Energy-Efficient Resource Allocation in Multi-UAV-Assisted Two-Stage Edge Computing for Beyond 5G Networks

Nway Nway Ei, Madyan Alsenwi, Yan Kyaw Tun, and Choong Seon Hong

Abstract—Unmanned aerial vehicle (UAV)-assisted multi-access edge computing (MEC) has become one promising solution for energy-constrained devices to run the applications with high computation demand and stringent delay requirement in beyond 5G era. In this work, we study a multi-UAV-assisted two-stage MEC system in which UAVs provide the computing and relaying services to the mobile devices. Due to the limited computing resources, each UAV executes only a portion of the offloaded tasks from its associated MDs in the first stage. Hence, in the second stage, each UAV relays the portions of the tasks to the terrestrial base station (TBS) which has rich computing resources enough to handle all the tasks relayed to it. In this regard, we formulate a joint task offloading, communication and computation resource allocation problem to minimize the energy consumption of MDs and UAVs by considering the limited resources of UAVs and the tolerable latency of the tasks. The formulated problem is a mixed-integer non-convex problem which is NP hard. To solve the formulated optimization problem, we apply the Block Successive Upper-bound Minimization (BSUM) method which guarantees to obtain the stationary points of the non-convex objective function. Finally, the extensive evaluation results are conducted to show the superior performance of our proposed framework.

Index Terms—Multi-access edge computing, unmanned aerial vehicle, block successive upper-bound minimization.

I. INTRODUCTION

A. Background and Motivation

With the unprecedented growth in the development of technology, the functionalities of smart devices such as smartphones, Internet of Things (IoT) devices, etc., have become more advanced. Moreover, the applications running on them for online gaming, augmented reality (AR), virtual reality (VR), video streaming and infotainment require high traffic demand and generate more processing data. Therefore, the cloud computing that provides the computing resources as well as the storage space has been introduced as a promising paradigm to lessen the burdens on mobile devices [1]. However, offloading the tasks of mobile devices to the central cloud server that is generally distant from them incurs high latency and degrades the system efficiency. To address this problem, multi-access edge computing (MEC) system that brings the computing resources near to the devices has been further introduced [1]. In particular, by providing a distributed computing environment, the cloud servers that are installed at the edge of the network such as the access points or base stations mitigate the energy consumption and communication/computation delay experiencing at the mobile devices [2]. However, constructing the new terrestrial network infrastructures in the temporary events (such as football matches or concerts) or in the disaster areas might not be cost-effective merely to assist the existing terrestrial network.

Recently, the integration of unmanned aerial vehicles (UAVs) such as balloons, airships or drones into 5G or beyond 5G network has drawn much attention to the researchers since they can not only extend the wireless coverage but also bring the computing resources near to the ground network devices. Due to the flexibility of on-demand deployment for the temporary events or emergency situations, UAVs are generally deployed as the assistance of the existing terrestrial networks in order to fulfill the unprecedented traffic demand and to provide the global internet connectivity [3], [4]. It is also anticipated that they are being used in various applications to bring fruitful business opportunities and green communication in the upcoming years [5]–[8]. Leveraging the good attributes of UAVs such as on-demand deployment and cost effectiveness, MEC-enabled UAVs can be deployed as the aerial computing platforms to offer the computing services to the energy-constrained MDs which are generally unable to completely execute the computation-intensive and delay-sensitive tasks locally. In addition, owing to its reliable line-of-sight link, they can relay the tasks of the mobile devices to the TBS when the MDs cannot directly offload their tasks to the TBS due to the severe link blockage or poor channel condition.

B. Challenges and Contributions

When UAVs are considered as the edge computing platforms, it is challenging to determine the amount of tasks...
to be offloaded from mobile devices (MDs) to the UAVs and decide the optimal allocation of communication and computation resources of UAVs to their associated devices in an energy-efficient manner. Moreover, as for UAV being an energy-limited and resource-constrained device, it is hard to accomplish all the tasks offloaded from the MDs. To address that problem, we propose a multi-UAV-assisted two-stage MEC system in which MEC-enable UAVs and TBS cooperatively execute the offloaded tasks of the MDs. In particular, UAVs compute part of MDs’ offloaded tasks and relay the rest to the TBS which has rich computing resources. In order to realize the energy efficiency of the proposed system, we investigate an energy minimization problem by jointly managing the task offloading and allocating the communication/computation resources. The main contributions of this paper are as follows:

- Firstly, we introduce a multi-UAV-assisted two-stage MEC system in which multiple MEC-enable UAVs offer the computing and relaying services to the MDs.
- We then propose a joint resource allocation and task offloading problem in order to minimize the energy consumption of MDs and UAVs. The proposed mixed integer non-convex problem which is NP hard.
- We further relax the channel allocation variable into the continuous form and then derive the upper bound of our objective function. To solve the formulated problem, we apply the block successive upper bound minimization (BSUM) algorithm which can tackle non-convex and non-smooth optimization problems.
- Finally, we perform an extensive simulation to verify that our proposed approach can yield the better solution compared to the other baseline schemes, namely, equal offloading, and offloading all.

The remainder of this paper is as follows. We describe the recent works in Section II. In Section III, we present our system model in detail. The communication and computation models for the proposed MEC system are described in Sections III-A and III-B, respectively. Then, our formulated optimization problem and the proposed solution approach are provided in Section IV. The simulation results are illustrated in Section V. Finally, we conclude the paper in Section VI.

II. RECENT WORKS

Regarding the intelligent edge computing in terrestrial network, the authors in [9] studied a three-layer network model which consists of IoT devices, edge and cloud servers. In this work, with the aim of maximizing the system’s benefit, TTL-constrained flood strategy transmission mechanism-based hierarchical game was proposed to determine the optimal offloading strategy of IoT devices while intelligently differentiating the tasks into data-intensive and CPU-intensive tasks. To make the edge computing network more adaptable, the authors in [10] and [11] studied an intelligent task offloading scheme to minimize the delay. To be specific, exploiting the deep Q-network, [10] jointly optimized the communication mode (V2I or V2V)/edge servers selection and task offloading strategy to minimize the execution time of the vehicle’s task. The authors in [11] studied an intelligent edge computing system in which the energy consumption and processing latency of the devices are minimized by determining the optimal offloading scheme based on the supervised decision tree offloading algorithm. Whereas the authors in [12] minimized the charging scheduling delay and charging price of electric vehicles in SDN-enabled vehicular networks by leveraging the deep reinforcement learning-based optimal fast charging station selection and path planning scheme. However, these works mainly focused on terrestrial mobile edge computing networks that limit the geographical locations of the edge servers.

Therefore, employing MEC-enabled UAV is another promising approach that brings fruitful advantages over a typical terrestrial MEC scenario due to its line-of-sight connectivity to the ground network terminals and flexible deployment. The work in [13] studied a three-dimensional UAV-aided MEC system for computation offloading of mobile users. Introducing a proactive deep reinforcement learning scheme, in this work, the expected long-term computation performance of the network is maximized by modeling the stochastic game among mobile users. The authors in [14] studied a single UAV-assisted MEC system in which the task offloading decision, task allocation and UAV trajectory are jointly optimized to minimize the total energy consumption of the system. Similarly, with the aim of minimizing the total energy consumption of UAV and ground user equipment, the works in [2], [15]–[18] studied the problem of managing the allocation of communication and computation resources to the user equipment and optimizing the trajectory of UAV. The authors in [19] maximized the energy efficiency of UAV by optimally determining the offloading strategy, transmission power of ground users and UAV trajectory. The work in [20] investigated the response time minimization problem in an aerial MEC system in which an MEC-enable UAV is deployed to serve a swarm of UAVs with the communication and computation resources.

Leveraging the different learning-based approaches, the works in [21] and [22] introduced an intelligent task offloading problem to mitigate the computational complexity of the UAV-assisted MEC system. In this regard, the authors in [21] proposed a deep reinforcement learning-based computation offloading scheme to maximize the users’ offloading tasks while taking into account the energy budget of the UAV. The work in [22] investigated a multi-UAV-aided cloud and edge computing network in which the weighted-sum of the time and energy consumption of the mobile devices is minimized by applying the Q-learning based computation offloading algorithm. Most of the existing works have mainly focused on resource allocation and UAV trajectory optimization problem in a single UAV-assisted MEC system for maximizing the energy efficiency or minimizing the delay. The multiple UAV-aided or ground-air integrated MEC system has been less explored. Different from the existing works, in this paper, we propose a multi-UAV-assisted two-stage MEC system in which UAVs offer computing and relaying services to MDs to help execute their tasks.
III. SYSTEM MODEL

In this work, we study a multi-UAV-assisted two-stage MEC system which can help the MDs to timely execute their computation-intensive and delay-sensitive tasks. There are many application scenarios where this kind of system can be leveraged, such as online gaming, AR/VR, video streaming, and infotainment. Let us suppose some VR/AR users need to process some content and obtain ultra-responsive service experiences. Due to their computing resource insufficiency to execute the task timely, they need to offload their tasks to the TBS equipped with MEC servers. However, offloading directly to TBS is not always favorable due to severe link blockage or being the cell edge users. To address this issue, UAVs acting as the computing platforms as well as relays are deployed to help the MDs for completing the tasks timely.

In our system model, as shown in Fig. 1, we propose a multi-UAV assisted two-stage MEC system in which there are $M$ MEC-enabled UAVs, a number of $U$ mobile devices and a terrestrial base station. A set $M = \{1, 2, \ldots, M\}$ of MEC-enabled UAVs are deployed for providing computing and relaying services to a set of MDs $U = \{1, 2, \ldots, U\}$ which are distributed in an area of interest. In this work, we assume that MDs cannot directly offload their tasks to the TBS due to the low signal strength or poor channel condition. Since UAVs are constrained by power and size, the available computing and communication resources on board are very limited. In that case, it is impossible for the UAVs to execute all the tasks offloaded from their associated MDs. The promising approach is that UAV can relay part of the MDs’ offloaded tasks to the TBS which provides a high-speed transmission rate with grid power supply and is empowered with an ultra-high performance processing server. Here, UAVs are assumed to be hovering or circling at the minimum fixed altitude enough to provide sufficient coverage without suffering severe path loss.

Since the position of UAVs can significantly affect the network performance, we exploit the k-means clustering algorithm for the deployment of UAVs and assignment of MDs to them. Although we consider the scenario where MDs cannot directly offload the tasks to TBS due to the poor channel condition or being cell-edge users or severe link blockage, the TBS also acting as a central controller is supposed to be able to collect the location information of MDs by itself or via the UAVs and control the UAVs’ deployment. With the prior knowledge of the locations of MDs and the available number of UAVs in the considered area, MDs are grouped into different clusters, and the cluster centers are obtained by applying the k-means clustering algorithm. Then, the UAVs that are at a fixed altitude are deployed at the cluster centers. With this approach, the MDs within each cluster can associate to their centroid UAV which has the minimum distance to them.

Here, we present the mathematical representation of k-means method-based MD association to UAVs. Given a set of MDs, $\mathcal{U}$, the k-means clustering algorithm intends to partition MDs into $M$ clusters, $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_M$, which are mutually exclusive and collectively exhaustive sets, i.e., $\mathcal{U}_a \cap \mathcal{U}_b = \emptyset$, $a \neq b$ and $\mathcal{U}_1 \cup \mathcal{U}_2 \cup \ldots \cup \mathcal{U}_M = \mathcal{U}$. Moreover, the path loss and transmit power between UAVs and MDs can also be reduced by minimizing the squared deviation of MD’s distance from its cluster’s centroid. Denoting the two-dimensional coordinates of MD $u$ and UAV $m$ as $s_u = (x_u, y_u)$ and $s_m = (x_m, y_m)$, respectively, where $u \in \mathcal{U}$ and $m \in M$, the association between MDs and UAVs can be obtained by solving the problem below:

$$\min_{\mathcal{U}_1, \ldots, \mathcal{U}_M} \sum_{m=1}^{M} \sum_{u \in \mathcal{U}_m} \|s_m - s_u\|^2. \quad (1)$$

After determining the association of MDs to UAVs in (1), we present the mathematical representation of our proposed system model and problem formulation in the following subsections.
A. Communication Model

In order to manage the communication resources, we consider that the total available system bandwidth is orthogonally divided into two portions for the MD-to-UAV data transmission which is used for offloading tasks from MDs to UAVs and UAV-to-TBS data transmission which is reserved for relaying tasks from UAVs to TBS, respectively. Then, the total available bandwidth for MD-to-UAV data transmission is further divided into $N$ subchannels, denoted by a set $\mathcal{N} = \{1, 2, \ldots, N\}$, each with a bandwidth of $\omega = 180$ kHz. The subchannels are shared by the mobile devices while transmitting their tasks to the associated UAVs.

1) MD-to-UAV Data Transmission: Each MD offloads its computation-intensive and delay-sensitive tasks to the associated UAV so that the tasks can be executed timely. To model the data transmission link between MDs and UAVs, we consider that the line-of-sight link is available and adopt the Rician channel fading model. In this work, we consider that the orthogonal frequency division multiple access (OFDMA) system is leveraged among MDs associated with each UAV to avoid intra-cell interference. We now define $\delta_{u,m} \in \{0, 1\}$ as a subchannel assignment variable, which indicates whether or not subchannel $n$ is allocated to MD $u$ associated with UAV $m$ as follows:

$$\delta_{u,m}^n = \begin{cases} 1, & \text{if subchannel } n \text{ is assigned to MD } u \text{ to transmit task data to UAV } m, \\ 0, & \text{otherwise}. \end{cases}$$

Then, adopting the free-space path loss model, the channel gain between MD $u$ and UAV $m$ over subchannel $n$ is given in [23], [24]:

$$h_{u,m}^n = \frac{h_0}{(d_{u,m})^\alpha},$$

where $h_0$ is the channel gain at a reference distance of 1 m and $\alpha$ is the path loss exponent. $d_{u,m}$ is the euclidean distance between MD $u$ and UAV $m$, i.e., $d_{u,m} = \sqrt{(x_u - x_m)^2 + (y_u - y_m)^2 + (z_u - z_m)^2}$, where $(z_u - z_m)$ means the vertical distance between MD $u$ and UAV $m$. The signal to interference plus noise ratio (SINR) for MD $u$ and UAV $m$ is also assumed to be dominated by the line-of-sight link as in MD-to-UAV data transmission. The channel gain between UAV $m$ and TBS which is located at $(x_0, y_0, z_0)$ is given by

$$h_{m,0} = \frac{h_0}{(d_{m,0})^\alpha},$$

where $d_{m,0} = \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2 + (z_m - z_0)^2}$ is the distance between UAV $m$ and TBS.

For the transmission link between UAVs and TBS, we consider that the available bandwidth $B$ is proportionally allocated to $M$ UAVs so that there is no interference among them when they relay MDs’ offloaded tasks to the TBS. Therefore, the signal to noise ratio (SNR) of UAV $m$ for relaying task data to the TBS can be calculated as

$$\gamma_{m,0} = \frac{P_{m,0} h_{m,0}}{N_0},$$

where $P_{m,0}$ the transmit power of UAV $m$ to the TBS. The data rate achieved by UAV $m$ for transmission to the TBS is given by

$$R_{m,0} = \beta_{m,0} \log_2 \left(1 + \gamma_{m,0}\right),$$

where $\beta_{m,0} = \frac{B}{M}$ is the bandwidth allocated to UAV $m$ for communication with the TBS.

B. Computation Model

Let us denote the computing task of MD $u$ associated with UAV $m$ is denoted as a tuple $(I_{u,m}, O_{u,m}, T_{u,m})$, where $I_{u,m}$ is the data size of the computing task, $O_{u,m}$ is the amount of required computing resources to execute 1-bit of input data and $T_{u,m}$ denotes the maximum tolerable delay for the completion of task. Each MD is assumed to be able to perform local computing and computation offloading simultaneously.

1) Local Computing at MD: Since each MD has very limited energy and computing resources, it is impossible to complete the tasks in time if it only relies on the local computing. Moreover, noting that the tasks of MDs are computation-intensive, delay-sensitive and bit-wise independent, we consider that MD $u$ offloads $I_{u,m}^{\text{off}} I_{u,m}$ (in bits) to UAV $m$, where $I_{u,m}^{\text{off}} \in [0, 1]$ and computes the amount of task $(1 - I_{u,m}^{\text{off}}) I_{u,m}$ locally. It is assumed that the task transmission and local computing at the MD can be done in a simultaneous manner [2]. The time taken for MD $u$ associated with UAV $m$ to compute the task locally is given by

$$t_{\text{local}}^{u,m} = \frac{(1 - I_{u,m}^{\text{off}}) I_{u,m} O_{u,m}}{f_{u,m}},$$

where $f_{u,m}$ is the CPU frequency of MD $u$ associated with UAV $m$. Thus, the energy consumption of MD $u$ associated with UAV $m$ can be expressed as [2],

$$E_{u,m}^{\text{local}} = (1 - I_{u,m}^{\text{off}}) I_{u,m} O_{u,m} k f_{u,m}^2.$$
where $k$ is the constant that depends on the processor’s chip architecture.

2) Computation Offloading to UAV and TBS: In this work, we study a two-stage edge computing model in which the offloaded tasks from the MDs will partially be computed at the UAV and TBS. In the first stage, after receiving the tasks offloaded from its associated MDs, each UAV executes only the partial portion of the task from each MD due to its limited computing resources and the task completion deadline. Therefore, in the second stage, the TBS executes the rest of the tasks relayed via UAVs. It is noteworthy that the energy consumption and task execution delay at the TBS can be neglected owing to its rich resources. In this regard, we only need to consider the energy consumption and transmission delay of UAVs for relaying the tasks to the TBS in the second stage. The detailed description of the two-stage edge computing is presented as follows.

a) First stage (computing at UAV): Since the resource-limited MDs cannot completely execute the tasks locally in a timely manner, they offload portion of the tasks to their associated UAVs. When MD $u$ offloads tasks to its associated UAV $m$ over subchannel $n$ for remote computing, its transmission delay is calculated by

$$t_{u,m}^{off} = \frac{\phi_{u,m}^{off} I_{u,m}}{n \log_2 (1 + \gamma_{u,m}^n)}. \quad (11)$$

The energy consumption of MD $u$ when it transmits the task to its associated UAV $m$ over subchannel $n$ is given by

$$E_{u,m}^{off,n} = p_{u,m}^{n} t_{u,m}^{off,n} = \frac{\phi_{u,m}^{off} I_{u,m} p_{u,m}^{n}}{n \log_2 (1 + \gamma_{u,m}^n)}. \quad (12)$$

UAV itself is an energy and resource constrained device, it cannot handle all the tasks offloaded from its associated MDs. Therefore, it collaborates with the TBS to reduce the computation burden and the energy consumption on them. In essence, each UAV executes a portion of MD $u$’s offloaded tasks and relays the rest to the TBS to save its energy consumption as well as computing resources. Hence, the time taken for UAV $m$ to compute portion of MD $u$’s offloaded task is calculated as

$$t_{u,m}^{e} = \frac{(1 - \phi_{u,m}^{off}) I_{u,m} O_{u,m}}{f_{u,m}^{e}}. \quad (13)$$

where $f_{u,m}^{e}$ is the CPU frequency of UAV $m$ allocated to its associated MD $u$ for the task execution and $\phi_{u,m,0} \in [0, 1]$ denotes the portion of MD $u$’s offloaded task at UAV $m$ that will be further relayed to the TBS. The energy consumed by UAV $m$ to partially compute the offloaded tasks of its associated MD $u$ is expressed as

$$E_{u,m}^{e} = (1 - \phi_{u,m,0}) I_{u,m} O_{u,m} k'(f_{u,m}^{e})^2. \quad (14)$$

b) Second stage (relaying to TBS): Here, we consider that all the associated MDs of UAV offload their tasks, but the UAV does not have enough computing resources to meet the task completion deadline of the MDs. Therefore, part of MDs’ offloaded task-input data at the UAV is assumed to be further relayed to the TBS. Since the TBS has enough computing resources and energy, the energy consumption and execution delay of the task at TBS are omitted in this work. The latency incurred by UAV $m$ to relay MD $u$’s offloaded tasks to the TBS is calculated by

$$t_{u,m,0}^{off} = \frac{\phi_{u,m}^{off} I_{u,m}}{\beta_{m,0} \log_2 (1 + \gamma_{m,0})}. \quad (15)$$

Then, the energy consumption of UAV $m$ when it relays part of its associated MD $u$’s task-input data to the TBS can be calculated as

$$E_{u,m,0}^{off} = P_{m,0} t_{u,m,0}^{off} = \frac{\phi_{u,m}^{off} I_{u,m} P_{m,0}}{\beta_{m,0} \log_2 (1 + \gamma_{m,0})}. \quad (16)$$

C. Energy Consumption Model

In this subsection, we present the energy consumption of the MDs and UAVs for our proposed two-stage edge computing system. The main objective of this work is to minimize the energy consumption of the system while considering the resource limitations and task completion deadline. Hence, the energy consumption of MDs and UAVs is presented as follows.

1) MDs’ Energy Consumption: The total energy consumption of MD $u$ associated with UAV $m$ for local computing and task offloading can be expressed as

$$E_{u,m}^{tot} = E_{u,m}^{local} + \sum_{n=1}^{N} \delta_{u,m}^{n} E_{u,m}^{off,n}. \quad (17)$$

2) UAVs’ Energy Consumption: Both MDs and UAVs are assumed to be able to perform computing and offloading the tasks simultaneously. It should be noted that the UAV needs to hover at a fixed altitude over the area of interest until all of its associated MDs’ tasks have been processed completely. Here, we consider the scenario where the UAVs don’t need to fly around over the area of interest since multiple UAVs are hovering above the MDs. In essence, the UAVs are already near the MDs enough to offer reliable computing and communication services. Therefore, similar to the work in [25], we don’t consider the propulsion energy of the UAVs in this work. However, in addition to the computing and transmission energy, the hovering energy of the UAVs is also taken into account in minimizing the total energy consumption. The time taken by UAV $m$ to hover over the area while providing communication and computing services to its associated MDs is denoted as

$$t_{m}^{rov} = \max_{u \in U_u} \left\{ \sum_{n=1}^{N} \delta_{u,m}^{n} t_{u,m}^{off,n} + \max \left( t_{u,m}^{e}, t_{u,m,0}^{off} \right) \right\}. \quad (18)$$

The power consumed by UAV $m$ to hover over the area of interest is given by [26], [27]:

$$P_{m}^{rov} = \frac{\zeta \sqrt{\zeta}}{\eta_m \sqrt{0.5 \pi q r^2 \rho}}, \quad (19)$$

where $\zeta$ is the thrust that depends on the mass of UAV, $\eta_m$ denotes the power efficiency of UAV $m$, and $q$ is the number of rotors in each UAV. $r$ and $\rho$ are the diameter of rotor and air...
density, respectively. Hence, the hovering energy consumption of UAV $m$ is calculated as

$$E_{mhov}^m = \rho_{mhov}^m.$$  \hfill (20)

The total energy consumption of UAV $m$ for remote computing, relaying and hovering is expressed as

$$E_{m}^\text{tot} = E_{m}^{\text{hov}} + \sum_{u=1}^{\|U\|} \left( E_{u,m}^f + E_{u,m,0}^o \right).$$  \hfill (21)

3) Total Task Execution Delay: Since the MDs and UAVs are assumed to perform computation and task offloading simultaneously, the total time taken for MD $u$ to finish the task can be expressed as

$$t_{u,m} = \max \left( t_{u,m}^{\text{local}} + \sum_{n=1}^{N} \delta_{u,m}^{n} t_{m}^{\text{off},n} + \max \left( t_{u,m}^{f}, t_{u,m}^{o} \right) \right).$$  \hfill (22)

IV. PROBLEM FORMULATION AND SOLUTION APPROACH

A. Problem Formulation

In this section, we present our proposed joint task offloading, communication and computation resource allocation problem. The objective is to minimize the energy consumption of MDs and UAVs in the system and the optimization problem is formulated as follows:

$$\min_{\delta, t, f, \phi} \sum_{m=1}^{M} \sum_{u=1}^{\|U\|} \left( E_{u,m}^{\text{local}} + \sum_{n=1}^{N} \delta_{u,m}^{n} E_{u,m}^{\text{off},n} \right) + \sum_{m=1}^{M} E_{m}^\text{tot}$$  \hfill (23)

s.t. $t_{u,m} \leq T_{u,m}, \ \forall u \in U_m, \ \forall m \in M,$ \hspace{1cm} (23a)

$0 < t_{u,m}^{\text{off}} \leq 1, \ \forall u \in U_m, \ \forall m \in M,$ \hspace{1cm} (23b)

$\sum_{u=1}^{\|U\|} f_{u,m} \leq f_{m}, \ \forall m \in M,$ \hspace{1cm} (23c)

$f_{u,m}^c \geq 0, \ \forall u \in U_m, \ \forall m \in M,$ \hspace{1cm} (23d)

$0 \leq \phi_{u,m,0} \leq 1, \ \forall u \in U_m, \ \forall m \in M,$ \hspace{1cm} (23e)

$\sum_{n=1}^{N} \delta_{u,m}^{n} \leq 1, \ \forall u \in U_m, \ \forall m \in M,$ \hspace{1cm} (23f)

$\delta_{u,m}^{n} \in \{0, 1\}, \ \forall u \in U_m, \ \forall n \in N, \ \forall m \in M,$ \hspace{1cm} (23g)

where $\delta = \{\delta_{u,m}^{n}\}_{u \in U, m \in M}, f = \{f_{u,m}\}_{u \in U_m, m \in M}$ and $\phi = \{\phi_{u,m,0}\}_{u \in U_m, m \in M}.$ Constraint (23a) ensures that the task of MD $u$ has completed during the tolerable amount of time. Constraint (23b) means that the offloaded task input data size is less than the total input data size. Constraints (23c) and (23d) guarantee that the total allocated computing resources to its associated MDs does not exceed the maximum computing capacity of each UAV. Constraint (23e) states that the task data size offloaded to the TBS is less than MD $u$’s offloaded task at UAV $m$. Constraints (23f) and (23g) ensure that an associated MD of each UAV can only be allocated at most one subchannel.

The formulated optimization problem in (23) is a mixed-integer non-convex problem which cannot be solved in a polynomial-time due to its combinatorial complexity [28]. Moreover, the subchannel assignment variable and the existence of coupling among the variables make it more challenging to solve. Therefore, we apply the BSUM framework to solve our proposed problem and the detail theoretical analysis is presented by leveraging a simple block-structured optimization problem in Section IV-B.

B. Theoretical Background of BSUM

1) Overview: BSUM is a general type of block coordinate descent (BCD) algorithm [29], [30]. It can be applied to address the separable convex optimization problems with smooth or non-smooth linear coupling constraints. Specifically, it successively optimizes the upper bound approximation of the original objective function in a block-by-block manner. One of the key advantages of the BSUM algorithm over BCD is that it can provide a good approximate solution of a non-convex objective function enough for the algorithm to keep going under the practical and theoretical considerations. Let us first introduce the following block-structured optimization problem to shed light on how BSUM algorithm can be applied [29]:

$$\min_{v} g(v_1, v_2, \ldots, v_B),$$  \hspace{1cm} (24)

s.t. $v_i \in \mathcal{V}_i, \ \forall i \in B, \ i = 1, 2, \ldots, B,$

where $g(.)$ is a continuous function of block of variables $v_i$. For every index $i \in B$, $\mathcal{V}_i$ is the close convex set and $\mathcal{V} := \mathcal{V}_1 \times \mathcal{V}_2 \times \ldots \times \mathcal{V}_B$. It is challenging to tackle the above problem in (24) when the objective function is non-convex and non-smooth. Therefore, BSUM algorithm is exploited instead of simply applying BCD which cannot ensure the convergence for every time. Firstly, the upper-bound approximation function $\tilde{g}(v_i, a)$ of the objective function $g(v_i, a_{-i})$ is defined for every feasible point $v \in \mathcal{V}$. We can choose a certain upper-bound approximation function from three common ones: quadratic upper bound, linear upper bound, and Jensen’s upper bound since optimizing the upper-bound approximation function can assure some descent of the original objective function [29].

The following assumptions should be met for the chosen upper-bound approximation function $\tilde{g}(v_i, a)$:

1) $\tilde{g}(v_i, v) = g(v), \ \forall v \in \mathcal{V}, \ \forall i,$

2) $\tilde{g}(v_i, a) \geq g(v_i, a_{-i}), \ \forall v_i \in \mathcal{V}_i, \ \forall a \in \mathcal{V}, \ \forall i,$

3) $\tilde{g}(v_i, a; e)|_{e=a_i} = g'(a; e) + c_i, \ \forall v_i \in \mathcal{V}_i,$

where assumptions 1) and 2) infer that $\tilde{g}(v_i, a)$ is the global upper-bound approximation function of $g(v)$. Assumption 3) ensures that both $\tilde{g}(.)$ and $g(.)$ have the same first-order behavior at the approximation point. To define the upper-bound approximation function, the quadratic penalty term is added to the objective function and it can be expressed as follow:

$$\tilde{g}(v_i, a) := g(v_i, a_{-i}) + \frac{\vartheta}{2} ||v_i - a_i||^2, \ \ (25)$$

where $\vartheta$ is a non-zero and non-negative parameter. Then the problem in (25) can be solved by applying the BSUM algorithm which performs the following update at every
Algorithm 1 BSUM-Based Joint Resource Allocation and Task Offloading

1) Initialization: Set $r = 0$, $\epsilon > 0$, and find the initial feasible points, $(\hat{g}^{(0)}, I^{(0)}, f^{(0)}, \phi^{(0)})$;
2) repeat
3) Choose index set $B_r'$;
4) Let $\delta_i^{(r+1)} = \min_{\delta_i \in D} E_i(\delta_i, \delta^{(r)}, I^{(r)}, f^{(r)}, \phi^{(r)})$;
5) Set $\delta_j^{(r+1)} = \delta_j^{(r)}$, $\forall j \notin B_r'$ and solve $\min_{\delta_i \in D} E_i(\delta_i, \delta^{(r)}, I^{(r)}, f^{(r)}, \phi^{(r)})$;
6) Similarly, solve (31), (32), and (33) to obtain $I_i^{(r+1)}$, $f_i^{(r+1)}$, and $\phi_i^{(r+1)}$ by using steps 1 through 5;
7) Update $r = r + 1$;
8) until $||E_i^{(r)} - E_i^{(r+1)}|| \leq \epsilon$;
9) Apply rounding technique on $\delta_i^{(r+1)}$ to ensure the binary values;
10) Solve $E_i + \tau \Delta$ and evaluate $\mu_i$ until $\mu_i \leq 1$;
11) Finally, $\delta^* = \delta_i^{(r+1)}$, $I^* = I_i^{(r+1)}$, $f^* = f_i^{(r+1)}$, and $\phi^* = \phi_i^{(r+1)}$ are set as the desired solutions.

In essence, it minimizes the upper-bound approximation function by iteratively updating the variable blocks until it obtains the coordinatewise minimum or stationary solution. However, getting the stationary solution, i.e., the global optimal solution is more preferable since the whole vector cannot move to a better direction at that point.

2) Convergence Speed and Complexity: It is implied that BSUM can converge to the $\epsilon$-optimal solution, $\delta^*_i \in \{v_1, v_2, ..., v_N\} \subset \delta_i$, $\tilde{g}(v_i, v', a') - \tilde{g}(v_i, v, a') \leq \epsilon$ in at most $O(\log(1/\epsilon))$ iterations, where $\tilde{g}(v_i, v', a')$ is the global objective value of $\tilde{g}(v_i, a)$. In other words, BSUM can achieve a sublinear rate of convergence. However, a linear rate of convergence can be guaranteed when the objective function is strongly convex or convex [29].

The BSUM-based Joint Resource Allocation and Task Offloading algorithm which is a standard BSUM algorithm as shown in Algorithm 1 minimizes the upper-bound approximation function by iteratively updating the variable blocks until it reaches the stationary point at which the algorithm cannot move to the better minimum direction. Hence, Algorithm 1 can be implied to have a sublinear iteration complexity $O(1/\epsilon)$, where $r$ is the iteration index which has been verified in [29].

C. BSUM-Based Joint Resource Allocation and Offloading

In this section, we present our solution approach to the non-convex problem in (23). The following steps are summarized to achieve the solution of the proposed problem:

- Firstly, we reformulate problem (23) into (27) by relaxing the channel assignment variable.
- Then, we propose the upper bound approximation function of the relaxed problem (27) in (29).
- After that, instead of minimizing problem (27), we minimize the approximation function in (29).
- Finally, we apply the rounding technique to enforce the subchannel assignment variable to be binary value.

First, our proposed problem is reformulated by relaxing the channel assignment variable $\delta_{u,m}$ in constraint (23g) into a continuous form as follows:

$$\min_{\delta, I, f, \phi} E(\delta, I, f, \phi) \triangleq \sum_{m=1}^{M} \left( E_{local}^{m} + \sum_{n=1}^{N} \delta_{u,m}^{n} E_{off}^{n,m} \right) + \sum_{m=1}^{M} E_{tot}^{m}$$

s.t. (23a) – (23f), (27a) – (27b)

Then, to put our proposed problem into the framework of BSUM, the objective function in (27) is rewritten in a simple form as

$$\min_{\delta \in D, I \in L, f \in F, \phi \in \Phi} E(\delta, I, f, \phi)$$

where

$$E(\delta, I, f, \phi) \triangleq \sum_{m=1}^{M} \left( E_{local}^{m} + \sum_{n=1}^{N} \delta_{u,m}^{n} E_{off}^{n,m} \right) + \sum_{m=1}^{M} E_{tot}^{m}$$

is the objective function with the feasible sets of $\delta, I, f, \phi$ given below,

$$D \triangleq \{ \delta : t_u \leq T_{u,m}, \forall u \in U_m, \forall m \in M, \sum_{n=1}^{N} \delta_{u,m}^{n} \leq 1, \forall u \in U_m, \forall m \in M, \delta_{u,m}^{n} \in [0, 1], \forall u \in U_m, \forall n \in N, \forall m \in M, \}$$

$$L \triangleq \{ I : t_u \leq T_{u,m}, \forall u \in U_m, \forall m \in M, 0 < f_{off}^{u,m} \leq 1, \forall u \in U_m, \forall m \in M, \}$$

$$F \triangleq \{ f : t_u \leq T_{u,m}, \forall u \in U_m, \forall m \in M, \sum_{n=1}^{\delta_{u,m}} f_{u,m}^{n} \leq f_{m}^{e}, \forall m \in M, f_{m}^{e} \geq 0, \forall u \in U_m, \forall m \in M, \}$$

$$\Phi \triangleq \{ \phi : t_u \leq T_{u,m}, \forall u \in U_m, \forall m \in M, 0 \leq \phi_{u,m,0} \leq 1, \forall u \in U_m, \forall m \in M. \}$$

The problem in (28) is still non-convex due to the existence of coupling among the variables such as $I, f, \phi$. Hence, to address this problem, we exploit the BSUM algorithm, a general type of block coordinate descent (BCD) algorithm [29], [30]. Literally, it successively minimizes the upper-bound approximation function by updating the blocks of variables in turn and can guarantee a few descent of the original objective function. Here, we define the convex surrogate function $\tilde{E}_i(\delta, \tilde{I}, f, \tilde{\phi})$ by adding the quadratic penalty term to the objective function and it can be described as

$$\tilde{E}_i(\delta; \tilde{I}, f, \tilde{\phi}) := E(\delta; \tilde{I}, f, \tilde{\phi}) + \frac{\partial I}{2} \left\| \delta_i - \tilde{\delta}_i \right\|^2,$$

where $\partial I$ is the positive penalty parameter.
Given the initial feasible points $\hat{\delta}$, $\hat{l}$, $\hat{f}$, and $\hat{\phi}$, instead of minimizing the intractable problem in (28), we minimize the surrogate function in (29) by separating into blocks. It is noted that the problem in (29) is strictly convex because of the quadratic penalty term [31]. Let us suppose $i \in B^c$, where $B^c$ is the set of index blocks at iteration $r$. The similar approach can be applied for other variable blocks $I$, $f$, and $\phi$. At each iteration $r + 1$, we solve the following optimization problems to get the solution of (29),

$$\delta_{i,r+1} = \min_{\delta \in \mathcal{D}} E_i(\delta_i^r, I_i^r, f_i^r, \phi_i^r),$$

$$l_{i,r+1} = \min_{l_i \in \mathcal{L}} E_i(l_i, \delta_i^{r+1}, I_i^r, f_i^r, \phi_i^r),$$

$$f_{i,r+1} = \min_{f_i \in \mathcal{F}} E_i(f_i, \delta_i^{r+1}, l_i^{r+1}, I_i^r, \phi_i^r),$$

$$\phi_{i,r+1} = \min_{\phi_i \in \Phi} E_i(\phi_i, \delta_i^{r+1}, l_i^{r+1}, f_i^{r+1}, I_i^r).$$

Since the solution of the relaxed problem in (27) cannot guarantee the subchannel assignment variable, $\delta_{u,m}$, to be binary value, the rounding technique is adopted to enforce the binary value of $\delta_{u,m}$ [32], [33]. Let suppose the rounding threshold be $\psi \in (0, 1)$. The optimal subchannel assignment value, $\delta_{u,m}^*$ is determined as follows:

$$\delta_{u,m}^* = \begin{cases} 1, & \text{if } \delta_{u,m}^r \geq \psi, \\ 0, & \text{otherwise}. \end{cases}$$

To address the problem of violating the communication resource constraint, we solve $\hat{E}_i + \tau \Delta$ by modifying the communication constraint in (23f) by

$$\sum_{n=1}^{N} \delta_{u,m}^n \leq 1 + \Delta, \quad \forall u \in U_m, \forall m \in \mathcal{M},$$

where $\Delta$ is the maximum violation of the communication constraint and $\tau$ is the penalty parameter of $\Delta$. Then, the value of $\Delta$ is expressed as

$$\Delta = \max \left\{ 0, \sum_{n=1}^{N} \delta_{u,m}^n - 1 \right\}, \quad \forall u \in U_m, \forall m \in \mathcal{M}. \quad (36)$$

Using the value of $\Delta$ and solving $\hat{E}_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*) + \tau \Delta$, we can obtain the integrality gap to verify that the solution achieved from the rounding technique is the best one. The integrality gap can be calculated by [33]

$$\mu_i = \min_{\delta} \frac{\hat{E}_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*)}{E_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*) + \tau \Delta},$$

where $\hat{E}_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*)$ is the solution obtained from the relaxed solution whereas $E_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*) + \tau \Delta$ is the solution achieved after rounding. The best solution can be guaranteed when the value of $\mu_i$ approaches to 1, i.e., $\mu_i \leq 1$. For every relaxation, given $\hat{E}_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*)$ whose instances form a convex set, the oblivious rounding scheme defined as $\hat{E}_i(\delta_i^*, l_i^*, f_i^*, \phi_i^*)$ is individually tight [32].

### V. Simulation Results

#### A. Algorithm Design

In this section, we present the detailed procedures of the proposed approach shown in Algorithm 1. In the initialization step of the proposed algorithm, we determine the initial feasible points, $(\delta^{0}, l^{0}, f^{0}, \phi^{0})$, of problem (29) by setting $r = 0$ and $\epsilon$ as a small positive number. Then, at each iteration $r$, the index set $B^c$ is selected to begin the iterative process. The updated solution is obtained at every iteration $r + 1$ by solving problems (30), (31), (32), and (33) until the convergence condition is met, i.e., $\|E^{(r)} - E^{(r+1)}\| \leq \epsilon$.

To enforce the solution obtained from (30) to be a binary value, we apply rounding technique to it and solve $\hat{E}_i + \tau \Delta$. Finally, $\delta_{i,r+1}^r$, $l_{i,r+1}^r$, $f_{i,r+1}^r$, and $\phi_{i,r+1}^r$ are considered as the desired solutions.

#### B. Simulation Environment

We consider the area of interest to be $300 \times 300$ m in which there are 5 MEC-enabled UAVs and 30 MDs. The location of the TBS is set at $(0, 0, 0)$. The MDs are randomly distributed in the considered area and the association between UAVs and MDs is determined by using k-means clustering algorithm. UAVs are assumed to be hovering at the fixed altitude of 150 m during the considered time interval. Unless stated otherwise, the values of simulation parameters are listed in Table II.

In Fig. 2, we present the association of MDs to UAVs by exploiting the k-means clustering algorithm. As we can see from Fig. 2 that the number of associated MDs to UAV 1, UAV 2, UAV 3, UAV 4, and UAV 5 are 8, 6, 5, 6, and 5, respectively. Since, the association of MDs to UAVs is determined based on the distance, the MDs can experience better line-of-sight link as well as minimize the transmission energy consumption to offload their tasks.

#### C. Convergence Analysis

In Fig. 3, we illustrate the convergence of our proposed algorithm by applying three coordinate selection rules [29], namely, cyclic, Gauss-Southwell and randomized for two scenarios: $\vartheta = 0.1$ and $\vartheta = 10$. As we can observe from Fig. 3, the proposed algorithm converges to a coordinate-wise minimum and stationary point at which the vectors

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| $h_0$     | -50 dB| $\alpha$  | 2     |
| $P_{u,m}$ | 1 mW  | $P_{m,0}$ | 1 W   |
| $N_0$     | -170 dBm| $B$  | 20 MHz |
| $J_{u,m}$ | [200, 700] MB | $\Omega_{u,m}$ | 1000 cycles |
| $f_{u,m}$ | [0.5, 3] MHz | $f_m$ | [1.2, 2] GHz |
| $k$       | $10^{-28}$ | $\omega$ | 180 kHz |
| $\zeta$   | 30 N | $\eta_m$ | 70% [26] |
| $\rho$    | 4 [26] | $\tau$ | 0.254 m [26] |
| $\rho$    | 1.225 kg/m$^3$ | $k'$ | $10^{-28}$ |
Fig. 2. Association of MDs with UAVs.

Fig. 3. Convergence analysis.

\[ \delta^* = \delta_i^{(r+1)}, f^* = f_i^{(r+1)}, \phi^* = \phi_i^{(r+1)} \]

cannot find the better minimum direction.

D. Offloaded Data Analysis

The variation of the offloaded data size of MDs depending on the tolerable task completion deadline is illustrated in Fig. 4. When the MDs are more tolerable to the task completion time, they will offload less task data to the UAVs so that the communication resources can be less consumed. Nevertheless, MDs will offload their tasks more when there are more UAVs in order to save their energy on the local computation.

The data size of the task relayed to the TBS by the UAVs versus the task input data size is plotted in Fig. 5. As we can see from Fig. 5 that the portion of task data relayed to TBS increases when the task input data size of the MDs increases. The reason is that the UAVs will relay more task to the TBS due to their limited CPU resources and task completion deadline. On the other hand, the offloaded portion to the TBS will decrease with the increasing number of UAVs. The reason is that UAVs will handle the offloaded tasks from the MDs to meet the task completion deadline constraint when they have sufficient resources to serve the associated MDs.

Fig. 4. The impact of task completion deadline on offloaded data size of MDs to UAVs.

Fig. 5. The impact of task input data size on the size of tasks relayed to the TBS by UAVs.

Fig. 6. Tasks relayed to the TBS by UAVs vs. task completion deadline.
to the TBS will increase. Due to UAVs’ limited computing resources, more task data will be relayed to the TBS by the UAVs to meet the stringent task completion deadline and save the energy consumption. When the more number of UAVs are deployed, the less task data will be relayed to the TBS. However, when the task completion deadline are more tolerable, the gap between them will become narrower.

E. Energy Consumption Analysis

In Fig. 7, we depict how the amount of computing resources (CPU cycles) of UAVs impacts their energy consumption. We can observe from Fig. 7 that the energy consumed by the UAVs increases with the amount of CPU resources. This is because the MDs tend to offload more tasks to UAVs which have rich computing resources and as a result, UAVs consume more energy on the processing of the tasks. Moreover, the number of MDs associated with the UAVs affects the energy consumption of the UAVs. To show that, we simulate by considering a different number of MDs. It is obvious that more energy will be consumed when there are more number of MDs to be served by the UAVs in the system.

As we can see from Fig. 8, the total energy consumption of the system increases with the task input data size of the MDs. However, our proposed scheme achieves the minimum energy consumption when compared to other baselines: 1) equal offloading where the same proportion of the MDs’ tasks is offloaded to the UAV and TBS in addition to local computing at the MD, and 2) offloading all where all the tasks of MDs are computed at the UAVs and TBS without considering the local computing at the MDs. The total energy consumption of the system in offloading all is the maximum since the MDs consume energy for transmitting all the tasks to the UAVs and the UAVs have to consume much more energy for the computing and relaying of all the tasks offloaded from the MDs. It is also observed that the gap between our proposed scheme and equal offloading becomes narrower when the total input data size of the task increases.

The effects of the number of available subchannels on the MDs’ energy consumption and offloaded data size are given in Fig. 9. As we can observe from Fig. 9, the energy dissipated by the MDs reduces with the increasing number of subchannels. This is because MDs can minimize their transmit power by selecting the more favorable subchannel while offloading their tasks to the UAVs. On the other hand, they can offload more data to the UAVs by saving energy consumption on data transmission.

Moreover, in Fig. 10, we have illustrated the comparisons of our proposed method with the following baselines,

- **Relaying**: In this method, UAV is only considered as the relaying platform which transmits all the MDs’ offloaded tasks to the TBS for the remote computing.
- **Baseline1**: In this approach, the tasks are exclusively processed by the UAV in addition to local computing at the MDs and the collaboration with TBS is not considered [15].
- **Baseline2**: In this approach, the tasks of MDs are computed locally and at the UAV. However, the hovering energy of the UAV is not taken into account [16].

In Fig. 10, we can observe that the total energy consumption of the system increases with the number of MDs.
The performance of our propose approach is much better than Baseline2 because the optimization of UAV’s hovering energy is not considered in [16]. Moreover, the total energy consumption in our proposed approach is lower than the other two baselines such as Relaying and Baseline1. The reason is that the UAV must consume much more energy to transmit the tasks to the TBS in Relaying. On the other hand, Baseline1 does not consider TBS to assist the UAV for computing the MDs’ tasks and hence the energy consumption of UAV becomes higher to compute the tasks timely. Therefore, we can summarize that the total energy consumption while considering UAVs only is higher than the UAVs and TBS collaboration under the task deadline constraint.

VI. CONCLUSION

In this paper, we have studied a multi-UAV-assisted two-stage MEC system in which MEC-enabled UAVs provide computing and relaying services to the MDs. Taking into account the tolerable delay of the tasks and the limited communication/computation resources of the UAVs, we have formulated a joint resource allocation and offloading problem with the objective of minimizing the total energy consumption of the MDs and UAVs. Since the formulated optimization problem is a mixed-integer non-convex problem which is NP-hard, we first relaxed the channel assignment variable and reformulated the problem. However, the reformulated problem is still non-convex due to the coupling among the variables. To address that problem, the BSUM algorithm has been deployed. The simulation results have shown that the proposed approach can reduce the energy consumption of the network and outperformed the baseline schemes. In our future work, we will investigate the resource allocation and trajectory optimization of multi-UAV-assisted MEC system by taking into account the flying/propulsion energy of the UAVs.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[2] X. Hu, K. K. Wong, K. Yang, and Z. Zheng, “UAV-assisted relaying and edge computing: Scheduling and trajectory optimization,” IEEE Trans. Wireless Commun., vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” IEEE Trans. Wireless Commun., vol. 15, no. 6, pp. 3949–3963, Jun. 2016.

[4] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, “Beyond 5G with UAVs: Foundations of a 3D wireless cellular network,” IEEE Trans. Wireless Commun., vol. 18, no. 1, pp. 357–372, Jan. 2019.

[5] Y. Zeng, “Accessing from the sky: A tutorial on UAV communications for 5G and beyond,” Proc. IEEE, vol. 107, no. 12, pp. 2327–2375, Feb. 2019.

[6] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, May 2016.

[7] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, “Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,” IEEE J. Sel. Areas Commun., vol. 35, no. 5, pp. 1046–1061, May 2017.

[8] X. Jiang, M. Sheng, N. Zhao, C. Wang, and X. Wang, “Green UAV communications for 5G: A survey,” Chin. J. Aeronom., May 2021, doi: 10.1016/j.cj.2021.04.025.

[9] M. Yu, A. Liu, N. N. Xiong, and T. Wang, “An intelligent game based offloading scheme for maximizing benefits of IoT-Edge-cloud ecosystems,” IEEE Internet Things J., early access, Nov. 23, 2020, doi: 10.1109/JIOT.2020.3039828.

[10] H. Guo, J. Liu, J. Ren, and Y. Zhang, “Intelligent task offloading in vehicular edge computing networks,” IEEE Wireless Commun., vol. 27, no. 4, pp. 126–132, Aug. 2020.

[11] H. Guo, J. Liu, and J. Lv, “Toward intelligent task offloading at the edge,” IEEE Netw., vol. 34, no. 2, pp. 128–134, Mar. 2020.

[12] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, “Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks,” IEEE J. Sel. Areas Commun., vol. 38, no. 1, pp. 217–228, Jan. 2020.

[13] X. Ren, T. Chen, Z. Zhao, H. Zhang, M. Bennis, and Y. Ji, “Resource awareness in unmanned aerial vehicle-assisted mobile-edge computing systems,” in Proc. IEEE 91st Veh. Technol. Conf., Antwerp, Belgium, May 2020, pp. 1–6.

[14] H. Guo and J. Liu, “UAV-enhanced intelligent offloading for Internet of Things at the edge,” IEEE Trans. Ind. Informat., vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[15] Y. K. Tun, Y. M. Park, N. H. Tran, W. Saad, S. R. Pandey, and C. S. Hong, “Energy-efficient resource management in UAV-assisted mobile edge computing,” IEEE Commun. Lett., vol. 25, no. 1, pp. 249–253, Jan. 2021.

[16] M. Alsenwi, Y. K. Tun, S. R. Pandey, N. N. Ei, and C. S. Hong, “UAV-assisted multi-access edge computing system: An energy-efficient resource management framework,” in Proc. Int. Conf. Inf. Netw., Barcelona, Spain, Jan. 2020, pp. 214–219.

[17] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in UAV-enabled mobile edge computing networks,” IEEE Trans. Wireless Commun., vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[18] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, “Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing,” IEEE Internet Things J., vol. 6, no. 2, pp. 3686–3699, Apr. 2019.

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

[20] Q. Zhang, J. Chen, L. Ji, Z. Feng, Z. Han, and Z. Chen, “Response delay optimization in mobile edge computing enabled UAV swarm,” IEEE Trans. Veh. Technol., vol. 69, no. 3, pp. 3280–3295, Mar. 2020.

[21] J. Li, Q. Liu, P. Wu, F. Shu, and S. Jin, “Task offloading for UAV-based mobile edge computing via deep reinforcement learning,” in Proc. IEEE Int. Conf. Commun. China, Aug. 2018, pp. 798–802.

[22] M. Wang, S. Shi, S. Gu, X. Gu, and X. Qin, “Q-learning based computation offloading for multi-UAV-enabled cloud-edge networking,” IEEE Trans. Veh. Technol., vol. 16, no. 15, pp. 2481–2490, Sep. 2020.

[23] Q. Wu and R. Zhang, “Common throughput maximization in UAV-enabled OFDMA systems with delay consideration,” IEEE Trans. Wireless Commun., vol. 66, no. 12, pp. 6614–6627, Dec. 2018.
Y. P. Hong, T.-C. Hsu, and P. Chennakesavula, “Wireless power transfer for distributed estimation in wireless passive sensor networks,” IEEE Trans. Signal Process., vol. 64, no. 20, pp. 5382–5395, Oct. 2016.

M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, “A unified algorithmic framework for block-structured optimization involving big data. With applications in machine learning and signal processing,” IEEE Signal Process. Mag., vol. 33, no. 1, pp. 57–77, Jan. 2016.

A. Ndikumana et al., “Joint communication, computation, caching, and control in big data multi-access edge computing,” IEEE Trans. Mobile Comput., vol. 19, no. 6, pp. 1359–1374, Jun. 2020.

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

U. Feige, M. Feldman, and I. Talgam-Cohen, “Oblivious rounding and the integrality gap,” in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, vol. 60, K. Jansen, C. Mathieu, J. D. P. Rolim, and C. Umans, Eds. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 1–23. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2016/6631

A. Ndikumana et al., “Deep learning based caching for self-driving cars in multi-access edge computing,” IEEE Trans. Intell. Transp. Syst., early access, Mar. 4, 2020, doi: 10.1109/TITS.2020.2976752.

Choong Seon Hong (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University in 1997. He joined KT, where he was involved in broadband networks, as a Member of Technical Staff. In 1993, he joined Keio University, Japan. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Engineering, Kyung Hee University. His research interests include future internet, ad-hoc networks, network management, and network security. He is a member of the ACM, the IEICE, the IPSI, the KIEEE, the KICS, the KIPS, and the OSIA. He has served as the General Chair, the TPC Chair/Membership, or an Organizing Committee Member for international conferences, such as NOMS, IM, APNOMS, E2EMON, CCNC, ADNS, ICPP, DIM, WISA, BnC, TINA, SAINT, and ICOIN. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the International Journal of Network Management, and the Journal of Communications and Networks; and an Associate Technical Editor of the IEEE Communications Magazine.

Madyan Alsenwi received the M.Sc. degree in electronics and communications engineering from Cairo University, Egypt, in 2016, and the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2021. Before joining KHU, he worked as a Researcher under several research projects funded by the Information Technology Industry Development Agency (ITIDA), Egypt. His research interests include resource management in next-generation wireless networks, ultra-reliable low-latency communications (URLLC), UAV-assisted wireless networks, and machine learning.

Yan Kyaw Tun (Member, IEEE) received the B.E. degree in marine electrical systems and electronics engineering from Myanmar Maritime University, Thanlyin, Myanmar, in 2014, and the Ph.D. degree in computer science and engineering from Kyung Hee University, Seoul, South Korea, in 2021. He is currently working as a Post-Doctoral Researcher with the Networking Intelligence Laboratory, Kyung Hee University. His research interests include network economics, game theory, network optimization, wireless communication, wireless network virtualization, mobile edge computing, and wireless resource slicing for 5G. He received the Best Ph.D. Thesis Award in engineering in 2020.