Energy Consumption Minimization in UAV-assisted Mobile Edge Computing Systems: Joint Resource Allocation and Trajectory Design

Jiequ Ji, Kun Zhu, Member, IEEE, Changyan Yi, Member, IEEE, and Dusit Niyato, Fellow, IEEE

Abstract—Unmanned aerial vehicles (UAVs) have been introduced into wireless communication systems to provide high-quality services and enhanced coverage due to their high mobility. In this paper, we study a UAV-assisted mobile edge computing (MEC) system in which a moving UAV equipped with computing resources is employed to help user devices (UDs) computing their tasks. The computing tasks of each UD can be divided into two parts: one portion is processed locally and the remaining portion is offloaded to the UAV for computing. Offloading is enabled by uplink and downlink communications between UDs and the UAV. On this basis, two types of access modes are considered, namely non-orthogonal and orthogonal multiple access. For both access modes, we formulate new optimization problems to minimize the weighted sum energy consumption of the UAV and UD by jointly optimizing the UAV trajectory and computation resource allocation, under the constraint on the number of computation bits. These problems are non-convex optimization problems that are difficult to solve directly. Accordingly, we develop alternating iterative algorithms to solve them based on the block alternating descent method. Specifically, the UAV trajectory and computation resource allocation are alternatively optimized in each iteration. Extensive simulation results demonstrate the significant energy savings of our proposed joint design over the benchmarks.

Index Terms—Mobile edge computing, computation offloading, local computation, trajectory optimization, resource allocation.

I. INTRODUCTION

With the development of Internet of Things (IoT), more and more mobile applications emerge in our life, such as face recognition, video processing and online gaming [1]. However, all of these applications are computation-intensive and latency-sensitive, which normally need a high computation capability [2]. The limited battery energy and low computation capability make it difficult for user devices (UDs) to execute these applications with their own computing resources [3]. To tackle this issue, mobile edge computing (MEC) has emerged as a promising solution, which can provide cloud computing services for UDs at the edge of wireless networks with low transmission delay and light access burden [4]–[6]. Different from the traditional cloud computing, MEC servers are deployed at the infrastructure-based edge of wireless networks in proximity to UDs. By offloading partial or total computation-intensive tasks to the MEC server, the quality of computation experience can be significantly improved.

However, traditional MEC techniques are not applicable to the situation that the network facilities are sparsely distributed or the number of UDs increases dramatically [7]. Fortunately, unmanned aerial vehicle (UAV)-enabled MEC has been proposed and envisioned as a promising technology to tackle these challenges [8]. Compared with traditional architectures, UAVs equipped with computing resources can bring many potential advantages to MEC systems [9]. Firstly, UAVs usually provide line-of-sight (LoS) channel links with UDs, which can mitigate signal blockage and shadowing. Secondly, UAVs can leverage their mobility to dynamically adjust their trajectories to move closer to the UDs so as to provide better channel conditions. In addition, when a UAV flies over a UD, the energy consumed by the UD to offload its computation tasks to the UAV can be significantly reduced.

A. Related Work

There have been a number of research works on MEC techniques [10]–[14]. The authors in [10] investigated a multi-cell MEC system, where the computation and radio resources were jointly optimized to minimize the total energy consumption. In [11], an optimal resource allocation algorithm that maximizes the sum of computation efficiency was developed. The authors in [12] proposed a task offloading policy to reduce the energy consumption under the data processing delay constraints. The authors in [13] investigated a multi-tasking MEC system in which multiple access points (APs) equipped with computing resources were employed to help local resource-limited UDs to reduce their task execution latency. To achieve a delay-energy tradeoff, the authors proposed a random arrival model in [14] for computation task offloading. However, conventional MEC techniques cannot cope with the situation that the number of UDs increases explosively or the communication facilities are sparsely distributed.

Due to the attractive advantages of unmanned aerial vehicles (UAVs) in terms of their flexible movement and easy deploy-
ment, the integration of UAV-aided communication with MEC can further enhance the computational performance [15]–[22]. In [15], a UAV-enabled MEC system was first proposed, where the computing performance of UDs can be obviously improved with UAVs. The author in [16] proposed a novel scheme to guarantee the security of UAV-relayed wireless networks with caching by jointly the time scheduling and UAV trajectory. The authors in [17] proposed a joint trajectory and bit allocation algorithm to minimize the energy consumption. The authors in [18] studied a space-air-ground integrated computing system, where UAVs and satellites were used to provide edge computing and cloud computing accesses, respectively. The energy efficiency in an MEC system was investigated in [19], with the objective of minimizing the computation energy and hovering energy of UAVs. In [20], a UAV-aided wireless powered MEC system was investigated, where the sum energy consumption at the UAV was minimized by optimizing the CPU frequency and UAV trajectory. A computation rate maximization problem was proposed in [21], subject to the constraints of UAV mobility and energy harvesting causality. The authors in [22] studied resource allocation for time division multiple access (TDMA) and orthogonal frequency division multiple access (OFDMA) based multiuser MEC systems. However, such access protocols may not be able to accommodate massive connectivity with high spectrum utilization. We note that non-orthogonal multiple access (NOMA) is a promising technique for 5G networks, which allows multiple users to exploit successive interference cancellation to alleviate the co-channel interference [23]–[26]. For uplink and downlink transmission in MEC systems, non-orthogonal and orthogonal multiple access modes can be used due to their superior spectral efficiency.

It is worth noting that [19]–[21] assume that all computation tasks are offloaded to the UAV for computation, whereas this may be impractical as the number of computation input bits is tremendously large. In fact, one portion of data bits computed locally at each UD and the other portion of data bits offloaded to the UAV for computing can efficiently save the total energy consumption. Specifically, less number of computation bits to be offloaded to the UAV will reduce the communication energy consumption of UDs. Moreover, UDs may prefer to compute their tasks locally when the UD-to-UAV channels are poor.

In addition, the energy consumption of UAV-MEC system is a major issue when tackling offloaded tasks and trajectory. However, most of the existing work has not resolved the energy consumption problem of UAV for computation offloading and flying. Although the total mobile energy consumption of the UAV was considered in [17], the weighted sum energy consumption of the UAV and UDs under the partial offloading mode is ignored.

B. Our Contributions

Motivated by the aforementioned shortcoming of the existing work, in this paper, we investigate a UAV-aided MEC system in which a fixed-wing UAV equipped with computing resources is used to provide offloading opportunities for UDs with limited local computing resources. Each UD has spartable computing tasks to complete, which can be partitioned into two parts: one portion is computed locally and the other portion is offloaded to the UAV. In our design, task offloading is executed by uplink and downlink communications between UDs and the UAV. Meanwhile, two types of access modes are considered, namely orthogonal or non-orthogonal multiple access. For both access modes, we formulate new problems to jointly design computation resource allocation and UAV trajectory in order to minimize the weighted sum energy consumption of the UAV and UDs under the computation-causality constraint and the UAV trajectory constraint. Note that the computation resource parameter consists of the CPU frequencies of the UAV and each UD as well as the offloading and downloading bits.

Some benefits can be achieved by our proposed joint design. On the one hand, by properly designing resource allocation, each UD can offload partial or all of its computation tasks to the UAV, which alleviates the computational burden on UDs and reduces the task execution latency. On the other hand, by designing flight trajectory, the UAV can get closer to its serving UD to obtain a better channel condition, which shortens the offloading distance and enhances the offloading efficiency.

However, the formulated problems for both orthogonal and non-orthogonal access modes are shown to be two non-convex optimization problems. To facilitate solving them, we propose alternating iterative algorithms based on the block alternating descent method. Specifically, we divide the entire optimization variables into two blocks, namely resource allocation and UAV trajectory. We then alternately optimize these variable blocks. Notice that the resource allocation problem with fixed UAV trajectory is a convex optimization problem, which can be optimally solved by using the Lagrange duality method. However, the UAV trajectory optimization problem with fixed resource allocation is challenging to solve due to its non-convexity. As such, we adopt the successive convex approximate method to solve it approximately. We further analyze the computational complexity of our proposed iterative algorithms.

The main novelty and contributions of this paper are summarized as follows:

- This is the first work to study the weighted sum energy consumption problem in a UAV-assisted MEC system by joint task offloading and local computing design.
- For both orthogonal and non-orthogonal multiple access modes, we aim at jointly designing computation resource allocation and UAV trajectory in order to minimize the weighted sum energy consumption of the UAV and UDs.
- To efficiently solve the formulated problems, alternating iterative algorithms are proposed, and their computational complexity are analyzed.
- Simulation results provide several key observations. Firstly, the great performance improvement is achieved by our proposed joint design compared with other benchmarks. Secondly, the weighted sum energy consumption of our proposed joint design decreases with the increase of task completion time, revealing a time and energy consumption tradeoff. Finally, the sum energy consumption for the OMA mode is lower than that for the NOMA mode.

The rest of this paper is organized as follows. In Section II, we introduce the system model and propose two optimization problems. Sections III and IV present joint resource allocation and trajectory design algorithms for both orthogonal and non-
orthogonal access modes, respectively. Simulation results are presented in Section V. Conclusions are drawn in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model

As shown in Fig. 1, we consider a UAV-assisted mobile edge computing system consisting of $S$ user devices (UDs) denoted as $S = \{1, \ldots, S\}$ and one UAV, all of which are equipped with a single antenna. It is assumed that each UD and the UAV have an onboard communication circuit and an onboard computing processor powered via their embedded batteries. Note that the computing processor of each UD is a microprocessor that only can perform simple tasks locally [17]. The UAV has a powerful processor that can act as an assistant to help local resource-limited UDs compute their tasks. Assume that the computation tasks of each UD $s \in S$ are bitwise independent and can be divided into two parts: one portion is computed locally and the other portion is offloaded to the UAV [27]. To complete the offloading of a given task for UD $s$, we consider the following phases: (i) uplink transmission of task-input data from UD $s$ to the UAV; (ii) computation of offloading tasks at the UAV; (iii) downlink transmission of task-output data from the UAV to UD $s$. Besides, the UAV is assumed to operate in a frequency division duplex mode with equal bandwidth $B$ for receiving and forwarding data. Note that all computation tasks need to be completely computed within a period of $T$ in seconds.

B. Coordinate System and Channel Model

Suppose that the position of each UD is fixed on the ground with zero height within a task completion period $T$. Then, a 2D cartesian coordinate system is utilized, where the horizontal coordinate of UD $s \in S$ is denoted by $\mathbf{w}_s = [x_s, y_s]^T$, which is known in advance at the UAV for designing its trajectory. During period $T$, the UAV is assumed to fly at a given height of $H$ in m above the ground. The horizontal coordinate of the UAV at time instant $t$ is given as $\mathbf{q}_u = [x_u, y_u]^T, t \in [0, T]$. For ease of illustration, we partition the finite period $T$ into $N$ time slots with equal duration $\delta_t = \frac{T}{N}$, where $\delta_t$ is chosen to be sufficiently small to ensure that the position of the UAV is considered to be static during each slot. Hence, the flight trajectory of the UAV in a period $T$ is denoted by the sequence $\mathbf{q}_u[n] = [x_u[n], y_u[n]], n \in \mathcal{N} = \{1, \ldots, N\}$. Define $V_{\text{max}}$ as the maximum speed of the UAV. To proceed, the maximum distance that the UAV can move during each slot is given by $D_{\text{max}} = V_{\text{max}} \delta_t$. Due to its launching and landing locations are usually predetermined, the start and end positions of the UAV are denoted by $\mathbf{q}_l = [x_l, y_l]$ and $\mathbf{q}_f = [x_f, y_f]$, respectively. Accordingly, the mobility constraints of the UAV are given by

$$\begin{align*}
\mathbf{q}_u[1] &= \mathbf{q}_l, \\
\mathbf{q}_u[N] &= \mathbf{q}_f, \\
|\mathbf{q}_u[n + 1] - \mathbf{q}_u[n]| &\leq D_{\text{max}}, \forall n.
\end{align*}$$

Unlike traditional communications, the UD-to-UAV and the UAV-to-UD channels are both dominated by the line-of-sight (LoS) link. Similar to [28] and [29], the Doppler shift due to the high mobility of the UAV can be assumed to be perfectly compensated at the UDs. Hence, the channel gain between UD $s$ and the UAV in time slot $n$ is described as the free-space wireless channel model, which is calculate as

$$g_{s,u}[n] = \beta d_{s,u}^{-\alpha_u / 2}[n] = \frac{\beta}{H^2 + \|\mathbf{q}_u[n] - \mathbf{w}_s\|^2}, n \in \mathcal{N},$$

where $\alpha_u$ denotes the path loss exponent, which is set to be 2 according to [30] and [31]; $d_{s,u}[n] = \sqrt{\|\mathbf{q}_u[n] - \mathbf{w}_s\|^2 + H^2}$ is the distance between UD $s$ and the UAV in time slot $n$; $\beta$ is the channel gain at the distance $d_{\text{ref}} = 1 \text{ m}$, which depends on the carrier frequency and antenna gains.

C. Computation Model and Execution Methods

The application of each UD $s \in S$ is characterized by the number $L_s$ of input data bits, the ratio $O_s$ of task-output data size to task-input data size, and the number $C_s$ of CPU cycles required for computing one bit of input data. It is worth noting that the task-input data of UD $s$ is bitwise independent and can be randomly separated to realize parallel computation between task offloading and local execution. After all UDs offload their tasks in time slot $n$, the UAV computes these offloaded tasks and sends the generated results back to the corresponding UDs. For downlink and uplink communications, we consider either orthogonal multiple access (OMA) or non-orthogonal multiple access (NOMA) modes, as shown in Fig. 2. In the following, we will introduce the operation of each UD under the partial offloading manner in detail.

1) Local Computation: Since the computation unit and the communication circuit are separated [32], [33], each UD can perform task offloading and local computing simultaneously. To fully utilize the energy for local computing, each UD adopts a dynamic frequency scaling (DFS) technology [34], and hence the energy consumed for performing local computation can be reduced by adjusting the CPU frequency. We define the CPU frequency of UD $s \in S$ in the $n$-th time slot as $f_s[n]$ (in the
unit of cycles per second). Then, the computation workload and energy consumption of UD \( s \) for local computing at the \( n \)-th time slot are respectively given by

\[
L_{s}^{\text{loc}}[n] = \frac{\delta_{f_s}[n]}{C_s^s}, \forall s, n,
\]

\[
E_{s}^{\text{loc}}[n] = \frac{\delta_{\psi_s}[f_s^3[n]]}{C_s^s}, \forall s, n,
\]

where \( \psi_s \) denotes the effective capacitance coefficient of UD \( s \), which depends on its processor chip structure [35, 36].

2) Computation Offloading: Let \( L_{s}^{\text{off}}[n] \) be the number of bits that UD \( s \) transmits to the UAV at time slot \( n \). It is worth noting that the energy consumption of each UD and the UAV for communication depends on whether OMA or NOMA is adopted. In OMA, each time slot is equally partitioned into \( S \) durations with \( \delta_f = \tau S \), while all operations related to UD \( s \) are performed during the \( s \)-th duration. For OMA, the energy consumption of UD \( s \) for computation offloading at time slot \( n \) is describe as

\[
E_{s,u}^{\text{off}}[n] = \frac{2 \left( \frac{L_{s}^{\text{off}}[n]}{m_s} - 1 \right)}{g_{s,u}[n]} \left( \sigma^2 B \delta_t + \sum_{j=1,j \neq s}^{S} g_{j,u}[n] E_{N,j,u}^{\text{off}} \right),
\]

where \( \sigma^2 \) is the noise power. In NOMA, all UDs transmit and receive data simultaneously during each time slot. When UD \( s \) offloads \( L_{s}^{\text{off}}[n] \) bits to the UAV at the \( n \)-th time slot, the corresponding energy consumption is given by

\[
E_{N,s,u}^{\text{off}}[n] = \frac{2 \left( \frac{L_{s}^{\text{off}}[n]}{m_s} - 1 \right)}{g_{s,u}[n]} \left( \sigma^2 B \delta_t + \sum_{j=1,j \neq s}^{S} g_{j,u}[n] E_{N,j,u}^{\text{off}} \right).
\]

To enhance its energy efficiency for computation, the UAV also exploits DFS technology. The CPU frequency of the UAV during time slot \( n \) for computing the offloaded data of UD \( s \) is defined as \( f_{u,s}[n] \). Thus, the computation workload and energy consumption at the UAV in time slot \( n \) are expressed as

\[
L_{\text{comp}}[n] = \frac{\delta_{f_u}[f_s[n]]}{C_s^u}, \forall s, n,
\]

\[
E_{u,s}[n] = \frac{\delta_{\psi_u}[f_{s,u}^3[n]]}{C_s^u}, \forall s, n,
\]

where \( \psi_u \) is the effective capacitance coefficient of the UAV. After the UAV completes the task calculation, the generated results will be sent back to the corresponding UDs. Let \( L_{\text{don}}[n] \) be the number of bits that the UAV sends to UD \( s \) at time slot \( n \). In OMA, the energy consumed by the UAV for transmitting \( L_{\text{don}}[n] \) bits to UD \( s \) at the \( n \)-th time slot is given as

\[
E_{\text{don}}[n] = \frac{2 \left( \frac{L_{\text{don}}[n]}{m_s} - 1 \right)}{g_{s,u}[n]} \left( \sigma^2 B \delta_t + \sum_{j=1,j \neq s}^{S} g_{j,u}[n] E_{\text{don}}[n] \right).
\]

In NOMA, when the UAV transmits \( L_{\text{don}}[n] \) bits to UD \( s \) at time slot \( n \), the corresponding energy consumption is given by

\[
E_{N,s,u}[n] = \frac{2 \left( \frac{L_{\text{don}}[n]}{m_s} - 1 \right)}{g_{s,u}[n]} \left( \sigma^2 B \delta_t + \sum_{j=1,j \neq s}^{S} g_{j,u}[n] E_{N,j,u}^{\text{don}} \right).
\]

Due to the high mobility of the UAV, its propulsion energy consumption in each time slot should be taken into account, which can be calculated as

\[
E_{u}^{\text{by}}[n] = \delta_t \left( \kappa_1 \nu^3[n] + \frac{\kappa_2}{\nu^2[n]} \right), \forall n,
\]

where \( \nu[n] \triangleq ||q_u[n+1]-q_u[n]|| \) is the flight speed of the UAV at time slot \( n \), which is constrained by the maximum speed \( V_{\text{max}} \), and \( \kappa_1 \) and \( \kappa_2 \) are constants that are related to the UAV weight, wing span efficiency and wing area, etc.

It should be noted that in any given time slot, the UAV can only process the tasks that have been received from UD \( s \in S \). For simplify, the processing delay at the UAV for decoding and computing operation is assumed to be one time slot. Thus, the computation-causality constraint should satisfy

\[
\sum_{i=2}^{n} \frac{\tau_{f_s}[i]}{C_s^u} \leq \sum_{i=1}^{n-1} L_{\text{don}}[i], n \in N_2 = \{2, \ldots, N-1\}.
\]

Similarly, in any given time slot, the UAV can only deliver the tasks that have been computed to the corresponding UDs. To this end, we have another computation-causality constraint

\[
\sum_{i=3}^{n} L_{\text{don}}[i] \leq O_s \sum_{i=2}^{n-2} \frac{\tau_{f_s}[i]}{C_s^u}, n \in N_3 = \{3, \ldots, N\}.
\]

We assume that all tasks need to be computed within \( T \). This implies the requirements that 1) UDs should not offload their computation tasks at the last two time slots; 2) the UAV cannot compute any task at the 1-st and \( N \)-th time slots; 3) the UAV is not able to deliver the task-output data to the UDs at the 1-st and 2-nd time slots. Thus, we have

\[
f_{u,s}[1] = f_{u,s}[N] = 0, \forall s,
\]

\[
f_{\text{don}}[s,u][1] = f_{\text{don}}[s,u][2] = 0, \forall s,
\]

\[
L_{\text{off}}[N-1] = L_{\text{off}}[N] = 0, \forall s.
\]

D. Problem Formulation

Based on the above analysis, we formulate new optimization problems, which minimize the weighted sum energy consumption of the UDs and UAV by jointly designing the computation resource allocation and UAV trajectory over a finite period.

1) For OMA Mode: When the OMA mode is used in uplink and downlink, the joint optimization problem is formulated as

\[
P_1: \text{minimize} \quad \sum_{n=1}^{N} \sum_{s=1}^{S} \omega_s E_{O,s}[n] + \omega_u E_{O,u}[n],
\]

\[
\text{s. t.} \quad \frac{\sum_{i=2}^{n} \tau_{f_s}[i]}{C_s^u} \leq \sum_{i=1}^{n-1} L_{\text{don}}[i], \forall s,
\]

\[
\frac{\sum_{i=3}^{n} L_{\text{don}}[i]}{C_s^u} \leq O_s \frac{\sum_{i=2}^{n-2} \tau_{f_s}[i]}{C_s^u}, \forall s,
\]

\[
\frac{\sum_{i=n}^{N-1} \tau_{f_s}[i]}{C_s^u} \leq \sum_{i=n+1}^{N-2} L_{\text{don}}[i], \forall s,
\]

\[
\frac{\sum_{i=1}^{N} \delta_{f_u}[f_s[n]]}{C_s^u} + \sum_{i=1}^{N-2} L_{\text{don}}[i] = L_{\text{off}}[N], \forall s,
\]

\[
f_{u,s}[1] = f_{u,s}[N] = 0, \forall s,
\]

\[
f_{\text{don}}[s,u][1] = f_{\text{don}}[s,u][2] = 0, \forall s,
\]

\[
L_{\text{off}}[N-1] = L_{\text{off}}[N] = 0, \forall s.
\]
\[
\|q_u[n+1] - q_u[n]\| \leq D_{\text{max}}, \forall n, \tag{13}\]
\[
q_u[1] = q_f, \quad q_u[N] = q_f, \tag{13k}\]
where \(E_{O,u,s}[n] = \sum_{s=1}^S (E_{\text{com}}^{n,s}[n] + E_{\text{don}}^{n,s}[n]) + E_{\text{fly}}^{n,s}[n]\) and \(E_{O,s}[n] = E_{s}^{\text{loc}}[n] + E_{O,u,s}^{\text{off}}[n]\) are the energy consumption of the UAV and UD s at slot n, \(w_s\) and \(w_n\) are the weights of UD s and the UAV, \(Q \triangleq \{q_u[n], \forall n\}\) is the trajectory of the UAV at slot n, \(F \triangleq \{f_s[n], u_s[n], \forall s, n\}\) is the CPU frequencies of the UAV and UD s at slot n, and \(L \triangleq \{L_{s,u,s}^{\text{off}}[n], L_{s,u,s}^{\text{on}}[n], \forall s, n\}\) is the task allocation of UD s at slot n. As can be seen, (13b) and (13c) are the task-causality constraints, (13d)–(13f) are the computation task allocation constraints, (13g)–(13i) ensure that each optimization variable is nonnegative, and (13j)–(13k) are the speed and trajectory constraints of the UAV.

2) For NOMA Mode: When uplink and downlink adopt the NOMA mode, the joint optimization problem is formulated as

**P2: minimize**

\[
F, L, Q
\]

subject to

\[
\begin{align}
\sum_{n=1}^N \left( \sum_{s=1}^S \omega_s E_{N,s}[n] + \omega_u E_{O,u,s}[n] \right), \tag{14a} \\
\begin{align}
\sum_{i=2}^{n-1} \frac{\delta f_{s,u,s}[i]}{C_s} & \leq \sum_{i=1}^n L_{s,u,s}[i], \tag{14b} \\
L_{s,u,s}[i] & \leq O_s N \frac{\delta f_{s,u,s}[i]}{C_s}, \tag{14c} \\
N \sum_{n=2}^N L_{s,u,s}[i] & = \sum_{n=1}^{N-2} L_{s,u,s}[i], \forall s, \tag{14d} \\
N \sum_{n=3}^N L_{s,u,s}[n] & = O_s \sum_{n=2}^{N-1} \frac{\delta f_{s,u,s}[n]}{C_s}, \tag{14e} \\
(13f) - (13k), \tag{14f}\end{align}
\]

where

\[
E_{N,s}[n] = \sum_{s=1}^S (E_{s}^{\text{loc}}[n] + E_{N,s,u,s}[n]) + E_{N,s}^{\text{off}}[n], \tag{15}\]
\[
E_{O,s}[n] = E_{s}^{\text{loc}}[n] + E_{O,u,s}^{\text{off}}[n]. \tag{15}\]

Note that P1 and P2 are non-convex optimization problems due to their objective functions are not convex with regard to the trajectory of the UAV. In addition, the variables \(L_{s,u,s}^{\text{off}}[n]\) and \(L_{s,u,s}^{\text{on}}[n]\) are severely coupled with the trajectory of the UAV. Consequently, P1 and P2 cannot be directly solved by existing optimization methods, which will be further discussed below.

### III. ENERGY CONSUMPTION MINIMIZATION FOR OMA

In this section, we propose an alternating iterative algorithm for solving problem P1. Specifically, P1 is divided into two subproblems by adopting the block alternating descent method [37], namely joint bit allocation and CPU frequency optimization under given trajectory and trajectory optimization under given bit allocation and CPU frequency, which can be solved alternately through an iterative manner till convergence. In the following, we will introduce these subproblems in detail.

#### A. Bit Allocation and CPU Frequency Optimization

We jointly optimize the bit allocation \(L\) and CPU frequency \(F\) with given trajectory \(Q\), for which the optimization problem can be described as

**P1.1: minimize**

\[
F, L
\]

subject to

\[
\begin{align}
\sum_{n=1}^N \omega_s E_{O,s}[n] + \omega_u E_{O,u,s}^{\text{(1)}}[n], \tag{20a} \\
(13b) - (13i), \tag{20b}\end{align}
\]

where \(E_{O,u,s}^{\text{(1)}}[n] = \sum_{s=1}^S (E_{s}^{\text{loc}}[n] + E_{O,u,s}^{\text{off}}[n])\). For any given trajectory \(Q\), the channel gain \(g_{s,u,n}[n] = Q\) is known. As can be observed, subproblem P1.1 relative to \(L\) and \(F\) is a convex optimization problem. To proceed, a Lagrange method [38] is exploited to obtain an optimal solution to subproblem P1.1, which can be derived in Theorem 1.

**Theorem 1:** With given trajectory \(Q\), the optimal bit allocation and CPU frequency related to UD s ∈ S are respectively expressed as (16)–(19) at the top of the next page, where

\[
\phi_{u,s} = \log_2 \frac{B g_{s,u,n}[n]}{\omega u \beta}, \quad n = 3, \cdots, N, \tag{21a}\]
\[
\phi_s = \log_2 \frac{B g_{s,u,n}[n]}{\omega u \beta}, \quad n = 1, \cdots, N - 2. \tag{21b}\]

Also, \([x]^+ = \max\{x, 0\}\), while \(\nu_s^\ast, \mu_s^\ast, \mu_s^\ast, \theta_s^\ast, \lambda_s^\ast\) denote the optimal Lagrange multipliers corresponding to the constraints given by (13b)–(13f), respectively.

**Proof:** See Appendix A.

**Remark 1:** It can be observed from Theorem 1 that \(L_{s,u,s}^{\text{off}}[n]\) and \(L_{s,u,s}^{\text{on}}[n]\) increase with increasing \(\phi_{u,s}[n]\) and \(\phi_{u,s}^\ast\[n]\). This means that more task-input data (or task-output data) need to be offloaded (or downloaded) with larger \(\phi_{u,s}[n]\) and \(\phi_{u,s}^\ast\[n]\), which corresponds to the scenarios with better channel conditions and smaller weights for energy consumption.

**Remark 2:** Theorem 1 reveals the fact that as the time slot index \(n\) increases, \(L_{s,u,s}^{\text{off}}[n]\) decreases while \(L_{s,u,s}^{\text{on}}[n]\) increases. This is because \(\sum_{i=n+1}^N \nu_s^\ast[i]\) and \(\sum_{i=n}^N \mu_s^\ast[i]\) in (18) and (19) are monotonically decreasing with \(n\), respectively. This means that as time goes by, the offloading bits in the uplink decrease, while the downloading bits in the downlink increase.

Note that it is essential to find the optimal values of the dual variables, i.e., \(\rho^\ast = \{\rho_s^\ast, \forall s\}\), \(\lambda^\ast = \{\lambda_s^\ast, \forall s\}\), \(g^\ast = \{g_s^\ast, \forall s\}\), \(\nu^\ast = \{\nu_s^\ast, \forall s, n\}\) and \(\mu^\ast = \{\mu_s^\ast, \forall s, n\}\), since they have a crucial effect on determining the optimal CPU frequency and bit allocation. Thus, a subgradient method [38] is exploited to obtain the optimal dual variables in \(\nu^\ast\) and \(\mu^\ast\) associated with inclusions (13b) and (13c), which is derived in Lemma 1.

**Lemma 1:** By exploiting the subgradient method, the dual variables \(\nu_{s,n}\) and \(\mu_{s,n}\) at the \((r+1)\)th iteration are given by

\[
\nu_{s,n}^{r+1} = \nu_{s,n}^{r} - \frac{\tau_{s,n}}{C_s} \Delta \nu_{s,n}^{r}, \forall s \in S, n \in N, \tag{22a}\]
\[
\mu_{s,n}^{r+1} = \mu_{s,n}^{r} - \frac{\tau_{s,n}}{C_s} \Delta \mu_{s,n}^{r}, \forall s \in S, n = 3, \cdots, N, \tag{22b}\]

where \(\tau_{s,n}\) and \(\tau_{s,n}\) are the iteration steps required to obtain the dual variables in \(\nu\) and \(\mu\) over \(r\)-th iteration [39]. Also, \(\Delta \nu_{s,n}^{r}\) and \(\Delta \mu_{s,n}^{r}\) are the corresponding gradients which are given by

\[
\Delta \nu_{s,n}^{r} = \frac{1}{n} \sum_{i=1}^n L_{s,u,s}^{\text{off}}[i] - \frac{1}{n} \sum_{i=2}^n \frac{\tau f_{s,u,s}[i]}{C_s}, \tag{23a}\]
\[
\Delta \mu_{s,n}^{r} = O_s \sum_{i=2}^n \frac{\tau f_{s,u,s}[i]}{C_s} - \frac{1}{n} \sum_{i=3}^n L_{s,u,s}^{\text{on}}[i], \tag{23b}\]
\[ f^*_n = \sqrt{\frac{[\lambda^*_n]}{3C_s^\varphi_s\omega_s}}, n \in \mathcal{N}, \]  
\[ f^*_{u,s}[n] = \begin{cases} \sqrt{\frac{\rho^*_s - \bar{O}_s + \sum_{j=1}^{N} \mu^*_s - \sum_{j=1}^{N} \nu^*_s}{3C_s^\varphi_s\omega_s}}, & n = 2, \ldots, N - 1, \\ 0, & n = 1 \text{ or } N \end{cases} \]  
\[ L^\text{off}^*_{u,s}[n] = \begin{cases} \tau B \left[ \phi^*_s + \log_2 \left( \sum_{j=1}^{N-1} \nu^*_s + \lambda^*_s - \rho^*_s \right) \right]^+, & n = 1, \ldots, N - 2, \\ 0, & n = 1 \text{ or } N \end{cases} \]

where \( L^\text{off}^*_{u,s}[n], f^*_{u,s,r}[n] \) and \( L^\text{off}^*_{u,s,r}[n] \) are the optimal solutions obtained by Theorem 1 with the dual variables obtained in \( r \)th iteration, denoted as \( \mu^r = \{ \mu^r_s, \forall s, n \}, \quad \nu^r = \{ \nu^r_s, \forall s, n \}, \) \( \rho^r = \{ \rho^r_s, \forall s \} \) and \( \lambda^r = \{ \lambda^r_s, \forall s \} \).

In addition, the bisection search method [40] in Lemma 2 can be utilized to derive the optimal dual variables in \( \rho^r, \nu^r, \lambda^r \) associated with equations (13d)–(13f).

Lemma 2: According to \( \mu^{r+1} \) and \( \nu^{r+1} \) given in (22a) and (22b), the corresponding \( \rho^{r+1}, \nu^{r+1} \) and \( \lambda^{r+1} \) can be obtained with the bisection search of \( \lambda^{r+1}_s \in [0, \lambda^{max}_s], \forall s \in \mathcal{S} \), where \( \lambda^{max}_s = 3C_s^\varphi_s\omega_s(L - \frac{C_s^\varphi_s}{L})^2 \). With a given \( \lambda^{r+1}_s \in [0, \lambda^{max}_s] \), the corresponding \( \rho^{r+1}_s \) and \( \nu^{r+1}_s \) can be achieved by another two bisection searches within \( \rho^{r+1}_s \in [\rho^{r+1,\text{low}}_s, \rho^{r+1,\text{up}}_s] \) and \( \nu^{r+1}_s \in [\nu^{r+1,\text{low}}_s, \nu^{r+1,\text{up}}_s] \), which make the expressions shown in Appendix B satisfy (43a)–(43b) and (43a)–(43c), where \( \rho^{r+1}_s, \nu^{r+1}_s, \rho^{r+1,\text{low}}_s, \rho^{r+1,\text{up}}_s, \nu^{r+1,\text{low}}_s, \nu^{r+1,\text{up}}_s \) can be obtained as in (44a)–(44d). The optimal \( \lambda^{r+1}_s, \rho^{r+1}_s, \) and \( \nu^{r+1}_s \) should make (43a)–(43d) hold.

Proof: See Appendix B.

Remark 3: Based on the discussion of [38], the subgradient can be guaranteed to converge to the optimal dual variables \( \nu^*, \mu^*, \rho^*, \nu^*, \lambda^* \) when the bisection search is terminated after a finite number of iterations.

B. UAV Trajectory Optimization

We optimize the UAV trajectory \( Q \) under the optimized bit allocation \( L \) and CPU frequency \( F \), for which the optimization problem can be formulated as

**P1.2:** minimize \( \sum_{n=1}^{N} \sum_{s=1}^{S} \omega_s E^{(2)}_{Q,O,s,u}[n] + \omega_u E^{(2)}_{Q,u}[n] \),  
subject to  
\[ \| q_u[n+1] - q_u[n] \| \leq D_{max}, \forall n, \]  
\[ q_u[1] = q_f, q_u[N] = q_f. \]

where \( E^{(2)}_{Q,u}[n] = \sum_{s=1}^{S} E^{(2)}_{Q,O,s,u}[n] + E^{\text{fly}}_{u}[n] \). Since \( E^{\text{fly}}_{u}[n] \) in the objective function (24a) is non-convex with respect to \( v[n] \), subproblem P1.2 is still a non-convex one. To efficiently solve this problem, we first introduce an auxiliary variable \( \chi[n] \) into \( E^{\text{fly}}_{u}[n] \), and rewrite \( E^{\text{fly}}_{u}[n] \) as

\[ E^{\text{fly}}_{u}[n] = \delta_1 \left( \kappa_1 v^3[n] + \kappa_2 \frac{v}{\chi[n]} \right), \forall n, \]  
with additional constraints:

\[ \| v[n] \|^2 \geq \chi^2[n], n \in \mathcal{N}, \]  
\[ \chi[n] \geq 0, n \in \mathcal{N}. \]

Notably, \( E^{\text{fly}}_{u}[n] \) is jointly convex with \( \chi[n] \) and \( v[n] \). However, the additional constraint in (26a) is still non-convex. As such, we adopt the successive convex approximation method to solve the non-convexity of (26a). Due to the left side expression in constraint (26a) is convex versus \( v[n] \), the following result can be obtained by defining the set \( \mathcal{V} = \{ v[n], \forall n \} \) as the given local point in \( j \)th iteration.

Lemma 3: With a local point \( v^j[n] \) in \( j \)th iteration, (26a) can be approximated to a convex one as

\[ f^{\text{flow}}(v^j[n]) \geq \chi^2[n], n \in \mathcal{N}, \]  
where

\[ f^{\text{flow}}(v^j[n]) = \| v[n] \|^2 + 2v[n] (v[n] - v^j[n]). \]

Proof: It is not difficult to see that \( \| v[n] \|^2 \) in (26a) is a convex quadratic. Recall that the first-order Taylor expansion of a convex function is its lower bound. Thus, with any given local point \( v^j[n] \) in \( j \)th iteration, we obtain

\[ \| v[n] \|^2 \geq 2v^j[n] (v[n] - v^j[n]) + \| v[n] \|^2 \]

Notably, \( f^{\text{flow}}(v^j[n]) \) is linear with \( v[n] \), verifying \( v^j[n] \) can be replaced by its lower bound \( f^{\text{flow}}(v^j[n]) \). Thus, the additional constraint (26a) is substituted as (27), which is convex.

After such a transformation, we find that the approximated problem of P1.2 is convex with respect to \( v[n] \) and \( \chi[n] \). Due to the horizontal positions of the UAV at different time slots are highly coupled, it is challenging to find a closed-form solution of \( Q \). Hence, we solve the approximated problem of P1.2 by exploiting the convex optimization software CVX [41].
C. Algorithm Design and Analysis

According to the above analysis of the alternating optimization for the UAV trajectory Q, CPU frequency F and bit allocation L in each subproblem, an efficient iterative algorithm denoted by Algorithm 1 is proposed for solving problem P1. Since the objective value of P1 with the solutions obtained by solving subproblem P1.1 and P1.2 is non-increasing after each iteration and has a finite upper bound, which thus guarantees the convergence of Algorithm 1.

Next, the computational complexity of Algorithm 1 mainly comes from optimizing Q, F and L in subproblems P1.1 and P1.2, respectively. The computational complexity for solving P1.1 can be denoted by $O(1/\xi^2 + S \log_2(1/\xi_\lambda)(\log_2(1/\xi_\rho) + log_2(1/\xi_\sigma)))$, where $\xi$ is the computational accuracy of the subgradient method, and $\xi_\lambda$, $\xi_\rho$, and $\xi_\sigma$ are the computational accuracy of the bisection search method. For subproblem P1.2, we can obtain an approximate solution by leveraging the CVX software, and its computational complexity is polynomial.

Algorithm 1: Iterative algorithm for solving P1

1: Setting $v_{\text{max}}$, $\beta$, $B$, $H$, $T$, $N$, $\mathbf{q}_1$, $\mathbf{q}_f$, $\Omega_1$, $\Omega_2$, $\varphi_1$, $\psi_1$, $C_s$, $\sigma_f^2$, $L_s$, $O_s$, the iterative steps $\theta^*_i$ and $\theta^*_f$, and the tolerance accuracies $\alpha_1$ and $\alpha_2$.
2: Initialize $\mathbf{q}^{(0)}_u[n]$ and the iteration index $j = 1$.
3: repeat
4: Initialize $\nu_I$, $\mu_I$, and $r = 1$.
5: repeat
6: Obtain $\rho^r$, $\varphi^r$, and $\lambda^r$ for given $\nu^r$ and $\mu^r$.
7: Obtain $f^r_u[n]$, $f^r_{u,s,r}[n]$, $L^r_{u,s,r}[n]$ and $I^r_{u,s,r}[n]$ by Theorem 1 for given $\rho^r$, $\varphi^r$, $\lambda^r$, $\nu^r$ and $\mu^r$.
8: Calculate the weighted sum energy consumption $E_r^{(1)}$ by substituting $F^r_{r,j}$, $L^r_{r,j}$, $\mathbf{q}^{(0)}_u[n]$ into (13a).
9: $r = r + 1$.
10: Update $\nu^r$ and $\mu^r$ by using the subgradient method.
11: until $|E_r^{(1)} - E_r^{(1-1)}| \leq \alpha$, then obtain $F^{r+1} = F^r_{r,j}$ and $L^{r+1} = L^r_{r,j}$.
12: repeat
13: Solve the approximated problem of P1.2 for given $F^{r+1}$ and $L^{r+1}$ by CVX, and obtain the optimal $Q^{r+1}$.
14: until
15: Update $j = j + 1$.
16: Calculate the weighted sum energy consumption by substituting $F^r$, $L^r$, $Q^r$ into the objective function of P1.
17: until $|E_j - E_{j-1}| \leq \alpha_1$, then obtain the minimum energy consumption $E_j$ with $F^r$, $L^r$ and $Q^r$.

IV. ENERGY CONSUMPTION MINIMIZATION FOR NOMA

In this section, we solve the problem P2 formulated for the NOMA mode. It can be seen that P2 is a non-convex problem, but its structure is similar to P1, which facilitates us to solve it with an alternating iterative algorithm. Similar to P1, P2 is also partitioned into two subproblems, namely joint bit allocation and CPU frequency optimization under given trajectory as well as trajectory optimization under given CPU frequency and bit allocation. Then, these subproblems are solved alternately via an iterative manner till convergence.

A. Bit Allocation and CPU Frequency Optimization

Let $E^{(3)}_{O,u,n}[n] = \sum_{s=1}^S (E_{\text{com},u,s}[n] + E_{\text{den},u,s}[n])$. For a given trajectory $Q$, the bit allocation and CPU frequency optimization problem is given by

P2.1: minimize $\mathbf{F}, \mathbf{L}$ subject to

\[
\begin{align*}
& \sum_{n=1}^N \left( \sum_{s=1}^S \omega_s E_{O,s,n}[n] + \omega_u E^{(3)}_{O,u,n}[n] \right), \\
& \text{s. t. } (13b) - (13i). 
\end{align*}
\]

Note that subproblem P2.1 has a convex constraint set and a convex objective function, and thus it is a convex optimization problem. To proceed, we resort to using the Lagrange duality method to solve this subproblem, which can obtain an optimal solution to P2.1. By defining $a = (a_{s,n}, \forall s, n), c = (c_s, \forall s), b = (b_{s,n}, \forall s, n), d = (d_s, \forall s)$ and $e = (e_s, \forall s)$ as the dual variables corresponding to constraints (13b)–(13f), we obtain the following result.

Theorem 2: For given multipliers $a^r_{s,n}, b^r_{s,n}, c^r_s, d^r_s$ and $e^r_s$, the optimal bit allocation and CPU frequency are respectively expressed as (31)–(34) at the top of the next page.

Proof: The proof is similar to Theorem 1, and thus, it is omitted here due to space limitation.

Since the optimal Lagrange multipliers $a^r, b^r, c^r, d^r$ and $e^r$ play vital roles in determining the optimal bit allocation $L$ and CPU frequency $F$, it is necessary to obtain their optimal values. Here, a subgradient method is leveraged to derive the optimal multipliers in $a^r$ and $b^r$ associated with inequations (13b) and (13c), as shown in Lemma 4.

Lemma 4: Based on the subgradient method, the Lagrange multipliers $\nu_{s,n}$ and $\mu_{s,n}$ at the $(r+1)$th iteration are given as

\[
\begin{align*}
& a^r_{s,n} = a_{s,n}^r - \theta^r_s \Delta a^r_{s,n}, \forall s, n \in N, \\
& b^r_{s,n} = b_{s,n}^r - \theta^r_s \Delta b^r_{s,n}, \forall s, n \in N,
\end{align*}
\]

(35a)

(35b)

where $\theta^r_s$ and $\theta^r_b$ are the iteration steps required to obtain the Lagrange multipliers in $a$ and $b$ over $r$-th iteration; $\Delta a^r_{s,n}$ and $\Delta b^r_{s,n}$ denote the corresponding gradients, which are given by

\[
\begin{align*}
& \Delta a^r_s = \sum_{i=1}^n \tilde{I}_{s,u,s}[i] - \sum_{i=2}^n \tilde{I}_{u,s,r}[i], \\
& \Delta b^r_s = O_s \sum_{i=2}^n \frac{\tilde{I}_{u,s,r}[i]}{C_s} - \sum_{i=3}^n \tilde{I}_{u,s,r}[i],
\end{align*}
\]

(36a)

(36b)

where $\tilde{I}_{u,s,r}[n], \tilde{I}_{u,s,r}[n], \tilde{I}_{u,s,r}[n]$ are the optimal solutions achieved by Theorem 1 with the Lagrange multipliers obtained in $r$-th iteration, denoted as $a^r = (a^r_{s,n}, \forall s, n), c^r = (c^r_s, \forall s), b^r = (b^r_{s,n}, \forall s, n), d^r = (d^r_s, \forall s)$ and $e^r = (e^r_s, \forall s)$.

Meanwhile, the bisection search method in Lemma 2 can be exploited to derive the optimal dual variables in $c^r$, $d^r$ and $e^r$ associated with equations (13d)–(13f).

Lemma 5: For given $a^r+1$ and $b^r+1$, the corresponding $c^r+1$, $d^r+1$ and $e^r+1$ can be obtained with the bisection search of $e^r+1 \in [0, \{e^r_{s}\}_{s \in S})$, where $e_{\max} = 3C_s/\omega_s \varphi_s (L_\text{wave})^2$. With a given $e_{r+1}^r \in [0, e_{\max}^r]$, the corresponding $d_{r+1}^r$ and $c_{r+1}^r$ can be achieved by another two bisection searches within $d_{r+1}^r \in [d_{r+1}^{\text{low}}, d_{r+1}^{\text{up}}]$ and $c_{r+1}^r \in [c_{r+1}^{\text{low}}, c_{r+1}^{\text{up}}]$.

Proof: The proof is similar to that in Lemma 2, and thus, it is omitted here due to space limitation.
of P1.2 with the additional constraint (39a) is jointly convex
After such a conversion, we see that the approximated problem
we first introduce an auxiliary variable
objection function of P2.2 is non-convex. To solve this issue,
rewrite \[ E \]
Due to
\[ \∥v[n+1]−v[n]∥≤D_{\max}, \forall n \]
\[ v[0] = q_f, v[N] = q_F. \]

Due to \( E^\text{fly} [n] \) in (9) is not a convex function of \( v[n] \), the
objection function of P2.2 is non-convex. To solve this issue,
we first introduce an auxiliary variable \( \bar{v}[n] \) into \( E^\text{fly}[n] \), and
rewrite \( E^\text{fly}[n] \) as
\[ E^\text{fly}[n] = \delta_i \left( \kappa_1 v^3[n] + \frac{\kappa_2}{v^2[n]} \right), \forall n, \]
with additional constraints:
\[ \|v[n]\|^2 ≥ \bar{v}^2[n], n ∈ N, \]
\[ \bar{v}[n] ≥ 0, n ∈ N. \]

As can be observed, \( E^\text{fly}[n] \) is jointly convex with \( v[n] \) and
\( \bar{v}[n] \). However, the additional constraint in (39a) is not convex.
Thus, we exploit the successive convex optimization method to solve the non-convexity of (39a). The left side expression in
constraint (39a) is convex with \( v[n] \) and can be approximated as
its lower bound via adopting its first-order Taylor expansion
at a given local point \( v^i[n] \) over \( i \)th iteration. Thus, the non-
convex constrain (39a) is converted to a convex one as
\[ \|v[n]\|^2 + v^T[n](v[n] − v^i[n]) ≥ \bar{v}[n]. \]
After such a conversion, we see that the approximated problem
of P1.2 with the additional constraint (39a) is jointly convex
with respect to \( v[n] \) and \( \bar{v}[n] \). Since the horizontal positions of
the UAV at different time slots are coupled, it is challenging
to obtain a closed-form solution of \( Q \). As such, we solve the
approximated problem of P1.2 by CVX [41].

\begin{algorithm}[ht]
\caption{Iterative algorithm for solving P2}
1: Setting \( V_{\max}, \beta, B, H, T, N, q_1, q_F, \omega_u, \omega_s, \kappa_1, \kappa_2, \varphi_s, \)
\( ψ_u, C_s, \sigma^2, L_s, O_s, \) the iterative steps \( \theta_{n}^r \) and \( \theta_{n}^l \), and
the tolerant accuracies \( \varepsilon \) and \( \varepsilon_1 \).
2: Initialize \( q_0^{(0)}[n], v_0^{(0)}[n] \) and the iteration index \( i = 1. \)
3: repeat
4: Initialize \( a_1, b_1, \) and \( r = 1. \)
5: repeat
6: Obtain \( c^*, \ d^* \) and \( e^* \) for given \( a^* \) and \( b^* \).
7: Obtain \( \bar{f}[n], \bar{f}_u,s,r[n], \bar{L}_n^\text{off} \) \( \bar{L}_{u,s,r}^\text{off} \) by
Theorem 1 for given \( a^*, \ b^*, \ c^*, \ d^* \) and \( e^* \).
8: Calculate the weighted sum energy consumption \( E^\text{f}(1) \)
by substituting \( F_{r,n}, L_{r,n}, q_0^{(0)}[n] \) into (14a).
9: \( r = r + 1. \)
10: Update \( a^* \) and \( b^* \) by using the subgradient method.
11: until \( |E^\text{f}(1) − E^\text{f}(1)| ≤ \varepsilon \), then obtain the optimal \( F_i+1 = F_{r,i} \) and \( L_i+1 = L_{r,i}. \)
12: repeat
13: Solve the approximated problem of P2.2 for given
\( F_i+1 \) and \( L_i+1 \) by CVX, and obtain the optional \( Q_i+1. \)
14: until
15: Update \( i = i + 1. \)
16: Calculate the weighted sum energy consumption by
substituting \( F^i, \ L^i, \ Q^i \) into the objective function of P2.
17: until \( |E_i − E_{i−1} | ≤ \varepsilon_1 \), then obtain the minimum energy
consumption \( E_i \) with \( F^i, \ L^i \) and \( Q^i \).
\end{algorithm}

C. Algorithm Design and Analysis
According to the aforementioned two subproblems, an
alternating iterative algorithm is developed to solve the original
problem P2, as shown in Algorithm 2. Specifically, the entire
optimization variables in problem P2 can be divided into two
blocks, namely computation resource scheduling \( L \) and \( F \) as
well as UAV trajectory $Q$, which are alternately optimized by solving subproblems P2.1 and P2.2. Since the proposed iterative algorithm consists of the successive convex approximation technique and Lagrange duality method, we can at last obtain a locally optimal solution for P2 through Algorithm 2.

As shown in Section IV-A and IV-B, the objection value of P2 with the solutions obtained via solving subproblems P2.1 and P2.2 is monotonically non-increasing over each iteration of Algorithm 2. In addition, the optimal value of P2 is upper-bounded through a finite of iterations, which thus guarantees the convergence of Algorithm 2.

The computational complexity of Algorithm 2 comes from optimizing the computation resource scheduling $L$ and UAV trajectory $Q$ in subproblems P2.1 and P2.2, respectively. Notably, P2.1 is a convex optimization problem and can be solved by the subgradient and bisection methods with computational complexity $O(1/\varepsilon^2 + S \log_2(1/\varepsilon))[\log_2(1/\varepsilon_\omega)+\log_2(1/\varepsilon_\rho)]$, where $\varepsilon$ denotes the computational accuracy of the subgradient method, and $\varepsilon_\omega, \varepsilon_\rho$ are the computational accuracies of the bisection search method. For subproblem P2.2, we can obtain an approximate solution by applying the CVX software, and its complexity is polynomial.

V. Simulation Results and Discussion

In this section, simulation results are provided for examining the performance of the proposed joint optimization algorithms for both OMA and NOMA modes. In the simulation, we will consider a UAV-assisted MEC system with $S = 4$ UDs, which are randomly scattered in a square area of $1.5 \times 1 \text{km}^2$. The initial and final locations of the UAV are set as $q_I = [0, 0]$ and $q_F = [10, 0]$, respectively. To show the effectiveness of our proposed joint design scheme, we consider the following three cases: i) the computation bits of the task-input data at UDs are $[L_1, L_2, L_3, L_4] = [7, 3, 5, 3] \times 10^2$ Mbits; ii) $[L_1, L_2, L_3, L_4] = [7, 5, 7, 3] \times 10^2$ Mbits; iii) $[L_1, L_2, L_3, L_4] = [3, 3, 7, 5] \times 10^2$ Mbits. For both OMA and NOMA modes, we first illustrate the UAV trajectory under different computation task requirements, and then analyze the convergence of our proposed algorithms. In addition, the impacts of the various parameters, such as the computation bits of the task-input data at UDs $L_s, s \in S$, the size ratio of task-output data to task-input data $O_s$, the weight for energy consumption of the UAV $\omega_u$, the task completion time $T$, are investigated on the performance evaluation metrics. For comparison, we consider the following three schemes.

• Scheme 1: All UDs just offload their tasks to the UAV for computing without local computing themselves.
• Scheme 2: The UAV directly flies from the initial position to the final position with a fixed velocity.
• Scheme 3: Each UD adopts its own computation resource to complete its task-input data without offloading.

In Table I, we summarize the basic system parameters.

![Table I](https://example.com/table1.png)

**Fig. 3 and Fig. 4** demonstrate the UAV trajectory for OMA and NOMA modes under three different cases, namely Case 1: $[L_1, L_2, L_3, L_4] = [7, 3, 5, 3] \times 10^2$, Case 2: $[L_1, L_2, L_3, L_4] = [7, 5, 7, 3] \times 10^2$ and Case 3: $[L_1, L_2, L_3, L_4] = [3, 3, 7, 5] \times 10^2$. It is worth noting that the total computation input bits at UDs are same for the cases in 3(a), 3(c), 4(a) and 4(c), e.g., 1800 Mbits, while the cases in 3(b) and 4(b) have larger total computation input bits, e.g., 2200 Mbits. It is not difficult to see that the computation bits of the task-input data at each UD has a great impact on designing the UAV trajectory. As shown in Fig. 3 and Fig. 4, under both access modes, the UAV prefers moving closer the UDs with higher task requirement for calculating. The reason is that the UDs with large computation input bits are inclined to offload their task-input data to the UAV for computing, and thus the UAV needs to move closer these UDs so as to improve the UD-to-UAV channel condition. All these results suggest that the distribution of computation input bits at UDs plays a critical role on the trajectory design. Moreover, it is clear that when the UAV gets close to the UDs with large number of computation input bits, the offloading and downloading energy consumption are both reduced.

Fig. 5 exhibits the convergence behaviours of our proposed algorithms for both OMA and NOMA modes, where we also consider the following three cases with different computation requirements, namely Case 1: $[L_1, L_2, L_3, L_4] = [7, 3, 5, 3] \times 10^2$, Case 2: $[L_1, L_2, L_3, L_4] = [7, 5, 7, 3] \times 10^2$ and Case 3: $[L_1, L_2, L_3, L_4] = [3, 3, 7, 5] \times 10^2$. It is clear that the weighted sum energy consumptions for all the cases are non-increasing after each iteration of Algorithm 1 and Algorithm 2. It can be seen that for the two access modes with different computation requirements, Algorithm 1 and Algorithm 2 converge in about 4 and 5 iterations, respectively. Such results suggest that our proposed Algorithm 1 and Algorithm 2 are quite efficient and have faster convergence speed. Moreover, for $[L_1, L_2, L_3, L_4] = [7, 5, 7, 3] \times 10^2$ Mbits, the proposed joint scheme requires a weighted sum energy consumption of 710 Joule for orthogonal access and 730 Joule for non-orthogonal access, respectively. It is worth noting that the performance of NOMA is worse than that of OMA due to the more severe interference generated by
the UAV for NOMA. Moreover, in the OMA mode, the UAV intends to fly more closely to each UD so that less power is consumed by UDs in uplink and downlink transmissions.

In Fig. 6, we compare the weighted sum energy consumption achieved by the proposed iterative algorithm with the penalty dual decomposition (PDD) algorithm [42]. It can be seen from Fig. 6 that the proposed iterative algorithm converges at around 5 iterations while the PDD-based algorithm converges after 8 iterations. Besides, the proposed algorithm achieves a better performance in minimizing the weighted sum energy consumption than that of the PDD-based algorithm.

Fig.7(a) and Fig.7(b) respectively demonstrate the relationship between the weighted sum energy consumption and the uniform task size $L = L_s, s \in \mathcal{S}$ under OMA and NOMA modes,
where we also compare the proposed joint design scheme with the above several benchmark schemes. According to Fig. 7, we can make several important observations. Firstly, as expected, the weighted sum energy consumption increases largely for all the four schemes as $L$ becomes large. The reason is that the UAV and UDs will consume more energy to complete the task-input data with large bits. Secondly, for varying $L$, the lowest energy consumption can be achieved by exploiting the proposed joint design scheme in comparison with other three benchmark schemes under both access modes. Moreover, the local computing scheme achieves a significant higher energy consumption than other schemes with computation offloading. It means that edge computing through offloading is essential for the computational performance improvement. Particularly, as shown in Fig. 7(b), the weighted sum energy consumption of our proposed joint design scheme is one thousandth of that of the local computing scheme, which demonstrates the enormous advantages achieved via employing the UAV as an assistant for calculating. Moreover, the weighted sum energy consumption of the direct trajectory scheme is one quarter higher than that of our proposed joint design scheme. When $L$ is small, e.g., $L = 400$ Mbits, the curves of the only offloading scheme and the proposed joint design scheme are close to each other. However, as $L$ increases, the performance gap between the only offloading scheme and the proposed joint design scheme is even larger than that between the direct trajectory scheme and the proposed joint design scheme. Such results suggest that our proposed joint resource allocation and trajectory design have prominent impacts on minimizing the weighted sum energy consumption. Thirdly, it is not difficult to see that the performance gaps between the proposed joint design scheme and other three benchmark schemes become larger with increasing $L$, which further demonstrates that our proposed joint design scheme is more effective in tackling the computationally intensive tasks.

Fig. 8(a) and Fig. 8(b) demonstrate the weighted sum energy consumption versus different values of task completion time $T$ under both OMA and NOMA modes, respectively. According to Fig. 8(a) and Fig. 8(b), we find that the weighted sum energy consumption decreases for the four schemes as $T$ grows, which verifies that there exist a tradeoff between time consumption and energy consumption for executing the same task, while the energy consumption decreases as the consumed time increases. For both access modes, it is clear that the proposed joint design scheme is superior to the three benchmark schemes in terms of energy consumption, while the performance improvement will become more significant under the strict time constraint. Such results further confirm that the proposed joint design scheme is effective for tackling the delay-sensitive task-input data and can realize a better delay-energy tradeoff. In particular, as $T$ becomes large, the UAV has more freedom to get closer to its serving UD to obtain a better channel condition and more tasks can be offloaded to the UAV, thereby reducing task execution latency. In addition, some similar observations can be obtained from Fig. 7(a) and Fig. 7(b).

Fig. 9(a) and Fig. 9(b) demonstrate the weighted sum energy consumption versus the uniform ratio of output-data to input-data $O$ for both OMA and NOMA.
NOMA modes, respectively. As shown in Fig. 9(a) and Fig. 9(b), the proposed joint design scheme achieves significantly lower energy consumption compared with three benchmark schemes under both access modes. For different values of $O$, the weighted sum energy consumption of the local computing scheme remains unchanged, while the weighted sum energy consumption increases for all the other schemes as $O$ becomes large. This is because more task-output data will be downloaded to the UDs when with a large $O$. By comparing the curves of the proposed joint design scheme and the direct trajectory scheme, we find that the energy consumption gap between the two schemes decreases as $O$ increases. However, the energy consumption gap between the proposed joint design scheme and the only offloading scheme increases with increasing $O$. The reason is that locally computing a portion of tasks at UDs and offloading the rest to the UAV can save the weighted sum energy consumption when $O$ is large.

Fig. 10(a) and Fig. 10(b) illustrate the weighted sum energy consumption versus the weight for energy consumption of the UAV $\omega_u$ for OMA and NOMA, respectively. It is clear that a better performance can be achieved by leveraging the proposed joint design scheme in comparison with all the benchmark schemes in both access modes. All the curves in Fig. 10(a) and Fig. 10(b) increase with $\omega_u$, except for the local computing scheme, since a higher energy consumption of the UAV will be counted into the weighted sum energy consumption when $\omega_u$ gets larger. It is notable that energy consumption gap between the proposed joint design scheme and the direct trajectory scheme becomes larger with increasing $\omega_u$ under both access modes. The reason is that the propulsion energy consumption of the UAV contributes a large portion of the weighted sum energy consumption for the direct trajectory scheme without trajectory design, and hence its weighted sum energy consumption increases much faster with respect to $\omega_u$ than the other schemes.

According to the above results, we observe that the weighted sum energy consumption for the non-orthogonal multiple access mode is much higher than that for the orthogonal multiple access mode for all the schemes. Notably, such a performance gain for the orthogonal multiple access mode is attributed to the fact that its performance mainly depends on the mutual interference between UDs, which is affected through the computation resource allocation. In addition, when $T$ is small, the orthogonal multiple access mode always achieves significantly lower weighted sum energy consumption than that of the non-orthogonal multiple access mode.

Fig. 11(a), Fig. 11(b) and Fig. 11(c) demonstrate the energy consumption of the UDs (considering that $\omega_s = 1, s \in S$), the weighted energy consumption and the energy consumption of the UAV versus $\omega_u$, respectively. According to Fig. 11(a) and Fig. 11(b), we observe that the weighted energy consumption of the UAV and UDs increases for all the schemes as $\omega_u$ gets larger, except that for the local computing scheme. It can be seen from Fig. 11(c) that the energy consumption of the UAV decreases as the value of $\omega_u$ increases. The reason is that our goal is to minimize the weighted sum energy consumption, while the optimal objective value increases with $\omega_u$, which is
Thus, the solutions of all dual variables are optimal, the corresponding solutions are \(\omega\) of the computation input bits at each UD greatly affect the to help local resource-limited UDs compute their tasks. Moreover, we considered two types access modes for the uplink and downlink communications required for computation migration. For both access modes, the weighted sum energy consumption was minimized by jointly optimizing the computation resource allocation and UAV trajectory. However, the formulated problems have been shown to be non-convex optimization problems. For efficiently solving them, alternating iterative algorithms were proposed based on the block alternating descend method. Specifically, the resource allocation and UAV trajectory were alternately optimized in an iterative manner till our proposed algorithms converge. Simulation results verified the significant energy saving of our proposed joint design compared to three benchmark schemes. It can also be concluded that the number of the computation input bits at each UD greatly affect the flight trajectory of the UAV.

VI. CONCLUSION

In this paper, we investigated a UAV-assisted MEC system where a UAV equipped with computing resources is employed to help local resource-limited UDs compute their tasks. Moreover, we considered two types access modes for the uplink and downlink communications required for computation migration. For both access modes, the weighted sum energy consumption was minimized by jointly optimizing the computation resource allocation and UAV trajectory. However, the formulated problems have been shown to be non-convex optimization problems. For efficiently solving them, alternating iterative algorithms were proposed based on the block alternating descend method. Specifically, the resource allocation and UAV trajectory were alternately optimized in an iterative manner till our proposed algorithms converge. Simulation results verified the significant energy saving of our proposed joint design compared to three benchmark schemes. It can also be concluded that the number of the computation input bits at each UD greatly affect the flight trajectory of the UAV.

APPENDIX A: PROOF OF THEOREM 1

Define \(\nu = \{\nu_{s,n}, \forall s, n\}, \mu = \{\mu_{s,n}, \forall s, n\}, \rho = \{\rho_{s,n}\}, \varphi = \{\varphi_s, \forall s\} \) and \(\lambda = \{\lambda_s, \forall s\}\) as the dual variables related to (13b)–(13f), respectively, where \(\nu_{s,n}\) and \(\mu_{s,n}\) are nonnegative. Mathematically, the Lagrange function of P1 is calculated as (41) at the top of this page, where \(\bar{\nu}_{s,n} = \sum_{i=n=1}^{N-1} \nu_{s,i}, \bar{\mu}_{s,n} = \sum_{i=n=1}^{N-1} \mu_{s,i}\) and \(\bar{\mu}_{s,n} = \sum_{i=n=1}^{N-1} \mu_{s,i}\). Thus, the Lagrangian dual function of P1 is given by

\[
\text{D1: } g^\dagger(\nu, \mu, \rho, \varphi, \lambda) = \min \{\mathcal{L}(\mathbf{L}, \mathbf{F}, \nu, \mu, \rho, \varphi, \lambda) \mid \text{s.t. } (13g)–(13i)\}.
\]

Thus, the solutions of \(\mathbf{F}\) and \(\mathbf{L}\) with any given \(\nu, \mu, \rho, \varphi\) and \(\lambda\) can be obtained by solving D1. Particularly, if the values of all dual variables are optimal, the corresponding solutions are optimal. It is not difficult to see that the dual problem D1 can be equivalently decomposed into \(S\) subproblems with respect to each UD to facilitate parallel operation. By leveraging the Karush-Kuhn-Tucker (KKT) conditions [38] and setting the first-order derivatives of \(\mathcal{L}(\mathbf{L}, \mathbf{F}, \nu, \mu, \rho, \varphi, \lambda)\) with respect to \(f_{u,s}[n], L_{s,u}[n], \tilde{f}_{u,s}[n]\) and \(f_{s}[n]\) to zero, the corresponding optimal solutions can be easily obtained as in (16)-(19). Hence, Theorem 1 is proved.

APPENDIX B: PROOF OF LEMMA 1 AND LEMMA 2

Based on the multipliers \(\mu^{r+1}\) and \(\varphi^{r+1}\) obtained in Lemma 1, we can obtain \(\rho^{r+1}\), \(\varphi^{r+1}\) and \(\lambda^{r+1}\) accordingly. According to the constraints in (13d)–(13f) and the expressions in (16)–(19), the value of \(\sum_{n=1}^{N-1} L_{s,u,r+1}[n]\) is calculated as

\[
\sum_{n=1}^{N-1} L_{s,u,r+1}[n] = L_s \left[ T \right] \frac{\lambda^{r+1}}{C_s} \sqrt{\frac{3C_s \varphi_s \omega_s}{\lambda^{r+1} + \varphi^{r+1} + \lambda^{r+1} - \rho^{r+1}}} + \varphi^{r+1} \sqrt{3C_s \varphi_s \omega_s}.
\]

where \(\tilde{\nu}_{s,n} = \sum_{i=n=1}^{N-1} \nu_{s,i}, \tilde{\mu}_{s,n} = \sum_{i=n=1}^{N-1} \mu_{s,i}\) and \(\mu_{s,n} = \sum_{i=n=1}^{N-1} \mu_{s,i}\). It is worth noting that (43a) is derived from (13f), (43b) is given by (19), (43c) comes from (13d) and (13e) with \(\sum_{n=1}^{N-1} L_{s,u,r+1}[n] = \sum_{n=1}^{N-2} L_{s,u,r+1}[n] + \sum_{n=1}^{N-1} L_{s,u,r+1}[n]\) and (43d) is derived from (13d).

Considering that \(f_{s}[n] \geq 0\) and \(\sum_{n=1}^{N-1} L_{s,u,r+1}[n] = \{0, I_s\}\), the range of \(\lambda^{r+1} = [0, \lambda^{max}]\) for \(s \in S\) can be derived. From (43a)-(43c), we find that \(\rho^{r+1}\) is a non-decreasing function of \(\lambda^{r+1}\), and \(\varphi^{r+1}\) is a non-increasing function of \(\lambda^{r+1}\), which further verifies that (43d) is a non-decreasing function of \(\varphi^{r+1}\). Thus, with a given \(\lambda^{r+1} = [0, \lambda^{max}]\) and the obtained \(\mu^{r+1}\) and \(\nu^{r+1}\) in Lemma 1, the correspondingly \(\rho^{r+1}\) and \(\varphi^{r+1}\) can be derived from (43a)-(43c) by leveraging the bisection search method with \(\rho^{r+1} = [\rho^{r+1}_{low}, \rho^{r+1}_{up}]\) and \(\varphi^{r+1} = [\varphi^{r+1}_{low}, \varphi^{r+1}_{up}]\), where \(\rho^{r+1}_{low}, \rho^{r+1}_{up}, \varphi^{r+1}_{low}, \varphi^{r+1}_{up}\) and \(\varphi^{r+1}_{up}\) are respectively given by

\[
\rho^{r+1}_{low} = \tilde{\mu}_{s,n},
\]

\[
\rho^{r+1}_{up} = \tilde{\mu}_{s,n} N.
\]
\[ r_{s,1}^{\tau+1} = \rho_{s,1}^{\tau+1} + \lambda_{s}^{\max}, \] (44b) 
\[ r_{s,\text{up}}^{\tau+1} = \tilde{r}_{s,\text{up}}^{\tau+1} - 2^{\frac{\tau}{2^n}} \sum_{n=1}^{N} \theta_{s,n}[\eta], \] (44c) 
\[ \tilde{r}_{s,\text{up}}^{\tau+1} = \tilde{r}_{s,\text{up}}^{\tau} + 2^{\frac{\tau}{2^n}} \sum_{n=1}^{N} \phi_{s,n}[\eta], \] (44d) 

The optimal \( \lambda_{s}^{\tau+1}, \rho_{s}^{\tau+1}, \) and \( q_{s}^{\tau+1} \) obtained at \((r+1)\)th iteration should make \((43a) = (43d)\) hold, indicating the termination of the bisection search of \( \lambda_{s}^{\tau+1} \) for \( \lambda \in \mathcal{S} \).

REFERENCES

Jiequ Ji received the M.E. degree in 2017 from College of Computer Science and Technology, Anhui University of Science and Technology, Huainan, China. She is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From October 2018 to January 2020, she has been a visiting student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her research interests include UAV-enabled wireless communications, wireless content caching, resource allocation in 5G and beyond, mobile edge computing, and physical layer security.

Kun Zhu (M’16) received the Ph.D. degree from School of Computer Engineering, Nanyang Technological University, Singapore, in 2012. He was a research fellow with the Wireless Communications Networks and Services Research Group in University of Manitoba, Canada, from 2012 to 2015. He is currently a Professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He is also a Jiangsu specially appointed professor. His research interests include resource allocation in 5G, wireless virtualization, and self-organizing networks. He has published more than fifty technical papers and has served as TPC for several conferences. He won several research awards including IEEE WCNC 2019 Best paper awards, ACM China rising star chapter award.

Changyan Yi (S’16-M’18) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Manitoba, MB, Canada, in 2018. From September 2018 to August 2019, he worked as a Research Associate in University of Manitoba, MB, Canada. Since September 2019, he has been with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China, where he is currently a Professor. He was awarded Chinese Government Award for Outstanding Students Abroad in 2017, University of Manitoba Graduate Fellowship (UMGF) for 2015-2018 and IEEE ComSoc Student Travel Grant for IEEE Globecom 2016. His research interests include mechanism design, game theory, queueing theory and their applications in various wireless networks including edge/fog computing, IoT, 5G and beyond.

Dusit Niyato (M’09–SM’15–F’17) received the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published more than 400 technical articles in the area of wireless and mobile computing. He received the Best Young Researcher Award of the IEEE Communications Society Asia Pacifica and the 2011 IEEE Communications Society Fred W. Ellersick Prize Paper Award. He is also serving as a Senior Editor of the IEEE Wireless Communication Letters, an Area Editor of the IEEE Transactions on wireless Communications and the IEEE Communications Surveys and Tutorials, an Editor of the IEEE Transactions on Communications, and an Associate Editor of the IEEE Transactions on Mobile Computing, the IEEE Transactions on Vehicular Technology, and the IEEE Transactions on Cognitive Communications and Networking. He was a Distinguished Lecturer of the IEEE Communications Society from 2016 to 2017. He was named a highly cited researcher in computer science.