Multi-UAV-Enabled Mobile Edge Computing for Time-Constrained IoT Applications

Cheng Zhan, *Member, IEEE,* Han Hu, *Member, IEEE,* Zhi Liu, *Member, IEEE,* Zhi Wang, *Member, IEEE,* Shiwen Mao, *Fellow, IEEE*

***Abstract*—Unmanned aerial vehicle (UAV)-enabled mobile edge computing (MEC) has emerged as a promising paradigm to extend the coverage of computation service for Internet of Things (IoT) applications, which are usually time-sensitive and computation-intensive. In this paper, a novel design framework is proposed for a multi-UAV-enabled MEC system, where edge servers are equipped on multiple UAVs to provide flexible computation assistance to IoT devices with hard deadlines. The aim is to maximize the number of served IoT devices through jointly optimizing UAV trajectory and service indicator as well as resource allocation and computation offloading, where the chosen IoT devices will complete their computation tasks on time under given energy budgets and co-channel interference is taken into account. We formulate the optimization problem as a mixed integer nonlinear programming (MINLP), which is challenging to solve directly. The problem is first reformulated to a more mathematically tractable form by adding a penalty term to the objective function. We then decouple the problem into two sub-problems and develop an iterative algorithm by solving the two sub-problems with alternating optimization and successive convex approximation techniques, where the proposed algorithm converges to a Karush-Kuhn-Tucker (KKT) solution. In addition, an efficient initialization scheme is proposed based on multiple traveling salesman problem with time windows (m-TSPTW) method. Finally, simulation results are provided to demonstrate that the proposed joint design achieves significant performance gains over baseline schemes.**

***Index Terms*—Unmanned aerial vehicles, timely edge computing, Internet of Things (IoT), resource allocation.**

I. Introduction

With the rapid development of software and hardware technologies, an increasing number of devices (e.g. smart cameras, smartphones, smart vehicles, etc) can access the Internet for various services (e.g., game, video, smart city etc.), and Internet of Things (IoT) devices as well as their applications have become more pervasive. Such explosive growth of devices generate a large amount of data which needs storage and computation processing. As reported in [1] that global mobile data traffic is expected to reach 77.5 exabytes per month by 2022. Although this will come with positive impacts including new applications, e.g., virtual reality (VR), augmented reality (AR), and high-definition (HD) video streaming, it also poses pressure on the existing network infrastructure with critical requirements on communication and computation.

However, most of the IoT devices, such as sensors and mobile devices, have limited capacity for communication, computing, and storage, which has great impact on the performance of the above resource-hungry IoT applications. As such, computation cannot be performed solely on the IoT devices and mobile edge computing (MEC) is introduced [2], [3]. MEC provides computational service near the edge of structured core network. As such, tasks and data generated from IoT devices no longer need to be forwarded to remote IoT clouds, but just be analyzed and processed immediately near the edge of the mobile network, which not only eases backhaul impacts but also facilitates low latency. In addition, by offloading computation tasks to edge servers, the energy consumption of IoT devices can be saved compared to the traditional cloud computing approach [4], [5]. However, the geographic areas of IoT applications may be still far from the the edge of core network, e.g., video surveillance for precision agriculture, smart factory, underwater monitoring, etc. It would be difficult for IoT devices to access the remote MEC server with reliable links due to the long distance [6]. Moreover, IoT devices are flexibly and widely distributed in and it is difficult for the conventional ground base station to provide enough service coverage.

In contrast to edge servers at fixed locations, unmanned aerial vehicle (UAV)-enabled MEC has recently emerged as a promising paradigm to extend the coverage of computation service for IoT applications and provide more flexible and cost-efficient computing services [7], [8]. Such UAV-enabled MEC platforms can also provide fast and flexible deployment in emergency scenarios, e.g., traffic disruptions in intelligent transport systems, etc. In particular, by deploying MEC servers on UAVs, computation tasks can be offloaded from ground IoT devices to the aerial MEC servers. Due to the agility for convenient deployment and its desired channel conditions to devices, UAV-enabled MEC can be deployed with flexible mobility and support reliable line-of-sight (LoS) connections to devices. Consequently, more data can be transmitted from IoT devices when the UAVs are close enough through the LoS

This work was supported in part by the National Natural Science Foundation of China under Grants 61702426 and 61971457, by the Innovation Support Program for Chongqing Overseas Returnees under Grant cx2020122, Shenzhen Science and Technology Program under Grant No. RCYX20200714114523079, and by the NSF under Grant ECCS-1923163. Han Hu is the corresponding author.

Cheng Zhan is with the School of Computer and Information Science, Southwest University, Chongqing, China. Email: zhanc@swu.edu.cn

Han Hu is with the School of Information and Electronics, Beijing Institute of Technology, Beijing, China. Email: hhu@bit.edu.cn.

Zhi Liu is with the Department of Mathematical and Systems Engineering, Shizuoka University, Hamamatsu 432-8561, Japan. Email: liu@ieee.org.

Zhi Wang is with the Graduate School at Shenzhen, Tsinghua University, China. Email: wangzhi@sz.tsinghua.edu.cn

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, USA. Email: smao@ieee.org.

channels, such that the computation offloading related energy consumption at devices can be greatly reduced [9].

On the other hand, many tasks generated by IoT devices usually need to be accomplished with strict time constraints. In other words, the computation tasks should be finished before their deadlines; otherwise, the input data becomes outdated and loses its value. Such tasks are typically energy-consumptive, computing-intensive, and time-sensitive. In general, it is difficult to satisfy the latency requirements for all IoT devices. As a result, the effective design of task offloading and device selection scheme becomes important. Furthermore, a single UAV usually has limited computation capability, which motivates the deployment of multiple UAVs to serve IoT devices cooperatively to achieve more efficient offloading service. As such, IoT devices are served simultaneously with higher computation capacity and lower access delay. Such multi-UAV-enabled MEC should be carefully designed to work efficiently and cooperatively, i.e., the resources (both computation and communication) should be efficiently allocated.

Motivated by the issues discussed above, in this paper, we propose a multi-UAV-enabled network architecture to provide edge computing for IoT devices with target hard deadlines. More flexible services to the area can be provided by dynamically adjusting the UAVs' positions. Our aim is to maximize the number of served IoT devices through jointly optimizing UAV trajectory and service indicator as well as resource allocation and computation offloading, while the chosen IoT devices will complete their computation task on time under given energy budgets. We summarize the main contributions of this paper as follows.

- First, a novel design framework is proposed for multi-UAV-enabled MEC to provide computation services for time-constrained IoT applications. The UAVs cooperatively serve the offloaded tasks to meet their deadlines. It is assumed that all UAVs operate at the same frequency band, where co-channel interference is considered.
- Second, an optimization problem is formulated to maximize the number of served IoT devices by optimizing UAV trajectory jointly with service indicators as well as resource allocation and computation offloading. Our problem formulation considers both IoT devices' energy budget and dynamic allocation of CPU frequency for computing. The formulated optimization problem is mixed integer nonlinear programming (MINLP), which is challenging to solve for optimal solutions.
- Third, we reformulate the problem to a more tractable form using deductive penalty functions, and then decompose it into two sub-problems: (i) computation offloading and service indicator optimization as well as resource allocation, and (ii) UAV trajectory optimization. An efficient iterative algorithm is proposed by jointly solving the two sub-problems with alternating optimization and successive convex approximation (SCA) techniques. We also prove that the proposed algorithm is guaranteed to converge to a solution satisfying the Karush-Kuhn-Tucker (KKT) conditions.
- Finally, a systematic initialization scheme is proposed based on multiple traveling salesman problem with time windows (m-TSPTW) method. Simulations results are provided to demonstrate that the proposed scheme is more advantageous and effective compared with several benchmark schemes. Furthermore, new insights for multi-UAV movement for time-constrained IoT applications have been revealed.

The rest of this paper is organized as follows: Section II introduces the related work. Section III presents the multi-UAV-enabled MEC system model and problem formulation. Section IV reformulates the problem and develops an effective iterative algorithm, and analyzes its convergence and complexity. Section V presents our simulation study and the conclusion is given in Section VI.

II. Related Work

In this section we introduce the related work on MEC and UAV-enabled MEC.

A. Mobile Edge Computing

Compared to the traditional cloud computing, MEC puts forward new requirements for equipment requirements and related technologies, such as computation offloading techniques that characterize the network with low-latency. Computation offloading and resource allocation are very critical aspects in affecting communication and service quality over MEC system. The problem of delay minimization for task offloading ultra-dense network was studied in [10], subject to the battery constraints. In [11], an energy-efficient computation offloading scheme was proposed for MEC system in heterogeneous cellular Networks networks, such that the total energy consumption of all ground devices is minimized. The work in [12] studied the computation offloading problem over a software-defined access network, where edge computing services are provided by multi-hop access-points (APs). The work in [13] analyzed effective capacity of MEC with a two-stage tandem queue, based on which a joint computation and bandwidth resource allocation problem was solved such that the total revenue of the network was maximized. The work in [14] maximized the throughput and minimized the operational cost in a mobile edge cloud network, where a specified network function is requested by each offloading task with a tolerable delay.

Delay-sensitive computation offloading over MEC system should be taken into account to satisfy the quality-of-service (QoS) requirements of heterogeneous devices. The joint transmission scheduling and computation offloading for delay-sensitive applications was investigated in [17] by leveraging game theory. The work in [18] studied task delay minimization problem in non-orthogonal multiple access (NOMA) enabled MEC networks, where computation tasks can be offloaded simultaneously for multiple users. The work in [19] minimized the weighted sum of average number of requested resources and average completion time of jobs in IoT edge computing with deep reinforcement learning. In [20], the cost for periodic tasks in MEC system was minimized based on game theory, where the cost is represented by the linear combination of the task completion time and energy consumption. In summary,

MEC over traditional cellular networks has been well studied in the literature.

B. UAV-Enabled Mobile Edge Computing

Due to the flexibly and high mobility of the UAV, UAV communication has attracted significant research as UAVs can substitute for the terrestrial base stations such that the network coverage area can be further enlarged [21]. Unlike ground MEC networks, UAV-enabled MEC provides wide coverage and flexible mobility management for computation offloading. The work in [22] considered a UAV-enabled MEC wireless powered system and studied the computation rate maximization problem with UAV's speed and energy-harvesting constraints. In [23], a ground-air-space integrated network edge computing architecture was proposed, where edge computing was provided by flying UAVs. In [24], orthogonal frequency-division multiple access scheme was employed in UAV-enabled network for serving multiple ground users with different delay requirements. The authors in [25] investigated a UAV-enabled wireless powered cooperative MEC system, where an MEC server and energy transmitter were equipped at a UAV to provide both computing and energy services to ground devices. The economics of UAV-enabled service provisioning was studied in [26], where the dynamic service, capacity allocation, and UAV-network deployment were taken into account. In [27], the queuing delay behavior was investigated with software-defined coexisting WiFi AP and UAV-mounted base station, via designing the AP traffic offloading and UAV positioning. However, only a single flying UAV MEC server was employed in these existing works to provide computation service.

Multiple UAVs can cooperate to provide MEC services for ground IoT nodes [28]. The authors in [29] integrated a UAV swarm and edge/cloud computing together and proposed a hybrid UAV-edge-cloud computing model, such that high quality of service could be guaranteed. A new architecture was proposed in [30] for UAV clustering to enable efficient multi-task multi-modal offloading. In [31], a differential evolution based multi-UAV deployment scheme was presented by considering the average transmission cost and load balancing requirement. The work [6] proposed a UAV-enabled MEC framework where each UAV was able to turn on and off the onboard computing elements. However, these works only considered one snapshot optimization without considering the UAV trajectory optimization. Different from the UAV-enabled coordinated multi-points (CoMP) transmission [15], [16], the delay-sensitive computation offloading should be taken into account for the trajectory design over multi-UAV enabled MEC. The above existing works relied on the assumption that edge computing can complete the computation task without considering its strict deadline. Unlike the aforementioned studies, we study the problem of multi-UAV-enabled MEC in this paper for time-constrained IoT applications with target hard deadlines, the cooperation among multiple UAVs in the MEC network, as well as co-channel interference.

Notations: In this paper, matrices and vectors are represented by boldface upper-case and lower-case letters, respectively. The Euclidean norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|$. The space of $m \times n$ real matrices is denoted by $\mathbb{R}^{m\times n}$. $\mathbb{E}[\cdot]$ and $Var(\cdot)$ denote the statistical expectation and variation operations, respectively. The cardinality of a set X is denoted by $|X|$. $O(\cdot)$ denotes the standard big-O notation.

III. System Model and Problem Statement

In this section, we first give the system model and then formulate the optimization problem. We summarize the main notations used in this paper in Table I.

TABLE I
SUMMARY OF MAIN NOTATIONS AND DEFINITIONS

Notation	Definition
B	Signal bandwidth (Hz)
H	The altitude of UAVs (m)
M	The number of UAVs
u_m	The m-th UAV
K	The number of IoT devices
s_k	The k-th device
T	Mission completion time required for all UAVs (s)
N	The number of time slots for T
δ_t	The element slot length (s)
$\mathbf{g}_k$	The horizontal coordinate of device s_k
I_k	The size of computation task-input data for device s_k (bit)
C_k	The number of CPU cycles required for computing 1-bit of input data for s_k
D_k	The maximum tolerable latency for task completion at device s_k (s)
$P_k^{\max}$	The maximum transmit power of device s_k (W)
$f_k[n]$	CPU frequency of s_k at time slot n
$f_{m,k}^U[n]$	Allocated CPU frequency of UAV u_m for task offloaded from s_k at time slot n
$f_k^{\max}$	The maximum allowable CPU frequency of device s_k
$f_m^{U,\max}$	The maximum allowable CPU frequency of UAV u_m
$\mathbf{u}_I$	The initial/final positions of all UAVs
$\mathbf{u}_m[n]$	The horizonal location of UAV u_m at time slot n
$d_{\min}$	The minimum inter-UAV distance to avoid collision
$V_{\max}$	The maximum UAV speed (m/s)
$a_k[n]$	The fraction of time allocated for device s_k to offload task to UAV u_m at time slot n
$p_k[n]$	The transmit power used by device s_k for computation offloading
$h_{m,k}[m]$	The channel coefficient between s_k and UAV u_m at time slot n
$R_{m,k}[m]$	The achievable rate between s_k and UAV u_m at time slot n (bps)
θ_k	The service indicator to illustrate whether computation task at device s_k can be finished before its deadline or not
κ	The effective switched capacitance of CPU
$E_k^{\max}$	The energy budget of device s_k

A. Network Model

As shown in Fig. 1, we consider a multi-UAV enabled MEC system, which consists of M UAVs and K ground IoT devices (e.g., smart cameras). The set of UAVs and IoT devices are represented by $\mathcal{M} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{K} = \{s_1, s_2, \ldots, s_K\}$, respectively. Denote $\mathbf{g}_k \in \mathbb{R}^{2\times 1}$ as the horizontal coordinates of the kth device on the ground. We consider a time-constrained application in the IoT system, e.g., video analysis or face recognition, etc. Each IoT device s_k has a computation task S_k to be completed. In particular, S_k can be described as a triple tuple $S_k = (I_k, C_k, D_k)$, where I_k denotes the task-input data size measured in bits (e.g., input data and program codes), C_k denotes the number of CPU cycles required to finish one bit of the computation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

task, and D_k denotes the maximum tolerable latency measured in seconds for computation task S_k. In other words, S_k should be finished before its deadline D_k; otherwise, the task-input data loses its significance and becomes irrelevant (e.g., the person to be recognized is out of the surveillance area).

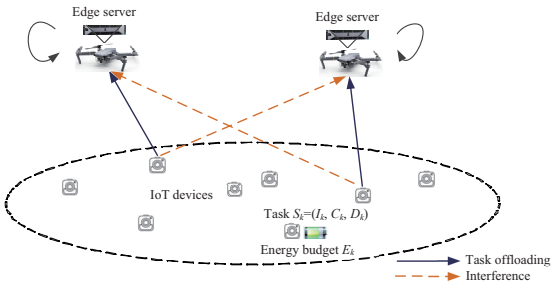


Fig. 1. System model for the multi-UAV enabled MEC system

To mitigate the limited computing capabilities of IoT devices, the M UAVs equipped with MEC servers are deployed to provide edge computing services within time horizon T. In practice, T can be set based on the limited flying/hovering time of UAVs [32]. The initial/final locations of all UAVs are pre-determined, denoted by $\mathbf{u}_I \in \mathbb{R}^{2\times 1}$. In general, $\mathbf{u}_I$ corresponds to the depot where UAVs can replenish energy. Since level flight is less energy-consuming than frequent ascending or descending [33], [35], all UAVs are assumed to fly at a constant altitude of H, corresponding to the the minimum UAV flying altitude for building avoidance. We assume that the UAVs are assigned with an exclusive frequency band such that there exists no inter-cell interference from co-channel base stations. To facilitate the trajectory design, we discretize the time horizon T into N equal time slots with slot length δ_t. In practice, δ_t is chosen to be small enough such that the distances between all devices and UAVs are considered to be constant within each time slot. Denoting the horizontal location of UAV u_m at time slot n by $\mathbf{u}_m[n] \in \mathbb{R}^{2\times 1}$, the flying trajectory of UAV u_m can be approximated by the discrete set $\{\mathbf{u}_m[n], 1 \le n \le N\}$. As such, a continuous UAV trajectory can be easily constructed by connecting the locations in $\{\mathbf{u}_m[n]\}$ with line-segments.

The distance between UAV u_m and IoT device s_k at time slot n can be calculated as $d_{m,k}[n] = \sqrt{H^2 + \|\mathbf{u}_m[n] - \mathbf{g}_k\|}$. Furthermore, the UAV's maximum speed is denoted as $V_{\max}$ due to the mechanical limitation. Then we have the following flying constraints.

$$\|\mathbf{u}_m[n+1] - \mathbf{u}_m[n]\| \le V_{\max}\delta_t, \forall m, n, \quad (1)$$

$$\mathbf{u}_m[1] = \mathbf{u}_I, \mathbf{u}_m[N] = \mathbf{u}_I, \forall m, \quad (2)$$

$$\|\mathbf{u}_m[n] - \mathbf{u}_i[n]\| \ge d_{\min}, \forall m, i > m, n = 2, \ldots, N-1, \quad (3)$$

where $d_{\min}$ is the minimum inter-UAV distance for collision avoidance. We assume that the partial offloading strategy is adopted [36], [37], where each task can be divided into two parts: one for local computing at IoT devices, and the other for offloading to the associated UAV for remote computing.

B. Communication Model

The channels between UAVs and IoT devices are assumed to follow the quasi-static fading model, where the corresponding channel coefficients remain the same within each time slot, but may vary across different time slots [38], [39]. Denote $h_{m,k}[n]$ as the channel coefficient between UAV u_m and device s_k at time slot n. Then $h_{m,k}[n]$ can be represented by $h_{m,k}[n] = \sqrt{\beta_{m,k}[n]}\tilde{h}_{m,k}[n]$, where $\beta_{m,k}[n]$ and $\tilde{h}_{m,k}[n]$ account for large-scale channel coefficient and small-scale fading, respectively. $\beta_{m,k}[n]$ can be calculated by $\beta_{m,k}[n] = \beta_0 d_{m,k}^{-\alpha}[n]$, where α and β_0 denote the path-loss exponent average channel power gain at 1 m, respectively. $\tilde{h}_{m,k}[n]$ is represented by a complex-valued random variable with $\mathbb{E}[|\tilde{h}_{m,k}[n]|^2] = 1$. Typically, $\tilde{h}_{m,k}[n]$ can be modelled by Rician fading with Rician factor K_c [39]–[41]. In particular, $\tilde{h}_{m,k}[n] = \sqrt{\frac{K_c}{K_c+1}}\bar{g} + \sqrt{\frac{1}{K_c+1}}\tilde{g}$, where $\bar{g}$ is the deterministic LoS component with $|\bar{g}| = 1$ and $\tilde{g}$ is random Non-line-of-sight (NLoS) component modelled by a zero-mean unit-variance circularly symmetric complex Gaussian (CSCG) random variable. Therefore, the variance for $\tilde{h}_{m,k}[n]$ is given as $Var(\tilde{h}_{m,k}[n]) = \frac{1}{K_c+1}$. It is assumed that the channel distribution information (CDI) (including β_0, α, K_c) is available prior to the UAVs' flight.

We assume that all UAVs operate at the same frequency and the K IoT devices offload their computation tasks to the associated UAVs with time-division multiple access (TDMA) scheme [25], [42]. Let $a_{m,k}[n]$ be the fraction of time allocated to device s_k for offloading computation task to UAV u_m at time slot n, $0 \le a_{m,k}[n] \le 1, \forall k, n$. Thus, $a_{m,k}[n]$ specifies not only computation offloading but also UAV-device association at each time slot. Due to the TDMA scheme, at each time instant, each UAV receives the offloaded task from at most one IoT device. Furthermore, each IoT device is assumed to connect to at most one UAV at each time instant. Thus, we have

$$\sum_{k=1}^{K} a_{m,k}[n] \le 1, \forall m, \quad (4)$$

$$\sum_{m=1}^{M} a_{m,k}[n] \le 1, \forall k. \quad (5)$$

As a result, a device can be served by different UAVs but not at the same time.

Denote $p_k[n]$ as the transmit power used by device s_k for computation offloading, $0 \le p_k[n] \le P_k^{\max}$, where $P_k^{\max}$ represents for the maximum transmit power of IoT device s_k. As such, the achievable rate at time slot n for computation offloading from device s_k to UAV u_m, measured in bits per second (bps), can be expressed by

$$R_{m,k}[n] = B\log_2\left(1 + \frac{p_k[n]|h_{m,k}[n]|^2}{\sum_{j=1,j\neq k}^{K} p_j[n]|h_{m,j}[n]|^2 + \sigma^2}\right), \quad (6)$$

where B and σ^2 denote the channel bandwidth and noise power, respectively. The term $\sum_{j=1,j\neq k}^{K} p_k[n]|h_{m,j}[n]|^2$ is the interference from all other IoT devices at time slot n. Therefore, the amount of data that could be offloaded from device s_k to UAV u_m at time slot n is calculated as $R_{m,k}[n]a_{m,k}[n]\delta_t$.

C. Computation Model

We assume that task computation adopts the dynamic frequency and voltage technique [43]. In particular, at time slot n, let the CPU frequency of IoT device s_k be $f_k[n]$, then $0 \leq f_k[n] \leq f_k^{\max}$, where $f_k^{\max}$ represents for the maximum allowable CPU frequency of s_k. The amount of computation bits executed by device s_k at time slot n is calculated as $\frac{f_k[n]\delta_t}{C_k}$. Similar to [36], the power consumption of IoT device s_k for local computing at time slot n is modelled as $\kappa f_k^3[n]$, where κ denotes the effective switched capacitance of CPU, and its value is relevant to the chip architecture. Note that the IoT device's energy consumption includes both computation and communication related energy, then the energy consumption of device s_k at time slot n can be calculated as $\kappa f_k^3[n]\delta_t + p_k[n]\delta_t$. It follows that

$$\sum_{n=1}^{N}(\kappa f_k^3[n]\delta_t + p_k[n]\delta_t) \leq E_k, \forall k, \tag{7}$$

where E_k denotes the energy budget of IoT device s_k. Denoting $f_{m,k}^U[n]$ as the CPU frequency allocated to the task of device s_k by UAV u_m at time slot n, then we have

$$\sum_{k=1}^{K} f_{m,k}^U[n] \leq f_m^{U,\max}, \forall m, \tag{8}$$

where $f_m^{U,\max}$ denotes the maximum CPU frequency of UAV u_k. Similarly, the amount of computation bits executed by UAV u_m for the device s_k task at time slot n is given by $\frac{f_{m,k}^U[n]\delta_t}{C_k}$.

It is worth noting that the amount of computation bits executed at each UAV's MEC server for a given device should not exceed the total amount of bits that have already been offloaded from that device. By assuming that processing delay for offloaded task (e.g., decoding and computation preparation) is one time slot, the following information-causality constraints for computation offloading holds, i.e.,

$$\sum_{n=1}^{t-1} R_{m,k}[n]a_{m,k}[n]\delta_t \geq \sum_{n=2}^{t}\frac{f_{m,k}^U[n]\delta_t}{C_k}, \quad \forall m, k, t = 2, \ldots, N. \tag{9}$$

The left-hand-side (LHS) and right-hand-side (RHS) of (9) denote the amount of task-input data at UAV u_m that have been offloaded from s_k before time slot t and those that have been executed at time slot t, respectively. It is observed in (9) that devices should not offload tasks to UAVs at the last time slot N, since offloaded tasks are no longer executed after time slot N. On the other hand, edge computing at UAVs should not be activated in the first time slot since no task has been offloaded yet. In general, the one time slot processing delay can be neglected when N is large enough [9], [44].

We assume that the size of computation results is sufficiently small (e.g., as in video analysis or face recognition), and thus the downloading time for computation results is negligible as in [8], [45]. Note that the computation tasks can be done by local computing at IoT devices or remote computing at UAVs, and thus the total number of computation bits executed for task S_k within time slot n can be written as $\frac{f_k[n]\delta_t}{C_k} + \sum_{m=1}^{M}\frac{f_{m,k}^U[n]\delta_t}{C_k}$. Recall that I_k and D_k are the data size and deadline for computation task S_k, respectively. If $\sum_{n=1}^{\lfloor D_k/\delta_t \rfloor}\left(\frac{f_k[n]\delta_t}{C_k} + \sum_{m=1}^{M}\frac{f_{m,k}^U[n]\delta_t}{C_k}\right) \geq I_k$, then the computation task S_k for IoT device s_k can be finished before its deadline; otherwise, the deadline for S_k is violated.

D. Problem Formulation

Define $\theta_k \in \{0,1\}$ as a service indicator variable to illustrate whether computation task S_k can be finished before its deadline or not. If task S_k can be finished on time, then $\theta_k = 1$; otherwise, $\theta_k = 0$. Thus, the total number of served IoT devices can be expressed as $\sum_{k=1}^{K}\theta_k$. In this paper, we aim to maximize the total number of served IoT devices in the multi-UAV-enabled MEC system with given time horizon T, while ensuring that the time sensitive computation tasks for served IoT devices can be finished before their deadlines and the energy budget constraints of the devices are also satisfied, via jointly optimizing service indicator $\Theta \triangleq \{\theta_k\}$, computation offloading and association $\mathcal{A} \triangleq \{a_{m,k}\}$, transmit power allocation $\mathcal{P} \triangleq \{p_k[n]\}$, computation resource allocation $\mathcal{F} \triangleq \{f_k[n], f_{m,k}^U[n]\}$, as well as UAV trajectory $\mathcal{U} \triangleq \{\mathbf{u}[n]\}$. The optimization problem can be expressed as follows:

$$\begin{aligned}
(\text{P1}): \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{U}} \quad & \sum_{k=1}^{K}\theta_k \\
\text{s.t.} \quad & (1)-(5), (7)-(9), \\
& \sum_{n=1}^{\lfloor D_k/\delta_t \rfloor}\left(\frac{f_k[n]\delta_t}{C_k} + \sum_{m=1}^{M}\frac{f_{m,k}^U[n]\delta_t}{C_k}\right) \geq \theta_k I_k, \forall k, && (10)\\
& \theta_k \in \{0,1\}, \forall k, && (11)\\
& 0 \leq a_{m,k}[n] \leq \theta_k, \forall m, k, n, && (12)\\
& 0 \leq p_k[n] \leq \theta_k P_k^{\max}, \forall k, n, && (13)\\
& 0 \leq f_k[n] \leq \theta_k f_k^{\max}, \forall k, n. && (14)
\end{aligned}$$

In (P1), the objective is to maximize the total number of served IoT devices. Constraints in (10) guarantee that the computation task for each served IoT device can be finished before its deadline. Constraints in (12)-(14) are imposed to prevent wasting communication and computation resources for tasks on devices that cannot be served within their deadlines, which are valid for both $\theta_k = 0$ and $\theta_k = 1$. For example, if $\theta_k = 0$, then $a_{m,k}$, $p_k[n]$, and $f_k[n]$ can all be set to 0. Due to the lack of instantaneous channel state information (CSI) between the IoT devices and UAVs, $h_{m,k}[n]$ is a random variable. Furthermore, Problem (P1) consists of binary constraints in (11) and non-convex constraints in (3) and (9). Therefore, this is a mixed-integer nonlinear programming (MINLP) problem, and is challenging to solve directly.

IV. Proposed Solution Method

In this section, we first reformulate the problem into a more tractable form using deductive penalty functions, and then decompose the reformulated problem into two sub-problems. We then develop an iterative algorithm to obtain a suboptimal solution through solving the two sub-problems jointly by employing alternating optimization and SCA techniques.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

A. Problem Reformulation

Since only CDI knowledge for the random channel $h_{m,k}[n]$ is available prior to the UAV's flight, we focus on the expected communication rate as in [38], [46], denoted by $\mathbb{E}[R_{m,k}[n]]$. It is shown in [47], [48] that $\mathbb{E}[R_{m,k}[n]]$ can be lower bounded as $\mathbb{E}[R_{m,k}[n]] \geq B\log_2\left(1+\frac{p_k[n]\mathbb{E}[|h_{m,k}[n]|^2]}{p_k[n]Var(h_{m,k}[n])+\sum_{j=1,j\neq k}^{K}p_j[n]\mathbb{E}[|h_{m,j}[n]|^2]+\sigma^2}\right) = B\log_2\left(1+\frac{p_k[n]\beta_{m,k}[n]}{p_k[n]\beta_{m,k}[n]/(K_c+1)+\sum_{j=1,j\neq k}^{K}p_j[n]\beta_{m,j}[n]+\sigma^2}\right) \triangleq \bar{R}_{m,k}[n]$. We adopt $\bar{R}_{m,k}[n]$ in resource allocation and trajectory design as in [41], [46], such that the worst-case achievable rate can be achieved. As a result, Problem (P1) can be rewritten as

$$\text{(P2)}: \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{U}} \quad \sum_{k=1}^{K}\theta_k$$
$$\text{s.t.} \quad (1)-(5),(7),(8),(10)-(14),$$
$$\sum_{n=1}^{t-1}\bar{R}_{m,k}[n]a_{m,k}[n] \geq \sum_{n=2}^{t}\frac{f_{m,k}^{U}[n]}{C_k}, \forall m,k,t=2,\ldots,N. \quad (15)$$

To tackle the binary constraints (11), we redefine (11) with an equivalent continuous representation while enforcing variables θ_k to take binary values. Specifically, we can express (11) as intersection of the following regions,

$$0 \leq \theta_k \leq 1, \forall k, \quad (16)$$
$$\sum_{k=1}^{K}(\theta_k-\theta_k^2) \leq 0. \quad (17)$$

It can be verified that any feasible point in (11) satisfies both constraints in (16) and (17) and vice versa. Thus, we can rewrite Problem (P2) as

$$\text{(P3)}: \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{U}} \quad \sum_{k=1}^{K}\theta_k$$
$$\text{s.t.} \quad (1)-(5),(7),(8),(10),(12)-(17).$$

Note that we need binary solutions for variables in Θ. To ensure this feasible condition, we add a cost function to the objective function to penalize the objective for non-binary values of Θ. As such, the problem can be rewritten as

$$\text{(P4)}: \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{U}} \quad \sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k-\theta_k^2)$$
$$\text{s.t.} \quad (1)-(5),(7),(8),(10),(12)-(16),$$

where $\lambda \gg 1$ is the penalty factor which defines the penalty when variables θ_k are set to values other than 0 or 1.

Proposition 1. *When λ is sufficiently large, Problem (P3) is equivalent to Problem (P4) .*

Please refer to Appendix A for the proof of this proposition.

It can be seen that Problem (P4) remains non-convex due to constraints (3) and (15). In the following, we propose an effective method to obtain a sub-optimal solution to Problem (P4) by solving its two sub-problems iteratively. Specifically, in the first sub-problem, the computation offloading, service indicator, and resource allocation (i.e., $\mathcal{A}$, Θ, $\mathcal{P}$, $\mathcal{F}$) are optimized with given UAV trajectory $\mathcal{U}$. For any feasible ($\mathcal{A}$, Θ, $\mathcal{P}$, $\mathcal{F}$), the second sub-problem is only the UAV trajectory optimization problem. These sub-problems are solved in the following subsections. Afterwards, we will provide the overall algorithm, and examine its convergence and complexity.

B. Computation Offloading, Service Indicator Optimization, and Resource Allocation

We first consider computation offloading and service indicator optimization as well as resource allocation with given UAV flight $\mathcal{U}$. The problem is written as

$$\text{(P5)}: \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F}} \quad \sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k-\theta_k^2)$$
$$\text{s.t.} \quad (4),(5),(7),(8),(10),(12)-(16).$$

With given UAV trajectory $\mathcal{U}$, Problem (P5) is still a non-convex optimization problem. By employing the slack variables $\mathcal{X} \triangleq \{x_{m,k}[n]\}$, Problem (P5) can be reformulated as

$$\text{(P6)}: \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{X}} \quad \sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k-\theta_k^2)$$
$$\text{s.t.} \quad (4),(5),(7),(8),(10),(12)-(14),(16),$$
$$\sum_{n=1}^{t-1}x_{m,k}[n]a_{m,k}[n] \geq \sum_{n=2}^{t}\frac{f_{m,k}^{U}[n]}{C_k}, \forall m,k,t=2,\ldots,N, \quad (18)$$
$$\bar{R}_{m,k}[n] \geq x_{m,k}[n], \forall m,k,n. \quad (19)$$

It can be shown that in the optimal solution to (P6), all constraints in (19) are met with equality, i.e., $\bar{R}_{m,k}[n] = x_{m,k}[n], \forall m,k,n$. Otherwise, $x_{m,k}[n]$ can always be increased to satisfy the equality, while the objective value remains unchanged with all other constraints are met. Therefore, there always exists an optimal solution to Problem (P5) and $\bar{R}_{m,k}[n] = x_{m,k}[n], \forall m,k,n$. By substituting $x_{m,k}[n]$ with $\bar{R}_{m,k}[n]$ in (18) and (19), Problem (P6) is equivalent to Problem (P5).

Problem (P6) is still non-convex since the objective function is non-concave and both constraints in (18) and (19) are non-convex. We find the SCA technique can be utilized to effectively solve Problem (P6): the non-convex constraints can be replaced with convex approximations, and a sequence of improved solutions for the original problem can be derived [33]. In particular, to tackle the non-convex term in (18), we rewrite the LHS of (18) as the difference of two convex functions, i.e., $x_{m,k}[n]a_{m,k}[n] = \frac{1}{2}(x_{m,k}[n]+a_{m,k}[n])^2 - \frac{1}{2}(x_{m,k}[n]^2 + a_{m,k}[n]^2)$. By applying first-order Taylor expansion on the convex term $(x_{m,k}[n]+a_{m,k}[n])^2$, the following inequalities hold at the rth iteration, i.e.,

$$x_{m,k}[n]a_{m,k}[n] \geq (x_{m,k}[n]^r + a_{m,k}[n]^r)(x_{m,k}[n]+a_{m,k}[n]) - \frac{1}{2}(x_{m,k}[n]^r + a_{m,k}[n]^r)^2 - \frac{1}{2}(x_{m,k}[n]^2 + a_{m,k}[n]^2) \triangleq y_{m,k}[n]. \quad (20)$$

To tackle the non-convex term in (19), we first rewrite $\bar{R}_{m,k}[n]$ as the difference of two concave

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

functions, i.e., $\bar{R}_{m,k}[n] = \check{R}_{m,k}[n] - \hat{R}_{m,k}[n]$, where $\check{R}_{m,k}[n] = B\log_2\left(\gamma_{m,k}[n] + \sum_{j=1}^{K} p_j[n]\beta_{m,j}[n]\right)$, $\hat{R}_{m,k}[n] = B\log_2\left(\gamma_{m,k}[n] + \sum_{j=1,j\neq k}^{K} p_j[n]\beta_{m,j}[n]\right)$, and $\gamma_{m,k}[n] \triangleq \frac{p_k[n]\beta_{m,k}[n]}{K_c+1} + \sigma^2$. Note that $\beta_{m,k}[n]$ is fixed due to the given UAV trajectory. By applying first-order Taylor expansion on the concave term $\hat{R}_{m,k}[n]$, the following inequalities hold at the rth iteration, i.e.,

$$\hat{R}_{m,k}[n] \leq \hat{\Phi}^r_{m,k}[n] + \hat{\Psi}^r_{m,k,k}(p_k[n] - p^r_k[n]) + \sum_{j=1,j\neq k}^{K} \hat{\Psi}^r_{m,k,j}(p_j[n] - p^r_j[n]) \triangleq \hat{R}^{ub}_{m,k}[n], \quad (21)$$

where $\hat{\Phi}^r_{m,k} = B\log_2(\hat{\Omega}^r_{m,k}[n])$, $\hat{\Omega}^r_{m,k}[n] = \frac{p^r_k[n]\beta_{m,k}[n]}{K_c+1} + \sum_{j=1,j\neq k}^{K} p^r_j[n]\beta_{m,j}[n] + \sigma^2$, $\hat{\Psi}^r_{m,k,k}[n] = \frac{B\beta_{m,k}[n]\log_2(e)}{(K_c+1)\hat{\Omega}^r_{m,k}[n]}$, and $\hat{\Psi}^r_{m,k,j}[n] = \frac{B\beta_{m,j}[n]\log_2(e)}{\hat{\Omega}^r_{m,k}[n]}$. Due to (21), we have the following lower bound for $\bar{R}_{m,k}[n]$, i.e.,

$$\bar{R}_{m,k}[n] \geq \check{R}_{m,k}[n] - \hat{R}^{ub}_{m,k}[n], \forall m,k,n. \quad (22)$$

For the objective function, by applying first-order Taylor expansion on the convex term θ_k^2 at the rth iteration, we have $\theta_k^2 \geq (\theta_k^r)^2 + 2\theta_k^r(\theta_k - \theta_k^r), \forall k$. Therefore, we have the following lower bound for the objective function, i.e.,

$$\sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k - \theta_k^2) \geq \sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k - (\theta_k^r)^2 - 2\theta_k^r(\theta_k - \theta_k^r)). \quad (23)$$

By replacing term $x_{m,k}[n]a_{m,k}[n]$, $\bar{R}_{m,k}[n]$ and the objective function with lower bounds expressions derived in (20), (22), and (23), Problem (P6) can be approximated as

$$\begin{aligned}
(\text{P7}): \max_{\Theta,\mathcal{A},\mathcal{P},\mathcal{F},\mathcal{X}} \ & \sum_{k=1}^{K}\theta_k - \lambda\sum_{k=1}^{K}(\theta_k - (\theta_k^r)^2 - 2\theta_k^r(\theta_k - \theta_k^r)) \\
\text{s.t.} \ & (4),(5),(7),(8),(10),(12)-(14),(16), \\
& \sum_{n=1}^{t-1} y_{m,k}[n] \geq \sum_{n=2}^{t}\frac{f^U_{m,k}[n]}{C_k}, \forall m,k,t=2,\ldots,N, \quad (24) \\
& \check{R}_{m,k}[n] - \hat{R}^{ub}_{m,k}[n] \geq x_{m,k}[n], \forall m,k,n. \quad (25)
\end{aligned}$$

In Problem (P7), the objective function is linear and all the constraints are convex constrains. Therefore, Problem (P7) is a standard convex optimization problem. As a result, existing solvers such as CVX [49] and many standard convex optimization techniques can be leveraged to solve Problem (P7). Note that the feasible set of Problem (P7) is always a subset of Problem (P6) due to the lower bound approximations. Therefore, the objective value of Problem (P7) serves as a lower bound of that in Problem (P6). The accuracy of such approximation will be evaluated in Section V.

C. UAV Trajectory Optimization

In this subsection, we consider the sub-problem of UAV trajectory optimization problem with given ($\mathcal{A}$, Θ, $\mathcal{P}$, $\mathcal{F}$). In this case, the objective function is fixed. Note that after solving Problem (P5), the constraints in (15) are satisfied with equality. Since otherwise, we can always increase $f^U_{m,k}[n]$ until the equality holds, and the objective value remains unchanged with all the other constraints are met. In the UAV trajectory optimization problem, we maximize the minimum ratio between the amount of offloaded data and those that have been computed at the UAVs, which is written as problem (P8). As a result, we can relax the constraints in (15) such that more optimization space can be preserved for further maximizing the number of served devices in Problem (P5). Detailed proof of the convergence of the alternating optimization technique will be given in Section IV-D.

$$\begin{aligned}
(\text{P8}): \max_{\rho,\mathcal{U}} \ & \rho \\
\text{s.t.} \ & (1)-(3), \\
& \frac{\sum_{n=1}^{t-1}\bar{R}_{m,k}[n]a_{m,k}[n]}{\sum_{n=2}^{t}\frac{f^U_{m,k}[n]}{C_k}} \geq \rho, \forall m,k,t=2,\ldots,N. \quad (26)
\end{aligned}$$

Problem (P8) is non-convex since both constraints (3) and (26) are non-convex. To tackle the non-convex term in $\bar{R}_{m,k}[n]$, $\bar{R}_{m,k}[n]$ is rewritten as $\bar{R}_{m,k}[n] = \check{R}_{m,k}[n] - \hat{R}_{m,k}[n]$, where $\check{R}_{m,k}[n] = B\log_2\left(\gamma_{m,k}[n] + \sum_{j=1}^{K}\frac{p_j[n]\beta_0}{(H^2+\|\mathbf{u}_m[n]-\mathbf{g}_j\|^2)^{\alpha/2}}\right)$, $\hat{R}_{m,k}[n] = B\log_2\left(\gamma_{m,k}[n] + \sum_{j=1,j\neq k}^{K}\frac{p_j[n]\beta_0}{(H^2+\|\mathbf{u}_m[n]-\mathbf{g}_j\|^2)^{\alpha/2}}\right)$, and $\gamma_{m,k}[n] \triangleq \frac{p_k[n]\beta_0/(K_c+1)}{(H^2+\|\mathbf{u}_m[n]-\mathbf{g}_k\|^2)^{\alpha/2}} + \sigma^2$. By employing slack variables $\mathcal{W} \triangleq \{\omega_m[n]\}$ and letting $\hat{R}^{\omega}_{m,k}[n] = B\log_2\left(\frac{p_k[n]\beta_0/(K_c+1)}{(H^2+\omega_{m,k}[n])^{\alpha/2}} + \sum_{j=1,j\neq k}^{K}\frac{p_j[n]\beta_0}{(H^2+\omega_{m,j}[n])^{\alpha/2}} + \sigma^2\right)$, Problem (P8) can be reformulated as follows:

$$\begin{aligned}
(\text{P9}): \max_{\rho,\mathcal{U},\mathcal{W}} \ & \rho \\
\text{s.t.} \ & (1),(2), \\
& \sum_{n=1}^{t-1}\left((\check{R}_{m,k}[n] - \hat{R}^{\omega}_{m,k}[n])a_{m,k}[n]\right)/\sum_{n=2}^{t}\frac{f^U_{m,k}[n]}{C_k} \geq \rho, \\
& \qquad \forall m,k,t=2,\ldots,N, \quad (27) \\
& \|\mathbf{u}_m[n] - \mathbf{g}_j\|^2 \geq \omega_{m,j}[n], \forall m,j,n, \quad (28) \\
& \|\mathbf{u}_m[n] - \mathbf{u}_i[n]\|^2 \geq d^2_{\min}, \forall m,i>m,n=2,\ldots,N-1. \quad (29)
\end{aligned}$$

It is not hard to see that equality holds in all constraints (28) in the optimal solution to Problem (P9). Otherwise, $\omega_{m,j}[n]$ can always be increased to achieve equality. Then the objective value remains unchanged with all other constraints are met. Furthermore, the constraints in (3) and (29) are equivalent. Thus Problem (P9) is equivalent to Problem (P8).

Due to the non-convex constraints (27)–(29), Problem (P9) is a non-convex optimization problem. To deal with such challenges, we use SCA technique to derive convex approximation. In particular, for constraints in (28) and (29), by applying first-order Taylor expansion on convex terms $\|\mathbf{u}_m[n] - \mathbf{u}_i[n]\|^2$ and $\|\mathbf{u}_m[n] - \mathbf{g}_j\|^2$, the following inequalities hold, respectively, at the rth iteration:

$$\|\mathbf{u}_m[n] - \mathbf{g}_j\|^2 \geq \|\mathbf{u}^r_m[n] - \mathbf{g}_j\|^2 + 2(\mathbf{u}^r_m[n] - \mathbf{g}_j)^T(\mathbf{u}_m[n] - \mathbf{u}^r_m[n]) \triangleq \check{\mu}_m[n], \forall m,j,n, \quad (30)$$

$$\|\mathbf{u}_m[n]-\mathbf{u}_i[n]\|^2 \geq -\|\mathbf{u}_m^r[n]-\mathbf{u}_i^r[n]\|^2 +2(\mathbf{u}_m^r[n]-\mathbf{u}_i^r[n])^T(\mathbf{u}_m[n]-\mathbf{u}_i[n]) \triangleq \hat{\mu}_{m,i}[n], \forall m,i,n, \quad (31)$$

In (27), it can be verified that $\check{R}_{m,k}[n]$ is a convex function with respect to term $\|\mathbf{u}_m[n]-\mathbf{g}_k\|^2$. Thus, the lower bound for $\check{R}_{m,k}[n]$ can be obtained at the given point $\mathbf{u}_m^r[n]$ as in [33], [34], i.e.,

$$\check{R}_{m,k}[n] \geq \check{\Phi}_{m,k}^r[n] - \sum_{j=1}^{K} \check{\Psi}_{m,k,j}^r(\|\mathbf{u}_m[n]-\mathbf{g}_k\|^2 - \|\mathbf{u}_m^r[n]-\mathbf{g}_k\|^2) - \frac{\check{\Psi}_{m,k,k}^r}{K_c+1}(\|\mathbf{u}_m[n]-\mathbf{g}_k\|^2 - \|\mathbf{u}_m^r[n]-\mathbf{g}_k\|^2) \triangleq \check{R}_{m,k}^{lb}[n], \quad (32)$$

where $\check{\Phi}_{m,k}^r = B\log_2(\check{\Omega}_{m,k}^r[n])$, $\check{\Omega}_{m,k}^r[n] = \sigma^2 + \frac{p_k[n]\beta_0/(K_c+1)}{(H^2+\|\mathbf{u}_m^r[n]-\mathbf{g}_k\|^2)^{\alpha/2}} + \sum_{j=1}^{K}\frac{p_j[n]\beta_0}{(H^2+\|\mathbf{u}_m^r[n]-\mathbf{g}_j\|^2)^{\alpha/2}}$, $\check{\Psi}_{m,k,j}^r[n] = \frac{Bp_j[n]\beta_0(\alpha/2)\log_2(e)}{\check{\Omega}_{m,k}^r[n](H^2+\|\mathbf{u}_m^r[n]-\mathbf{g}_k\|^2)^{(1+\alpha/2)}}$.

By substituting the lower bounds derived in (30) and (31) into the LHSs of (28) and (29), respectively, and substituting $\check{R}_{m,k}^{lb}[n]$ into the LHS of (27), Problem (P9) can be approximated as:

$$\begin{aligned}
(\text{P10}): \quad & \max_{\rho,\mathcal{U},\mathcal{W}} \rho \\
\text{s.t.} \quad & (1),(2), \\
& \sum_{n=1}^{t-1}\Big((\check{R}_{m,k}^{lb}-\hat{R}_{m,k}^{\omega}[n])a_{m,k}[n]\Big)/\sum_{n=2}^{t}\frac{f_{m,k}^U[n]}{C_k} \geq \rho, \\
& \qquad \forall m,k,t=2,\ldots,N, \quad (33)\\
& \check{\mu}_m[n] \geq \omega_{m,j}[n], \forall m,j,n, \quad (34)\\
& \hat{\mu}_{m,i}[n] \geq d_{\min}^2, \forall m,i>m,n=2,\ldots,N-1. \quad (35)
\end{aligned}$$

Problem (P10) is now a convex optimization problem, and many existing solvers, e.g., CVX, can be used to solve it.

D. Overall Iterative Algorithm, Convergence, and Complexity

Based on the results obtained in the previous two subsections, we now propose an iterative algorithm to solve Problem (P4) for suboptimal solutions, by applying SCA and alternating optimization, as shown in Fig. 2. The details are summarized in Algorithm 1. The computation offloading and service indicator as well as resource allocation ($\mathcal{A}$, Θ, $\mathcal{P}$, $\mathcal{F}$), and UAV trajectory $\mathcal{U}$ are optimized alternately in each iteration, by solving Problems (P7) and (P10) correspondingly.

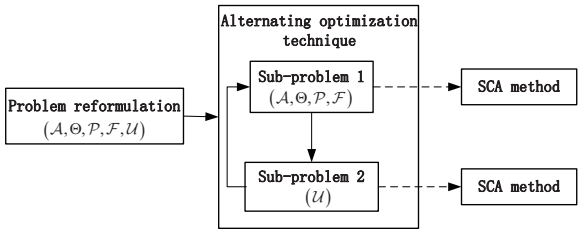


Fig. 2. The procedure of Algorithm 1.

Theorem 1. *The proposed Algorithm 1 is convergent.*

Algorithm 1 Iterative Computation Offloading, Service Indicator, Resource Allocation, and Trajectory Optimization Algorithm for Problem (P4)

1: Initialize $\mathcal{U}^0$; let $r \leftarrow 0$.
2: **repeat**
3: With given $\mathcal{U}^r$, obtain the optimal solution $\mathcal{A}^{r+1}$, $\mathcal{P}^{r+1}$, $\mathcal{F}^{r+1}$, and Θ^{r+1} by solving convex problem (P7);
4: With given $\mathcal{A}^{r+1}$, $\mathcal{P}^{r+1}$, $\mathcal{F}^{r+1}$, Θ^{r+1}, as well as $\mathcal{U}^r$, obtain the optimal solution $\mathcal{U}^{r+1}$ by solving convex problem (P10);
5: $r \leftarrow r+1$;
6: **until** The objective value of (P4) converges

Proof. Let $\mathcal{R}^r \triangleq \{\mathcal{A}^r,\mathcal{P}^r,\mathcal{F}^r,\Theta^r\}$, and denote $\Lambda(\mathcal{R}^r,\mathcal{U}^r)$ as the objective value of problem (P4) in the rth iteration. Let $\hat{\Lambda}$ be the objective value of Problem (P7), then we have

$$\Lambda(\mathcal{R}^r,\mathcal{U}^r) \overset{(a)}{=} \hat{\Lambda}(\mathcal{R}^r,\mathcal{U}^r) \overset{(b)}{\leq} \hat{\Lambda}(\mathcal{R}^{r+1},\mathcal{U}^r) \overset{(c)}{\leq} \Lambda(\mathcal{R}^{r+1},\mathcal{U}^r),$$

where (a) holds due to the tightness of first-order Taylor expansions at given local points in Problem (P7); (b) holds since Problem (P7) is optimally solved. Since the objective value of Problem (P7) is a lower bound of that of Problem (P4), the inequality (c) holds. Similarly, for the SCA-based UAV trajectory optimization problem, we have

$$\rho(\mathcal{R}^{r+1},\mathcal{U}^r) \leq \rho(\mathcal{R}^{r+1},\mathcal{U}^{r+1}), \quad (36)$$

where $\rho(\mathcal{R}^{r+1},\mathcal{U}^r)$ denotes the objective value of Problem (P8) with given $(\mathcal{R}^{r+1},\mathcal{U}^r)$. After Step 3 in Algorithm 1, denote $\Gamma_{m,k,t}(\mathcal{R}^r,\mathcal{U}^r) = \frac{\sum_{n=1}^{t-1}\bar{R}_{m,k}[n]a_{m,k}[n]}{\sum_{n=2}^{t}\frac{f_{m,k}^U[n]}{C_k}}$ for $(\mathcal{R}^r,\mathcal{U}^r)$. It follows that

$$\begin{aligned}
1 &\overset{(d)}{\leq} \min_{m,k,t}\Gamma_{m,k,t}(\mathcal{R}^{r+1},\mathcal{U}^r) \overset{(e)}{=} \rho(\mathcal{R}^{r+1},\mathcal{U}^r) \\
&\overset{(f)}{\leq} \rho(\mathcal{R}^{r+1},\mathcal{U}^{r+1}) \overset{(e)}{=} \min_{m,k,t}\Gamma_{m,k,t}(\mathcal{R}^{r+1},\mathcal{U}^{r+1}), \quad (37)
\end{aligned}$$

where (d) holds since $\mathcal{R}^{r+1}$ is a feasible solution of Problem (P5) with given $\mathcal{U}^r$, and constraints in (15) should be satisfied; (e) and (f) hold due to the definition of Problem (P8) and (36), respectively. Since $\min_{m,k,t}\Gamma_{m,k,t}(\mathcal{R}^{r+1},\mathcal{U}^{r+1}) \geq 1$, $(\mathcal{R}^{r+1},\mathcal{U}^{r+1})$ is a feasible solution of Problem (P4), and $\Lambda(\mathcal{R}^{r+1},\mathcal{U}^r) = \Lambda(\mathcal{R}^{r+1},\mathcal{U}^{r+1})$ since the objective value of Problem (P4) only depends on $\mathcal{R}^{r+1}$. As a result, $\Lambda(\mathcal{R}^r,\mathcal{U}^r) \leq \Lambda(\mathcal{R}^{r+1},\mathcal{U}^{r+1})$, then the objective value of Problem (P4) is non-decreasing over iterations and it is upper bounded by a finite value. Therefore, the proposed Algorithm 1 is convergent. Furthermore, it is worthwhile to note the first-order Taylor expansions have identical gradients as original functions, then Algorithm 1 with SCA technique converges to a KKT solution [33]. □

Theorem 2. *The computational complexity of Algorithm 1 is $O((MKN)^{3.5}\log^2(1/\epsilon))$, where ϵ corresponds to the given solution accuracy.*

Proof. Note that the overall iterative algorithm requires solving Problems (P7) and (P10) alternatively, which are both

standard convex optimization problems and can be solved by using the convex solver CVX with interior-point method, whose complexity are given by $O((MKN)^{3.5}\log(1/\epsilon))$ with given solution accuracy ϵ [50], [51], since the number of optimization variables depend on M, K, and N. Therefore, accounting for the number of iterations with the order of $O(\log(1/\epsilon))$, the computational complexity of Algorithm 1 is given as $O((MKN)^{3.5}\log^2(1/\epsilon))$. □

E. Trajectory Initialization Scheme

Note that the system performance and the converged solution is relevant to the initialization scheme in general. In this section, an efficient initialization scheme is proposed for $\mathcal{Q}^0$. Note that some straightforward relationships between the total number of served IoT devices and UAV trajectory can be obtained as follows:

- Observation 1: The UAV should move close to IoT devices to provide better computation offloading service with better communication channel.
- Observation 2: Multiple UAVs may serve different groups of IoT devices simultaneously.
- Observation 3: Each UAV may fly with the maximum speed and tries to serve as many devices as possible within given time horizon.
- Observation 4: The UAV should fly to and complete computation offloading for IoT devices before given deadlines.

As a result, the initial trajectory should consider the above four aspects. Note that multiple traveling salesman problem with time windows (m-TSPTW) [52] is to find a set of routes for a group of vehicles such that a set of locations are served, each one within a specified time window. Thus, it is reasonable to set the UAV initial trajectory $\mathcal{Q}^0$ based on the m-TSPTW path to minimize the total traveling distance of multiple UAVs to visit each IoT device s_k within time window $[0, D_k]$ with the maximum speed, which starts from $\mathbf{q}_I$ and also ends with $\mathbf{q}_I$. The m-TSPTW is a generalization of the classical TSP which is a well-known NP-hard problem, and thus m-TSPTW is also a NP-hard problem and more difficult to solve [52]. Note that the m-TSPTW belongs to the class of time-constrained vehicle routing problems [53], which have been well studied in the past few decades, and various efficient exact and heuristic algorithms have been proposed to find optimal solutions for small scale scenarios and high-quality approximate solutions for large scale scenarios, respectively [53], [54].

After solving the m-TSPTW over IoT devices set $\mathcal{K}$, we obtain the total flying time T_I and the m-TSPTW path which are denoted by device groups $\{G_m, 1 \le m \le M\}$ and permutation orders $\{\Gamma_m, 1 \le m \le M\}$. In particular, G_m represents for IoT devices visited by UAV u_m, where $G_{m_1} \bigcap G_{m_2} = \emptyset, 1 \le m_1 < m_2 \le M$. Γ_m represents for the visiting order in device group G_m for UAV u_m, and $\Gamma_m = (\pi_1, \ldots, \pi_{|G_m|})$, with $1 \le \pi_i \le \mathcal{K}, 1 \le i \le |G_m|$ representing the index of the ith node s_i in $\mathcal{K}$ to be visited by UAV u_m. Note that the given time horizon T may be different from T_I for m-TSPTW solution. As a result, we propose a m-TSPTW based initial trajectory applied in two-cases:

- Case 1: $T \ge T_I$. In this case, each UAV u_m is able to reach the top of IoT devices before deadlines in G_m within time horizon T_I, and then the remaining $T - T_I$ can be utilized for the UAV u_m to hover above the final location $\mathbf{u}_I$.
- Case 2: $T < T_I$. In this case, each UAV u_m is not able to reach the top of devices in G_m within time duration T. As a result, with given visiting order Γ_m, we should determine the waypoint for UAV u_m to visit each device s_{π_i}, denoted as $\mathbf{u}_{m,\pi_i}, 1 \le i \le |G_m|$. Similar to [55], we construct a disk-shaped region for devices in G_m with radius r, and UAV u_m only needs to to reach the disk region of devices in G_m within time duration T. Note that with given r and visiting order Γ_m, the minimum flying distance of UAV u_m can be obtained by solving the following problem (P11), which is a standard convex optimization problem and can be efficient solved.

$$\begin{aligned} \text{(P11)}: \quad & \min_{\{\mathbf{u}_{m,\pi_i}\}} \sum_{i=1}^{|G_m|+1} \left\|\mathbf{u}_{m,\pi_i} - \mathbf{u}_{m,\pi_{i-1}}\right\| \\ \text{s.t.} \quad & \left\|\mathbf{u}_{m,\pi_i} - \mathbf{w}_{\pi_i}\right\| \le r, 1 \le i \le |\mathcal{D}|, \end{aligned} \quad (38)$$

where $\mathbf{w}_{\pi_i}$ is the horizonal location of device s_{π_i}, and we let $\mathbf{u}_{m,\pi_0} = \mathbf{u}_I$ and $\mathbf{u}_{m,\pi_{|G_m|+1}} = \mathbf{u}_I$. It is worthwhile to note that the objective value of (P11) is non-increasing with r, since the optimal solution to (P11) with r is always a feasible solution to (P11) when r increases to $r + \epsilon, \epsilon > 0$. Thus, the waypoints $\{\mathbf{u}_{m,\pi_i}\}$ of UAV u_m are obtained by solving (P11) with fixed r, and then efficient bisection search can be employed to find the optimal radius r with given time horizon T.

Using the results obtained above, the details of UAV trajectory initialization algorithm is summarized in the following Algorithm 2.

In Algorithm 2, we obtain the visiting order with m-TSPTW method at step 1, where various efficient algorithms exist for m-TSPTW with polynomial complexity, e.g., $O(K^2)$ [54]. From step 6 to step 16, the bisection search approach is employed to find the optimal radius r with given T, with complexity $O(\log((r_2 - r_1)/\epsilon))$, which is unrelated to the number of IoT devices K and r_2 can be set as the maximum radius of the considered region. At step 9 during each iteration, standard convex optimization problem (P11) can be solved by using interior-point method based CVX solver, whose complexity is given by $O(K^{3.5}\log(1/\epsilon))$ with a solution accuracy of ϵ [50], [51]. To sum up, the computation complexity of Algorithm 2 can be roughly given by $O(K^{3.5}\log^2(1/\epsilon))$.

V. Simulation Study

In this section, we provide the system setup and evaluate the performance of the proposed algorithms via simulations with detailed performance analysis. We also investigate the impact of important parameters on the performance of the proposed algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

Algorithm 2 UAV trajectory initialization algorithm

Given the locations of $\mathcal{K}$, obtain devices set G_m and visiting order Γ_m for each UAV u_m as well as T_I with m-TSPTW method [54].
if $T \geq T_I$ **then**
  Construct the UAV trajectory $\mathcal{Q}^0$ by letting each UAV u_m fly to the top devices in G_m with maximum speed $V_{\max}$ and visiting order Γ_m, and hover over final location $\mathbf{q}_I$ for duration $T - T_I$.
else
  Let $r_1 = 0$, r_2 be sufficiently large, and let tolerance $\epsilon > 0$.
  repeat
    Update $r = \frac{r_1 + r_2}{2}$.
    for $m = 1$ to M **do**
      Given r, solve convex optimization problem (P11) for UAV u_m to obtain the objective value T_m and optimal solution $\{\mathbf{u}_{m,\pi_i}\}$.
    end for
    if $\forall m, T_m \leq T$ **then**
      Let $r_2 = r$, and update $\{\mathbf{q}^*_{m,\pi_i}\} = \{\mathbf{q}_{m,\pi_i}\}$.
    else
      Let $r_1 = r$.
    end if
  until $(r_2 - r_1) \leq \epsilon$.
  Construct the UAV trajectory $\mathcal{Q}^0$ by letting each UAV u_m fly to the waypoint $\mathbf{q}^*_{m,\pi_i}$ of device s_{π_i} in G_m with maximum speed $V_{\max}$ by following the visiting order Γ_m, $1 \leq i \leq |G_m|$.
end if

A. Simulation Setting

We consider a multi-UAV-enabled MEC system, where there are $M = 2$ UAVs and $K = 20$ IoT devices. The IoT devices are randomly and uniformly distributed in a square area of 2.0×2.0 km^2. The task deadlines of the IoT devices are randomly generated in $[0, D_{max}]$, where D_{max} denotes the maximum deadline. We assume that all devices have identical energy budget, task-input data size, maximum CPU frequency, maximum transmit power, and required CPU cycles per bit, i.e., $E_k = \bar{E}, I_k = \bar{I}, f_k^{\max} = \bar{f}^{\max}, P_k^{\max} = \bar{P}^{\max}, C_k = \bar{C}, \forall k$. All UAVs are assumed to have identical maximum CPU frequency, i.e., $f_m^{U,\max} = \bar{f}^{U,\max}$. Unless otherwise stated, the common parameters adopted in the simulations are set as follows: $H = 100$ m, $V_{\max} = 50$ m/s, $d_{\min} = 10$ m, $\bar{P}_{\max} = 0.1$ W, $\bar{E} = 2$ Joule, $D_{\max} = 200$ s, $T = 200$ s, $B = 1$ MHz, $\alpha = 2.2$, $\beta_0 = -60$ dB, $\sigma^2 = -110$ dBm, $K_c = 20$, $\delta_t = 1$ s, $\bar{f}^{U,\max} = 4$ GHz, $\bar{f}^{\max} = 0.5$ GHz, $\bar{C} = 10^3$ cycles/bit, $\kappa = 10^{-28}$, $\lambda = 10^5$.

B. Convergence of the Proposed Algorithm

In order to verify the convergence of the proposed iterative algorithm, the obtained objective value versus iteration number is given in Fig. 3. Recall that the objective value of Problem (P7) can be regarded as a lower bound of the objective value of Problem (P6). Thus, the objective value obtained by Algorithm 1 is a lower bound of the exact value, which is calculated from the objective function of Problem (P4). In Fig. 3, both the lower bound and the exact value become closer as iteration number increases, and the two curves meet when the objective value converges, which demonstrates that the lower bound for the number of served IoT devices with SCA technique is tight in Algorithm 1. Moreover, it can be seen that the proposed algorithm is quite efficient since it converges within 5 iterations with a prescribed accuracy $\epsilon = 10^{-4}$.

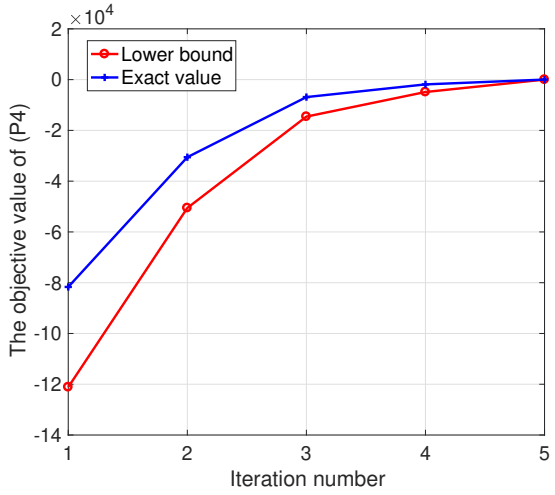


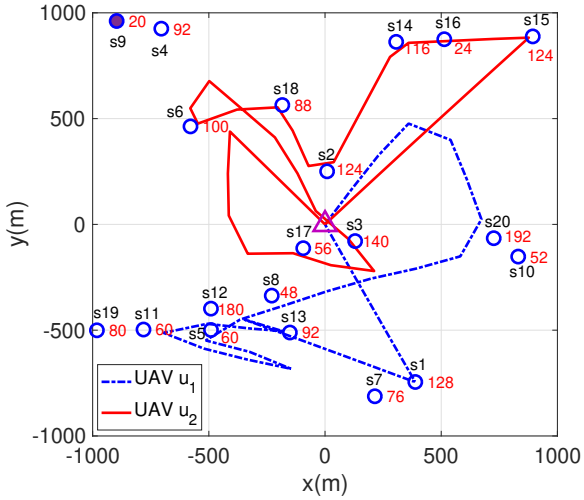
Fig. 3. Convergence of the proposed iterative Algorithm 1.

C. Optimized Trajectory, Service Indicator, Computation Offloading and Resource Allocation

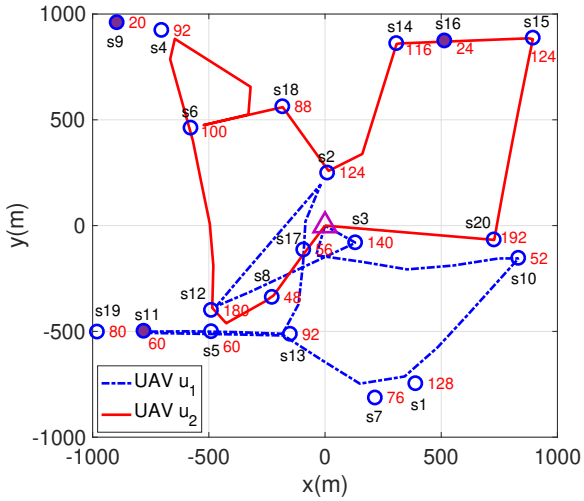
The optimized trajectories and service indicator with different task-input data size $\bar{I}$ are shown in Fig. 4. The hollow circles and solid circles denote the served IoT devices and non-served IoT devices, respectively. It is observed that with the increase of task-input data size $\bar{I}$, fewer IoT devices can be served on time since more computation resource is required, which is expected. For example, in Fig. 4(a) only s_9 cannot be served before its deadline, while in Fig. 4(b) s_9, s_{11} and s_{16} cannot be served on time. The UAVs will adjust their trajectories to move closer to the IoT devices to obtain more offloaded data for computation before deadlines, since better channel quality can be achieved when UAVs are close to devices. It can be seen in Fig. 4(b) that there exist some devices which are served by different UAVs, e.g., s_2, s_5, s_8, s_{12}, s_{17}, s_{20}, and UAVs u_1 and u_2 work cooperatively to serve these devices before their deadlines. This can also be verified in Fig. 5(a), which illustrates the computation offloading for all devices.

Fig. 5 illustrates the computation offloading and resource allocation results when $\bar{I} = 100$ Mbits. The corresponding results for other values of $\bar{I}$ follow similar trends and thus are omitted for brevity. Unless otherwise stated, we set $\bar{I} = 100$ Mbits in the following. From Fig. 5(a), it can be seen that different devices can offload tasks to different UAVs at the same time, and each device offloads computation task when the corresponding UAV flies close to it, as expected.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

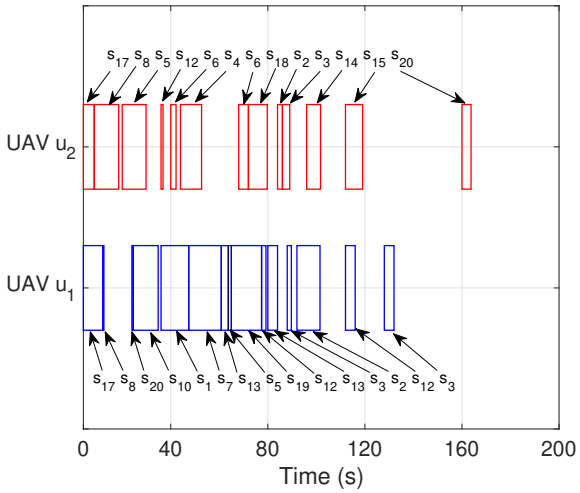


(a) $\bar{I} = 60$ Mbits

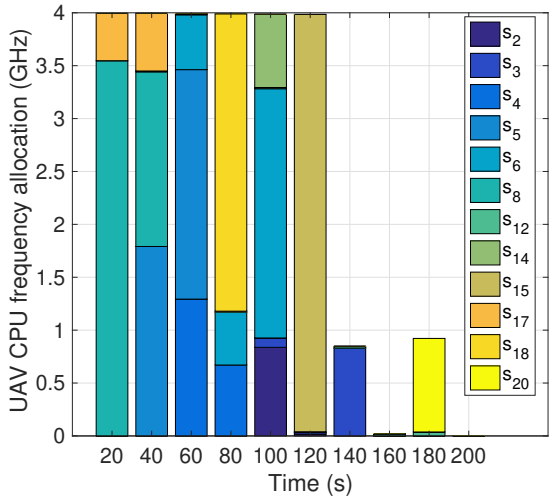


(b) $\bar{I} = 100$ Mbits

Fig. 4. Optimized trajectories with different task-input data size $\bar{I}$. The triangle denotes the initial and final UAV locations. The hollow circles and solid circles denote the served and non-served IoT devices, respectively. The red number to the right of an IoT device denotes its deadline.



(a) Computation offloading among all IoT devices.



(b) CPU frequency allocation among all IoT devices for UAV u_2.

Fig. 5. Computation offloading and resource allocation with $\bar{I} = 100$ Mbits.

After receiving the data offloaded from device, computation resource (i.e., CPU frequency) of the UAV can be allocated to the corresponding task, which is required by computation offloading constraints in (9) and verified in Fig. 5(b). The CPU frequency allocation of UAV u_2 over time for all IoT devices is shown in Fig. 5(b), the result for UAV u_1 follows similar trends and is thus omitted for brevity. Since the UAV's computation resources are shared by different devices, then it can be seen from Fig. 5(b) that the UAV's CPU frequency is allocated to tasks that are offloaded from different devices simultaneously. From Fig. 5(b), it can be seen that the computation resource of UAV u_2 is no longer allocated to devices whose deadlines are expired, and non-served IoT devices (e.g., s_9, s_{11}, s_{16}) are not allocated with any resource, such that the computation resource can be efficiently utilized.

D. Performance Comparison

Before performance comparisons over benchmarks, let us first investigate the effect of various initialization schemes. Fig. 6 shows the percentage of served IoT devices versus the task-input data size $\bar{I}$ with different initialization schemes when $\bar{E} = 2$ Joule. Different from our proposed m-TSPTW based trajectory initialization scheme, m-TSP based trajectory initialization scheme constructs the UAV initial trajectory with m-TSP methods [57] applying on the devices set $\mathcal{K}$ without considering the effect of time windows or deadlines, while hovering based trajectory initialization constructs the UAV initial trajectory by letting the UAV hover over the centers of M non-overlap clusters of devices all the time which are determined by the K-means clustering strategy. It is observed that the performance gap among the three initialization scheme is not large, which demonstrates the effectiveness of our proposed iterative algorithm. It can be seen that m-TSP based initialization scheme performs worse than hovering based initialization scheme. The reason is that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

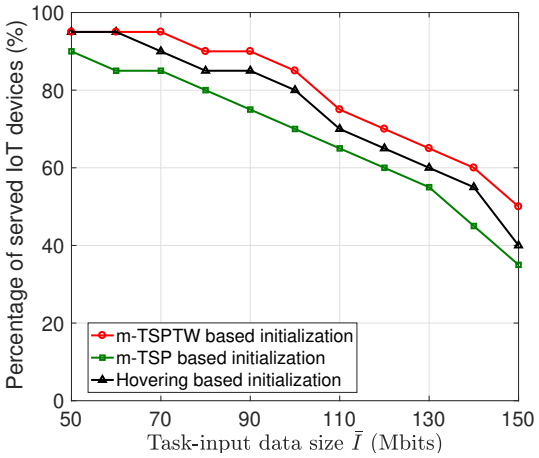
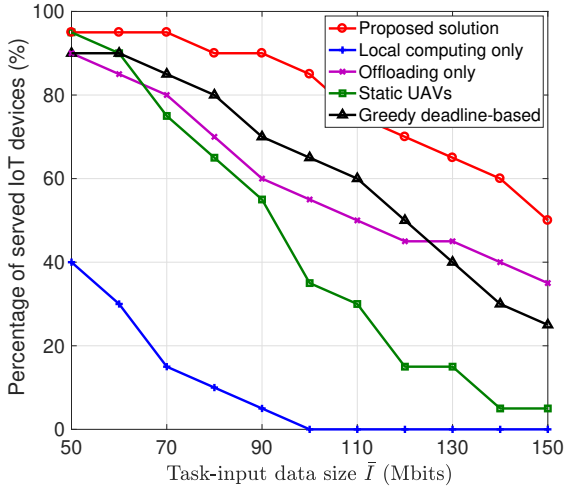


Fig. 6. Performance comparisons for different initialization schemes.

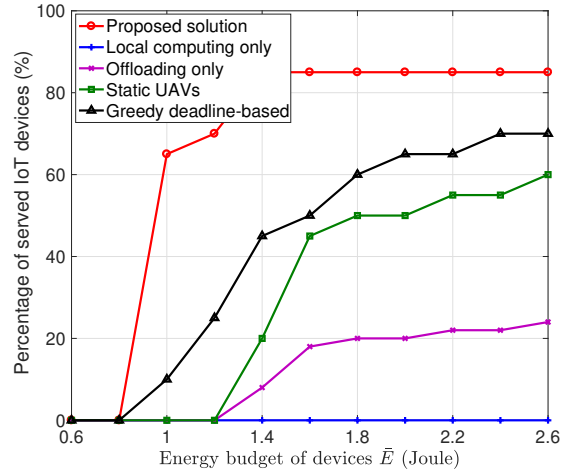
the visiting order determined by m-TSP based initialization scheme may not be appropriate and then has negative effect for path planning with time windows, while no visiting order has been determined by hovering based initialization scheme and it can be further obtained by our proposed iterative algorithm. The proposed m-TSPTW trajectory initialization outperforms the other two. This is expected since the m-TSPTW based trajectory initialization scheme takes the time windows into account, based on which the computation and communication resources can be more efficiently allocated to satisfy the deadline constraints.

To show the performance gain achieved by the proposed joint optimization scheme, we compare the percentage of served IoT devices between our proposed scheme and four benchmark schemes, which are referred to *Local computing only*, *Offloading only*, *Static UAVs*, and *Greedy deadline-based*. In the local computing only scheme, the IoT devices only perform local computing without offloading, while in the offloading only scheme, the task computation are only performed on UAVs, as in [36]. In the static UAVs scheme, M UAVs are fixed over the centers of M non-overlap clusters of devices to provide edge computation services as in [30], where the M clusters can be determined by the K-means clustering strategy. In the greedy deadline-based scheme, the UAVs serve the IoT devices by following the order of deadline from small to large, and allocates resource to maximize the computation service rate of each device.

From Fig. 7, it can be seen that our proposed scheme can serve more IoT devices than other benchmark schemes, and the percentage of served IoT devices increases when task-input data size $\bar{I}$ decreases or energy budget $\bar{E}$ increases, as expected. More performance gain can be achieved with large $\bar{I}$. When $\bar{E}$ is sufficiently large, the percentage of served IoT devices saturates. The reason is that when $\bar{E}$ is sufficiently large, it is not the system bottleneck anymore, while the maximum computation capacity of UAVs and devices (i.e., $\bar{f}^{U,\max}$, $\bar{f}_{\max}$) become the bottleneck. By comparing local computing only benchmark and offloading only benchmark with our proposed scheme, the gain brought by collaborative resource allocation with UAV offloading and local computing is shown.



(a) Percentage of served IoT devices versus $\bar{I}$ ($\bar{E} = 2$ Joule)



(b) Percentage of served IoT devices versus $\bar{E}$ ($\bar{I} = 100$ Mbits)

Fig. 7. Percentage of served IoT devices versus the task-input data size $\bar{I}$ or device's energy budget $\bar{E}$.

The gain brought by the mobility of UAVs is demonstrated by the performance gap between the static UAVs scheme and our proposed scheme. The performance gap between the greedy deadline-based benchmark and our proposed scheme demonstrates the additional gain of efficient collaborative UAV trajectory design. Fig. 8 shows the percentage of served IoT devices versus the number of IoT devices K when $\bar{I} = 100$ Mbits and $\bar{E} = 2$ Joule. For each K, the average performance of the algorithm is obtained over 100 independent settings of locations and deadlines of IoT devices. It is observed that the percentage of served IoT devices decreases with the increase of the number of IoT devices K. The reason is that the computation and communication resources as well as UAV flying/hovering time are limited, if more devices are deployed in the considered area, then less resources are allocated for each IoT device and thus results in less served devices. It can be seen that none of the IoT devices can be served before their deadlines with local computing only benchmark, which demonstrates the advantages of collaborative UAV-enabled computation offloading and local computing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3073208, IEEE Internet of Things Journal

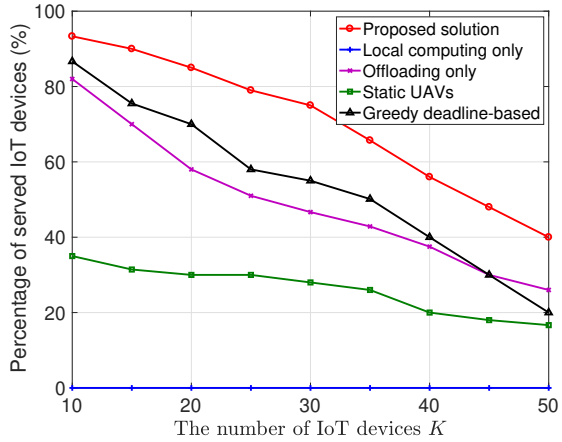


Fig. 8. Percentage of served IoT devices versus the number of IoT devices K.

VI. Conclusion

This paper considers a UAV-enabled MEC where multiple UAVs are employed as edge servers to provide computation assistance to IoT devices with hard deadlines. Our objective is to maximize the number of served IoT devices by jointly optimizing UAV trajectory and service indicator, as well as resource allocation and computation offloading, where the served IoT devices should complete their computation tasks before their deadlines with given energy budgets. To solve the formulated MINLP problem, we first reformulate the problem by penalizing the objective through a deductive penalty term, and then decompose the problem into two sub-problems. By alternately solving each of the two sub-problems with the SCA techniques, we propose an iterative algorithm to obtain a KKT solution. The performance is analyzed and then evaluated numerically as compared to several baseline schemes, where the superior performance of the proposed scheme is demonstrated.

Appendix A
Proof of Proposition 1

Let $\mathcal{V} \triangleq \{\mathcal{A}, \mathcal{P}, \mathcal{F}, \mathcal{U}\}$. Similar to [56], define Lagrangian $\mathcal{L}(\mathcal{V}, \Theta, \lambda) = \sum_{k=1}^{K} \theta_k - \lambda \sum_{k=1}^{K} (\theta_k - \theta_k^2)$ with Lagrangian multiplier λ to handle the non-convex constraint (17). Problem (P3) can be expressed as $\max\limits_{\mathcal{V},\Theta} \min\limits_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$, whose dual problem is $\min\limits_{\lambda} \max\limits_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$. Due to the max-min inequality, we have

$$\max_{\mathcal{V},\Theta} \min_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda) \leq \min_{\lambda} \max_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda). \tag{39}$$

Let $\vartheta(\lambda) \triangleq \max\limits_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$ and define $\vartheta(\lambda^*) \triangleq \min\limits_{\lambda} \vartheta(\lambda)$. Then we have

$$\vartheta(\lambda^*) \leq \max_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda). \tag{40}$$

Note that for all $\theta_k, 0 \leq \theta_k \leq 1$, it can be verified that $\sum_{k=1}^{K} (\theta_k - \theta_k^2) \geq 0$. If $\sum_{k=1}^{K} (\theta_k - \theta_k^2) > 0$, then $\vartheta(\lambda)$ tends to $-\infty$ at the optimal solution, since $\vartheta(\lambda)$ is monotonically decreasing with λ. This contradicts with (39), which shows that $\vartheta(\lambda)$ is lower bounded. Therefore, we should have $\sum_{k=1}^{K} (\theta_k - \theta_k^2) = 0$. In this case, $\mathcal{L}(\mathcal{V}, \Theta, \lambda) = \min\limits_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$. Due to (40), we have

$$\min_{\lambda} \max_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda) \leq \max_{\mathcal{V},\Theta} \min_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda). \tag{41}$$

Combining (39) and (41) yields $\max\limits_{\mathcal{V},\Theta} \min\limits_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda) = \min\limits_{\lambda} \max\limits_{\mathcal{V},\Theta} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$. On the other hand, $\vartheta(\lambda)$ is monotonically decreasing with λ. Thus, we can conclude that for any $\lambda > \lambda^*$, $\vartheta(\lambda) = \max\limits_{\mathcal{V},\Theta} \min\limits_{\lambda} \mathcal{L}(\mathcal{V}, \Theta, \lambda)$, where the LHS and RHS of the equality are from Problem (P4) and (P3), respectively. This completes the proof.

References

[1] Cisco company, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022", Mar. 2019. [online] Available: *https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf*

[2] 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.

[3] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE INFOCOM'19*, Paris, France, Apr./May, 2019, pp. 1459–1467.

[4] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *Proc. IEEE INFOCOM'19*, Paris, France, Apr./May, 2019, pp. 514–522.

[5] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *Proc. IEEE INFOCOM'19*, 2019, Paris, France, Apr./May, pp. 10–18.

[6] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5G network slice extension with MEC UAVs managed with reinforcement learning," *IEEE J. Sel. Areas Commun.*, to appear.

[7] N. Cheng, et al., "Air-ground integrated mobile edge networks: Architecture, challenges and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, Aug. 2018, pp. 26–32.

[8] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 140–146, Feb. 2020.

[9] C. Zhan, H. Hu, X. Sui, Z. Liu, and D. Niyato, "Completion time and energy optimization in UAV-enabled mobile edge computing system," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7808–7822, Aug. 2020.

[10] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587-597, Mar. 2018.

[11] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.

[12] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.

[13] Y. Wang, X. Tao, Y. T. Hou, and P. Zhang, "Effective capacity-based resource allocation in mobile edge computing with two-stage tandem queues," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6221–6233, Sept. 2019.

[14] Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao, "Task offloading with network function requirements in a mobile edge-cloud network," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2672–2685, Nov. 2019.

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

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

[17] C. Yi, J. Cai, and Z. Su, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 29-43, 1 Jan. 2020.

[18] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7867-7881, Dec. 2020.

[19] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1133-1146, Jun. 2020.

[20] S. Josilo and G. Dan, "Computation offloading scheduling for periodic tasks in mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 667-680, Apr. 2020.

[21] C. Caillouet, F. Giroire, and T. Razafindralambo, "Efficient data collection and tracking with flying drones," *Ad Hoc Networks*, vol. 89, pp. 35-46, Jun. 2019.

[22] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sept. 2018.

[23] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117-1129, May 2019.

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

[25] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, "UAV-assisted wireless powered cooperative mobile edge computing: Joint offloading, CPU control, and trajectory optimization," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2777–2790, Apr. 2020.

[26] X. Wang and L. Duan, "Economic analysis of unmanned aerial vehicle (UAV) provided mobile services," *IEEE Trans. Mobile Comput.*, to appear.

[27] M. A. Ali, Y. Zeng, and A. Jamalipour, "Software-defined coexisting UAV and WiFi: Delay-oriented traffic offloading and UAV placement," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 988–998, June 2020.

[28] R. Duan, J. Wang, C. Jiang, Y. Ren, and L. Hanzo, "The Transmit-Energy vs Computation-Delay Trade-Off in Gateway-Selection for Heterogenous Cloud Aided Multi-UAV Systems," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 3026-3039, Apr. 2019.

[29] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Netw.*, vol. 33, no. 2, pp. 36–43, Mar. 2019.

[30] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming," *IEEE Netw.*, vol. 33, no. 3, pp. 42–48, May 2019.

[31] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV enabled load-balance mobile edge computing for IoT networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6898–6908, Aug. 2020.

[32] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming endurance issue: UAV-enabled communications with proactive caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1231–1244, June 2018.

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

[34] Q. Wu, Y. Zeng, and R. Zhang, "Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109-2121, Mar. 2018.

[35] H. Wang, J. Wang, G. Ding, J. Chen, Y. Li, and Z. Han, "Spectrum sharing planning for full-duplex UAV relaying systems with underlaid D2D communications," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1986–1999, Sept. 2018.

[36] X. Hu, 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.

[37] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[38] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, June 2019.

[39] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020.

[40] F. Ono, H. Ochiai, and R. Miura, "A wireless relay network based on unmanned aircraft system with rate optimization," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7699–7708, Nov. 2016.

[41] C. Zhan and Y. Zeng, "Aerial-ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1937–1950, Mar. 2020.

[42] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[43] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sept. 2013.

[44] X. Zhang, J. Zhang, J. Xiong, L. Zhou, and J. Wei, "Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating noma," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5613–5627, June 2020.

[45] 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, Sept. 2019.

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

[47] G. Caire, "On the Ergodic rate lower bounds with applications to massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3258–3268, May 2018.

[48] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.

[49] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming," 2016. [Online] Available: http://cvxr.com/cvx.

[50] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, June 2019.

[51] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, vol. 2. Philadelphia, PA, USA: SIAM, 2001.

[52] M.W. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations research*, vol. 4, no. 1, pp. 285–305, 1985.

[53] P. Toth and D. Vigo, "The vehicle routing problem," *SIAM Monographs on Discrete Mathematics and Applications*. Philadelphia, PA: SIAM, 2002.

[54] M.M. Solomon, and J. Desrosiers, "Time window constrained routing and scheduling problems. *Transportation science*, vol. 22, no. 1, pp. 1-13, 1988.

[55] J. Zhang, Y. Zeng, and R. Zhang, "UAV-enabled radio access network: Multi-mode communication and trajectory design," *IEEE Trans. Signal Processing*, vol. 66, no. 20, pp. 5269-5284, 15 Oct.15, 2018.

[56] B. Soleimani and M. Sabbaghian, "Cluster-based resource allocation and user association in mmWave femtocell networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1746–1759, Mar. 2020.

[57] D. Kim, R. N. Uma, B. H. Abay, W. Wu, W. Wang, and A. O. Tokuta, "Minimum latency multiple data mule trajectory planning in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 838-851, Apr. 2014.

Cheng Zhan received the B.Eng. and Ph.D. degrees in Computer Science from the School of Computer Science, University of Science and Technology of China, Anhui, China, in 2006 and 2011 respectively. From 2009 to 2010, he was a Research Assistant with the Department of Computer Science, City University of Hong Kong. From 2016 to 2017, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore. He is currently an Associate Professor with the School of Computer and Information Science, Southwest University, China. His research interests include unmanned aerial vehicle communications, multimedia communications, wireless sensor networks, and network coding. He served as a TPC Member for the IEEE ICC, WCNC, IWCMC, and UIC.

Han Hu is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology, China. He received the B.E. and Ph.D. degrees from the University of Science and Technology of China, China, in 2007 and 2012 respectively. His research interests include multimedia networking, edge intelligence and space-air-ground integrated network. He received several academic awards, including Best Paper Award of IEEE TCSVT 2019, Best Paper Award of IEEE Multimedia Magazine 2015, Best Paper Award of IEEE Globecom 2013, etc. He served as an Associate Editor of IEEE TMM and Ad Hoc Networks, and a TPC Member of Infocom, ACM MM, AAAI, IJCAI, etc.

Zhi Liu (S'10-M'14-SM'19) received the B.E., from the University of Science and Technology of China, China and Ph.D. degree in informatics in National Institute of Informatics. He is currently an Assistant Professor at Shizuoka University. His research interest includes video network transmission, vehicular networks and mobile edge computing. He is now an editorial board member of Springer wireless networks and has been a Guest Editor of ACM/Springer Mobile Networks & Applications, Springer Wireless networks and IEICE Transactions on Information and Systems. He is a senior member of IEEE.

Zhi Wang (S'10-M'14) is currently an associate professor in Graduate School at Shenzhen, Tsinghua University. His research areas include multimedia network, mobile cloud computing and large-scale machine learning systems. He received the Outstanding Doctoral Dissertation Award from China Computer Federation in 2014, Best Paper Award at ACM Multimedia 2012, and Best Student Paper Award at MMM 2015. He is a recipient of the Second Prize of National Natural Science Award in 2018. He is Guest Editor of ACM TIST and JCST. His research has been covered by prestigious medias including MIT Technology Review.

Shiwen Mao (S'99-M'04-SM'09-F'19) received his Ph.D. in electrical and computer engineering in 2004 from Polytechnic University, Brooklyn, NY. Currently, he is a Professor and Earle C. Williams Eminent Scholar, and Director of the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL. His research interest includes wireless networks, multimedia communications, and smart grid. He is on the Editorial Board of IEEE Transactions on Wireless Communications, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Mobile Computing, IEEE Internet of Things Journal, IEEE Open Journal of the Communications Society, IEEE Multimedia, IEEE Networking Letters, ACM GetMobile, and KeAi Digital Communications and Networks Journal, etc. He is the TPC Co-Chair of IEEE INFOCOM 2018 and TPC Vice Chair of IEEE GLOBECOM 2022. He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019 and NSF CAREER Award in 2010. He is a co-recipient of the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE ComSoc MMTC 2018 Best Journal Award and 2017 Best Conference Paper Award, the Best Demo Award from IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2019, 2016 & 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.