Joint Computation Offloading and Trajectory Planning for UAV-Assisted Edge Computing

Chao Sun, *Graduate Student Member, IEEE*, Wei Ni, *Senior Member, IEEE*, and Xin Wang, *Senior Member, IEEE*

***Abstract*— With excellent flexibility, unmanned aerial vehicles (UAVs) can act as airborne computing servers to assist smart terminals (STs) with their computationally-intense and delay-sensitive tasks. This paper presents a new UAV-assisted edge computing framework, which jointly optimizes the trajectory and CPU frequency of a fixed-wing UAV, and the offloading schedule to minimize the energy consumption of the UAV. The key idea is that we reveal the condition for the convexity of the optimization, when the UAV flies a linear trajectory. Under the condition, alternating optimization- and successive convex approximation (SCA)-based algorithms are developed to efficiently achieve the globally optimal linear trajectory, CPU configuration, and offloading schedule. Another important aspect is that we prove the SCA-based algorithm can achieve a local optimum satisfying the Karush-Kuhn-Tucker (KKT) conditions, when the revealed condition is unmet or the UAV flies horizontally in two dimensions. By analyzing the KKT conditions, we also unveil the underlying patterns for the optimal CPU frequency and offloading schedule. Extensive simulations validate the patterns and corroborate the merits of our schemes.**

***Index Terms*— Edge computing, UAV, computation offloading, trajectory planning, time allocation.**

I. Introduction

THE fruition of the Internet-of-Things (IoT) has promoted the widespread proliferation of smart terminals (STs). Take cognitive radio for example. Each ST can sense the frequency utilization, and use the sensed data to train a recurrent neural network (RNN) to forecast future white spaces [1], [2]. In many cases, the STs are resource-restrained in terms of computing and energy, and may experience difficulty in producing instantaneous response or maintaining endurance [3], [4]. By deploying roaming computing servers with powerful computing processors, the STs can offload their computing tasks for timely processing, thereby saving the energy of the STs and preventing their buffers from overflowing. With attractive merits of on-demand deployment and excellent mobility, unmanned aerial vehicles (UAVs) have given rise to many applications [5], such as mobile base stations [6], [7], mobile relays [8], [9], caching networks [10], surveillance and monitoring [11], and wireless power transmission [12]. Recently, UAVs equipped with powerful CPUs have been nominated to be the roaming computing servers. By properly planning its trajectory, a UAV can establish line-of-sight (LoS)-dominant communication links with the STs to reduce their energy consumption and increase offloading speed.

Existing studies on airborne computing servers can be divided into two classes, based on quasi-static UAVs or mobile UAVs. The former is driven by rotary-wing UAVs, while the latter can be driven by either rotary-wing or fixed-wing UAVs. In the case of quasi-static UAVs [13]–[17], the UAV hovers at a fixed location or periodically adjusts the hovering locations, with the onboard server as a quasi-static access point to support the STs in an area. In addition to task schedule and resources allocation, the horizontal location and altitude of the UAV, the hovering period, as well as the antenna beamwidth have been optimized for quality-of-service provision.

By taking advantage of the excellent mobility and maneuverability of the UAVs, the distance between a mobile UAV and STs can be shortened with proper trajectory planning, achieving significant signal strengths at the UAV. In [18] and [19], the energy consumption of the STs was minimized for their task offloading to a UAV. Tasks offloading (i.e., communication and computation resources) and UAV trajectory were jointly optimized in [20] to maximize the energy efficiency of the STs. In addition, the computing throughput was maximized in [21], and the total of the maximum delay among all STs over time was minimized in [22].

Energy management is critical to a mobile UAV-based computing platform. In [23], the CPU frequencies of a UAV and STs, the time resources, and the UAV trajectory were jointly optimized to minimize the energy consumption of the UAV in regards of delay-insensitive tasks. [24] minimized the energy consumption of a UAV with a wireless power transfer function by jointly optimizing the CPU frequencies of the UAV and STs, the offloading amount, power allocation, and the UAV trajectory. [25] maximized the energy efficiency of a UAV-mounted cloudlet by jointly optimizing the offloading amount, channel and power allocation, and the UAV trajectory.

Manuscript received September 5, 2020; revised January 21, 2021; accepted March 14, 2021. Date of publication March 25, 2021; date of current version August 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62071126 and in part by the Innovation Program of Shanghai Municipal Science and Technology Commission under Grant 20JC1416400. The associate editor coordinating the review of this article and approving it for publication was M. Kaneko. *(Corresponding author: Xin Wang.)*

Chao Sun and Xin Wang are with the Key Laboratory of EMW Information (MoE), Department of Communication Science and Engineering, Fudan University, Shanghai 200433, China (e-mail: xwang11@fudan.edu.cn).

Wei Ni is with the Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, NSW 2122, Australia (e-mail: wei.ni@data61.csiro.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3067163.

Digital Object Identifier 10.1109/TWC.2021.3067163

Both [26] and [27] minimized the total energy consumption of the STs and a UAV, where the UAV served as both a computing server and a relay for an access point. In [24] and [26], the STs adopted time-division multi-access (TDMA) protocols to communicate with the UAV (whereas frequency-division multi-access (FDMA) was adopted in [25] and [27]). However, the slots allocated to every ST were even and not optimized.

This paper presents a new mobile UAV-assisted edge computing framework, where a computing server mounted on a fixed-wing UAV assists the STs with their computing tasks. We minimize the UAV's energy consumption and guarantee the deadlines of the ST tasks, by jointly optimizing the trajectory and CPU frequency of the UAV, and the offloading and computing schedules for the STs. The aeronautic maneuverability of the UAV is captured with its velocities and accelerations specified. The key contributions of the paper are summarized as follows.

- A new system model is proposed, which comprehensively considers the aeronautics and CPU reconfigurability of the fixed-wing UAV, and different deadlines/timelines of the STs' tasks; and optimizes the UAV's trajectory (including velocities and accelerations), CPU frequency, and offloading durations and schedules for UAV energy minimization;
- Geometric condition is revealed for the convexity of the considered problem, when the UAV flies a one-dimensional (1D) flight path. Under the condition, alternating optimization- and successive convex approximation (SCA)-based algorithms are developed to achieve the globally optimal 1D trajectory, CPU configuration, offloading durations and schedules;
- The proposed SCA-based algorithm is proved to converge to a locally optimal solution satisfying the Karush-Kuhn-Tucker (KKT) conditions, when the geometric condition is unmet or the UAV flies a two-dimensional (2D) flight path;
- Two underlying patterns are analytically revealed: The optimal CPU frequency of the UAV only changes at time slots when a causality or deadline constraint is met with equality, and the STs with higher offloading demands are allocated with longer time than the others in any slots whenever they are served.

The rest of this paper is organized as follows. Section II introduces the system model. Section III formulates the UAV energy minimization problem, and develops two efficient algorithms based on alternating optimization and SCA. Section IV generalizes the proposed approach to 2D UAV trajectory planning. In Section V, extensive simulations are provided to evaluate the proposed schemes. Multiple UAVs and the rotary-wing UAV are discussed in Section VI. Finally, conclusions are drawn in Section VII.

II. System Model

Fig. 1 illustrates the considered scenario which consists of a UAV (acting as an airborne computing server) and K STs at fixed locations. Each ST has tasks to process. The location of ST_k is $(x_k,\ y_k,\ 0)$, $k \in \mathcal{K} = \{1, \cdots, K\}$. It is assumed

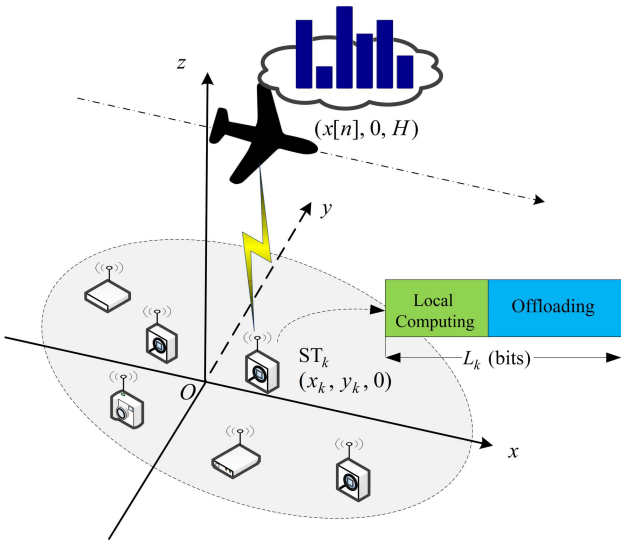

Fig. 1. A UAV-assisted edge computing system.

that the locations of the STs are known to the UAV [19], [21]. During a time period T, the UAV assists the STs with their computing tasks. We first consider that the UAV flies along a straight line at an altitude H. The extension to 2D UAV trajectories will be provided in Section IV. Without loss of generality (w.l.o.g.), we assume that the UAV flies from an initial location $(x_I,\ 0,\ H)$ to the final location $(x_F,\ 0,\ H)$ along the x-axis. For ease of exposition, T is split evenly into N time slots, denoted by $n \in \mathcal{N} = \{1, \cdots, N\}$. The slot duration $\delta = T/N$ is so short that the UAV location during a slot, i.e., $(x[n],\ 0,\ H)$, can be represented by a waypoint of the UAV.

A. Task Offloading Model

The channels between the UAV and STs are dominated by LoS links, as shown in field tests [28]. A free-space path loss model can be adopted with the channel gain between the UAV and ST_k at slot n, as given by [7], [29]

$$g_k[n] = \beta_0 d_k^{-2}(x[n]) = \frac{\beta_0}{H^2 + y_k^2 + (x[n] - x_k)^2}, \quad (1)$$

where β_0 is the reference channel gain at $d = 1$ m, and $d_k(x[n]) = \sqrt{H^2 + y_k^2 + (x[n] - x_k)^2}$ is the Euclidean distance between the UAV and ST_k at slot n. The generalization of our approach under a more generic channel model will be discussed in Section IV-D.

Any $\text{ST}_k, \forall k$, can have a different task size, denoted by L_k (in bits), and a different timeline for the task to be offloaded and processed within an interval $[t'_k,\ t''_k]$. The interval covers time slots $\{n'_k,\ n'_k + 1,\ \cdots,\ n''_k\}$. Part of the task which cannot be processed locally by ST_k in time is offloaded to the UAV to be completed before its deadline. We assume that the computing task is bit-wise independent and divisible, i.e., the STs work in the partial task offloading model [21], [30]. At a time slot, the UAV can be in a position accessible to multiple STs, and multiple STs with different task sizes and deadlines/timelines can be selected to offload tasks to the UAV. The UAV does not start processing the tasks until the

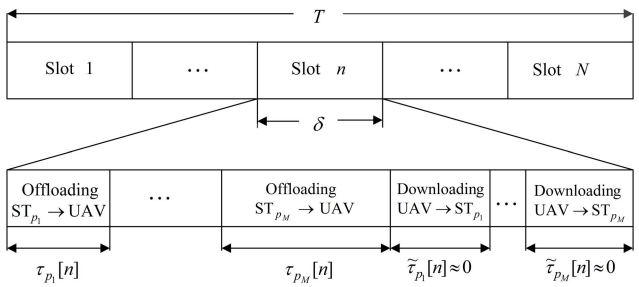

Fig. 2. An illustration of the TDMA protocol for STs computation offloading. $\mathrm{ST}_{p_1}, \cdots, \mathrm{ST}_{p_M}$ are terminals that can potentially offload tasks to the UAV at slot n.

next time slot when the CPU frequency of the UAV is adjusted to accommodate the tasks and deadlines [24]–[26].

As shown in Fig. 2, TDMA is adopted between the STs within a time slot. [1] As the computing results are usually comparatively negligible in size (cf., the tasks), it is reasonable to ignore the time to return the results, i.e., $\tilde{\tau}_k[n] \approx 0, \forall k \in \{p_1, \cdots, p_M\}$, and the corresponding energy consumption [27], [30]. The total offloading time of the STs cannot exceed the duration of the slot; i.e.,

$$\sum_{k \in \Theta_n} \tau_k[n] \leq \delta, \quad \forall n \in \mathcal{N}_1, \tag{2}$$

where $\Theta_n := \{p_1, \cdots, p_M\}$, and $\mathcal{N}_1 = \{1, \cdots, N-1\}$. Let $l_k[n]$ denote the amount of tasks (in bits) offloaded from ST_k to the UAV at slot n, then

$$l_k[n] \leq \tau_k[n] B \log_2 \left(1 + \frac{E_k g_k[n]}{\tau_k[n]\sigma^2}\right), \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}, \tag{3}$$

where B is the system bandwidth in Hertz (Hz); E_k is the emission energy of ST_k when offloading tasks to the UAV; σ^2 is the additive white Gaussian noise (AWGN) power at the UAV; and $\mathcal{M}_k = \{n'_k, \cdots, n''_k - 1\}$ collects the time slots that ST_k can potentially offload tasks to the UAV.

B. Computing Model

1) *Local Computing:* The processing density, denoted by c (in CPU cycles per bit), is the number of CPU cycles required to compute a bit of input data. The CPU frequency of the local processor at ST_k, denoted by f_{ST_k}, is assumed to be constant. Therefore, the number of bits computed locally by ST_k is given by

$$l_{\mathrm{ST}_k} = \frac{(t''_k - t'_k) f_{\mathrm{ST}_k}}{c}. \tag{4}$$

2) *Edge Computing:* The CPU of the UAV typically consumes the majority of data processing energy [31]. Assume that the CPU of the UAV adopts dynamic voltage and frequency scaling (DVFS) techniques [26], the UAV can improve its energy efficiency by adjusting the CPU frequency $f_{\mathrm{UAV}}[n]$ at each slot n. Hence, the number of computed bits and the computing energy at slot n are given respectively by [26]

$$l_{\mathrm{UAV}}[n] = \frac{\delta f_{\mathrm{UAV}}[n]}{c} \text{ and } E_{\mathrm{c}}[n] = \delta \kappa f^3_{\mathrm{UAV}}[n], \quad \forall n \in \mathcal{N}_2, \tag{5}$$

where $\mathcal{N}_2 = \{2, \cdots, N\}$, and κ is the effective capacitance coefficient [30].

C. The Propulsion Energy and Mobility Models of the UAV

A fixed-wing UAV is considered for its excellent flight endurance and payload (as compared to a rotary-wing UAV) [18], [29]. The propulsion energy of the UAV at slot n is modeled as [29]

$$E_{\mathrm{fly}}[n] = \delta \left(c_1 v[n]^3 + \frac{c_2}{v[n]} \left(1 + \frac{a[n]^2}{g^2}\right)\right), \quad \forall n \in \mathcal{N}_3, \tag{6}$$

where $\mathcal{N}_3 = \{0, 1, \cdots, N-1\}$; $v[n]$ and $a[n]$ denote the velocity and acceleration of the UAV at slot n, respectively; c_1 and c_2 are two known coefficients depending on the wing surface area, wing span efficiency, and the mass of the UAV; and $g = 9.8$ m/s^2 is the gravitational acceleration.

In practice, the UAV's location, velocity, and acceleration need to meet the mobility constraints:

$$x[0] = x_I, \quad x[N] = x_F, \tag{7}$$

$$v[0] = v_I, \quad v[N] = v_F, \tag{8}$$

$$x[n+1] = x[n] + v[n]\delta + \frac{1}{2} a[n] \delta^2, \quad \forall n \in \mathcal{N}_3, \tag{9}$$

$$v[n+1] = v[n] + a[n]\delta, \quad \forall n \in \mathcal{N}_3, \tag{10}$$

$$v_{\min} \leq v[n] \leq v_{\max}, \quad \forall n \in \mathcal{N}_1, \tag{11}$$

$$|a[n]| \leq a_{\max}, \quad \forall n \in \mathcal{N}_3, \tag{12}$$

where v_I and v_F are the initial and final velocities; $v_{\max}$ and $a_{\max}$ are the maximum speed and acceleration; and $v_{\min}$ is the stall speed. The UAV would fall if $v[n] < v_{\min}$. Also, $v_{\min} T \leq |x_F - x_I| \leq v_{\max} T$, as the UAV needs to fly from $(x_I, 0, H)$ to $(x_F, 0, H)$ within the T seconds.

III. Minimization of UAV's Energy Consumption With 1D Trajectory Planning

In this section, we minimize the energy consumption of the UAV with a 1D linear trajectory, while ensuring the deadlines of all STs' tasks. We analyze the convexity of the problem and prove that the problem is conditionally convex when the altitude of the UAV and the locations of STs satisfy a specific condition. Consider both the convex and non-convex cases. Two iterative algorithms based on alternating optimization and SCA are developed.

As will be discussed, the propulsion energy of the UAV (6) can be convex in variables $v[n]$ and $a[n]$, and the data transmission constraint is conditionally convex in UAV location $x[n]$ for the 1D trajectories; whereas both of them are non-convex for 2D trajectories. These differences can affect the algorithm optimality. The consideration of 1D trajectories also help shed insights on the patterns of the optimal CPU frequency and offloading schedules. Moreover, 1D trajectories can have practical applications. For example, UAVs can be employed

[1] The time allocation under the TDMA setting can be translated to the allocation of bandwidth if FDMA is adopted in the air-to-ground channels which are typically dominated by strong LoS paths and experience little frequency selectivity.

to fly along highways, railways, or powerlines, to provide computing services for infrastructure monitoring and fatigue modeling [32]. A UAV assigned for goods delivery may also provide computing services to the ground STs deployed along the flight path between the source and destination.

A. Problem Formulation

Let $\boldsymbol{f} = \{f_{\text{UAV}}[n], \forall n \in \mathcal{N}_2\}$, $\boldsymbol{l} = \{l_k[n], \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}\}$, $\boldsymbol{\tau} = \{\tau_k[n], \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}\}$, and $\boldsymbol{Z} = \{x[n], v[n], a[n], \forall n \in \mathcal{N}_3\}$. While ensuring that the STs' tasks are processed in time, we wish to minimize the UAV's energy consumption accounting for both computing and propulsion through joint optimization of the CPU frequency of the UAV, the offloading volume, offloading time, and the UAV's trajectory. The problem is formulated as[2]

$$\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Z}} \sum_{n=2}^{N} E_{\text{c}}[n] + \sum_{n=0}^{N-1} E_{\text{fly}}[n] \tag{13a}$$

$$\text{s.t.} \quad (2), (7) - (12),$$

$$\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c} \le \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i], \quad \forall n \in \mathcal{N}_2, \tag{13b}$$

$$\sum_{k\in\Omega_n} \sum_{i=n'_k}^{n''_k-1} l_k[i] \le \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c}, \quad \forall n \in \mathcal{N}_2, \tag{13c}$$

$$l_{\text{ST}_k} + \sum_{i=n'_k}^{n''_k-1} l_k[i] = L_k, \quad \forall k \in \mathcal{K}, \tag{13d}$$

$$\tau_k[n]\left(2^{\frac{l_k[n]}{\tau_k[n]B}} - 1\right) \le \frac{E_k \beta_0}{\left(H^2 + y_k^2 + (x[n] - x_k)^2\right)\sigma^2}, \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}, \tag{13e}$$

$$f_{\text{UAV}}[n] \ge 0, \quad \forall n \in \mathcal{N}_2, \tag{13f}$$

$$l_k[n] \ge 0, \tau_k[n] \ge 0, \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}, \tag{13g}$$

where Ω_n is the set of STs with task deadlines no later than slot n; (13b) preserves causality, i.e., the size of tasks processed by slot n, $\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c}$ (in bits), must not exceed the size received by the previous slots, i.e., $\sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i]$ (in bits); (13c) is the deadline constraint, i.e., $\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c}$ must be no smaller than the tasks (in bits) computed by their deadlines, i.e., $\sum_{k\in\Omega_n} \sum_{i=n'_k}^{n''_k-1} l_k[i]$; (13c) and (13d) guarantee that the STs' tasks can be completed by their deadlines; and (13e) provides the data transmission constraint based on (3).

The objective of (13) is convex, and all its constraints are affine except (13e). In (13e), $2^{l_k[n]}$ is convex in $l_k[n]$, and $\tau_k[n]2^{\frac{l_k[n]}{\tau_k[n]}}$ is its perspective which is a jointly convex function of $\{l_k[n], \tau_k[n]\}$ [33]. Since a non-negatively weighted summation preserves convexity, the left-hand side (LHS) of (13e) is convex. The right-hand side (RHS) of (13e) is generally non-concave in $x[n]$. Thus, problem (13) is in general non-convex.

[2] E_c grows cubically with the CPU frequency, and can contribute considerably to the energy cost, especially when the offloading demands are high. Therefore, both E_{fly} and E_c are considered, as done in [23]–[27].

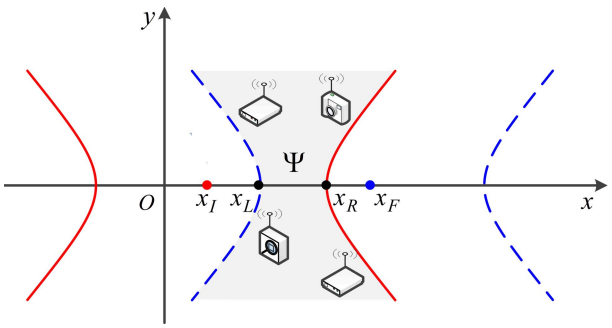

Fig. 3. An illustration of the geometric condition. The region between the solid curves is $S(x_I)$, and between the dashed curves is $S(x_F)$. The shaded part is Ψ, and its left and right intersections with x-axis are $x_L = x_F - \frac{H}{\sqrt{3}}$ and $x_R = x_I + \frac{H}{\sqrt{3}}$, respectively.

Next, by checking the second-order convexity condition of the RHS of (13e), we show that problem (13) becomes convex under the condition stated in the following lemma.

Lemma 1: Problem (13) *is convex in the flight range from* $(x_I, 0, H)$ *to* $(x_F, 0, H)$ *if and only if*

$$H^2 \ge 3(x_k - x[n])^2 - y_k^2, \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}. \tag{14}$$

Proof: See Appendix A.

Lemma 1 implies that (13) is convex if the location of a ST which offloads computations at slot n, is inside the following area:

$$S(x[n]) := \left\{ (x, y) \middle| \frac{(x - x[n])^2}{H^2/3} - \frac{y^2}{H^2} \le 1 \right\}. \tag{15}$$

The area is specified by a hyperbola. Given that the UAV flies along the straight line, (14) holds and the convexity of (13) is guaranteed, if

$$(x_k, y_k) \in \Psi := S(x_I) \cap S(x_F), \quad \forall k \in \mathcal{K}. \tag{16}$$

Fig. 3 visualizes the condition for the convexity of (13). The higher the altitude H is, the larger the area Ψ covers. In other cases where (14) does not hold, problem (13) is non-convex.

B. Proposed Solutions

We develop two iterative algorithms to solve problem (13) when it is convex and non-convex.

1) Alternating Optimization Algorithm: With the STs inside the region specified by (16), condition (14) is satisfied and problem (13) is convex. However, standard convex optimization solvers, such as MATLAB CVX, cannot recognize (13), because constraint (13e) is non-convex in $x[n]$ across the whole x-axis. We propose to alternately solve the subproblems of computation offloading and trajectory planning until convergence.

Given fixed $\boldsymbol{Z}$, (13) can be transformed into a computation offloading subproblem of the CPU frequency, offloading task, and time allocation, as given by

$$\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau}} \sum_{n=2}^{N} E_{\text{c}}[n] \quad \text{s.t.} \quad (2), (13\text{b}) - (13\text{g}). \tag{17}$$

Algorithm 1 Alternating Optimization Algorithm for Solving Problem (13)

1: **Initialize** $\{x^{(0)}[n], \forall n\}$ and iteration number $j = 0$.
2: **repeat**
3: Solve problem (17) at the given $\boldsymbol{Z}^{(j)}$, and obtain the optimal solution $\boldsymbol{f}^{(j)}, \boldsymbol{l}^{(j)}, \boldsymbol{\tau}^{(j)}$.
4: Solve problem (18) at the given $\boldsymbol{f}^{(j)}, \boldsymbol{l}^{(j)}, \boldsymbol{\tau}^{(j)}$, and obtain the optimal solution $\boldsymbol{Z}^{(j+1)}$.
5: Update $j = j + 1$.
6: **until** convergence.

Given fixed $\{\boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\tau}\}$, (13) can be transformed into a trajectory planning subproblem of the UAV's location, velocity, and acceleration, as given by

$$\min_{\boldsymbol{Z}} \sum_{n=0}^{N-1} E_{\text{fly}}[n] \tag{18a}$$

$$\text{s.t.} \quad (7)-(12),$$

$$\left(2^{\frac{l_k[n]}{\tau_k[n]B}} - 1\right)\left(H^2 + y_k^2 + (x[n] - x_k)^2\right) \leq \frac{E_k \beta_0}{\tau_k[n]\sigma^2}, \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}, \tag{18b}$$

where (18b) is implied by (13e), providing a quadratic convex constraint for variables $x[n]$.

The alternating optimization algorithm is summarized in Algorithm 1. In the case where problem (13) is convex, Algorithm 1 converges to the global optimum of the problem. Both problems (17) and (18) are convex, and can be efficiently solved to monotonically reduce the energy consumption of the UAV. On the other hand, the energy consumption of the UAV is lower bounded. It can then be inferred that Algorithm 1 is guaranteed to converge. Given the convexity of (13), the stationary point of the problem is the global optimum.

The computational complexity of Algorithm 1 is dominated by the interior point method used to solve (17) and (18) during each iteration. The worst-case complexity of the interior point method is $\mathcal{O}(\max\{N_c, N_v\}^4\sqrt{N_v})$, where N_c and N_v are the total numbers of constraints and variables, respectively [33], [34]. For problem (17), $N_c \leq 3NK + 4N + K - 4$, accounting for $(N-1)$ affine constraints in (2), $(N-1)$ linear constraints in (13b), $(N-1)$ linear constraints in (13c), K affine constraints in (13d), at most NK non-linear constraints in (13e), $(N-1)$ linear constraints in (13f), and at most $2NK$ linear constraints in (13g); and $N_v \leq 2NK + N - 1$, accounting for $(N-1)$ scalar variables in $\boldsymbol{f}$, at most NK scalar variables in $\boldsymbol{l}$, and at most NK scalar variables in $\boldsymbol{\tau}$. Therefore, the complexity of solving (17) is $\mathcal{O}((NK)^{4.5})$. For problem (18), $N_c \leq NK + 5N + 2$, accounting for $(5N+2)$ linear constraints in (7) – (12), and at most NK quadratic constraints in (18b); and $N_v = 3N$, accounting for $3N$ scalar variables in $\boldsymbol{Z}$. Therefore, the complexity of solving (18) is $\mathcal{O}((NK)^4\sqrt{N})$. As a result, the complexity of Algorithm 1 is $\mathcal{O}((NK)^{4.5})$ per iteration. Suppose that the convergence accuracy of the algorithm is ϵ. The overall complexity of Algorithm 1 is $\mathcal{O}((NK)^{4.5}\log(1/\epsilon))$.

2) SCA-Based Algorithm: In the case where condition (14) is unsatisfied, problem (13) is non-convex due to the non-convexity of (13e). To develop an efficient solver, we employ the SCA technique. At each iteration, (13e) is approximated by a convex constraint at given local points $\{x^{(j)}[n], \forall n\}$, where the superscript $^{(j)}$ indicates the j-th iteration of SCA.

Algorithm 2 SCA-Based Algorithm for Solving Problem (13)

1: **Initialize** $\{x^{(0)}[n], \forall n\}$ and iteration number $j = 0$.
2: **repeat**
3: Solve problem (21) at the given local points $\{x^{(j)}[n], \forall n\}$, and obtain the optimal solutions to $\boldsymbol{f}^{(j+1)}$, $\boldsymbol{l}^{(j+1)}, \boldsymbol{\tau}^{(j+1)}$, and $\boldsymbol{Z}^{(j+1)}$.
4: Update $j = j + 1$.
5: **until** convergence.

An important property is that, although the RHS of (13e) is not concave in $x[n]$, it is convex and differentiable with respect to $(x[n] - x_k)^2$. The first-order Taylor expansion of a convex function at any local point is its global lower bound [33]. With the given UAV trajectory $\{x^{(j)}[n], \forall n\}$ obtained at the j-th iteration, we can take the first-order Taylor expansion of the RHS of (13e) with respect to $(x[n]-x_k)^2$, as the desired lower bound; i.e.,

$$\frac{E_k\beta_0}{(H^2 + y_k^2 + (x[n] - x_k)^2)\sigma^2} \geq \frac{E_k\beta_0}{\sigma^2} \times \frac{H^2 + y_k^2 + 2(x^{(j)}[n] - x_k)^2 - (x[n] - x_k)^2}{(H^2 + y_k^2 + (x^{(j)}[n] - x_k)^2)^2} := \Lambda_k(x[n]). \tag{19}$$

Since $\Lambda_k(x[n])$ is concave in $x[n]$, (13e) can be approximated by the following convex form:

$$\tau_k[n]\left(2^{\frac{l_k[n]}{\tau_k[n]B}} - 1\right) \leq \Lambda_k(x[n]), \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}. \tag{20}$$

By replacing (13e) with (20), problem (13) can be approximated as

$$\min_{\boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\tau}, \boldsymbol{Z}} \sum_{n=2}^{N} E_{\text{c}}[n] + \sum_{n=0}^{N-1} E_{\text{fly}}[n] \tag{21}$$

$$\text{s.t.} \quad (2), (7)-(12), (13\text{b})-(13\text{d}), (13\text{f}), (13\text{g}), (20),$$

which is convex and can be solved by using the standard CVX tools. The value of $\{x^{(j+1)}[n], \forall n\}$ in the optimal solution to (21) are used for the local points in the next iteration of SCA.

The SCA-based algorithm is summarized in Algorithm 2. In the case where problem (13) is non-convex, Algorithm 2 converges to a solution satisfying the KKT conditions [35], [36]. Note that the non-convex constraint (13e) is approximated with the convex constraint (20). The RHSs of (13e) and (20) satisfy $\frac{E_k\beta_0}{(H^2+y_k^2+(x[n]-x_k)^2)\sigma^2} \geq \Lambda_k(x[n])$; and they also have the same value $\frac{E_k\beta_0}{(H^2+y_k^2+(x^{(j)}[n]-x_k)^2)\sigma^2}$ and the same gradient $\frac{-2E_k\beta_0(x^{(j)}[n]-x_k)}{\sigma^2(H^2+y_k^2+(x^{(j)}[n]-x_k)^2)^2}$ at the local point $x[n] = x^{(j)}[n], \forall n, j$. On the other hand, problem (21) is convex and satisfies the Slater's condition; i.e., there exists such a point in the domain of (21) that all non-linear constraints are satisfied with inequality. According to [34,

Thm 1], Algorithm 2 stops at a KKT point of problem (13), which is a local minimum if it in the relative interior of the feasible set of (21). In Algorithm 2, the interior point method is used to solve problem (21), where $N_c \leq 3NK+9N+K-2$ and $N_v \leq 2NK+4N-1$, as analyzed in the same way as those in Algorithm 1. As a result, the complexity of Algorithm 2 is $\mathcal{O}((NK)^{4.5}\log(1/\epsilon))$.

Remark 1: When (13) *is convex (i.e., when* (14) *holds),* (13) *can also be solved by Algorithm 2. Recall that any point satisfying the KKT conditions of a convex problem is its global minimizer [33]. Algorithm 2 can converge to the globally optimal solution. On the other hand, when* (13) *is non-convex, Algorithm 1 can also apply. Given that both subproblems* (17) *and* (18) *are convex, they can be efficiently solved to monotonically decrease the energy consumption of the UAV. With this monotonicity, it is easy to see that Algorithm 1 converges. However, Algorithm 1 may not converge to a (local) optimum for* (13) *[37]. As will be revealed in Section V, in the non-convex case, the solution of Algorithm 2 satisfying the KKT conditions can significantly outperform the sub-optimal solution yielded by Algorithm 1.*

C. Structural Properties

We analyze the structural properties of the optimal solution to shed insights on the optimal computation offloading and trajectory planning of the considered UAV-assisted edge computing. We reveal the following pattern of the optimal CPU frequency of the UAV.

Proposition 1: In the optimal solution to (13)*, the CPU frequency of the UAV only changes at the slots when the causality constraint* (13b) *or the deadline constraint* (13c) *is met with equality. The CPU frequency increases after a slot n when $\sum_{i=2}^{n}\frac{\delta f_{UAV}[i]}{c}=\sum_{i=1}^{n-1}\sum_{k=1}^{K}l_k[i]$, and decreases after a slot n when $\sum_{k\in\Omega_n}\sum_{i=n'_k}^{n-1}l_k[i]=\sum_{i=2}^{n}\frac{\delta f_{UAV}[i]}{c}$.*

Proof: See Appendix B.

Proposition 1 states that the optimal CPU frequency of the UAV follows a "string tautening" pattern like the data transmission policies under energy (or data) causality and deadline constraints in [38]–[40], as will be depicted in Fig. 4(a). Since $\sum_{n=2}^{N}E_{\mathrm{c}}[n]$ is convex in $\boldsymbol{f}$, a constant CPU frequency should be maintained whenever possible to minimize the computing energy. A causality constraint is active if the UAV completes all offloaded tasks. As a result, the CPU frequency is lower before slot n than after. Similarly, a deadline constraint is active if the tasks of a ST are completed by its deadline. The CPU frequency is higher before slot n than after.

The optimal time allocation of the STs also follows a pattern, as will be depicted in Fig. 4(b). Specifically, the STs with high offloading demands are allocated with more time than the others in any slots whenever they are served. This is because the RHS of (3) is a monotonically increasing function of $\tau_k[n]$. When ST_k has high offloading demands and offloads at slot n, its allocated time $\tau_k[n]$ needs to be large to ensure that constraint (13e) is met. According to (2), the allocated time of the STs with low offloading demands is small or even none at the slot. To ensure the deadlines of those STs' tasks, the UAV offers long offloading time in other time slots.

IV. Generalizations to 2D Trajectory and Online Operation

To further exploit the flexibility and mobility of the UAV, we generalize the proposed algorithms to optimize a 2D trajectory together with computation offloading.

A. Problem Formulation

Consider a 2D UAV flight trajectory on a horizontal plane at the altitude H. $\mathbf{q}[n]=(x[n],\,y[n])$ collects the UAV's coordinates on the horizontal plane at slot n. $\mathbf{q}_k=(x_k,\,y_k)$ collects the ST_k's coordinates on the ground. The channel gain between the UAV and ST_k at slot n is given by [7], [29]

$$g_k[n]=\beta_0 d_k^{-2}(\mathbf{q}[n])=\frac{\beta_0}{H^2+\|\mathbf{q}[n]-\mathbf{q}_k\|^2}, \tag{22}$$

where $d_k(\mathbf{q}[n])=\sqrt{H^2+\|\mathbf{q}[n]-\mathbf{q}_k\|^2}$ is the Euclidean distance between the UAV and ST_k at slot n, and $\|\cdot\|$ stands for Euclidean norm.

With reference to (6), the propulsion energy of the UAV at slot n can be modeled as [29]

$$E_{\mathrm{fly}}[n]=\delta\bigg(c_1\|\mathbf{v}[n]\|^3+\frac{c_2}{\|\mathbf{v}[n]\|}\Big(1+\frac{\|\mathbf{a}[n]\|^2}{g^2}\Big)\bigg),\ \forall n\in\mathcal{N}_1. \tag{23}$$

where $\mathbf{v}[n]=(v_x[n],\,v_y[n])$ and $\mathbf{a}[n]=(a_x[n],\,a_y[n])$ are the velocity and acceleration of the UAV at slot n, respectively.

Let $\boldsymbol{Q}=\{\mathbf{q}[n],\mathbf{v}[n],\mathbf{a}[n],\forall n\in\mathcal{N}_3\}$. To minimize the energy consumption of the UAV, the joint optimization of the CPU frequency of the UAV, the amount of offloaded tasks, offloading time, and the 2D trajectory of the UAV, is formulated as

$$\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q}}\ \sum_{n=2}^{N}\delta\kappa f_{\mathrm{UAV}}^3[n]+\sum_{n=0}^{N-1}\delta\bigg(c_1\|\mathbf{v}[n]\|^3+\frac{c_2}{\|\mathbf{v}[n]\|}\Big(1+\frac{\|\mathbf{a}[n]\|^2}{g^2}\Big)\bigg) \tag{24a}$$

$$\text{s.t.}\quad (2),(13\text{b})-(13\text{d}),(13\text{f}),(13\text{g}),$$

$$\mathbf{q}[0]=\mathbf{q}_I,\quad \mathbf{q}[N]=\mathbf{q}_F, \tag{24b}$$

$$\mathbf{v}[0]=\mathbf{v}_I,\quad \mathbf{v}[N]=\mathbf{v}_F, \tag{24c}$$

$$\mathbf{q}[n+1]=\mathbf{q}[n]+\mathbf{v}[n]\delta+\frac{1}{2}\mathbf{a}[n]\delta^2,\ \forall n\in\mathcal{N}_3, \tag{24d}$$

$$\mathbf{v}[n+1]=\mathbf{v}[n]+\mathbf{a}[n]\delta,\quad \forall n\in\mathcal{N}_3, \tag{24e}$$

$$\|\mathbf{a}[n]\|\leq a_{\max},\quad \forall n\in\mathcal{N}_3, \tag{24f}$$

$$\|\mathbf{v}[n]\|\leq v_{\max},\quad \forall n\in\mathcal{N}_1, \tag{24g}$$

$$v_{\min}\leq\|\mathbf{v}[n]\|,\quad \forall n\in\mathcal{N}_1, \tag{24h}$$

$$\tau_k[n]\Big(2^{\frac{l_k[n]}{\tau_k[n]B}}-1\Big)\leq\frac{E_k\beta_0}{(H^2+\|\mathbf{q}[n]-\mathbf{q}_k\|^2)\sigma^2},\quad \forall n\in\mathcal{M}_k,\ \forall k\in\mathcal{K}, \tag{24i}$$

where $\mathbf{q}_I$ and $\mathbf{q}_F$ are the initial and final locations of the UAV; $\mathbf{v}_I$ and $\mathbf{v}_F$ are the initial and final velocities of the UAV; (24b) – (24h) are generalized from (7) – (12); and

(24i) is the data transmission constraint generalized from (13e). Since the objective and constraints (24h) and (24i) are non-convex, problem (24) is non-convex. To solve it, we generalize the SCA-based Algorithm 2 to find a (locally) optimal computation offloading schedule and 2D trajectory.

B. SCA-Based Convexification and Solution

To tackle the non-convexity of the objective, we reformulate (24) by introducing slack variables $\boldsymbol{w} := \{w_n, \forall n\}$, as given by

$$\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q},\boldsymbol{w}} \sum_{n=2}^{N} \delta\kappa f_{\text{UAV}}^3[n] + \sum_{n=0}^{N-1} \delta\bigg(c_1\|\mathbf{v}[n]\|^3 + \frac{c_2}{w_n}\Big(1+\frac{\|\mathbf{a}[n]\|^2}{g^2}\Big)\bigg) \tag{25a}$$

$$\text{s.t.} \quad (2), (13b)-(13d), (13f), (13g), (24b)-(24g), (24i),$$

$$w_n^2 \le \|\mathbf{v}[n]\|^2, \quad \forall n \in \mathcal{N}_1, \tag{25b}$$

$$v_{\min} \le w_n, \quad \forall n \in \mathcal{N}_1. \tag{25c}$$

In the optimal solution for (25), w_n is equal to $\|\mathbf{v}[n]\|, \forall n$, or we can increase w_n to reduce the objective. Hence, problem (25) is equivalent to (24). The objective of problem (25) becomes convex. Constraint (24i) and the new constraint (25b) are non-convex. In light of Section III, (24i) and (25b) can be effectively tackled with the SCA and approximated by convex constraints at given local points $\{\mathbf{q}^{(j)}[n], \mathbf{v}^{(j)}[n], \forall n\}$.

The RHS of (24i) is a convex and differentiable function of $\|\mathbf{q}[n]-\mathbf{q}_k\|^2$. With the given UAV trajectory $\{\mathbf{q}^{(j)}[n], \forall n\}$ obtained at the j-th iteration, we can take the first-order Taylor expansion of the RHS of (24i) with respect to $\|\mathbf{q}[n]-\mathbf{q}_k\|^2$, as the desired lower bound; i.e.,

$$\begin{aligned}&\frac{E_k\beta_0}{\big(H^2+\|\mathbf{q}[n]-\mathbf{q}_k\|^2\big)\sigma^2}\\ &\quad\ge \frac{E_k\beta_0}{\sigma^2}\frac{H^2+2\|\mathbf{q}^{(j)}[n]-\mathbf{q}_k\|^2-\|\mathbf{q}[n]-\mathbf{q}_k\|^2}{(H^2+\|\mathbf{q}^{(j)}[n]-\mathbf{q}_k\|^2)^2}\\ &\quad:= \Phi_k(\mathbf{q}[n]).\end{aligned} \tag{26}$$

Since $\Phi_k(\mathbf{q}[n])$ is concave in $\mathbf{q}[n]$, (24i) can be approximated by the following convex form:

$$\tau_k[n]\Big(2^{\frac{l_k[n]}{\tau_k[n]B}}-1\Big) \le \Phi_k(\mathbf{q}[n]), \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}. \tag{27}$$

The RHS of (25b) is convex and differentiable in $\mathbf{v}[n]$. Its lower bound can be obtained by using the first-order Taylor expansion at the local point $\mathbf{v}^{(j)}[n]$ obtained at the j-th iteration, i.e.,

$$\begin{aligned}\|\mathbf{v}[n]\|^2 \ &\ge\ \|\mathbf{v}^{(j)}[n]\|^2 + 2{\mathbf{v}^{(j)}}^T[n](\mathbf{v}[n]-\mathbf{v}^{(j)}[n])\\ &:= \Upsilon(\mathbf{v}[n]),\end{aligned} \tag{28}$$

where $(\cdot)^T$ stands for transpose. Since $\Upsilon(\mathbf{v}[n])$ is linear in $\mathbf{v}[n]$, (25b) can be approximated by the following convex form:

$$w_n^2 \le \Upsilon(\mathbf{v}[n]), \quad \forall n \in \mathcal{N}_1. \tag{29}$$

Algorithm 3 SCA-Based Algorithm for Solving Problem (25)

1: **Initialize** $\{\mathbf{q}^{(0)}[n], \mathbf{v}^{(0)}[n], \forall n\}$ and iteration number $j=0$.
2: **repeat**
3: Solve problem (30) at the given local points $\{\mathbf{q}^{(j)}[n], \mathbf{v}^{(j)}[n], \forall n\}$, and obtain the optimal solutions to $\boldsymbol{f}^{(j+1)}$, $\boldsymbol{l}^{(j+1)}$, $\boldsymbol{\tau}^{(j+1)}$, $\boldsymbol{Q}^{(j+1)}$, and $\boldsymbol{w}^{(j+1)}$.
4: Update $j = j+1$.
5: **until** convergence.

By replacing (24i) with (27) and (25b) with (29), the approximate problem of (25) is given by

$$\begin{aligned}\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q},\boldsymbol{w}} \ &\sum_{n=2}^{N} \delta\kappa f_{\text{UAV}}^3[n] + \sum_{n=0}^{N-1} \delta\bigg(c_1\|\mathbf{v}[n]\|^3 + \frac{c_2}{w_n}\Big(1+\frac{\|\mathbf{a}[n]\|^2}{g^2}\Big)\bigg)\\ \text{s.t.} \ &(2), (13b)-(13d), (13f), (13g),\\ &(24b)-(24g), (25c), (27), (29),\end{aligned} \tag{30}$$

which is convex and can be solved by using the standard CVX tools. The value of $\{\mathbf{q}^{(j+1)}[n], \mathbf{v}^{(j+1)}[n], \forall n\}$ in the optimal solution to (30) are used for the local points in the next iteration of SCA.

The SCA-based algorithm solving (25) is summarized in Algorithm 3. The algorithm converges to a solution satisfying the KKT conditions of problem (25) [35], [36]. Specifically, the non-convex constraint (24i) are approximated with the convex constraint (27). The RHSs of (24i) and (27) satisfy $\frac{E_k\beta_0}{(H^2+\|\mathbf{q}[n]-\mathbf{q}_k\|^2)\sigma^2} \ge \Phi_k(\mathbf{q}[n])$; and they also have the same value $\frac{E_k\beta_0}{(H^2+\|\mathbf{q}^{(j)}[n]-\mathbf{q}_k\|^2)\sigma^2}$ and the same gradient $\frac{-2E_k\beta_0(\mathbf{q}^{(j)}[n]-\mathbf{q}_k)}{\sigma^2(H^2+\|\mathbf{q}^{(j)}[n]-\mathbf{q}_k\|^2)^2}$ at the local point $\mathbf{q}[n] = \mathbf{q}^{(j)}[n], \forall n, j$. Likewise, the non-convex constraint (25b) are approximated with the convex constraint (29). The RHSs of (25b) and (29) satisfy $\|\mathbf{v}[n]\|^2 \ge \Upsilon(\mathbf{v}[n])$; and they also have the same value $\|\mathbf{v}^{(j)}[n]\|^2$ and the same gradient $2\mathbf{v}^{(j)}$ at the local point $\mathbf{v}[n] = \mathbf{v}^{(j)}[n], \forall n, j$. On the other hand, problem (30) is convex and satisfies the Slater's condition; i.e., there exists such a point in the domain of (30) that all non-linear constraints are satisfied with inequality. According to [34, Thm 1], Algorithm 3 stops at a KKT point of problem (25), which is a local minimum if it is in the relative interior of the feasible set of (30). In Algorithm 3, the interior point method is used to solve problem (30), where $N_c \le 3NK+13N+K-4$ and $N_v \le 2NK+8N-1$, as analyzed in the same way as those in Algorithm 1. As a result, the complexity of Algorithm 3 is $\mathcal{O}((NK)^{4.5}\log(1/\epsilon))$.

Note that the computing energy $\sum_{n=2}^{N} E_{\text{c}}[n]$, constraints (13b), (13c) and (13f) are all about variables $\boldsymbol{f}$ and exist in both of problems (13) and (24). They shape the optimal CPU frequency of the UAV. Therefore, the pattern of the CPU frequency revealed under the 1D trajectory in Section III-C remains valid under the 2D trajectories. Moreover, constraint (2) exists in both of problems (13) and (24). Given the analogy of (13e) and (24i), we can confirm that the optimal time allocation pattern revealed in Section III-C is also valid under the 2D trajectories.

Algorithm 4 Online Implementation of the Energy Minimization of the UAV

Let $\tilde{n}$ denote the current slot and $S_{\tilde{n}}$ denote the set of STs being currently scheduled.
while $\tilde{n} < N$ **do**
  if a new ST sends the computing request at the current slot $\tilde{n}$ **then**
    Add the ST to $\mathcal{S}_{\tilde{n}}$, and update the sizes of tasks yet to be satisfied and their timelines.
    Run Algorithm 1, 2 or 3 to update the computing and offloading schedules, and the CPU frequency and trajectory of the UAV for slots $\{\tilde{n}+1, \tilde{n}+2, \cdots N\}$.
  end if
  Update $\tilde{n} = \tilde{n} + 1$.
end while

C. Online Implementation of the Proposed Schemes

Algorithms 1 to 3 entail the offline optimization of the trajectory and CPU frequency of the UAV, and the task offloading schedule and volume, when the locations, task sizes and execution intervals of the STs are known a-priori. The algorithms can operate online, where the UAV optimizes on-the-fly based on the requests of the STs. When an ST needs the assistance of the UAV in computing a task, it sends a request to the UAV to specify the size and timeline of the task. Upon the receipt of the request, the UAV updates its list of requests (including the earlier unsatisfied requests and their timelines, and the new one); and produces the new offloading and computing schedules, and the trajectory and CPU frequency of the UAV by running the proposed algorithms, i.e., Algorithms 1 to 3. The UAV can notify the STs of the schedules, and then proceed as scheduled until another new computing request is received at the UAV.

The online implementation is summarized in Algorithm 4. To reserve enough maneuver time for the UAV, we assume that the STs report their task sizes, timelines, and locations by t_a time slots ahead of their execution intervals. When t_a is large enough, the performance of the online implementation can approach that of the offline, whereas the latter offers the (global or local) optimum, as discussed in Sections III-B and IV-B.

D. Extension Under Probabilistic Path Loss Model

The proposed algorithms can be generalized under a more generic probabilistic path loss model to taking into account the impact of the UAV's altitude on large- and small-scale fading. The probability of an LoS UAV-ST_k link is given by [11], [41]

$$p_{\text{LoS},k}(\mathbf{q}[n]) = \frac{1}{1 + C\exp(-D(\theta_k[n] - C))}, \quad \forall n, \quad (31)$$

where C and D are two constants depending on the propagation environment; and $\theta_k[n] = \frac{180}{\pi}\arcsin(H/d_k(\mathbf{q}[n]))$ is the elevation angle in degree and it depends on the UAV's altitude and distance to ST_k.

Let $\gamma_k[n]$ denote the small-scale fading, which is a complex random variable with $\mathbb{E}\{|\gamma_k[n]|^2\} = 1$. Also let $\beta_k(\mathbf{q}[n])$ be the large-scale path loss attenuation, as given by [6]

$$\beta_k(\mathbf{q}[n]) = \begin{cases} \beta_0 d_k^{-2}(\mathbf{q}[n]), & \text{LoS link}, \\ \varrho\beta_0 d_k^{-2}(\mathbf{q}[n]), & \text{NLoS link}, \end{cases} \quad (32)$$

where ϱ is the additional attenuation factor due to the non-line-of-sight (NLoS) condition. As a result, the average channel gain between the UAV and ST_k depends on the LoS probability, and can be written as [41]

$$g_k(\mathbf{q}[n]) = \hat{p}_{\text{LoS},k}(\mathbf{q}[n])\beta_0 d_k^{-2}(\mathbf{q}[n]), \quad \forall n, \quad (33)$$

where $\hat{p}_{\text{LoS},k}(\mathbf{q}[n]) = p_{\text{LoS},k}(\mathbf{q}[n]) + \varrho(1 - p_{\text{LoS},k}(\mathbf{q}[n]))$ is the regularized LoS probability, which could be obtained by measuring or modeling empirically [41].

Problem (24) remains unchanged, except that constraint (24i) is modified as

$$\tau_k[n]\left(2^{\frac{l_k[n]}{\tau_k[n]B}} - 1\right) \le \frac{E_k\beta_0\hat{p}_{\text{LoS},k}(\mathbf{q}[n])}{(H^2 + \|\mathbf{q}[n] - \mathbf{q}_k\|^2)\sigma^2}, \quad \forall n \in \mathcal{M}_k, \ \forall k \in \mathcal{K}. \quad (34)$$

Algorithm 3 remains valid. The only change is that, at the $(j+1)$-th iteration, we can take the value of $\hat{p}_{\text{LoS},k}(\mathbf{q}[n])$ based on the outcome of the j-th iteration, i.e., $\mathbf{q}^{(j)}[n]$, and convexify (34) in the same way as constraint (24i) by employing the SCA technique.

V. Simulation and Numerical Results

In this section, numerical results are provided to verify the proposed joint optimization of computation offloading and trajectory planning for UAV-assisted edge computing. The altitude of the UAV is $H = 100$ m. The duration of each slot is $\delta = 0.5$ s [18]. The noise power is $\sigma^2 = -100$ dBm at the UAV. The reference channel gain is set to $\beta_0 = -50$ dB at $d = 1$ m. The system bandwidth is $B = 1$ MHz. The required number of CPU cycles to execute a bit is $c = 10^3$. The effective capacitance coefficient is $\kappa = 10^{-28}$ [23], [25]. The maximum and minimum speeds of the UAV are $v_{\max} = 50$ m/s and $v_{\min} = 3$ m/s, respectively [23]. The maximum acceleration is $a_{\max} = 5$ m/s^2 [18]. We assume that $c_1 = 0.002$ and $c_2 = 70.698$ for the fixed-wing UAV [18]. W.l.o.g, we set the CPU frequency of the local processor as $f_{\text{ST}_k} = 200$ MHz at $\text{ST}_k, \forall k$, and the transmit energy of ST_k per slot as $E_k = 0.001$ J, $\forall k$. To evaluate the offloading intensity, we define the average offloading per slot (AOPS) for ST_k:

$$\text{AOPS}_k = \frac{L_k - l_{\text{ST}_k}}{n_k'' - n_k'}. \quad (35)$$

We simulate two scenarios. Scenario 1: The UAV flies along a straight line. We assume that the UAV's altitude and the STs' locations meet the special geometric condition proclaimed by (16), and that the STs' locations are arbitrarily distributed. Scenario 2: The UAV flies on a horizontal plane. Four different task sizes are studied.

A. UAV Flies Along a Straight Line

We first consider the case where condition (16) is satisfied, i.e., problem (13) is convex. With $T = 10$ s, the UAV flies from the location (0, 0, 100) m to (100, 0, 100) m and assists 6 STs

with their computing tasks. The shadow region Ψ in Fig. 3 intersects the x-axis at $x_L = 42.3$ m and $x_R = 57.7$ m, and all STs are located in this region. Both the initial and final velocities of the UAV are set to 10 m/s. The task sizes of the STs are $L_1 = 2.5$, $L_2 = 5.5$, $L_3 = 10$, $L_4 = 4$, $L_5 = 3$, and $L_6 = 3$ Mbits. The AOPS of the STs are $\text{AOPS}_1 = 0.243$, $\text{AOPS}_2 = 1.700$, $\text{AOPS}_3 = 1.880$, $\text{AOPS}_4 = 0.160$, $\text{AOPS}_5 = 0.123$, and $\text{AOPS}_6 = 0.164$ Mbits/s. ST_2 and ST_3 are terminals with high offloading demands, and the others have low offloading demands. The execution intervals of the STs are $[t_1', t_1''] = [0, 4]$ s, $[t_2', t_2''] = [7, 9]$ s, $[t_3', t_3''] = [2, 5]$ s, $[t_4', t_4''] = [2, 10]$ s, $[t_5', t_5''] = [3, 10]$ s, and $[t_6', t_6''] = [2, 8]$ s. The locations of the STs are $(x_1, y_1) = (50, 40)$ m, $(x_2, y_2) = (43, 0)$ m, $(x_3, y_3) = (45, 5)$ m, $(x_4, y_4) = (57, -40)$ m, $(x_5, y_5) = (50, -40)$ m, and $(x_6, y_6) = (54, 10)$ m.

Simulations show that both Algorithms 1 and 2 need no more than two iterations to converge to the same results, when problem (13) is convex. The UAV flies at a constant speed. This is because the flight distance is relatively short when (16) holds, thus the locations of the UAV have a weak impact on the channel capacity. The constant speed helps energy saving.

Fig. 4(a) shows the task arrival, deadline, and departure curves of the UAV. The same task arrival and departure curves are obtained by Algorithms 1 and 2. The task arrival curve describes the cumulative amount of the tasks received by the UAV by the time slot indicated by the x-axis. The slope of the departure curve specifies the CPU frequency. The CPU frequency increases after slot 5, decreases after slot 10, and remains unchanged at other slots, which is consistent with Proposition 1 in Section III-C. At slot 5, the task arrival curve intersects the departure curve. The causality constraint is met with equality, i.e., all tasks received by slot 5 have been computed, and the CPU frequency increases after the intersection. At slot 10, the departure curve intersects the deadline curve. The deadline constraint is met with equality, i.e., ST_3's tasks are completed by its deadline, and the CPU frequency decreases after the intersection. At other slots, the departure curve intersects neither of the task arrival and deadline curves, thus the causality and deadline constraints are met with inequality and the CPU frequency remains unchanged.

Fig. 4(b) depicts the allocated time of the STs, which verifies the regular pattern of time allocation revealed in Section III-C. ST_2 and ST_3 with high offloading demands occupy more time resources within the time slots during which they are served, i.e., $\{15, 16, 17\}$ and $\{5, 6, 7, 8, 9\}$, respectively. ST_1, ST_4, ST_5, and ST_6 with low offloading demands are allocated little or none during these slots and obtain the majority of their required time resources in the rest of the slots.

Next, we consider that the locations of the STs are arbitrarily distributed, i.e., problem (13) is non-convex. With $T = 40$ s, the UAV flies from the location $(-500, 0, 100)$ m to (500, 0, 100) m and assists 6 STs with their computing tasks. Both the initial and final speeds of the UAV are set to 25 m/s. The task sizes of the STs are $L_1 = 16$, $L_2 = 15$, $L_3 = 20$, $L_4 = 24$, $L_5 = 29$, and $L_6 = 12$ Mbits. The

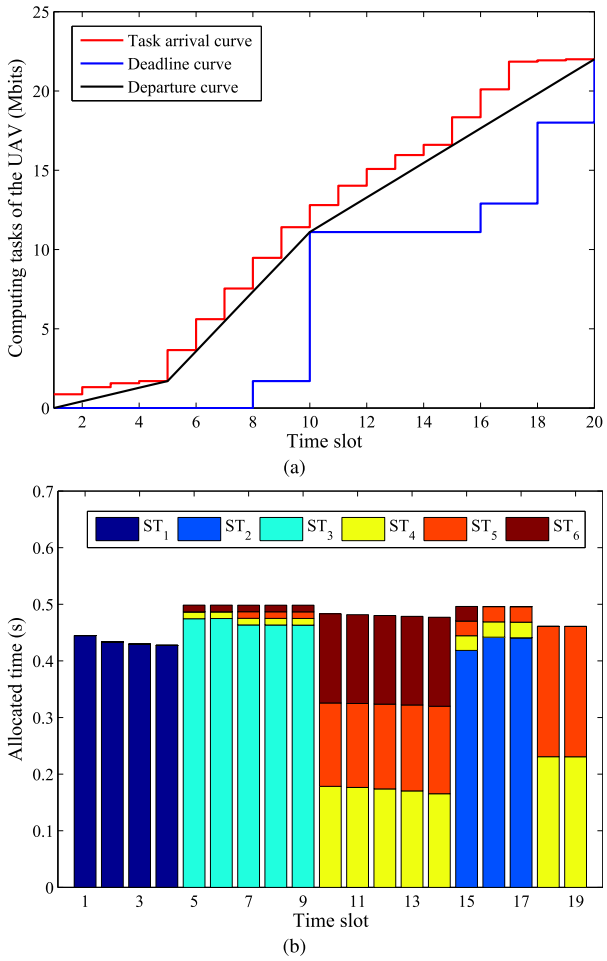

Fig. 4. (a) Task arrival, deadline, and departure curves of the UAV. (b) Time allocations for the STs in different time slots.

AOPS of the STs are $\text{AOPS}_1 = 0.737$, $\text{AOPS}_2 = 0.548$, $\text{AOPS}_3 = 2.111$, $\text{AOPS}_4 = 0.351$, $\text{AOPS}_5 = 2.123$, and $\text{AOPS}_6 = 0.205$ Mbits/s. ST_3 and ST_5 are terminals with high offloading demands, and the others have low offloading demands. The execution intervals of the STs are $[t_1', t_1''] = [0, 10]$ s, $[t_2', t_2''] = [5, 17]$ s, $[t_3', t_3''] = [13, 18]$ s, $[t_4', t_4''] = [12, 39]$ s, $[t_5', t_5''] = [25, 32]$ s, and $[t_6', t_6''] = [20, 40]$ s. The locations of the STs are $(x_1, y_1) = (-350, 30)$ m, $(x_2, y_2) = (-200, -20)$ m, $(x_3, y_3) = (-20, 0)$ m, $(x_4, y_4) = (120, -10)$ m, $(x_5, y_5) = (220, 15)$ m, and $(x_6, y_6) = (420, 5)$ m. The patterns revealed in Section III-C can also be validated, as done in Fig. 4. The results are suppressed for conciseness.

Fig. 5 shows the convergence behaviors of Algorithms 1 and 2. Compared with the sub-optimal solution obtained by the alternating optimization-based Algorithm 1, the solution achieved by the SCA-based Algorithm 2 and satisfying the KKT conditions is superior in terms of energy saving. In addition, we observe that Algorithm 1 converges in ten iterations, while Algorithm 2 only needs three iterations.

Fig. 6 plots the trajectory and speed of the UAV obtained by Algorithm 2. In Fig. 6(a), the UAV's trajectory, the STs' locations and task execution time show that the UAV reaches the vicinities of the STs with high offloading demands, i.e., ST_3 and ST_5, before they start to offload tasks at $t = 13$ s and

TABLE I

TASK SIZES L_k, AOPS$_k$, EXECUTION INTERVALS $[t'_k, t''_k]$, AND LOCATIONS (x_k, y_k) OF THE STS IN DIFFERENT CASES

Smart terminal		ST_1	ST_2	ST_3	ST_4	ST_5	ST_6	ST_7	ST_8
Task size L_k (Mbit) /AOPS$_k$ (Mbit/s)	Case 1	100 / 0.286	50 / 0.461	50 / 1.621	50 / 0.532	35 / 1.737	40 / 2.000	35 / 1.737	40 / 0.478
	Case 2	80 / 0.209	30 / 0.236	50 / 1.621	30 / 0.279	35 / 1.737	40 / 2.000	35 / 1.737	20 / 0.188
	Case 3	100 / 0.286	50 / 0.461	40 / 1.276	50 / 0.532	25 / 1.211	30 / 1.474	25 / 1.211	40 / 0.478
	Case 4	60 / 0.131	30 / 0.236	10 / 0.241	30 / 0.279	10 / 0.421	10 / 0.421	10 / 0.421	20 / 0.188
Execution interval $[t'_k, t''_k]$ (s)		[0, 130]	[15, 60]	[40, 55]	[50, 90]	[75, 85]	[90, 100]	[105, 115]	[90, 125]
Location (x_k, y_k) (m)		(875, 200)	(750, 650)	(950, 950)	(500, 1100)	(150, 900)	(125, 500)	(100, 75)	(475, 250)

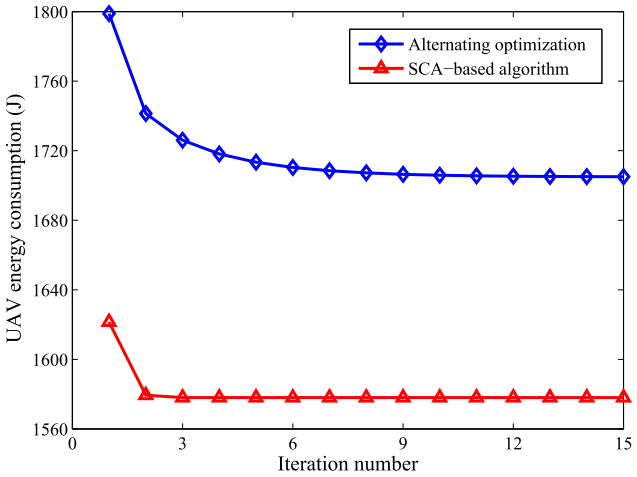

Fig. 5. Convergence behaviors of the proposed Algorithms 1 and 2.

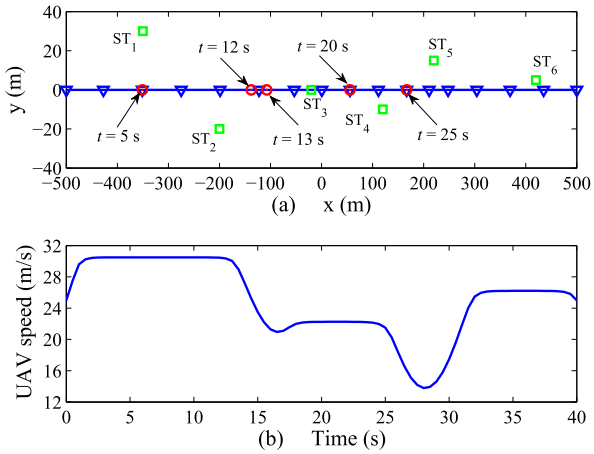

Fig. 6. The trajectory and speed of the UAV. The trajectory is sampled every 2.5 s and the sampled locations are marked with blue ▽. The locations of the UAV at $t = 5$ s, 12 s, 13 s, 20 s, and 25 s are marked with red ○. The locations of the STs are marked with green □.

25 s, respectively. For the STs with low offloading demands, e.g., ST_2, ST_4, and ST_6, the UAV can be far away from when their execution intervals approach. In Fig. 6(b), the speed of the UAV decreases at $t = 13$ s and 25 s. This is because a low speed allows the UAV to remain close to ST_3 and ST_5, so that the STs can offload tasks properly.

We further consider the online implementation of Algorithm 2. Fig. 7 shows the change of the UAV's energy consumption with the increase of t_a (by which the STs send their requests in advance). The offline result is provided as a reference. With the increase of t_a, the UAV can assist the STs with their tasks at a lower energy consumption. This is because the UAV can capitalize on requests received in advance for better computation offloading and trajectory planning. When $t_a \geq 26$, the online implementation achieves the same UAV's energy consumption as the offline.

B. UAV Flies on a Horizontal Plane

Consider next that the UAV flies on a horizontally 2D plane. The initial horizontal location is $\mathbf{q}_I = (500, 500)$ m, and the UAV returns to the initial location after the mission. The initial and final velocities are $\mathbf{v}_I = (15, -15)$ m/s and $\mathbf{v}_F = (15, 15)$ m/s, respectively. The mission completion time is $T = 130$ s, during which the UAV provides computing for 8 STs. As shown in Table I, four cases with different task sizes are considered under the same execution intervals and ST locations. In Case 1, ST_1, ST_2, ST_4, and ST_8 are set to have low offloading demands, while the others are set to have high offloading demands. In Case 2, the task sizes of ST_1, ST_2, ST_4, and ST_8 are reduced by 20 Mbits each, as compared to Case 1. In Case 3, the task sizes of the other STs are reduced by 10 Mbits each, as compared to Case 1. In Case 4, the task sizes of all the STs are relatively small and there is no ST with high offloading demands. The patterns revealed in Section III-C can also be validated for the 2D trajectories, as done in Fig. 4. The results are suppressed for conciseness.

Fig. 8(a) shows the trajectories of the UAV in the four cases. In Cases 1, 2, and 3, the UAV reaches the vicinities of the STs with high offloading demands, i.e., ST_3, ST_5, ST_6, and ST_7, before they start to offload tasks at $t = 40$ s, 75 s, 90 s, and 105 s, respectively, to get good channel conditions for the offloading. For the STs with low offloading demands, such as ST_2, ST_4, and ST_8, the UAV is far away when their execution intervals approach. We also see that the UAV's trajectories are closer to ST_1 in Cases 1, 2, and 3, although its offloading demand is low. Most of ST_1's execution interval overlaps with that of the other STs, as shown in Table I. The UAV allows ST_1 to offload as much as possible within the relatively less busy interval [0, 40] s (during which no ST has high offloading demands). However, the UAV is too far away from ST_1 at $t = 0$ s. For this reason, the UAV's trajectories are deviated towards ST_1. Further, by comparing the trajectories in Cases 2 and 3 with that in Case 1, we see that the task sizes have a non-negligible impact on the trajectories. With the decreasing task sizes, the distance between the UAV and STs can increase during the time when computing services are provided. The STs with high offloading demands have

TABLE II
ENERGY CONSUMPTION OF THE UAV IN DIFFERENT CASES

Energy	Case 1	Case 2	Case 3	Case 4
Computing energy (J)	251.90	111.76	173.49	10.56
Propulsion energy (J)	3642.78	3271.77	2210.82	1205.61

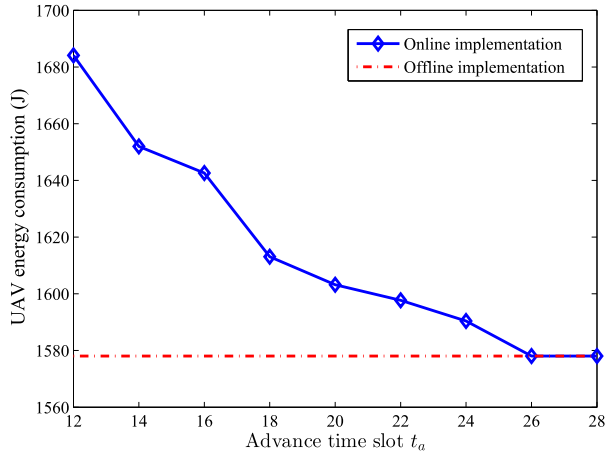

Fig. 7. Energy consumption of the UAV in online and offline implementations.

a stronger impact on the UAV's trajectories than those with low offloading demands. For instance, with all STs' offloading tasks being relatively small in Case 4, the UAV wanders near the initial location.

Fig. 8(b) plots the speeds of the UAV in the four cases. In Cases 1, 2, and 3, the UAV's speeds hit the locally lowest points around $t = 17$ s, 45 s, 80 s, and 110 s, which correspond to the direction changes of the UAV around ST_1, ST_3, ST_5, and ST_7, respectively, as shown in Fig. 8(a). The larger the turning angle is, the slower the speed is. The slower speeds also allow the UAV to remain close to these STs, to benefit their task offloading. We also see that within [90, 100] s, the UAV's speed drops sharply and then increases rapidly in Cases 1 and 2, and fluctuates slightly in Case 3. In Cases 1 and 2, to accomplish ST_6's tasks in time, the UAV flies to the vicinity of ST_6 at high speeds before it starts offloading at $t = 90$ s. Then the UAV flies slowly to remain close to ST_6. When ST_6's computations are completed at $t = 100$ s, the UAV flies to the next ST with high offloading demands, i.e., ST_7, at a higher speed. In Case 3, with the reduced task size, ST_6's computations can be finished in time even if the UAV's speed is not much adjusted within the time interval. In Case 4, the UAV's speed remains unchanged for most of the time, while the direction changes continuously, as already shown in Fig. 8(a).

Table II lists the computing and propulsion energy consumption of the UAV in the four cases. We see that most of energy is consumed for flight. The energy consumption for computing is considerable, especially when there are the STs with high offloading demands. When the STs' task sizes are reduced, both the computing and propulsion energy consumption decrease. By comparing the propulsion energy in Cases 2 and 3 with that in Case 1, we conclude that the task sizes of the STs with high offloading demands affect the

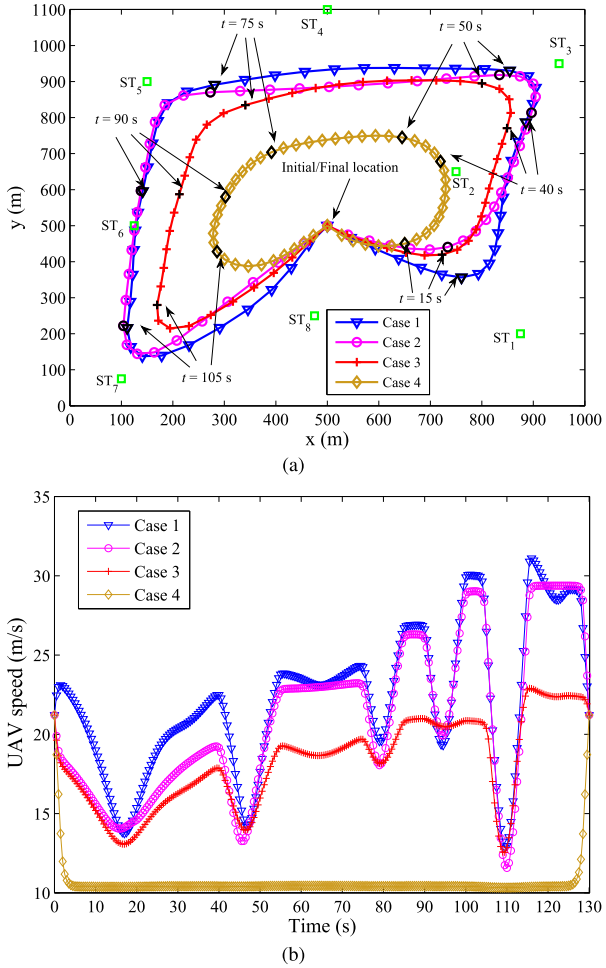

Fig. 8. The trajectories and speeds of the UAV in different cases. All trajectories are sampled every 2.5 s. The locations of the UAV at $t = 15$ s, 40 s, 50 s, 75 s, 90 s, and 105 s are marked with black symbols. The locations of the STs are marked with green □.

UAV's propulsion energy more markedly than those of the STs with low offloading demands.

Fig. 9(a) reveals the impact of the STs' density on the UAV trajectory. The ST settings are based on Case 1 in Table I. The legend ST1$-k$ indicates that only $ST_1, \cdots, ST_k$ are active in the system. It can be seen that the UAV's trajectory changes as the number of STs grows, especially when the newly added ST has high offloading demands. For example, the trajectories for ST1$-$2 and ST1$-$3 are significantly different, since ST_3 has high offloading demands. In contrast, the trajectories for ST1$-$3 and ST1$-$4 are similar, since ST_4 has low offloading demands.

Fig. 9(b) evaluates the impact of the UAV's altitude on its energy consumption based on the extension of Algorithm 3 in Section IV-D. The relevant parameters in (31) and (32) are set to $C = 10$, $D = 0.6$, and $\varrho = 0.35$ [41]. With an increasing UAV's altitude, the probability of an LoS link between the STs and UAV grows. On the other hand, the increased altitude enlarges the propagation distance. There can be an optimal altitude for the UAV, depending on the placement of the STs and their task demands. As shown in Fig. 9(b), the optimal flight altitudes of the UAV are different under different STs'

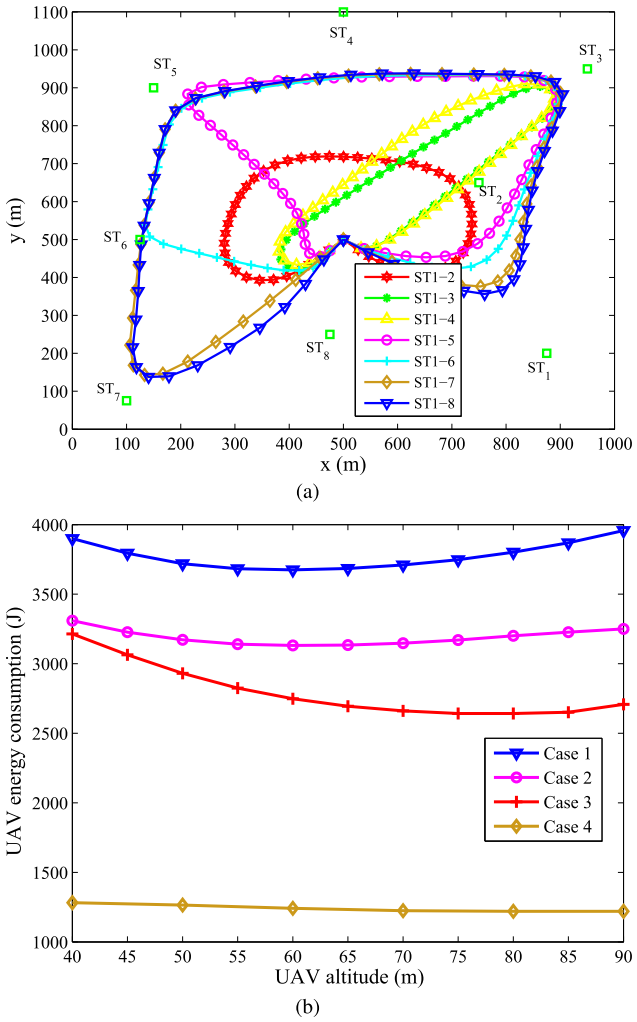

Fig. 9. (a) The impact of the STs' density on the UAV trajectory. (b) The impact of the UAV's altitude on its energy consumption.

task sizes. In Cases 1 and 2, the tasks of the STs with high offloading demands are relatively high and consistent, and the optimal altitude of the UAV is 60 m. In Case 3, the tasks of the STs with high offloading demands are reduced, and the optimal altitude of the UAV increases to 80 m. In Case 4, with all STs' offloading tasks being relatively small, the energy consumption of the UAV changes very slowly with the increase of the UAV's altitude, and the optimal altitude is above 90 m and not plotted in the figure.

VI. Extension and Generalization

A. Extension Under Multiple Fixed-Wing UAVs

Consider a system with M UAVs. Within each time slot, the STs can offload their computing tasks to multiple UAVs. The frequency bands of the UAVs, denoted by $B_1, \cdots, B_M$, are orthogonal to each other. Each ST can operate under different frequencies to offload tasks to different UAVs at different time slots. Multiple STs operating under the same frequency band can offload tasks to the UAV operating under the frequency band; and the STs run TDMA, as in the single-UAV scenario. The UAVs also keep a minimum distance, denoted by $d_{\min}$, for collision avoidance, as given by

$$(H_i - H_m)^2 + \|\mathbf{q}_i[n] - \mathbf{q}_m[n]\|^2 \geq d_{\min}^2, \quad \forall n \in \mathcal{N}_1,\ \forall i, m \in \mathbb{M}, \tag{36}$$

where H_i and H_m are the altitudes of UAV_i and UAV_m, respectively; and $\mathbb{M} = \{1, \cdots, M\}$. The other settings of the system are consistent with the single-UAV scenario, including the total offloading time of the STs to each UAV in (2), and the constraints of each UAV's positions, velocities, and accelerations in (24b) – (24h).

As with (24) under a single UAV, the minimization of the energy consumption of M UAVs can be formulated as

$$\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q}} \sum_{m=1}^{M} \left\{ \sum_{n=2}^{N} \delta \kappa f_{\text{UAV}_m}^3[n] + \sum_{n=0}^{N-1} \delta \left(c_1 \|\mathbf{v}_m[n]\|^3 + \frac{c_2}{\|\mathbf{v}_m[n]\|} \left(1 + \frac{\|\mathbf{a}_m[n]\|^2}{g^2}\right)\right)\right\} \tag{37a}$$

$$\text{s.t.} \quad (2), (24b) - (24h), (36),$$

$$\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}_m}[i]}{c} \leq \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_{m,k}[i],\ \forall n \in \mathcal{N}_2, \forall m \in \mathbb{M}, \tag{37b}$$

$$\sum_{k \in \Omega_n} \sum_{m=1}^{M} \sum_{i=n'_k}^{n''_k - 1} l_{m,k}[i] \leq \sum_{m=1}^{M} \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}_m}[i]}{c}, \quad \forall n \in \mathcal{N}_2, \tag{37c}$$

$$l_{\text{ST}_k} + \sum_{m=1}^{M} \sum_{i=n'_k}^{n''_k - 1} l_k[i] = L_k, \quad \forall k \in \mathcal{K}, \tag{37d}$$

$$\tau_{m,k}[n] \left(2^{\frac{l_{m,k}[n]}{\tau_{m,k}[n] B_m}} - 1\right) \leq \frac{E_{m,k} \beta_0}{(H_m^2 + \|\mathbf{q}_m[n] - \mathbf{q}_k\|^2)\sigma^2}, \quad \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}, \forall m \in \mathbb{M}, \tag{37e}$$

$$f_{\text{UAV}_m}[n] \geq 0, \quad \forall n \in \mathcal{N}_2,\ \forall m \in \mathbb{M}, \tag{37f}$$

$$l_{m,k}[n] \geq 0, \quad \tau_{m,k}[n] \geq 0, \quad \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}, \quad \forall m \in \mathbb{M}, \tag{37g}$$

where $f_{\text{UAV}_m}[n]$ is the CPU frequency of UAV_m at slot n; $l_{m,k}[n]$ is the offloading task of ST_k to UAV_m at slot n; $\tau_{m,k}[n]$ is the offloading time of ST_k to UAV_m at slot n; and $E_{m,k}$ is the emission energy of ST_k when offloading tasks to UAV_m. With a little abuse of notations, we denote $\boldsymbol{f} = \{f_{\text{UAV}_m}[n], \forall m \in \mathbb{M}, \forall n \in \mathcal{N}_2\}$; $\boldsymbol{l} = \{l_{m,k}[n], \forall m \in \mathbb{M}, \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}\}$; $\boldsymbol{\tau} = \{\tau_{m,k}[n], \forall m \in \mathbb{M}, \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}\}$; and $\boldsymbol{Q} = \{\mathbf{q}_m[n], \mathbf{v}_m[n], \mathbf{a}_m[n], \forall m \in \mathbb{M}, \forall n \in \mathcal{N}_3\}$. Constraints (37b) – (37g) are consistent with (13b) – (13g) in the single-UAV scenario.

Compared to problem (24), the only new constraint in (37) is the non-convex collision avoidance constraint (36), which can be convexified by using the SCA technique in the same way as (25b). Specifically, the second term on the LHS of (36) is convex and differentiable in $(\mathbf{q}_i[n] - \mathbf{q}_m[n])$. Its lower bound can be obtained by taking the first-order Taylor expansion at

the local points obtained at the j-th iteration, i.e., $\mathbf{q}_i^{(j)}[n]$ and $\mathbf{q}_m^{(j)}[n]$, as given by

$$\begin{aligned}\|\mathbf{q}_i[n]-\mathbf{q}_m[n]\|^2 &\geq \|\mathbf{q}_i^{(j)}[n]-\mathbf{q}_m^{(j)}[n]\|^2 \\ &+2(\mathbf{q}_i^{(j)}[n]-\mathbf{q}_m^{(j)}[n])^T(\mathbf{q}_i[n]-\mathbf{q}_m[n]-\mathbf{q}_i^{(j)}[n]+\mathbf{q}_m^{(j)}[n]) \\ &:= R(\mathbf{q}_i[n],\mathbf{q}_m[n]). \end{aligned} \tag{38}$$

Since $R(\mathbf{q}_i[n],\mathbf{q}_m[n])$ is linear to both $\mathbf{q}_i[n]$ and $\mathbf{q}_m[n]$, constraint (36) can be approximated by the following convex form:

$$(H_i-H_m)^2+R(\mathbf{q}_i[n],\mathbf{q}_m[n])\geq d_{\min},\ \forall n\in\mathcal{N}_1,\ \forall i,m\in\mathbb{M}. \tag{39}$$

By letting $\boldsymbol{w} := \{w_{m,n},\forall m,n\}$ denote slack variables, problem (37) can be approximated to the following convex problem

$$\begin{aligned}\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q},\boldsymbol{w}} \sum_{m=1}^{M}\Bigg\{\sum_{n=2}^{N}\delta\kappa f_{\mathrm{UAV}_m}^3[n]+\sum_{n=0}^{N-1}\delta\Bigg(c_1\|\mathbf{v}_m[n]\|^3 \\ +\frac{c_2}{w_{m,n}}\Big(1+\frac{\|\mathbf{a}_m[n]\|^2}{g^2}\Big)\Bigg)\Bigg\} \\ \text{s.t.}\quad (2),(24b)-(24g),(37b)-(37d), \\ (37f),(37g),(25c),(27),(29),(39),\end{aligned} \tag{40}$$

which can be solved by using the interior point method. Algorithm 3 can readily solve problem (37), i.e., by only replacing step 3 to solve problem (40) at the given local points $\{\mathbf{q}_m^{(j)}[n],\ \mathbf{v}_m^{(j)}[n],\forall m,n\}$, and obtain the optimal solutions to $\boldsymbol{f}^{(j+1)},\boldsymbol{l}^{(j+1)},\boldsymbol{\tau}^{(j+1)},\boldsymbol{Q}^{(j+1)},$ and $\boldsymbol{w}^{(j+1)}$.

B. Extension to Rotary-Wing UAV

By replacing the mobility and propulsion energy models of a fixed-wing UAV with those of a rotary-wing UAV, our approach is applicable to the rotary-wing UAV. The velocity of a rotary-wing UAV at time slot n can be modeled as [11], [41]

$$\mathbf{v}[n]=\frac{\mathbf{q}[n+1]-\mathbf{q}[n]}{\delta}. \tag{41}$$

The propulsion energy model of the rotary-wing UAV at slot n is given by [11], [41]

$$\begin{aligned}E_{\text{fly}}[n] &= P_0\Big(1+\frac{3\|\mathbf{v}[n]\|^2}{U_{\text{tip}}^2}\Big) \\ &+P_1\Bigg(\sqrt{1+\frac{\|\mathbf{v}[n]\|^4}{4v_0^4}}-\frac{\|\mathbf{v}[n]\|^2}{2v_0^2}\Bigg)^{1/2}+\frac{d_{\text{f}}\rho s A\|\mathbf{v}[n]\|^3}{2},\end{aligned} \tag{42}$$

where P_0 and P_1 are two constants and stand for the blade profile power and induced power under hovering mode, respectively; U_{tip} is the tip speed of the rotor blade; v_0 denotes the average rotor induced velocity in hover; d_{f} and s are the fuselage drag ratio and rotor solidity; and ρ and A are the atmospheric density and rotor disc area, respectively.

The mobility model of the rotary-wing UAV is given by [11], [41]

$$\mathbf{q}[0]=\mathbf{q}_I,\ \mathbf{q}[N]=\mathbf{q}_F,\ \|\mathbf{q}[n+1]-\mathbf{q}[n]\|\leq\delta v_{\max},\ \forall n. \tag{43}$$

As with (24) under a fixed-wing UAV, the minimization of the rotary-wing UAV's energy consumption is given by

$$\begin{aligned}\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q}} \sum_{n=2}^{N}\delta\kappa f_{\text{UAV}}^3[n]+\sum_{n=0}^{N-1}E_{\text{fly}}[n] \\ \text{s.t.}\quad (2),(13b)-(13d),(13f),(13g),(24i),(43),\end{aligned} \tag{44}$$

where (2), (13b) – (13d), (13f), (13g), and (24i) remain the same as they are under the fixed-wing UAV. The objective is non-convex, as the second term on the RHS of (42) is non-convex. Constraint (24i) is non-convex, and can be convexified to (27) by employing the SCA technique.

To convexify $E_{\text{fly}}[n]$ in (42), we introduce slack variables $\boldsymbol{\gamma}:=\{\gamma_n\geq 0,\forall n\}$ such that

$$\gamma_n^2=\sqrt{1+\frac{\|\mathbf{v}[n]\|^4}{4v_0^4}}-\frac{\|\mathbf{v}[n]\|^2}{2v_0^2},\quad\forall n, \tag{45}$$

which is equivalent to

$$\frac{1}{\gamma_n^2}=\gamma_n^2+\frac{\|\mathbf{v}[n]\|^2}{v_0^2},\quad\forall n. \tag{46}$$

As a result, the second term on the RHS of (42) is replaced by the linear component $P_1\gamma_n$, with the additional constraint (46). After substituting (41), (46) is relaxed as [11]

$$\frac{1}{\gamma_n^2}\leq\gamma_n^2+\frac{\|\mathbf{q}[n+1]-\mathbf{q}[n]\|^2}{\delta^2v_0^2},\quad\forall n, \tag{47}$$

which is non-convex and can be convexified by using the SCA. Specifically, the RHS of (47) is convex and differentiable in both γ_n and $(\mathbf{q}[n+1]-\mathbf{q}[n])$. Its lower bound can be obtained by taking the first-order Taylor expansion at the local points obtained at the j-th iteration, i.e., $\gamma_n^{(j)}$, $\mathbf{q}^{(j)}[n]$, and $\mathbf{q}^{(j)}[n+1]$, as given by

$$\begin{aligned}\gamma_n^2+\frac{\|\mathbf{q}[n+1]-\mathbf{q}[n]\|^2}{\delta^2v_0^2} &\geq (\gamma_n^{(j)})^2+2\gamma_n^{(j)}(\gamma_n-\gamma_n^{(j)}) \\ &+\frac{\|\mathbf{q}^{(j)}[n+1]-\mathbf{q}^{(j)}[n]\|^2}{\delta^2v_0^2}+\frac{2(\mathbf{q}^{(j)}[n+1]-\mathbf{q}^{(j)}[n])^T}{\delta^2v_0^2} \\ &\times(\mathbf{q}[n+1]-\mathbf{q}[n]-\mathbf{q}^{(j)}[n+1]+\mathbf{q}^{(j)}[n]) \\ &:= G(\gamma_n,\mathbf{q}[n],\mathbf{q}[n+1]).\end{aligned} \tag{48}$$

Since $G(\gamma_n,\mathbf{q}[n],\mathbf{q}[n+1])$ is linear in γ_n, $\mathbf{q}[n]$, and $\mathbf{q}[n+1]$, (47) can be approximated by the following convex form:

$$\frac{1}{\gamma_n^2}\leq G(\gamma_n,\mathbf{q}[n],\mathbf{q}[n+1]),\quad\forall n. \tag{49}$$

As a result, problem (44) can be approximated to the following convex problem

$$\begin{aligned}\min_{\boldsymbol{f},\boldsymbol{l},\boldsymbol{\tau},\boldsymbol{Q},\boldsymbol{\gamma}} \sum_{n=2}^{N}\delta\kappa f_{\text{UAV}}^3[n]+\sum_{n=0}^{N-1}\delta\Bigg(P_0\Big(1+\frac{3\|\mathbf{q}[n+1]-\mathbf{q}[n]\|^2}{\delta^2\ U_{\text{tip}}^2}\Big) \\ +P_1\gamma_n+\frac{d_{\text{f}}\rho s A\|\mathbf{q}[n+1]-\mathbf{q}[n]\|^3}{2\delta^3}\Bigg) \\ \text{s.t.}\quad (2),(13b)-(13d),(13f),(13g),(27),(43),(49),\end{aligned} \tag{50}$$

which can be solved by using the interior point method. Algorithm 3 can solve problem (44), by only replacing step 3 to solve problem (50) at the given local points $\{\mathbf{q}^{(j)}[n], \gamma_n^{(j)}, \forall n\}$, and obtain the optimal solutions to $\boldsymbol{f}^{(j+1)}$, $\boldsymbol{l}^{(j+1)}, \boldsymbol{\tau}^{(j+1)}, \boldsymbol{Q}^{(j+1)}$, and $\boldsymbol{\gamma}^{(j+1)}$.

VII. Conclusion

We minimized the energy consumption of a fixed-wing UAV playing a mobile computer server, by jointly optimizing the trajectory and CPU frequency of the UAV, and offloading schedule for STs with delay-sensitive tasks. We proved that the problem is convex when the UAV flies a 1D trajectory and the altitude of the UAV and the locations of the STs satisfy a specific condition. We also developed two iterative algorithms based on alternating optimization and SCA to achieve the globally optimal solution to the problem in the case where the problem is convex, and a high-quality solution satisfying the KKT conditions in the more general case where it is non-convex. We also generalized our approach under 2D UAV trajectories. Online implementation of the approach was described. Extensions to multiple UAVs and a rotary-wing UAV were discussed. Numerical results demonstrated the merits of the proposed schemes. As our future work, we will extend the proposed algorithms to three-dimensional flight trajectories.

Appendix

A. Proof of Lemma 1

The second-order derivative of the RHS of (13e) with respect to $x[n]$ is given by

$$\frac{-2E_k\beta_0(H^2+y_k^2-3(x[n]-x_k)^2)}{\sigma^2(H^2+y_k^2+(x[n]-x_k)^2)^3} := \nabla_k''(x[n]), \quad \forall n \in \mathcal{M}_k,\ \forall k \in \mathcal{K}. \tag{51}$$

According to the second-order concavity condition [33], the RHS of (13e) is concave in the flight range from $(x_I, 0, H)$ to $(x_F, 0, H)$ if and only if $\nabla_k''(x[n]) \le 0$, i.e.,

$$H^2 \ge 3(x_k - x[n])^2 - y_k^2, \quad \forall n \in \mathcal{M}_k,\ \forall k \in \mathcal{K}. \tag{52}$$

When (52) holds, constraint (13e) is convex in $\{l_k[n], \tau_k[n], x[n], \forall n \in \mathcal{M}_k, \forall k \in \mathcal{K}\}$, and problem (13) is convex.

B. Proof of Proposition 1

In problem (13), the variables $\boldsymbol{f}$ are decoupled from $\{\boldsymbol{\tau}, \boldsymbol{Z}\}$. Given fixed $\boldsymbol{l}$, the optimization problem for $\boldsymbol{f}$ is given by

$$\min_{\boldsymbol{f}} \sum_{n=2}^{N} \delta\kappa f_{\text{UAV}}^3[n] \quad \text{s.t.} \quad (13b), (13c), (13f). \tag{53}$$

Problem (53) is convex. Let $\boldsymbol{\Gamma} := \{\alpha_n, \lambda_n, \forall n \in \mathcal{N}_2\}$, where α_n and λ_n are the Lagrange multipliers associated with the data causality constraint (13b) and deadline constraint (13c), respectively. The Lagrangian of (53) is given by

$$\begin{aligned} L(\boldsymbol{f}, \boldsymbol{\Gamma}) &= \sum_{n=2}^{N} \delta\kappa f_{\text{UAV}}^3[n] \\ &\quad + \sum_{n=2}^{N} \alpha_n \left(\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c} - \sum_{i=1}^{n-1}\sum_{k=1}^{K} l_k[i] \right) \\ &\quad + \sum_{n=2}^{N} \lambda_n \left(\sum_{k\in\Omega_n} \sum_{i=n_k'}^{n_k''-1} l_k[i] - \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c} \right) \\ &= \sum_{n=2}^{N} \left(\delta\kappa f_{\text{UAV}}^3[n] + \frac{\delta f_{\text{UAV}}[n]}{c} \sum_{i=n}^{N} (\alpha_i - \lambda_i) \right) + C(\boldsymbol{\Gamma}), \end{aligned} \tag{54}$$

where $C(\boldsymbol{\Gamma}) := \sum_{n=2}^{N} \lambda_n \sum_{k\in\Omega_n} \sum_{i=n_k'}^{n_k''-1} l_k[i] - \sum_{n=2}^{N} \alpha_n \times \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i]$ for notation simplicity.

Let $\boldsymbol{f}^*$ denote the optimal solution to (53), and $\boldsymbol{\Gamma}^*$ the optimal Lagrange multiplier to the dual problem of (53). The KKT optimality conditions of (53) dictate that the non-negative α_n^* and λ_n^* satisfy the complementary slackness conditions: $\forall n \in \mathcal{N}_2$,

$$\begin{cases} \alpha_n^* = 0, \ \text{ if } \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}^*[i]}{c} < \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i], \\ \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}^*[i]}{c} = \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i], \ \text{ if } \alpha_n^* > 0, \end{cases} \tag{55}$$

$$\begin{cases} \lambda_n^* = 0, \ \text{if } \sum_{k\in\Omega_n} \sum_{i=n_k'}^{n_k''-1} l_k[i] < \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}^*[i]}{c}, \\ \sum_{k\in\Omega_n} \sum_{i=n_k'}^{n_k''-1} l_k[i] = \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}^*[i]}{c}, \ \text{if } \lambda_n^* > 0; \end{cases} \tag{56}$$

and $\forall n \in \mathcal{N}_2$,

$$\begin{aligned} f_{\text{UAV}}^*[n] &= \arg\min_{f_{\text{UAV}}[n]\ge 0} \left(\delta\kappa f_{\text{UAV}}^3[n] + \frac{\delta f_{\text{UAV}}[n]}{c} \sum_{i=n}^{N} (\alpha_i^* - \lambda_i^*) \right) \\ &= \sqrt{\frac{\theta_n}{3\kappa c}}, \end{aligned} \tag{57}$$

where $\theta_n = \sum_{i=n}^{N} (\lambda_i^* - \alpha_i^*)$. Clearly, $f_{\text{UAV}}^*[n]$ changes only when θ_n changes its value, and θ_n changes only when α_n^* or λ_n^* is positive at any slot n.

From the complementary slackness conditions (55) and (56), $\alpha_n^* > 0$ indicates that data causality constraint is met with equality, and $\lambda_n^* > 0$ indicates that deadline constraint is met with equality. If $\alpha_n^* > 0$, we have $\theta_{n+1} - \theta_n = \alpha_n > 0$, which indicates that $f_{\text{UAV}}^*[n+1] > f_{\text{UAV}}^*[n]$. Similarly, if $\lambda_n^* > 0$, we have $\theta_{n+1} - \theta_n = -\lambda_n < 0$, which indicates that $f_{\text{UAV}}^*[n+1] < f_{\text{UAV}}^*[n]$. As a result, we can conclude that the CPU frequency increases after a slot n when $\sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c} = \sum_{i=1}^{n-1} \sum_{k=1}^{K} l_k[i]$, and decreases after a slot n when $\sum_{k\in\Omega_n} \sum_{i=n_k'}^{n_k''-1} l_k[i] = \sum_{i=2}^{n} \frac{\delta f_{\text{UAV}}[i]}{c}$.

References

[1] Y. Xu, J. Yu, and R. M. Buehrer, "The application of deep reinforcement learning to distributed spectrum access in dynamic heterogeneous environments with partial observations," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4494–4506, Jul. 2020.

[2] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[3] 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.

[4] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart., 2014.

[5] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.

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

[7] 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.

[8] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 2068–2085, Jul. 2011.

[9] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1377–1386, Jun. 2016.

[10] N. Zhao *et al.*, "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.

[11] S. Hu, Q. Wu, and X. Wang, "Energy management and trajectory optimization for UAV-enabled legitimate monitoring systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 142–155, Jan. 2021.

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[13] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.

[14] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2019.

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

[16] 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.

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

[18] M. Hua, Y. Wang, C. Li, Y. Huang, and L. Yang, "UAV-aided mobile edge computing systems with one by one access scheme," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 3, pp. 664–678, Sep. 2019.

[19] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[20] J. Zhang *et al.*, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[21] 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, Sep. 2018.

[22] 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.

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

[24] 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.

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

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

[27] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020.

[28] *LTE Unmanned Aircraft Systems Trial Report*, Qualcomm, San Diego, CA, USA, 2017.

[29] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[30] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[31] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, no. 2, pp. 203–221, 1996.

[32] Y. Hu, X. Yuan, J. Xu, and A. Schmeink, "Optimal 1D trajectory design for UAV-enabled multiuser wireless power transfer," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5674–5688, Aug. 2019.

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

[34] Z.-Q. Luo, W.-K. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[35] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for non-convex mathematical programs," *Oper. Res.*, vol. 26, no. 2, pp. 681–683, Aug. 1978.

[36] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Minnesota, Minneapolis, MN, USA, 2014.

[37] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[38] X. Wang and Z. Li, "Energy-efficient transmissions of bursty data packets with strict deadlines over time-varying wireless channels," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2533–2543, May 2013.

[39] X. Wang, Z. Nan, and T. Chen, "Optimal MIMO broadcasting for energy harvesting transmitter with non-ideal circuit power consumption," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2500–2512, May 2015.

[40] X. Chen, W. Ni, X. Wang, and Y. Sun, "Optimal quality-of-service scheduling for energy-harvesting powered wireless communications," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3269–3280, May 2016.

[41] 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.

Chao Sun (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering from Shandong Agricultural University, China, in 2015, and the M.Eng. degree in communication engineering from Lanzhou University, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Communication Science and Engineering, Fudan University, Shanghai, China. His research interests include mobile edge computing, unmanned aerial vehicle, intelligent reflecting surface, and signal processing.

Wei Ni (Senior Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively.

He is currently a Group Leader and a Principal Research Scientist with CSIRO, Sydney, NSW, Australia, and an Adjunct Professor with the University of Technology Sydney and an Honorary Professor with Macquarie University, Sydney, NSW, Australia. He was a Post-Doctoral Research Fellow with Shanghai Jiaotong University from 2005 to 2008, the Deputy Project Manager with the Bell Labs, Alcatel/Alcatel-Lucent from 2005 to 2008, and a Senior Researcher with Devices Research and Development, Nokia, from 2008 to 2009. His research interests include signal processing, optimization, learning, and their applications to network efficiency and integrity. He has been the Chair of the IEEE Vehicular Technology Society (VTS) New South Wales (NSW) Chapter since 2020 and an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2018. First, he served as the Secretary and then Vice-Chair for IEEE NSW VTS Chapter from 2015 to 2019, the Track Chair for VTC-Spring 2017, the Track Co-Chair for IEEE VTC-Spring 2016, the Publication Chair for BodyNet 2015, and the Student Travel Grant Chair for WPMC 2014.

Xin Wang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree from Auburn University, Auburn, AL, USA, in 2004, all in electrical engineering.

From September 2004 to August 2006, he was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. In August 2006, he joined the Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL, USA, as an Assistant Professor, where he was promoted to a tenured Associate Professor in 2010. He is currently a Distinguished Professor and the Chair of the Department of Communication Science and Engineering, Fudan University. His research interests include stochastic network optimization, energy-efficient communications, cross-layer design, and signal processing for communications. He is a member of the Signal Processing for Communications and Networking Technical Committee of IEEE Signal Processing Society and a Distinguished Lecturer of the IEEE Vehicular Technology Society. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE SIGNAL PROCESSING LETTERS and an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is also a Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.