 \le x_{i,j} \le 1$. Let $\psi_3(x_{i,j})$ and $\psi_4(x_{i,j})$ denote the objective functions of $\mathscr{P}_3$ and $\mathscr{P}_4$. We note that $\frac{d\psi_4}{dx_{i,j}} = q_i = 1 > 0$, $\psi_4(x_{i,j})$ is a convex function, and the optimal solution is easily achieved by convex optimization [24]. Let $OPT(\mathscr{P}_3)$ and $OPT(\mathscr{P}_4)$ denote the optimal value of $\psi_3(x_{i,j})$ and $\psi_4(x_{i,j})$. Then, we have $OPT(\mathscr{P}_3) \le OPT(\mathscr{P}_4)$, $\psi_3(x_{i,j}) \le OPT(\mathscr{P}_3)$, and $\psi_4(x_{i,j}) \le OPT(\mathscr{P}_4)$. We assume the number of users is more than the number of BSs ($|\mathscr{U}| > |\mathscr{B}|$), and each BS is enough to serve any single user. To continue the proof, we consider two different workload scenarios: (1) heavy workload (at least one user is not served) and (2) light workload (all users are served).

1) For heavy workload, $\sum_i \sum_j x_{i,j} < |\mathscr{U}|$. In its operation, Algorithm 1 sorts the users in descending order based on their weight factors. Let n be the index of the first user not selected to be served, i.e., $x_{n,j} = 0, \forall j$. As noted earlier, two solutions ($x_{i,j} \in \Gamma_2$, $x_{i,j} \in \Gamma_4$) are produced internally by Algorithm 1, and the one with a bigger objective value is returned as the final solution ($\max(\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2}, \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4})$).

Then, we can write $\psi_4(x_{i,j}^*) = OPT(\mathscr{P}_4) = \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} + \psi_3(\delta_j x_{i,j})|_{x_{i,j} \in \Gamma_2}$, where $\delta_j = (\omega_j^{max} - \sum_{i=1}^{n-1} x_{i,j} \omega_{i,j})/\beta_{i,j}$, $i = n-1+j$, and $0 < \delta < 1$. Now, we have that $\psi_3(\delta_j x_{i,j})|_{x_{i,j} \in \Gamma_2} \le \psi_3(\delta_j x_{i,j})|_{x_{i,j} \in \Gamma_4} \le \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}$, because Γ_4 comprises the users with the maximum QoE. Moreover, we also have that $\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} + \psi_3(\delta_j x_{i,j})|_{x_{i,j} \in \Gamma_2} \le \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} + \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}$, and $OPT(\mathscr{P}_4) \le \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} + \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}$. Then, it

Algorithm 1: Approximation Algorithm for the Joint-CUE Problem (*AA-CUE*).

Input : $\mathcal{B}, \mathcal{B}', \mathcal{U}, C_j, A_i, P^M, P^U, \omega_j^{max}$, and v_j;
Output: $x_{i,j}$ and $\omega_{i,j}$;

1 $\Gamma_0 = \mathcal{U}$;
2 **for** $i \in \Gamma_0$ **do**
3   **for** $j \in \mathcal{B}$ **do**
4      $\omega_{i,j} = \underset{\omega_{i,j}}{\operatorname{argmin}} (t_{i,j}^T + t_{i,j}^C - T_i^{th} \le 0)$;
5   calculate $y_{i,j} = \omega_{i,j}\kappa_j$;
6   calculate $W_{i,j} = q_i/\omega_{i,j}$;
7   compute weight $W_i^{max} = \max(W_{i,j})$;
8   obtain $\omega'_{i,j} = \underset{\omega_{i,j}}{\operatorname{argmax}} W_{i,j}$;

9 put all users in descending order by W_i^{max};
10 $i = 1, \omega_j^{used} = 0, \Gamma_1 = \mathcal{U}$, and $\Gamma_2 = \varnothing$;
11 **while** $\omega_j^{used} \le \omega_j^{max}$ & $\Gamma_1 \ne \varnothing$ **do**
12   **if** $\omega_j^{used} + \omega'_{i,j} \le \omega_j^{max}, \forall i \in \omega_1$ **then**
13      $x_{i,j} = 1$;
14      $\Gamma_1 = \Gamma_1 \setminus \{i\}$ and $\Gamma_2 = \Gamma_2 \cup \{x_{i,j}\}$;
15      $\omega_j^{used} = \omega_j^{used} + \omega'_{i,j}$;
16   **else**
17      $\Gamma_0 = \Gamma_1$;
18   go to step 2;
19    $i = i + 1$;
20 $b = 1, \Gamma_3 = \mathcal{U}$, and $\Gamma_4 = \varnothing$;
21 **for** $b \le |\mathcal{B}|$ **do**
22    $i' = \underset{i}{\operatorname{argmax}} q_i$ and $x'_{i,j} = \underset{x_{i,j}}{\operatorname{argmax}} x_{i,j} q_i, \forall i \in \Gamma_3$;
23    $x'_{i,j} = 1$;
24    $\Gamma_3 = \Gamma_3 \setminus \{i'\}$ and $\Gamma_4 = \Gamma_4 \cup \{x'_{i,j}\}$;
25 calculate $\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2}$ and $\psi_3(x'_{i,j})|_{x_{i,j} \in \Gamma_4}$;
26 **if** $\psi_3(x_{i,j}) \ge \psi'_3(x'_{i,j})$ **then**
27   return $x_{i,j} \in \Gamma_2$, and $\psi_3(x_{i,j})$;
28 **else**
29   replace $x_{i,j}$ by $x'_{i,j} \in \Gamma_4$, and return $\psi'_3(x'_{i,j})$;
30 obtain $x_{i,j}$ and $\omega_{i,j}$.

holds that $\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} \ge \frac{1}{2}OPT(\mathscr{P}_4)$ or $\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4} \ge \frac{1}{2}OPT(\mathscr{P}_4)$. Thus, $\max(\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2}, \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}) \ge \frac{1}{2}OPT(\mathscr{P}_4)$ and $\max(\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2}, \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}) \ge \frac{1}{2}OPT(\mathscr{P}_3)$, because $OPT(\mathscr{P}_3) \le OPT(\mathscr{P}_4)$, implying that the worst objective value produced by Algorithm 1 is more than $\frac{1}{2}$ of the optimal objective value of problem $\mathscr{P}_3$ ($OPT(\mathscr{P}_3)$). In sum, the approximation ratio of Algorithm 1 is $\frac{1}{2}$.

2) Here, $\sum_i \sum_j x_{i,j} = |\mathcal{U}|$. We have $\psi_3(x_{i,j}) = |\mathcal{U}|$ and $\psi_4(x_{i,j}) = |\mathcal{U}|$. Meanwhile, $OPT(\mathscr{P}_3) = OPT(\mathscr{P}_4) = |\mathcal{U}|$, implying that $\psi_3(x_{i,j}) = OPT(\mathscr{P}_3)$. Thus, the AA-CUE algorithm produces the optimal solution in this case. ∎

Theorem 2: The approximation ratio of the AA-CUE algorithm is $(1 - \varepsilon)$ if $\omega_{i,j} \le \varepsilon\omega^{max}$. In other words, the AA-CUE algorithm is a $(1 - \varepsilon)$-approximation algorithm of problem $\mathscr{P}_3$ and problem $\mathscr{P}_2$. Here, $\varepsilon \le \frac{1}{2}$.

Proof: We assumed n is the index of the first user not being served, and the prior $(n - 1)$ users are served. Let the achieved

QoE of these n users be $q_1, q_2, \dots, q_{n-1}, q_n$, and the required bandwidth be $\omega_1, \omega_2, \dots, \omega_{n-1}, \omega_n$. We have $\omega_1 + \omega_2 + \cdots + \omega_{n-1} \le \omega^{max}, \omega_1 + \omega_2 + \cdots + \omega_{n-1} + \omega_n \ge \omega^{max}$, and $\frac{q_1}{\omega_1} \ge \frac{q_2}{\omega_2} \cdots \ge \frac{q_n}{\omega_n}$. Then,

$$\frac{q_1 + q_2 + \cdots + q_{n-1} + q_n}{\omega_1 + \omega_2 + \cdots + \omega_{n-1} + \omega_n} \ge \frac{q_n}{\omega_n}$$

$$\frac{q_1 + q_2 + \cdots + q_{n-1} + q_n}{\omega^{max}} \ge \frac{q_n}{\omega_n}$$

$$\frac{\omega_n}{\omega^{max}}(q_1 + q_2 + \cdots + q_{n-1} + q_n) \ge q_n$$

$$\frac{\varepsilon\omega^{max}}{\omega^{max}}(q_1 + q_2 + \cdots + q_{n-1} + q_n) \ge q_n$$

$$\varepsilon \sum_{i=1}^{n-1} q_i \ge (1 - \varepsilon)q_n$$

$$q_n \le \frac{\varepsilon}{1 - \varepsilon} \sum_{i=1}^{n-1} q_i \quad . \tag{9}$$

Note that $(n - 1)$ users are served and n is the index of the first user not provisioned. Then, we have

$$q_1 + q_2 + \cdots + q_{n-1} + q_n \ge OPT(\mathscr{P}_3)$$

$$(q_1 + q_2 + \cdots + q_{n-1}) + \frac{\varepsilon}{1 - \varepsilon} \sum_{i=1}^{n-1} q_i \ge OPT(\mathscr{P}_3)$$

$$q_1 + q_2 + \cdots + q_{n-1} \ge (1 - \varepsilon)OPT(\mathscr{P}_3)$$

$$\psi_3(x_{i,j})|_{x_{i,j} \in \mathcal{U}_1} \ge (1 - \varepsilon)OPT(\mathscr{P}_3) \quad . \tag{10}$$

After that, $\max(\psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2}, \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_4}) \ge \psi_3(x_{i,j})|_{x_{i,j} \in \Gamma_2} \ge (1 - \varepsilon)OPT(\mathscr{P}_4)$. Thus, the approximation ratio of the AA-CUE algorithm is $(1 - \varepsilon)$. ∎

B. Joint-CAL Problem

After solving the Joint-CUE problem, we try to solve the Joint-CAL problem by focusing on the 360° content layer assignment for all users. For given UAV positions v_j, we can reformulate (6) as problem $\mathscr{P}_5$ below. Note that $\mathscr{P}_5$ is different from the multi-knapsack problem and the Max-GAP problem because of constraint C2, which requires a lower content layer to be served first, before a higher layer.

$$[\textit{Joint-CAL}] \ \mathscr{P}_5 : \max_{x_{i,j}^k} \ \sum_i \sum_j \sum_k x_{i,j}^k q_i^k \ ,$$

$$\text{s.t.} \quad C1 : \sum_j x_{i,j}^k \le 1, \quad \forall i \in \mathcal{U}, \forall j \in \mathcal{B},$$

$$C2 : x_{i,j}^{k+1} \le x_{i,j}^k, \quad \forall i, j, k, k < q^{max},$$

$$C3 : \sum_i \sum_k x_{i,j}^k \omega_{i,j}^k \le \omega_j^{max}, \quad \forall j \in \mathcal{B},$$

$$C4 : x_{i,j}^k \in \{0, 1\}, \quad \forall i, j, k. \tag{11}$$

Algorithm 2: Approximation Algorithm for the Joint-CAL Problem (*AA-CAL*).

> **Input:** $\mathscr{B}, \mathscr{B}', \mathscr{U}, C_j, A_i, P^M, P^U, \omega_j^{max}$, and v_j;
> **Output:** $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k$, and $y_{i,j}^k$;

1: map scalable 360-degree video layers to sub-user, $i = i \cdot k$;
2: run Algorithm 1;
3: obtain $x_{i,j}$ and $\omega_{i,j}$;
4: calculate $x_{i,j}^k$ and $\omega_{i,j}^k$ by $x_{i,j}$ and $\omega_{i,j}$;
5: compute $p_{i,j}^k$ and $y_{i,j}^k$;
6: return $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k$ and $y_{i,j}^k$.

Algorithm 3: The AA-UAV-MV Algorithm.

> **Input :** $\mathscr{B}, \mathscr{B}', \mathscr{U}, C_j, A_i, P^M, P^U, \omega_j^{max}$, and Θ;
> **Output:** $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k, y_{i,j}^k$, and v_j;

1: **for** $v_j \in \Theta$ **do**
2:    update the horizontal positions of UAVs;
3:    calculate $x_{i,j}$ and $\omega_{i,j}$ by Algorithm 1;
4:    obtain $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k, y_{i,j}^k$ by *Algorithm 2*;
5:    calculate $\psi_7(v_j)$;
6: get $v_j^* = \underset{v_j}{\operatorname{argmax}} \, \psi_7(v_j)$;
7: update $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k$, and $y_{i,j}^k$;
8: return $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k, y_{i,j}^k$, and v_j.

To leverage Algorithm 1 (solution of problem $\mathscr{P}_3$), we transform problem $\mathscr{P}_5$ to $\mathscr{P}_6$ by removing constraint C2. $\mathscr{P}_6$ becomes problem $\mathscr{P}_3$ when we map each 360° content layer k of user i to a sub-user $i = i \cdot k$. Using this advance, we can adopt Algorithm 1, to formulate Algorithm 2 to solve $\mathscr{P}_6$. After that, we prove that any solution generated by Algorithm 2 for problem $\mathscr{P}_6$ is also a solution for $\mathscr{P}_5$.

$$\mathscr{P}_6 : \max_{x_{i,j}^k} \quad \sum_i \sum_j \sum_k x_{i,j}^k q_i^k \ ,$$

$$\text{s.t.} \quad C1, C3, C4 \ \text{ in } \ \mathscr{P}_5. \tag{12}$$

Theorem 3: Algorithm 2 is an approximation algorithm for problem $\mathscr{P}_6$, and its approximation ratio is $\frac{1}{2}$.

Proof: $\mathscr{P}_6$ has the same solution as $\mathscr{P}_3$, if we map content layer k of user i to a sub-user $i = i \cdot k$ and assign resources to sub-users. Moreover, Algorithm 2 is designed based on Algorithm 1 and the solution $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k, y_{i,j}^k$ of Algorithm 2 is computed based on the solution of Algorithm 1. The approximation ratio of Algorithm 1 was proved in Theorem 1. Thus, the approximation ratio of Algorithm 2 is $\frac{1}{2}$. ■

Theorem 4: Any solution generated by Algorithm 2 for problem $\mathscr{P}_6$ is also a valid solution for problem $\mathscr{P}_5$.

Proof: 1) Algorithm 2 is designed based on Algorithm 1 that employs a delivered QoE vs. resource demand weight for a user, prioritizing users with higher weights. Algorithm 2 applies the same approach to prioritize sub-users. The weight of sub-user $(i \cdot k - 1)$ is bigger than that of sub-user $(i \cdot k)$. This is because lower layers have higher normalized QoE and every layer has the same data size, as introduced earlier. Hence, a sub-user $(i \cdot k)$ will always be served after sub-user $(i \cdot k - 1)$ and constraint C2 in problem $\mathscr{P}_5$ will be automatically satisfied.

2) Assume for the sake of contradiction that one solution generated by Algorithm 2 for user i is $x_{i,j}^{k-1} = 0$ and $x_{i,j}^k = 1$. However, this cannot represent a solution to $\mathscr{P}_5$, as it violates its constraint C2. Moreover, this also cannot represent a solution produced by Algorithm 2 because sub-user $(i \cdot k - 1)$ has a higher weight than sub-user $(i \cdot k)$ and Algorithm 2 serves sub-users with higher weights preferentially. Thus, a feasible solution generated by Algorithm 2 can only be: $x_{i,j}^{k-1} = 1$ and $x_{i,j}^k = 0$.

In sum, a solution achieved by Algorithm 2 for problem $\mathscr{P}_6$ is also an appropriate solution for problem $\mathscr{P}_5$. ■

C. The UAV Placement Problem

After solving $\mathscr{P}_5$, the assignment of 360° video layers, and computing, bandwidth, and power resources is computed, for given UAV positions. Now, we need to find the best UAV positions. We transform (5) into problem $\mathscr{P}_7$ below, by adopting the solution to $\mathscr{P}_5$ to omit the constraints in (5) that include $x_{i,j}^k, \omega_{i,j}^k, p_{i,j}^k$, and $y_{i,j}^k$.

$$\mathscr{P}_7 : \max_{v_j} \quad \psi_7(v_j) \ ,$$

$$\text{s.t.} \quad C1 : v_j \in \Theta, \quad \forall j \in \mathscr{B}, j > 1. \tag{13}$$

Here, $\psi_7(v_j) = \psi_0|_{x_{i,j}^k = \tilde{x}_{i,j}^k, \omega_{i,j}^k = \tilde{\omega}_{i,j}^k, p_{i,j}^k = \tilde{p}_{i,j}^k, y_{i,j}^k = \tilde{y}_{i,j}^k}$, ψ_0 is the objective function in (5), and $\tilde{x}_{i,j}^k, \tilde{\omega}_{i,j}^k, \tilde{p}_{i,j}^k, \tilde{y}_{i,j}^k$ is a solution produced by Algorithm 2. As the number of UAVs is small, we solve (13) via an exhaustive search [16], [25].

D. The UAV-MV Problem

Using the solutions to the three considered subproblems, we formulate an approximation algorithm (*AA-UAV-MV*) to solve (5), included in Algorithm 3 below.

Theorem 5: Algorithm 3 is a $(1 - \epsilon)$-approximation for problem $\mathscr{P}_1$ if $\omega_{i,j}^k \leq \varepsilon \omega^{max}$, and the optimal result is achieved when all layers of entire users are provisioned.

Proof: Based on the previous analysis carried out in Theorem 1 – Theorem 4, we can conclude that Algorithm 3 has a $(1 - \epsilon)$-approximation ratio. Concretely, the worst solution of Algorithm 3 is better than $(1 - \epsilon)$ of the optimal solution of (5). Moreover, the optimal solution is achieved when all 360° content layers for entire users can be served. ■

The complexity of solving the UAV-MV problem is $O(|\mathscr{B}|^{N_1} N_1^3 C_{N_2}^{N_3})$. Here, $N_1 = |\mathscr{U}| \cdot |\mathscr{Q}|$, $N_2 = |\mathscr{B}| + |\mathscr{B}'| - 1$, and $N_3 = |\Theta|$. The complexity of Algorithm 3 is $O((|\mathscr{B}| + \log(N_1) + 2) N_1 C_{N_2}^{N_3})$, that of Algorithm 2 is $O(N_1 |\mathscr{B}| + N_1 \log(N_1) + 2N_1)$, that of solving the UAV placement problem is $O(C_{N_2}^{N_3})$, and that of Algorithm 1 is $O(|\mathscr{U}||\mathscr{B}| + |\mathscr{U}| \log(|\mathscr{U}|) + 2|\mathscr{U}|)$. For the detailed complexity of Algorithm 1, it is $O(|\mathscr{U}||\mathscr{B}|)$ of Steps 1–8, it is $O(|\mathscr{U}| \log(|\mathscr{U}|))$ of

Step 9, it is $O(|\mathcal{U}|)$ of Steps 10–19, and it is $O(|\mathcal{U}|)$ of Steps 20–30.

We briefly outline the deployment and implementation aspects of the proposed UAV-MEC network. As shown in Fig. 1, the UAV-MEC network proposed in this article is a centralized network, and the server in the MBS provides computing service and also runs as a controller for the whole network. We only considered one time slot in this paper and we assume the user demands change slowly (fixed) in this paper. In the beginning of a time slot, the controller generates the scheduling results by leveraging Algorithm 3 after obtaining the basic information of the network, e.g., the locations of users, the channel information between users and the MBS, the demands of users, and the available resources in the MBS and UAVs. The scheduling results include: the UAV placement, the user association, the power and bandwidth assignment and the computing resource assignment. Then, the UAVs will fly to the target locations and be hovering on that location in the whole duration of this time slot. Meanwhile, the MBS and UAVs assign the computing and communication resources to users according to the schedule.

V. PERFORMANCE EVALUATION

We use MATLAB to carry out simulation experiments. We set the coverage area of the MBS to $500\,m \times 500\,m$, divided into 25 equal areas for positioning UAVs in the horizontal plane. Three UAVs are utilized for edge computing and one UAV operates as a relay node. The altitude of all UAVs is set to $100\,m$. For the frequency spectra assignment for each BS, the MBS is allocated with half of the total spectra and the rest BSs with computing facilities ($j \in \mathcal{B}$, $j > 1$) equally share the rest half of the total spectra, $\omega_1^{max} = \omega^{max}/2$ and $\omega_j^{max} = \frac{1}{2}\omega^{max}/(|\mathcal{B}| - 1)$, $j > 1$. For the rest BSs without computing facilities ($j' \in \mathcal{B}'$), they equally share the total spectra of the MBS because they are deployed as relay nodes for the MBS. Three different 360° videos, Wingsuit, Roller Coaster, and Angel Falls, and their normalized QoE values are adopted from [15]. They feature $4K$ resolution and 30 fps frame rate, which are encoded into 5 scalable layers using SHVC [26] with a GOP size of 30 frames. A user requests at random one 360° video as its content of choice. Our optimization is carried out per GOP of the content and thus we set τ_i as $1\ s$. The user distribution is generated according to the Matérn cluster process; the parent points represent the clusters and daughter points represent the users, which are generated according to a Poisson process and the uniform distribution, respectively [27]. Our main simulation parameters are summarized in Table II.

We use two reference methods to benchmark Algorithm 3. The first one is named *Best-SINR-VR* and employs the best SINR strategy for the user to BS assignment and the UAV placement is the same as in Algorithm 3. The second method is named *S-MBS-VR* and serves all users by the MBS. Both reference methods employ the same approach to assing 360° content layers to users: The base layer for all users is served first, followed by the first enhancement layer, second enhancement layer, etc. This process continues until all layers are served or the system

TABLE II
SIMULATION PARAMETERS

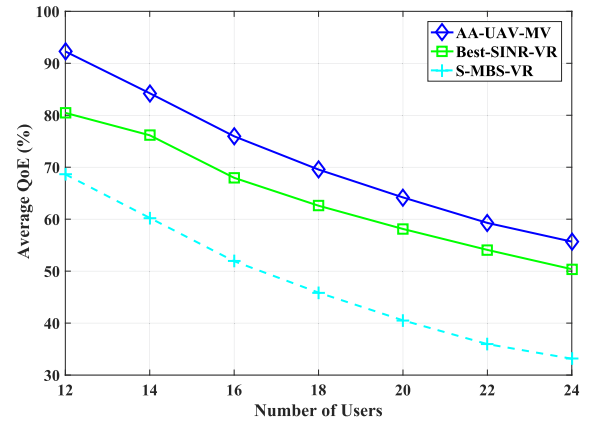|	
the coverage area	$500m \times 500m$
$|\mathcal{U}|$	$\{12, 14, \cdots, 24\}$
Q	$\{29\%, 25\%, 21\%, 16\%, 9\%\}$ [15]
r_i, full video i data rate	$\{10, 12, 14\}$ Mbps
e_i, computing requirement	$[500, 1500]$ CPU cycle/bit
UAV computing capacity	1×10^{10} CPU cycle/s
edge computing capacity	2×10^{10} CPU cycle/s
$(a_1, a_2, \eta_L, \eta_N)$	$(9.61, 0.16, 1, 20)$
path loss for the MBS	$131.1 + 42.8 log 10(d_{i,j})$, $d_{i,j}$ in km [28]
rayleigh fading for the MBS	8 dB
N_0	-174 dBm/Hz
P^M	33 dBm
P^U	27 dBm
ω^{max}, the total bandwidth	50 RB (10 MHz)
β_0, bandwidth of one RB	180 kHz

Fig. 2. Average QoE versus number of users.

resources are exhausted. The complexity of the *AA-UAV-MV* Algorithm, the *Best-SINR-VR* algorithm, and the *S-MBS-VR* algorithm are $O((N_1|\mathcal{B}| + N_1 \log(N_1) + 2N_1)C_{N_2}^{N_3})$, $O((N_1|\mathcal{B}| + N_1 \log(N_1) + N_1)C_{N_2}^{N_3})$ and $O(2N_1 + N_1 \log(N_1))$, respectively. Here, $N_1 = |\mathcal{U}| \cdot |\mathcal{Q}|$, $N_2 = |\mathcal{B}| + |\mathcal{B}'| - 1$, and $N_3 = |\Theta|$.

The average QoE is the mean value of the received QoE of all users. For the received QoE of each user, it is defined in Section II. Here, $Q = \{29\%, 25\%, 21\%, 16\%, 9\%\}$ is the set of normalized QoE for all five layers, including the obtained normalized QoE for each served layer. Fig. 2 shows the average QoE versus the number of users. We can see that the average QoE of all three methods decreases as the number of users increases, as an increasing number of enhancement layers for different users cannot be served. *AA-UAV-MV* has the highest average QoE and *S-MBS-VR* has the lowest average QoE. The average QoE of *AA-UAV-MV* reaches up to 15% and 68% improvement relative to *Best-SINR-VR* and *S-MBS-VR*. This is because the optimization *AA-UAV-MV* employs most effectively use of the available system resources.

Fig. 3 shows the cumulative distribution function (CDF) of the average QoE for 18 users. We can see that the QoE values for
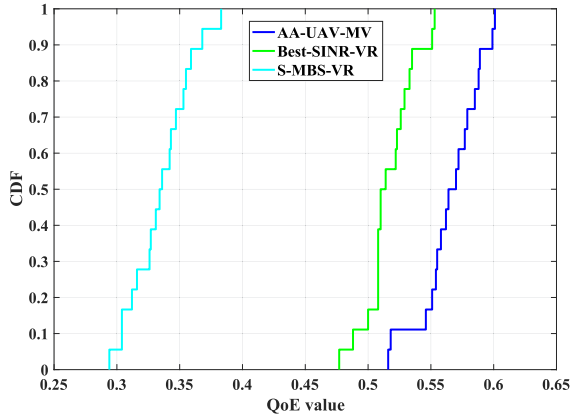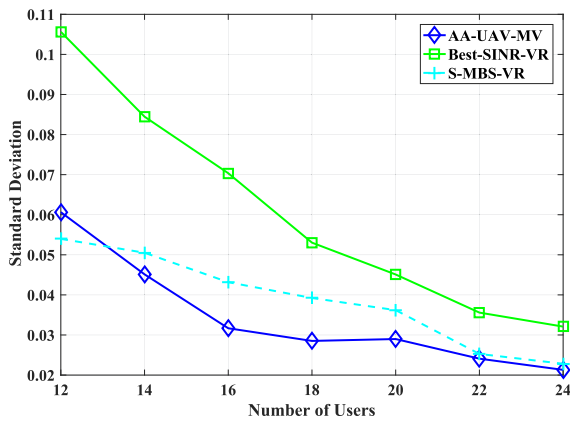
Fig. 3.　Cumulative Probability.

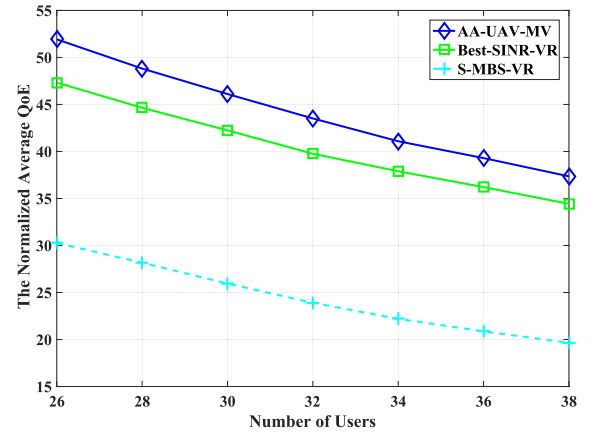Fig. 4.　Standard deviation vs number of users.

Fig. 5.　Normalized average QoE versus more users.

Fig. 6.　Normalized average QoE versus the scale of computing resource and the various number of users.

AA-UAV-MV, *Best-SINR-VR*, and *S-MBS-VR* fall in the ranges $[51\%, 60\%]$, $[48\%, 55\%]$, and $[29\%, 38\%]$, respectively. Thus, *AA-UAV-MV* provides the highest expected QoE and smallest QoE variation (standard deviation) across the user population.

Fig. 4 shows the average QoE standard deviation (*STD*) across the served users versus number of users. The STD for all three methods decreases with the latter. This is because the system resources are limited, and an increasing number of users experience low QoE, as the user load increases. Still, *AA-UAV-MV* demonstrates the lowest STD, 55% and 27% lower relative to *Best-SINR-VR* and *S-MBS-VR*. This indicates that *AA-UAV-MV* can better balance the user load.

Fig. 5 shows the results of the normalized average QoE versus more users. The same trend as Fig. 2 can be observed: the normalized average QoE of all algorithms decreases as the number of users increases because the network cannot support the services of the base layer and enhancement layers of all users under heavy workload, and more users receive the base layer service as compared to Fig. 2. The normalized average QoE of the *AA-UAV-MV* algorithm reaches up to 10% and 90% improvement as compared to the *Best-SINR-VR* algorithm and the *S-MBS-VR* algorithm. For 38 users, the normalized average QoE of the *S-MBS-VR* algorithm and the *AA-UAV-MV* algorithm is 19.7% and 37.4%, implying that all users nearly receive the services of the base layer by the *S-MBS-VR* algorithm and all

users roughly receive the services of two layers by the *AA-UAV-MV* algorithm.

Fig. 6 shows the results of the normalized average QoE versus different scales of computing resources and the various number of users. The scale factor of computing resources is set as $\gamma \in \{1, 1.5, 2\}$, implying that the total available computing resources in all BSs are multiplied by a scale factor, $C_j = \gamma C_j$. The normalized average QoE of all algorithms increases as the scale factor increases under a given number of users. The normalized average QoE improvement of the *AA-UAV-MV* algorithm with $\gamma = 2$ is up to 1.7 as compared to $\gamma = 1$; that of the *Best-SINR-VR* algorithm is up to 2 as compared to $\gamma = 1$, and that of the *S-MBS-VR* algorithm is up to 7.4% as compared to $\gamma = 1$. The reason why the *AA-UAV-MV* algorithm does not have much normalized average QoE improvement is that the *AA-UAV-MV* algorithm has efficiently assigned both communications and computing resources to users; only increasing the computing resource does not lead to an increase of the normalized average QoE.

The simulation results have demonstrated that the *AA-UAV-MV* algorithm is able to effectively solve the UAV-MV problem with guaranteed performance and it can efficiently assign both communications and computing resources to users.

VI. CONCLUSION

We explored the UAV-MV problem that aims to optimize the delivery of scalable 360° video content to mobile VR users, by intelligently allocating ground/air-based edge computing/communication resources and positioning the UAV BSs. We show that the UAV-MV problem is NP-hard, decompose it into three subproblems: the Joint-CUE problem, the Joint-CAL problem, and the UAV placement problem. We have proposed approximation algorithms to solve the subproblems sequentially. Then, another approximation algorithm, named AA-UAV-MV algorithm, has been proposed to solve the UAV-MV problem based on the solutions to the subproblems. Our approach demonstrates 15% and 90% improvement in average QoE relative to two competitive reference methods. Our results also demonstrate that the user load is better balanced by our algorithm.

REFERENCES

[1] J. Chakareski, "UAV-IoT for next generation virtual reality," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5977–5990, Dec. 2019.

[2] Grand View Research, "Virtual reality market size, share & analysis report, 2020–2027," Jun. 2020. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market

[3] L. Zhang and N. Ansari, "A framework for 5G networks with in-band full-duplex enabled drone-mounted base-stations," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 121–127, Oct. 2019.

[4] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: Multiaccess edge computing for 5G and internet of things," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6722–6747, Aug. 2020.

[5] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10 573–10580, Jun. 2020.

[6] J. Wang, K. Liu, and J. Pan, "Online UAV-mounted edge server dispatching for mobile-to-mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1375–1386, Feb. 2020.

[7] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[8] F. Guo, F. R. Yu, H. Zhang, H. Ji, V. C. M. Leung, and X. Li, "An adaptive wireless virtual reality framework in future wireless networks: A distributed learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8514–8528, Aug. 2020.

[9] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9517–9529, Oct. 2020.

[10] 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, Jul. 2019.

[11] J. Chakareski, "Viewport-adaptive scalable multi-user virtual reality mobile-edge streaming," *IEEE Trans. Image Process.*, vol. 29, pp. 6330–6342, May 2020.

[12] S. Petrangeli, V. Swaminathan, and M. Hosseini, "Improving virtual reality streaming using HTTP/2," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 225–228.

[13] S. Gupta, J. Chakareski, and P. Popovski, "Millimeter wave meets edge computing for mobile VR with high-fidelity 8K scalable 360° video," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process.*, 2019, pp. 1–6.

[14] M. Chen, W. Saad, and C. Yin, "Echo state learning for wireless virtual reality resource allocation in UAV-enabled LTE-U networks," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[15] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360° video streaming," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.

[16] L. Zhang and N. Ansari, "Approximate algorithms for 3-D placement of IBFD enabled drone-mounted base-stations," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7715–7722, Aug. 2019.

[17] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[18] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.

[19] M. S. Elbamby, M. Bennis, W. Saad, M. Debbah, and M. Latva-aho, "Resource optimization and power allocation in in-band full duplex-enabled non-orthogonal multiple access networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2860–2873, Dec. 2017.

[20] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[21] K. Zhang et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, Aug. 2016.

[22] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, Jun. 2017.

[23] L. Fleischer et al., "Tight approximation algorithms for maximum general assignment problems," in *Proc. 17th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2006, pp. 611–620.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[25] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 38–41, Feb. 2018.

[26] J. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jul. 2015.

[27] M. Afshang, C. Saha, and H. S. Dhillon, "Nearest-neighbor and contact distance distributions for matérn cluster process," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2686–2689, Dec. 2017.

[28] 3GPP TR 36.828 version 11.0.0, Release 11, 3GPP Tech. Rep., Jun. 2012. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2507

Liang Zhang (Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from the University of Science and Technology of China, Hefei, China, in 2014, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, in 2020. His research interests include machine learning, mobile edge computing, UAV communications, wireless communications, and Internet of Things (IoT). He was the recipient of the Hashimoto Prize for the best doctoral dissertation in 2020. He was also the recipient of the Travel Grant Award from IEEE GLOBECOM in 2016, Best Paper Award from IEEE ICNC in 2014, and National Scholarship of Graduate Students in China, in 2013.

Jacob Chakareski (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Rice Unviversity, Houston, TX, USA, and Stanford University, Stanford, CA, USA, and held research appointments with Microsoft, HP Labs, and EPFL. He is currently an Associate Professor with the Ying Wu College of Computing, the New Jersey Institute of Technology, Newark, NJ, USA, where he holds the Panasonic Chair of sustainability. He leads the Laboratory for AI-enabled Virtual and Augmented Reality Communications and Networked Systems, NJIT. His research interests include networked virtual and augmented reality, UAV-IoT sensing and networking, fast domain-aware reinforcement learning, 5G wireless edge computing and caching, ubiquitous immersive communication, and societal applications. He was the recipient of the Adobe Data Science Faculty Research Award in 2017 and 2018, Swiss NSF Career Award Ambizione 2009, AFOSR Faculty Fellowship in 2016 and 2017, and Best/Fast-Track Paper Awards at IEEE ICC 2017/2018 and IEEE Globecom 2016. He is the Organizer of the first NSF Visioning Workshop on networked VR/AR communications. He was on the advisory board of Frame, Inc. His research has been supported by the NSF, NIH, AFOSR, Adobe, Tencent Research, NVIDIA, and Microsoft.