Federated Learning via Unmanned Aerial Vehicle

Min Fu\textsuperscript{a}, Member, IEEE, Yuanming Shi\textsuperscript{a}, Senior Member, IEEE, and Yong Zhou\textsuperscript{a}, Senior Member, IEEE

Abstract—Federated learning (FL) has emerged as a promising alternative to centralized machine learning for exploiting large amounts of data generated by networks while ensuring data privacy. Unlike previous FL works that rely on terrestrial base stations, this paper studies an unmanned aerial vehicle (UAV)-assisted FL system where a UAV collects local models from distributed ground devices. By leveraging the UAV’s high altitude and mobility, it can proactively establish short-distance line-of-sight links with devices to mitigate the communication straggler effect and improve communication efficiency in FL. Specifically, we present the convergence analysis of FL without convexity assumptions, demonstrating the effect of device scheduling on the global gradients. Based on the derived convergence bound, we aim to minimize the completion time of FL training by jointly optimizing device scheduling, UAV trajectory, and time allocation. This problem explicitly incorporates the devices’ energy budgets, dynamic channel conditions, and convergence accuracy of FL constraints. Despite the non-convexity of the formulated problem, we exploit its structure to decompose it into two sub-problems and further derive the closed-form solutions via the Lagrange dual ascent method. Simulation results show that the proposed design significantly improves the tradeoff between completion time and test accuracy compared to existing benchmarks.

Index Terms—Federated leaning, UAV communications, completion time minimization, device scheduling, UAV trajectory design.

I. INTRODUCTION

As the storage and computation capabilities of edge devices keep growing, it becomes more attractive to process the data locally and push network computation to the edge [1], [2], [3]. In the field of machine learning (ML), distributed learning frameworks [4], [5] that keep the training data locally are well developed to protect data privacy and reduce network energy/time costs. Recently, federated learning (FL) [1], [4], [5] has been proposed as a promising solution for distributed ML, which enables multiple devices to execute local training on their own dataset and collaboratively build a shared ML model with the coordination of a parameter server (PS) (e.g., access point and base station). Since only model parameters rather than raw data are exchanged between devices and the PS, FL significantly relieves the communication burden and protects data privacy [4], [6] with wide-field applications, e.g., vehicle-to-vehicle communications [7] and content recommendations for smartphones [5].

In contrast to the centralized ML, the PS in FL needs to exchange models with multiple devices over hundreds to thousands of communication rounds to achieve the desired training accuracy. However, the main challenge in realizing FL on wireless networks arises from communication stragglers with unfavorable links [8], [9]. For example, in over-the-air computation (AirComp)-based analog FL [6], [10], communication stragglers dominate the overall model aggregation error caused by channel fading and communication noise since the devices with better channel qualities have to reduce their transmit power for the local models’ alignment at the PS. Moreover, in digital synchronous FL [8], [11], communication stragglers significantly slow down the model aggregation process and dominate cumulative communication delay since the PS must wait until receiving the training updates from all participants. If the number of communication stragglers is high, the overall communication delay will be unacceptable. The straggler issue is thus the main bottleneck to design communication-efficient FL systems.

There have been many efforts to mitigate the communication straggler effect in FL, such as device scheduling [6], [10], [11]. For instance, to reduce model misalignment error incurred by stragglers in AirComp-based FL, the authors in [6] and [10] scheduled the devices with reliable channels for concurrent model uploading. To reduce the communication delay incurred by stragglers in digital FL, devices with large contributions to the global model [11] and/or with favorable channel conditions [12], [13] are generally selected. In addition, to minimize an FL loss function, the authors in [14] optimized the device scheduling by considering wireless network parameters such as packet errors and the availability of wireless resources. The authors in [15] presented an overview of practical distributed FL techniques and their interplay with advanced communication optimization designs. Nevertheless, because such device scheduling may be biased, which results in a smaller amount of training data utilized, this, in turn, may damage the update of the global model and decrease the learning performance of FL. To alleviate such communication-learning tradeoff, recent research has investigated the integration of advanced technologies (i.e., relays [16], reconfigurable intelligent
surfaces [17], [18], [19], [20]) into FL systems to improve stragglers’ communication qualities and thus further upgrade device scheduling policy for the reduction of communication errors. These existing frameworks require a terrestrial BS to provide network coverage to the devices for model aggregation. However, many FL tasks need to be performed under the circumstances when terrestrial networks are unavailable in remote areas. For example, devices from multiple regions (e.g., forests and woodlands) can collaborate through FL to build a learning model for fire monitoring [21]. Under these harsh environments, it is imperative to deploy a more flexible PS that proactively establishes favorable communication links among devices.

As a viable complementary alternative to terrestrial networks, unmanned aerial vehicles (UAVs) can provide coverage extension and seamless connectivity to support various FL tasks, especially in distant and underdeveloped areas [22], [23], [24], [25]. Inspired by this, this paper studies a UAV-enabled FL network, where a UAV is dispatched as a flying PS to aggregate and update the digital FL model parameters when no terrestrial BS is available. To mitigate the communication straggler effect in UAV-enabled FL networks, we propose to jointly design UAV trajectory and device scheduling. First, the qualities of communication channels between the UAV and devices still differ since all links’ channel conditions depend on the UAV’s location at each time slot. To address this issue, device scheduling is necessary to prevent communication stragglers. Furthermore, by utilizing its mobility, the UAV can establish short-distance LoS links to scheduled devices. As a result, each scheduled device achieves a high data rate for model uploading, resulting in faster model uploads/downloads per round compared to the static UAV. In addition, with its ability to dynamically adjust communication distances, the UAV can prevent any device from being a communication straggler all the time, ensuring that all devices have the opportunity to participate in FL training. Hence, incorporated with UAV trajectory planning, the optimized device scheduling strategy focuses on data exploitation maximization, thereby reducing cumulative model aggregation loss and accelerating FL convergence. Such joint design in UAV-enabled FL networks is an appealing solution to mitigate the straggler effect.

However, the UAV-enabled FL system also faces new challenges. In particular, UAVs usually have a limited endurance due to the practical physical constraints (e.g., 30 minutes for the typical rotary-wing UAV [26]), and need to be recalled for battery swap or recharging. Therefore, it is imperative to minimize the completion time for the FL training process (i.e., the flight time of the UAV). It is worth noting that for rotary-wing UAVs, minimizing completion time of UAVs will effectively improve their energy efficiency, particularly when UAVs in hovering status. There are fundamental tradeoffs between completion time and convergence accuracy as well as the energy consumption of devices in the UAV-enabled FL system. Assuming that each device transmits its local model parameter only when the UAV is closer to it, then the energy consumption of each device is reduced. However, due to the longer flying distance required to fly closer to each device, this will result in a longer mission completion time. In addition, the higher the target FL convergence accuracy, the more devices and communication rounds are required, resulting in a longer mission completion time. To balance such tradeoffs, this paper aims to minimize the completion time for the UAV-enabled FL by jointly designing the UAV trajectory and device scheduling together with time allocation, while considering the energy budget of devices and ensuring that the FL algorithm can converge to a target accuracy.

A. Contributions

In this paper, we present a novel UAV-enabled FL system, where a mobile UAV is dispatched as a PS to exchange model parameters with devices. This framework is particularly valuable in scenarios where terrestrial BSs are unavailable. This paper also develops the joint optimization of UAV trajectory and device selection, which could effectively alleviate the communication straggler effects on complete time and improve the energy efficiency of the UAV. The main contributions of this paper are summarized as follows.

- We analyze the convergence of the UAV-enabled FL algorithm, taking into account non-convex loss functions and model update errors caused by device selection in all rounds. The analytical convergence expression shows that, besides the number of selected devices, their data sizes also affect the convergence gap of the FL algorithm.
- Different from the existing works on delay consideration of FL, we formulate a new completion time minimization problem by jointly optimizing device scheduling, time allocation, and UAV trajectory, which incorporates the practical learning and communication constraints on the target convergence accuracy of FL and energy budget of devices as well as the mobility constraints of UAV. By further analyzing the formulated problem, we shed light on the tradeoff between the completion time and the target convergence accuracy due to the limited energy budgets.
- We develop a unified optimization framework to solve the formulated mixed-integer non-convex problem, based on the principle of block coordinate descent and Lagrange duality, which can update each variable in closed form. This approach is computationally efficient, making it useful for practical implementation.

Simulation results in realistic federated settings are provided to illustrate the learning-communication tradeoff and validate the effectiveness of the proposed scheme. Specifically, the proposed joint design scheme outperforms existing benchmark schemes in terms of mission completion time, which is appealing in light of the limited endurance of UAVs. Moreover, our proposed scheme provides comparable performance to the full-scheduling ideal benchmark in terms of prediction accuracy, even when the target convergence accuracy of FL is relatively large.

B. Related Work

1) UAV Assisted FL Networks: Upon the completion of this work, the application of UAV in FL networks was investigated in some parallel works [27], [28], [29], [30]. Specifically, the
authors in [27] developed a novel FL framework with UAV swarms to improve the FL learning efficiency. The authors in [28] proposed an FL-based sensing and collaborative learning approach for UAV-enabled internet of vehicles (IoVs), where UAVs as devices collect data and train ML models for IoVs. In addition, [29] described using UAVs as flight relays to support wireless communication between IoVs and the FL server, thus enhancing FL accuracy. The authors in [30] studied the deployment of multiple UAVs as flying BSs to minimize the weighted sum of FL execution time and function loss. Nevertheless, this work does not consider device’s energy budget issues that may affect FL performance and convergence cannot be guaranteed.

2) Completion Time Minimization Problems in FL Networks: A few works have studied the completion time minimization problems in different FL scenarios. For example, the authors in [31] formulated an FL framework over a wireless network as an optimization problem that minimizes the sum of FL aggregation latency and total device energy consumption. In addition, the authors in [32] investigated the tradeoff between the FL convergence time and devices’ energy consumption. However, in [31] and [32], all devices are assumed to be involved in each round. There exists a handful of studies that focus on the completion time minimization problem of FL with device scheduling [11], [33], [34]. For example, in [33], the authors developed a multi-armed bandit-based framework for online device scheduling to minimize the completion time of the FL training process, which ignores communication resource constraints. The authors in [11] optimized the bandwidth allocation and device scheduling, selecting only the devices with significant local model contributions. Moreover, the author in [34] studied a semi-asynchronous FL mechanism, which optimizes the number of participating devices to minimize the training completion time. Nonetheless, all the above works assumed that a static terrestrial BS is available and failed to take into account the limited energy of devices.

C. Organization

The remainder of this paper is organized as follows. Section II describes the FL via the UAV system model. In Section III, we provide the convergence analysis of FL and completion time minimization problem formulation. In Section IV, we propose a BCD method to solve the formulated problem. Section V presents the numerical results to evaluate the performance of the proposed algorithm. Finally, we conclude this paper in Section VI.

Notations: In this paper, scalars, column vectors and matrices are written in italic letters, boldfaced lower-case letters and boldfaced upper-case letters respectively, e.g., \( a \), \( a \) \( A \). \( \| a \| \) denotes the Euclidean norm of vector \( a \) and \( a^T \) represents its transpose. \( |S| \) denotes the cardinality of the set \( S \). \( \langle \cdot, \cdot \rangle \) represents the inner product. The important notations used throughout the paper are listed in Table I.

| Notation    | Description                      | Notation    | Description                      |
|-------------|----------------------------------|-------------|----------------------------------|
| \( k \)    | Set of all \( K \) devices       | \( N \)     | Set of all \( N \) communication slots |
| \( T \)    | Mission duration (s)             | \( \delta \) | Duration of time slot \( n \) |
| \( q[n] \) | Horizontal coordinate of UAV at time slot \( n \) | \( h_k[n] \) | Channel coefficient between UAV and device \( k \) at time slot \( n \) |
| \( \alpha_k[n] \) | Scheduling for device \( k \) at time slot \( n \) | \( \tau_k[n] \) | Allocated communication duration for device \( k \) at time slot \( n \) |
| \( E_{\text{comm}}^k[n] \) | Energy consumption of device \( k \) at time slot \( n \) for communication | \( E_{\text{comp}}^k[n] \) | Energy consumption of device \( k \) at time slot \( n \) for computation |
| \( E_k \) | Total energy budget of device \( k \) | \( c_k \) | Computation time of device \( k \) |
| \( f_k \) | Horizontal coordinate of device \( k \) | \( e \) | Convergence threshold of FL |
| \( D_k \) | Global model vector at round \( n \) | \( D_k \) | Local dataset size of device \( k \) |
| \( \gamma \) | Bound of sample-wise gradient norm | \( \gamma \) | Learning rate of FL |

![Fig. 1. FL over UAV-enabled wireless networks.](image)
Algorithm 1 FL Over UAV-Enabled Wireless Networks

1: Input: $\eta$, $w_0 = 0$, and $n = 0$;
2: repeat
3:   Update $n = n + 1$.
4:   Device Scheduling: Denote the set of scheduled devices $k$ in the $n$-th round as $A_n$.
5:   Computation: All devices in set $A_n$ receive $w_{n-1}$ from the server, and update their local FL models denoted as $w_{k,n}$, $k \in A_n$ according to (4) in parallel.
6:   Communication: All the scheduled devices $k \in A_n$ transmit $w_{k,n}$ to the UAV over TDMA.
7:   Aggregation and Broadcast: The UAV updates the global model $w_n$ as in (5), and then broadcast the updated global model to devices.
8: until $n = N$
9: Output: the global model $w_N$

sequel, we describe FL over UAV-enabled wireless networks, as summarized in Algorithm 1.

A. FL Training Model

Let $w \in \mathbb{R}^d$ and $w_k \in \mathbb{R}^d$ be the global model parameters at the UAV and the local model parameters at the $k$-th device, respectively. The local loss function of device $k$ is defined as

$$F_k(w) = \frac{1}{D_k} \sum_{i=1}^{D_k} f(w; x_{ki}, y_{ki}), \forall k \in K.$$  (1)

Accordingly, the global loss function at the UAV is given by

$$F(w) := \frac{1}{D} \sum_{k=1}^{K} D_k F_k(w),$$  (2)

where $D$ denotes the total size of data with $D = \sum_{k=1}^{K} D_k$.

The training of an FL algorithm aims to solve the following optimization problem:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{D} \sum_{k=1}^{K} \sum_{i=1}^{D_k} f(w; x_{ki}, y_{ki}) \\
\text{subject to} & \quad w_1 = \cdots = w_K = w,
\end{align*}$$  (3)

where constraint (3) is used to ensure that all devices and the server have the same model after the FL algorithm converges. To solve problem (3), the training procedure of FL consists of multiple communication rounds. Therein, in a communication round, the devices update the local models based on their own data and a common model initialization from the PS. After that, they contribute to the global model at the PS by uploading model updates over the wireless channel, as described below.

First, the UAV determines the set of devices that participate in the current round. Let $a_k[n] \in \{0, 1\}$ denote the scheduling variable for device $k$ in the $n$-th round. Therein, $a_k[n] = 1$ indicates that device $k$ sends its updated local model to the UAV at communication round $n$; otherwise we have $a_k[n] = 0$. $A_n$ is defined as the set of scheduled devices in the $n$-th round.

Then, the UAV broadcasts the current global model, denoted by $w_{n-1}$, to all the scheduled devices.

Each scheduled device $k \in A_n$ receives the global model and updates its local model by applying the gradient descent algorithm [8], [11] on its local dataset:

$$w_{k,n} = w_{n-1} - \frac{\eta}{D_k} \sum_{(x_{ki}, y_{ki}) \in D_k} \nabla f(w_{n-1}; x_{ki}, y_{ki}),$$  (4)

where $\eta$ is the learning rate and $\nabla f(w_{n-1}; x_{ki}, y_{ki})$ is the gradient of $f(w; x_{ki}, y_{ki})$ with respect to $w$ at $w_{n-1}$.

After all scheduled devices upload their local models, the UAV aggregates them to obtain the following consolidated global model

$$w_n = \frac{\sum_{k=1}^{K} a_k[n] w_{k,n}}{\sum_{k=1}^{K} a_k[n]},$$  (5)

where $w_n$ is the global model updated at the UAV in the $n$-th round. With (4) and (5), the UAV and the devices can repeatedly update their models until the maximum number of communication rounds reaches. As the local FL models are updated at the scheduled devices and transmitted over wireless cellular links, device scheduling determines both the learning performance and communication delay. Furthermore, the channel conditions between the UAV and devices vary over different time slots due to the mobility of the UAV. Hence, device scheduling should be optimized along with the time allocation and UAV trajectory to account for time-varying channel conditions and the target convergence accuracy required of FL. In the next subsection, we elaborate on the communication and computation models of FL over UAV-enabled networks.

B. UAV-Enabled Signal Transmission

All ground devices are assumed to be fixed to the ground, and their locations are known by the UAV. The horizontal coordinate of ground device $k$ is denoted as $u_k = [x_k, y_k] \in \mathbb{R}^{1 \times 2}$ with $x_k$ and $y_k$ being $x$- and $y$-coordinates, respectively. In this paper, we suppose that the total number of communication rounds denoted by $N$ is fixed. We focus on the time required for the UAV to complete the FL process, wherein the corresponding variable is denoted as $T$. Energy minimization of UAV is a also critical design consideration in UAV-enabled systems [26], but addressing this problem goes beyond the scope of our current study and will be the focus of future work. However, it is worth noting that for rotary-wing UAVs, minimizing completion time may enhance its energy efficiency, particularly when the UAV is in a hovering status. In a three-dimensional (3D) Cartesian coordinate system, the location of the UAV at time $t$ projected on the horizontal (ground) plane is denoted as $q(t) = [x(t), y(t)] \in \mathbb{R}^{1 \times 2}$, $0 \leq t \leq T$ with $x(t)$ and $y(t)$ being $x$- and $y$-coordinates at time instant $t$, respectively. We assume that the UAV flies at a fixed altitude $H$ above ground level and starts the mission at an initial location denoted as $q_I = [x_I, y_I] \in \mathbb{R}^{1 \times 2}$. Note that in practice, $H$ corresponds to the minimum altitude that ensures obstacle avoidance without the need for frequent aircraft ascending and descending, and the initial location is determined by various
at time slot $0 < t < T$ as shown in Fig. 2.

To facilitate UAV trajectory design, we adopt the time discretization technique to tackle the continuous UAV trajectory, which is widely applied in most existing works [24], [25], [35]. Specifically, the mission duration $T$ is divided into $N$ unequal time slots, with the $n$-th slot duration expressed as $\delta[n]$ and $\sum_{n=1}^{N} \delta[n] = T$, where $\delta[n]$ equals the sum of the computing time, model uploading time, and model download time in round $n$. Moreover, to make the time discretization effective, we assume that the maximum distance that the UAV can move in each time slot cannot exceed a maximum distance denoted as $\Delta q_{\text{max}}$, which satisfies $\Delta q_{\text{max}} \ll H$ so that the distance between the UAV and devices is approximately constant during each time slot. Based on the time discretization technique, the UAV trajectory over time $T$ is approximated by the $(N + 1)$-length sequence $\{q[n]\}_{n=0}^{N}$ with $q[n]$ denoting the UAV’s horizontal coordinate at time slot $n$. The UAV’s mobility constraints are given by

$$||q[n] - q[n - 1]||_2 \leq \min\{V_{\text{max}}\delta[n], \Delta q_{\text{max}}\}, n = 1, \ldots, N,$$

(6)

$$q[0] = q_I,$$

(7)

where (6) and (7) correspond to the UAV speed constraint and its initial location constraint, respectively. In additions, we assume that the link between each device and the UAV is dominated by the LoS channel because UAVs usually fly at high altitudes for safety reasons, as most existing studies on UAVs [24], [25], [36]. Following the free-space path loss model [26], the channel gain between device $k$ and the UAV at time slot $n$ is given by

$$h_k[n] = \beta_0 d_k^{-2}[n] = \frac{\beta_0}{H^2 + ||q[n] - u_k||_2^2},$$

(8)

where $\beta_0$ represents the channel gain at the reference distance of 1 meter and $d_k[n]$ is the distance between device $k$ and the UAV at time slot $n$.

In the following, we elaborate on the process of FL over UAV-enabled wireless transmission during each time interval $\delta[n]$, which consists of four stages, namely, the local computation stage, the local model uploading stage, the global computation stage, and the global model downloading stage, as shown in Fig. 2.

1) Local Computation: We denote the required number of processing cycles for computing one data sample at device $k$ by $c_k$, which can be measured offline and is known as a prior. Since all samples $\{x_{k,i}, y_{k,i}\}_{i \in D_k}$ have the same size (i.e., number of bits), the number of CPU cycles required for device $k$ to run one local round is $c_k D_k$. By denoting the CPU-cycle frequency of the device $k$ by $f_k$, the computation time per round of device $k$ is given by

$$t_{\text{comp}}[n] = \frac{c_k D_k}{f_k}.$$

(9)

The CPU energy consumption of device $k$ for the $n$-th round of local computation is [31]

$$E_{\text{comp}}^k[n] = a_k [n] \sum_{i=1}^{c_k D_k} \frac{\alpha_k}{2} f_k^2 = a_k [n] \frac{\alpha_k}{2} c_k D_k f_k^2,$$

(10)

where $\alpha_k/2$ is the effective capacitance coefficient of device $k$’s computing chipset. Both synchronous and asynchronous FL schemes are used to update global parameter, while the former is more widely used due to its simplicity and guaranteed FL convergence capability for resource-constrained devices. Therefore, the synchronous FL aggregation technique is adopted in this paper as in most existing FL studies [11], [12], [14], [31]. In particular, we consider the synchronous operation that requires all scheduled devices to simultaneously train their local models and complete their training before entering the communication phase. Thus, the time cost for local model training at the $n$-th round is given by [31]

$$t_{\text{comp}}[n] = \max_k \{a_k [n] t_{\text{comp}}[n] \}.$$

(11)

Hence, the computing time in one local round is determined by scheduled devices with large date sizes and low CPU frequency.

2) Local Model Uploading: In the local model uploading stage, to avoid interference among devices during the uploading process, TDMA [31] is adopted as shown in Fig. 2. Specifically, in the local model uploading stage, $\{A_k\}$ devices send their respective local models one by one during slot $n$. The achievable data rate (bit/s) of devices $k$ at the $n$-th round is defined as

$$r_k[n] = B \log_2 \left(1 + \frac{p_k[n] h_k[n]}{\sigma^2} \right),$$

(12)

where $B$ is the system bandwidth in Hertz. $p_k[n]$ is the transmit power of device $k$ at the $n$-th round, and $\sigma^2$ is the power of the additional complex Gaussian noise. Let $\tau_k[n]$ denote the duration in which device $k$ transmits its local model to the UAV at the $n$-th round. We assume that the dimension of the model parameter vector is fixed throughout the FL process, denoted by $s$ (in bits). The uploading time of device $k$ can be calculated as

$$\tau_k[n] = \frac{a_k[n] s}{r_k[n]}.$$

(13)

With a given data size, the larger the channel gain, the shorter the required uploading time. The UAV can shorten the uploading time of scheduled devices by establishing LoS connections and shortening communication distances with the
devices. By combining (12) and (13), to transmit $s$ within a time duration $\tau_k[n]$, the device $k$’s energy consumption is

$$E_k^{\text{comp}}[n] = \frac{\tau_k[n]s^2}{h_k[n]} \left(2^{\frac{\log_2{s}}{\tau_k[n]}} - 1\right). \quad (14)$$

Note that the time allocation optimization in the TDMA setting may be translated into bandwidth allocation optimization in the FDMA scheme, which is left to be our future work.

After all scheduled devices upload their local training models $\{w_k[n]\}$ at the $n$-th slot, the UAV performs the global model update and broadcasts the updated result of length $s$ bits to the devices. Let $f_0$ denote the UAV processor’s fixed computing speed, $c_0$ denote the required number of processing cycles for one local model at the UAV, and $P_0$ denote the transmit power of the UAV. The time that spent on updating global model and feeding the result back to devices at the UAV is expressed as

$$t_{\text{UAV}}[n] = \frac{Kc_0}{f_0} + \sum_{k \in A_n} \frac{s}{\min_{k \in A_n} B \log_2 \left(1 + \frac{P_0 h_k[n]}{\sigma_n^2}\right)} \quad (15)$$

One aspect, comparing to the time of local model computation at any device $k$ denoting $\frac{D_k c_0}{f_0}$, the first term in (15) is negligible. The reasons are that the UAV has a higher computation capability $f_0$ than that of the devices $f_k$ [36]. Moreover, the number of devices $K$ is much smaller than the size of the local dataset $D_k$. The other aspect, the only need to broadcast the global model to devices while all the scheduled devices upload their local models via TDMA. Accordingly, we can infer that $t_{\text{UAV}}[n] \ll \sum_{k = 1}^K \tau_k[n] + t_{\text{comp}}[n]$, and the global computation time and the global model downloading time at the UAV are neglected as in [31] and [36]. Thus, the total uploading time of devices can occupy the rest of the time after local model training at the $n$-th round, i.e., $\sum_{k = 1}^K \tau_k[n] \leq \delta[n] - t_{\text{comp}}[n], \forall n$.

### III. CONVERGENCE ANALYSIS AND PROBLEM FORMULATION

In this section, we conduct the convergence analysis of UAV-enabled FL, which characterizes the impact of key system parameters on the convergence performance, and then formulate a completion time minimization problem, taking into account communication resource constraints, UAV deployment constraints, and target convergence accuracy of FL.

#### A. Convergence Analysis of FL

We follow the literature and make two standard assumptions on the loss function and local gradients as follows [19], [37], [38], [39], [40], [41], [42], [43].

**Assumption 1 (Smoothness):** The function $F: \mathbb{R}^d \to \mathbb{R}$ is $L$- smooth. That is, $\forall w, w' \in \mathbb{R}^d$,

$$F(w) \leq F(w') + \langle \nabla F(w'), w - w' \rangle + \frac{L}{2} \|w - w'\|^2_2. \quad (16)$$

And the loss function does not have to satisfy the convexity assumption, and only needs to be lower-bounded, which is the minimal assumption required for convergence [40], [41].

**Assumption 2 (Bounded Loss Function):** For any parameter vector $w$, the loss function $F(w)$ is lower-bounded by $F^*$.

The following assumption is also standard in the literature [19], [42], [43].

**Assumption 3 (Bounded Sample-Wise Gradient Norm):** For a constant $\kappa > 0$, the sample-wise gradients at local devices are bounded

$$\left\|\nabla f(w_n; x, y)\right\|_2^2 \leq \kappa. \quad (17)$$

We use the average gradient norm as an indicator of convergence for FL, which is widely adopted in the convergence analysis for non-convex loss function [38], [44]. According to the above assumptions, the average gradient norm can be bound, explained as follows.

**Theorem 1:** Suppose that the loss functions satisfy assumptions 1-3, given the learning rate $0 \leq \eta \leq \frac{1}{L}$, after $N$ rounds, the average norm of the global gradients is upper bounded by

$$\frac{1}{N} \sum_{n=0}^{N-1} \left\|\nabla F(w_n)\right\|^2 \leq \frac{2}{N\eta} (F(w_0) - F^*) + \frac{4K\kappa}{ND^2} \sum_{n=0}^{N-1} \sum_{k=1}^K (1 - a_k[n + 1])D_k^2. \quad (18)$$

**Proof:** Please refer to Appendix A. $\square$

Given rounds $N$, the FL algorithm achieves an $\epsilon$-approximation solution if [38]

$$\frac{1}{N} \sum_{n=0}^{N-1} \left\|\nabla F(w_n)\right\|^2 \leq \epsilon, \quad (19)$$

where $\epsilon > 0$ is the convergence threshold.

**Remark 1:** Expression (18) comprises two terms, i.e., the initial optimality gap and the time-average aggregation error resulting from the effect of device scheduling. For the first term, when the number of communication rounds, the learning step-size, and the initial model parameter are given, the initial optimality gap is a constant. For the second term, the time-average aggregation error decreases as the number of scheduled devices increases, particularly when large data-size devices are scheduled. However, scheduling more devices in one round for model uploading increases the communication burden and may significantly slow down the model aggregation process, especially if there are communication stragglers. Thus, it is crucial to schedule a proper subset of devices to balance training performance and time consumption.

#### B. Problem Formulation

Let $A = \{a_k[n], \forall n \in \mathcal{N}, \forall k \in \mathcal{K}\}$, $\Gamma = \{\tau_k[n], \forall n \in \mathcal{N}, \forall k \in \mathcal{K}\}$, $Q = \{q[n], \forall n = 0, \ldots, N\}$, and $\delta = \{\delta[n], \forall n = 1, \ldots, N\}$. In this paper, given the number of communication rounds $N$, we aim to minimize the completion time of the FL training process under the target convergence accuracy requirements by jointly optimizing the devices’ scheduling variables $A$, time allocation $\Gamma$, UAV trajectory $Q$, and slot intervals $\delta$. To satisfy the convergence requirement in (19), we ensure that the upper bound in Theorem 1 is less than the initial optimality gap.
than $\epsilon$. Therefore, the corresponding optimization problem is formulated as

\[
\begin{align*}
\text{minimize} & \quad \sum_{n=1}^{N} \delta[n] \\
\text{subject to} & \quad a_k[n] \in \{0, 1\}, \quad \forall k \in K, \quad \forall n \in \mathcal{N}, \quad (20a) \\
& \quad \sum_{n=1}^{N} (E^\text{comm}_k[n] + E^\text{comp}_k[n]) \leq E_k, \quad \forall k \in K, \quad (20b) \\
& \quad \sum_{k=1}^{K} T_k[n] + \max_{k} a_k[n] t^\text{comp}_k \leq \delta[n], \quad \forall n \in \mathcal{N}, \quad (20c) \\
& \quad 2 \frac{N\kappa}{N} (F(w_0) - F^*) \\
& \quad + \frac{4\kappa K}{ND^2} \sum_{n=0}^{N-1} \sum_{k=1}^{K} (1 - a_k[n + 1]) D_k \leq \epsilon, \quad (20d) \\
& \quad \text{Constraints (6), (7).} \quad (20e)
\end{align*}
\]

Constraints (20a) are integer constraints with respect to the devices’ scheduling variables. Constraints (20b) represent the devices’ energy budgets. Constraints (20c) ensure that the communication and computation time per round cannot exceed the duration of each time slot. Constraint (20d) denotes the target convergence accuracy requirement of FL. Problem (20) is challenging to solve due to the following reasons. First, the optimization variables $A$ for device scheduling are binary and thus (20a)-(20d) involve integer constraints. Second, $E^\text{comm}_k[n]$ in constraints (20b) are not jointly convex with respect to the optimization variables $A, \Gamma, Q,$ and $\delta$. Therefore, problem (20) is a mixed-integer non-convex problem, which is difficult to be optimally solved in general.

C. Problem Analysis

In this subsection, we take a close look at problem (20) and compare our formulation with those in the existing literature. Furthermore, we discuss properties of the considered problem formulation and emphasize the necessity of introducing flying UAVs into the FL system.

The existing works on wireless FL with static terrestrial BS [11], [33], [34] have made attempts to minimize the completion time under the convergence accuracy requirements by optimizing device scheduling. Comparing our proposed formulation with the frameworks in [11], [33], and [34], the following differences can be seen:

- Different from [11], [33], and [34], this paper investigated a UAV-enabled FL network, where a UAV is dispatched as a flying PS to aggregate and update the FL model parameters when no terrestrial BS is available.
- This paper not only optimizes device scheduling for completion time minimization but also takes into account the practical device’s energy budget and UAV trajectory design, the latter of which are not studied in [11], [33], and [34].
- Our formulated problem explicitly integrates the target convergence accuracy requirement into the completion time minimization, which is not directly shown in the previous works [11], [33], [34].

Prior to solving problem (20), we should check its feasibility. According to (18), when all devices are scheduled in every communication round, the convergence bound reaches the minimum value. Therefore, if the minimum value of the convergence bound is larger than the target convergence accuracy, problem (20) is infeasible. Moreover, if $E_k, \forall k$ is not big enough, the constraints in (20b) may not be satisfied even though the scheduled devices consume the lowest amount of energy. To sum up, the following proposition is given to check the feasibility of problem (20).

**Proposition 1:** Problem (20) is feasible if and only if

\[
N \geq \left[ \frac{2(F(w_0) - F^*)}{\epsilon \eta} \right] + \frac{\sum_{k=1}^{K} E_k}{|\mathcal{S}|} \geq \left[ \frac{1}{\beta_0 B} \right], \quad (21)
\]

where $|\mathcal{S}|$ denotes the required amount of scheduled devices during training process, given by $|\mathcal{S}| = \lceil KN - (\epsilon - 2) \frac{N^D (F(w_0) - F^*)}{4\kappa N \max_{k} D_k} \rceil$.

**Proof:** Please refer to Appendix B. □

We show the effect of the target convergence accuracy of FL on the completion time.

**Theorem 2:** For problem (20), the objective value is non-increasing with $\epsilon$.

**Proof:** Please refer to Appendix C. □

Theorem 2 sheds light on the tradeoff between the completion time of FL and the target convergence accuracy of FL due to the limited energy budgets. This is because imposing more stringent convergence accuracy requirements increases the required amount of scheduled devices in the FL training process, which can be observed from constraint (20d). From (20), we see that time used for the update of the local models depends on the device scheduling matrix $A$ and UAV trajectory $Q$. Fortunately, by exploiting the UAV’s mobility, UAV can dynamically establish favorable connections with scheduled devices in each round to mitigate the communication straggler issue, thereby reducing the completion time of the FL.

IV. DEVICE SCHEDULING AND TIME ALLOCATION ALONG WITH TRAJECTORY DESIGN

In this section, we present the BCD-LD method to design the device scheduling, time allocation, and UAV trajectory. Specifically, we first optimize the device scheduling and time allocation for a given UAV trajectory. Given device scheduling and time allocation solutions, we then optimize the UAV trajectory $Q$. To avoid the high computational complexity, we also use LD methods to derive the optimal and closed-form solutions for each optimization subproblem.

A. Joint Device Scheduling and Time Allocation

To eliminate the max function in constraints (20c), it is equivalently converted into

\[
\sum_{k=1}^{K} T_k[n] + a_k[n] t^\text{comp}_k \leq \delta[n], \quad \forall k, \quad \forall n \in \mathcal{N}. \quad (22)
\]
Given trajectory $Q$, for constraints (20d) and (6), they can be respectively rewritten as

$$\sum_{n=0}^{N-1} \sum_{k=1}^{K} a_k[n+1]D_k^2 \geq C,$$  \hspace{1cm} (23)

$$\delta[n] \geq ||q[n] - q[n-1]||_2/V_{\text{max}}, \forall n.$$ \hspace{1cm} (24)

where $C \triangleq N \sum_{k=1}^{K} D_k^2 - (\epsilon - \frac{2}{\eta_{\text{opt}}}(F(w_0) - F^*)) \frac{N D^2}{4k}$ is constant. By further relaxing the integer constraints, i.e., (20a), Problem (20) under fixed UAV trajectory $Q$ reduces to the following problem,

$$\text{minimize}_{A, \delta, \tau} \sum_{n=1}^{N} \delta[n]$$

subject to

$$\sum_{n=1}^{N} \left( \frac{\tau_k[n]}{h_k[n]} \left( 2 \frac{a_k[n]}{h_k[n]} - 1 \right) + a_k[n] E_k^{\text{comp}}[n] \right) \leq E_k, \quad \forall k \in K,$$

$$0 \leq a_k[n] \leq 1, \quad \forall k \in K, \quad \forall n \in N,$$

Constraints (22), (23), and (24).

(25a)

(25b)

Define $\tau_k[n] \left( 2 \frac{h_k[n]}{a_k[n]} - 1 \right) = 0$ when $\tau_k[n] = 0, \forall k, \forall n$, such that the left-hand-sides of constraints (25a) are continuous with respect to $\tau_k[n]$ with $\tau_k[n] \geq 0$. Accordingly, since $\tau_k[n] \left( 2 \frac{h_k[n]}{a_k[n]} - 1 \right)$ is convex with respect to $a_k[n]$, its perspective function $\tau_k[n] \left( 2 \frac{h_k[n]}{a_k[n]} - 1 \right)$ with $\tau_k[n] \geq 0$ in constraints (25a) is jointly convex with respect to $a_k[n]$ and $\tau_k[n]$. Furthermore, all remaining constraints in problem (25) are affine. It is concluded that problem (25) is convex, which can be optimally solved by the Lagrange duality method. The partial Lagrange function of problem (25) is given by

$$\mathcal{L}(\delta, A, \tau, \lambda, \mu, \xi) \triangleq$$

$$\sum_{n=1}^{N} \delta[n]$$

$$+ \sum_{k=1}^{K} \lambda_k \left( \sum_{n=1}^{N} \left( \frac{\tau_k[n]}{h_k[n]} \left( 2 \frac{a_k[n]}{h_k[n]} - 1 \right) + a_k[n] E_k^{\text{comp}}[n] \right) - E_k \right)$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} \mu_k[n] \left( \sum_{k=1}^{K} \tau_k[n] + a_k[n] E_k^{\text{comp}}[n] - \delta[n] \right)$$

$$+ \xi \left( C - \sum_{n=0}^{N-1} \sum_{k=0}^{K} a_k[n+1] D_k^2 \right),$$

(26)

where $\lambda = \{\lambda_k, \forall k\}$, $\mu = \{\mu_k[n], \forall k, \forall n\}$, and $\xi$ are the non-negative Lagrange multipliers associated with constraints (25a), (22), and (23), respectively. The other boundary constraints in problem (25) will be absorbed into the optimal solution in the following. Accordingly, the dual function is given by

$$g(\lambda, \mu, \xi) \triangleq \text{minimize}_{\delta, A, \tau} \mathcal{L}(\delta, A, \tau, \lambda, \mu, \xi)$$

subject to Constraints (25b) and (22).

(27)

To make $g(\lambda, \mu, \xi)$ lower-bounded (i.e., $g(\lambda, \mu, \xi) \geq -\infty$), the condition $\sum_{k=1}^{K} \mu_k[n] \leq 1$ must hold. Therefore, the dual problem of problem (25) is given by

$$\text{maximize}_{\lambda, \mu, \xi} g(\lambda, \mu, \xi)$$

subject to $\sum_{k=1}^{K} \mu_k[n] > 1, \forall n,$

$$\lambda \geq 0, \mu \geq 0, \xi \geq 0.$$ \hspace{1cm} (28a)

(28b)

Since the strong duality holds, we can solve problem (25) by equivalently solving its dual problem (28). In the following, we show how to obtain the dual function $g(\lambda, \mu, \xi)$ by solving problem (27) under any given $\lambda, \mu, \xi$. By ignoring the constant terms, problem (25) reduces to the following formulation,

$$\text{minimize}_{\delta, A, \tau} \sum_{n=1}^{N} \left( 1 - \sum_{k=1}^{K} \mu_k[n] \right) \delta[n]$$

$$+ \sum_{k=1}^{K} \lambda_k \sum_{n=1}^{N} \frac{\tau_k[n]}{h_k[n]} \left( 2 \frac{a_k[n]}{h_k[n]} - 1 \right) + a_k[n] E_k^{\text{comp}}[n]$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} \mu_k[n] \left( \sum_{k=1}^{K} \tau_k[n] + a_k[n] E_k^{\text{comp}}[n] \right)$$

$$- \xi \sum_{n=1}^{N} \sum_{k=1}^{K} a_k[n] D_k^2$$

subject to Constraints (25b) and (22).

(29a)

(29b)

Note that problem (29) can be decomposed into two sets of subproblems, i.e., optimizing $\delta$ and jointly optimizing $A$ and $\tau$. We first consider the following optimization problem for optimizing $A$ and $\tau$.

$$\text{minimize}_{A, \tau} \sum_{n=1}^{N} \sum_{k=1}^{K} \mu_k[n] \left( \sum_{k=1}^{K} \tau_k[n] + a_k[n] E_k^{\text{comp}}[n] \right)$$

$$+ \sum_{k=1}^{K} \lambda_k \sum_{n=1}^{N} \frac{\tau_k[n]}{h_k[n]} \left( 2 \frac{a_k[n]}{h_k[n]} - 1 \right) + a_k[n] E_k^{\text{comp}}[n]$$

$$- \xi \sum_{n=1}^{N} \sum_{k=1}^{K} a_k[n] D_k^2$$

subject to Constraints (25b),

(30)

Problem (30) is convex with respect to $\{a_k[n]\}$ and $\{\tau_k[n]\}$, so the optimal solution is the one that satisfies the Karush-Kuhn-Tucker (KKT) conditions. To obtain the optimal $\tau$ in problem (30), we have the following theorem.

Theorem 3: By setting the first derivative of the objective function of problem (30) with respect to $\tau_k[n]$ to zeros, the optimal time allocation can be written as

$$\tau_k^* \triangleq a_k[n] \left[ \frac{s \ln \frac{1}{e}}{B \left( 1 + \mathcal{W} \left( \frac{B h_k[n] \sum_{k=1}^{K} \mu_k[n]}{\lambda_k \sigma^2} - \frac{1}{e} \right) \right)) \right], \quad \forall k, \forall n,$$

(31)

where $\mathcal{W}$ is the Lambert W function with $\mathcal{W}(x)e^{\mathcal{W}(x)} = x$ and $[x]^+ \triangleq \max\{x, 0\}$.

Proof: Please see Appendix D.
By substituting \( \{\tau^*_k[n]\} \) into problem (30), we have

\[
\begin{align*}
\text{minimize}_{A} & \quad \sum_{k=1}^{K} \sum_{n=1}^{N} g(\tilde{\tau}_k[n]) a_k[n] \\
\text{subject to} & \quad 0 \leq a_k[n] \leq 1, \quad \forall k \in K, \quad \forall n \in N, \quad (32a)
\end{align*}
\]

where \( \tilde{\tau}_k[n] = \tau_k[n] a_k[n] \).

To solve the dual problem (28), specifically, dual variables are updated in (28). Specifically, dual variables are updated in (28). Specifically, dual variables are updated in (28). Specifically, dual variables are updated in (28). Specifically, dual variables are updated in (28). Specifically, dual variables are updated in (28).

Next, we consider the other subproblem for optimizing \( \delta \), which is given by

\[
\begin{align*}
\text{minimize}_\delta & \quad \sum_{n=1}^{N} \left( 1 - \sum_{k=1}^{K} \mu_k[n] \right) \delta[n] \\
\text{subject to} & \quad \delta[n] \geq \left( \|q[n] - q[n-1]\|_2 / V_{\max} \right), \quad \forall n. \quad (34)
\end{align*}
\]

It is evident that problem (34) is a linear programming. For problem (34), since \( \sum_{k=1}^{K} \mu_k[n] \leq 1 \) holds, the optimal time slot variables are given by

\[
\delta^*[n] = \begin{cases} 
(\|q[n] - q[n-1]\|_2 / V_{\max}), & \text{if } \sum_{k=1}^{K} \mu_k[n] < 1, \\
\max_{1 \leq k \leq K} a_k[n] \xi_{\text{comp}}, & \text{if } \sum_{k=1}^{K} \mu_k[n] = 1,
\end{cases} \quad (35)
\]

where \( b \) can be an arbitrary real number which is not smaller than \( \|q[n] - q[n-1]\|_2 / V_{\max} \) since the objective function of problem (34) is not affected in this case. For simplicity, we set \( b = \|q[n] - q[n-1]\|_2 / V_{\max} \).

The dual problem (28) can be solved based on \( (\tau^*, \delta^*, A^*) \). Despite the fact that the dual function \( g(\lambda, \mu, \xi) \) is always convex by definition, it is generally non-differentiable. To address this challenge, a common subgradient-based method, such as projected subgradient decent method, can be applied to solving the (28). Specifically, dual variables are updated in each iteration via

\[
\begin{align*}
\mu_k^2[n] & = [\mu_k[n] + \phi \sum_{k=1}^{K} \tau_k[n] + a_k[n] \xi_{\text{comp}} - \delta[n]]^+, \\
\mu_k[n] & = \frac{\mu_k^2[n]}{\max\{1, \sum_{k=1}^{K} \mu_k^2[n] \}}, \quad (36) \\
\lambda_k & = [\lambda_k + \phi \sum_{n=1}^{N} \tau_k[n] \sigma^2 / h_k[n] (2^{\tau_k[n] / \lambda_k[n]} - 1)]^+.
\end{align*}
\]

Algorithm 2 summarizes the details of obtaining the optimal solution to problem (25). Algorithm 2 includes three parts, i.e., solving problems (30) and (34), updating the dual variables, and solving problem (39). According to (31) and (33), the complexity of solving problem (30) is \( O(KN) \). According to (35) and (40), the complexity of both solving problem (34) and problem (39) is \( O(N) \). In addition, according to (37)-(36), the complexity of updating the dual variables is \( O(KN) \). Denote by \( l_1 \) the number of iterations required for convergence. Therefore, the total complexity of Algorithm 2 is given by \( O(l_1 KN) \).

Algorithm 2 Dual Method for Problem (25)

1: \textbf{Input:} \( K, N, Q \).
2: Initialize dual variables \( \{\lambda_k = 1\}, \{\mu_k[n] = 1/K\} \).
3: repeat
4: \quad Update the primal variables \( A, \tau, \) and \( \delta \) according to (33), (31), and (35).
5: \quad Update the dual variables \( \lambda, \mu, \xi \) according to (36)-(38).
6: until \( \lambda \) and \( \mu \) converge within a prescribed accuracy
7: Update \( \delta^* \) according to (40).
8: \textbf{Output:} \( A^*, \tau^*, \) and \( \delta^* \).
B. Trajectory Design

Given device scheduling and time allocation \{A, δ, τ\}, the trajectory optimization subproblem is reduced to the following feasibility checking problem

\[
\text{find } Q \quad \text{subject to } \sum_{n=1}^{N} b_k[n]||q[n] - u_k||_2^2 \leq \bar{E}_k, \quad \forall k \in K, \tag{41a}
\]

Constraints (6), (7),

\[
\text{where } \bar{E}_k \triangleq E_k - \sum_{n=1}^{N} a_k[n]E_k^{\text{comp}}[n] - \sum_{n=1}^{N} \frac{\tau_k[n]|\sigma^2|H_n^2}{\beta_0} (\frac{a_k[n]}{\alpha_k[n]})^2 - 1. \tag{41b}
\]

According to (13), \(\tau_k[n]\) is a non-increasing function with respect to channel gain \(h_k[n]\). Hence, to further reduce the time consumption while ensuring the feasibility of problem (41), we transform problem (41) into the following problem with an explicit objective function

\[
\text{minimize } \sum_{n=1}^{N} \sum_{k=1}^{K} b_k[n]||q[n] - u_k||_2^2 \quad \text{subject to Constraints (6), (7), (41a).} \tag{42a}
\]

Comparing problem (42) with problem (41), it is easily verified that the feasible set of problem (42) is the same as that of problem (41). Hence, the solution to problem (42) is also feasible for problem (41). Although the convex QCQP problem (42) can be solved using a general-purpose solver through interior-point methods, to further reduce the computational complexity, we exploit the specific structure of problem (42) and find its optimal solution by using a Lagrange dual ascent method in the sequel. The partial Lagrange function of problem (42) can be expressed as

\[
\mathcal{L}(Q, \gamma) = \sum_{n=1}^{N} \sum_{k=1}^{K} (\gamma_k + \gamma_k\bar{b}_k[n])||q[n] - u_k||_2^2 - \sum_{k=1}^{K} \gamma_k \bar{E}_k, \tag{43}
\]

where \(\gamma_k = \gamma_k \forall k\) are the non-negative Lagrange multipliers associated with constraints (41a). Accordingly, the dual function is given by

\[
g(\gamma) = \minimize_{Q} \mathcal{L}(Q, \gamma) \quad \text{subject to Constraints (6), (7).} \tag{44}
\]

The dual problem of problem (42) is given by

\[
\maximize_{\gamma \geq 0} g(\gamma) \tag{45}
\]

As the strong duality holds, we can solve problem (42) by equivalently solving its dual problem (45). In the following, we show how to obtain the dual function \(g(\gamma)\) by solving problem (44) under any given \(\gamma \geq 0\). By ignoring the constant terms, problem (44) reduces to

\[
\minimize_{Q} \sum_{n=1}^{N} \sum_{k=1}^{K} (b_k[n] + \gamma_k\bar{b}_k[n])||q[n] - u_k||_2^2 \quad \text{subject to Constraints (6), (7).} \tag{46}
\]

Since problem (46) is convex with respect to \(Q\), the solution that satisfies KKT conditions is also optimal. To obtain the optimal solution \(Q\) in problem (46), we have the following theorem.

**Theorem 5:** For problem (46), the optimal UAV trajectory can be denoted as

\[
q[n] = \begin{cases} q_1, & \text{if } n = 0, \\ \mathcal{P}(\sum_{k=1}^{K} (b_k[n] + \gamma_k\bar{b}_k[n])(u_k - q[n-1])) + q[n-1], & \text{otherwise,} \end{cases} \tag{47}
\]

where \(\mathcal{P}(x) := \min \{\min\{|\Delta q_{\text{max}}|/\|x\|, 1\}x\} \) is the projector associated with space \(C\).

After obtaining \(Q^*\) for given \(\gamma\), we next solve the dual problem (45) to find the optimal dual variables which maximize \(g(\gamma)\). Likewise in joint device scheduling and time allocation, the dual variable is determined by the subgradient based method, the updates of which in each iteration can be given by

\[
\gamma_k = \max \left[\gamma_{k-1} + \varphi (\sum_{n=1}^{N} b_k[n]||q[n] - u_k||_2^2 - \bar{E}_k)\right] \tag{48}
\]

where \(\varphi\) is a dynamically chosen step-size. The procedure for solving problem (42) is similar to that of solving problem (25). The details are omitted due to space limitations.

**Algorithm 3 BCD-LD for Problem (20)**

1: **Input:** \(M, N\).
2: Initialize \(Q^0\). Let \(r = 1\).
3: **repeat**
4: Solve problem (25) by applying Algorithm 2 for given \(Q^{r-1}\), and denote the optimal solution as \(\{A^r, \tau^r, \delta^r\}\).
5: Solve problem (42) for given \(\{A^r, \tau^r, \delta^r\}\), and denote the optimal solution as \(Q^r\).
6: Update \(r = r + 1\).
7: until the fractional decrease of the objective value is below a threshold \(\epsilon\).
8: **Output:** \(A, \tau, \delta, \text{ and } Q\).

C. Convergence and Complexity Analysis

In this section, based on the results in previous two subsections, we exploit BCD technique [45] to solve problem (20) by solving Problem (25) and Problem (42) alternately until convergence, which is summarized in Algorithm 3. Note that the subproblem for updating each block of variables is solved with optimality in each iteration. The convergence of Algorithm 3 is shown in the following proposition.

**Proposition 2:** With Algorithm 3, the objective value of problem (20) decreases as the number of iterations increases until convergence.

**Proof:** Please refer to Appendix E. \(\square\)

In the following, we investigate the complexity per iteration of Algorithm 3. Specifically, in step 4, as discussed in the last paragraph of Section IV-A, the computational complexity for updating \(\{A^r, \tau^r, \delta^r\}\) via Algorithm 2 is given by \(\mathcal{O}(lK\bar{N})\).
In step 5, for updating \( Q^r \), the computational complexity mainly lies in computing expression (47). According to (47), the complexity for computing \( Q \) is \( O(N) \). In summary, the total complexity of Algorithm 3 is thus dominated by \( O(l_1 K N) \) in each iteration.

V. NUMERICAL RESULTS

In this section, we present extensive numerical results to demonstrate the effectiveness of the proposed algorithm for UAV-enabled FL system. In the simulations, the service area of the UAV is restricted to a square area with the size of \([0, 400] \text{m} \times [0, 400] \text{m}\). The UAV is assumed to fly at a fixed altitude of \( H = 100 \) m, which complies with the practical rule, i.e., commercial UAVs should not fly over 400 feet (122 m) [26]. And the UAV is placed at a predetermined initial position, i.e., \((200, 0, 100) \text{m}\). The maximum speed of the UAV is \( V_{\text{max}} = 20 \text{m/s} \). In addition, considering channel heterogeneity, \( K \) devices are separated into two clusters allocated in two circles, i.e., cluster \( A \) with \([0.4K] \) devices and cluster \( B \) with \( K - [0.4K] \) devices. Specifically, the devices in cluster \( A \) and cluster \( B \) are randomly and uniformly distributed in circles centered at \((100, 100) \text{m} \) and \((300, 300) \text{m} \) with a radius of 80 m, respectively. The energy budget of devices in the cluster \( A \) and \( B \) are denoted as \( E_A \) and \( E_B \), respectively. The effective capacitance coefficient of all devices’ computing chipsets is \( \alpha_1 = \ldots = \alpha_k = 10^{-28} \).

We leverage the multi-nominal logistic regression to train the learning models. Specifically, the sample-wise loss function used for training is given by

\[
f(w; x_{ki}, y_{ki}) = \sum_{c=1}^{C} \mathbb{I}(y_{ki} = c) \log \frac{\exp(w^T x_{ki})}{\sum_{j=1}^{C} \exp(w^T x_{kj})},
\]

where \( C \) denotes the total number of label classes, \( w \) denotes the model parameter vector, i.e., \( w = [w_1^T, \ldots, w_C^T]^T \), and \( \mathbb{I}(y_{ki} = c) \) is defined as the indicator function. In addition, the partial gradient with respect to \( w_c \) is given by

\[
\nabla_{w_c} f(w; x_{ki}, y_{ki}) = -\left(\mathbb{I}(y_{ki} = c) - \frac{\exp(w^T x_{ki})}{\sum_{j=1}^{C} \exp(w^T x_{kj})}\right)x_{ki}
\]

and the entire gradient is given by \( \nabla_w f(w; x_{ki}, y_{ki}) = [\nabla_{w_1} f(w; x_{ki}, y_{ki}); \ldots; \nabla_{w_C} f(w; x_{ki}, y_{ki})] \). Unless specific state, we consider the image classification task on the widely-used CIFAR-10 datasets [46], which is composed of 60,000 \( 32 \times 32 \) RGB color images in 10 different classes. Specifically, images with data size \( D = 50,000 \) are assigned for training dataset while the remaining 10,000 images are assigned for test dataset. In addition, we first sort the dataset by the contained labels, then divide it into \( K \) shards, and finally assign one shard for each device without replacement. For this task, the dimension of model parameter vector \( d \) is given by \( d = 32 \times 32 \times 3 \times 10 \). Each parameter is assumed to be stored with \( I = 32 \) bits and thus \( s = I \times d \). Moreover, the learning rate \( \eta \) is set to \( \eta = 0.01 \). Other parameters are summarized in Table II. To evaluate the effects of different wireless factors on the completion time of FL training and test accuracy, we compare the performance of the proposed joint design with three benchmark schemes described below:

- **Full Scheduling:** The training process of the UAV-enabled FL applies full-scheduling transmission by setting \( a_k[n] = 1, \forall k, \forall n \), while optimizing \( \tau, \delta, Q \). This scheme serves as an upper bound of learning performance, as it does not introduce the model aggregation error.

- **Static UAV:** This scheme optimizes \( A, \tau, \delta \) and sets \( q[n] = q_I, \forall n \). The scheme highlights the significant communication straggler issue caused by the static UAV, which is positioned far from all devices. Therefore, it can be used to evaluate the effectiveness of UAV trajectory design in mitigating the impact of stragglers on the completion time cost.

- **Static UAV w/ HS:** This scheme utilizes a deterministic approach where both the UAV location \( Q \) and device scheduling \( A \) are fixed, with \( q[n] = q_I, \forall n \), and devices are selected using a heuristic method that prioritizes those with larger channel gains as in [12]. By satisfying constraint (20d) with equality, this scheme only requires optimizing \( \tau, \delta \) to minimize the completion time for FL. Its purpose is to assess the effectiveness of the joint design of UAV trajectory and device selection in reducing the straggler effect on completion time cost and improving learning performance.

The initial UAV trajectory \( Q^0 \) for BCD-LD and Full Scheduling is generated by using the predetermined initial position, i.e., \( q^0[n] = q_I, \forall n \), unless specified otherwise. Moreover, we initialize the global model parameters as a zero matrix.

A. Optimized UAV Trajectory

Fig. 3 shows the convergence behaviors of the proposed BCD-based algorithm (i.e., Algorithm 3) with different device energy budgets \( E_k, \forall k \) when \( \epsilon = 0.2 \). All devices have the same energy budgets. It is observed that for both schemes, the completion time with Algorithm 3 decreases quickly with the number of iterations, and the two algorithms converge within 5 iterations with prescribed accuracy \( \epsilon = 10^{-3} \). Besides, the proposed joint design achieves a smaller completion time under \( E_k = 30 \) J compared to \( E_k = 10 \) J.

Fig. 4 illustrates the trajectories obtained by Algorithm 3 under different energy budgets of devices \( E_k \). Each trajectory is sampled every 5 rounds and the sampled points are marked...
with ⋆ by using the same colors as their corresponding trajectories. The devices’ locations are marked by a dark black □. The predetermined locations of the UAV are red △. It is observed that by comparing the case of \( E_B = 5 \) J with that of \( E_B = 15 \) J, the controllable high-mobility UAV needs to approach devices if the energy budgets of devices become more stringent. Initially, one can observe that the UAV hovers over the initial position for an interval. Subsequently, the UAV proactively shortens the devices’ communication distance by visiting the devices in both clusters along an arc path. Finally, the UAV hovers over a certain location. Fig. 9 shows the percentage of scheduled devices over communication rounds and provides insights into the reasons behind the optimized trajectory’s behaviors. It can be observed that there are three stages for behaviors of scheduled devices, as described below.

- **Stage I**: As shown in Fig. 9, from round 1 to about 1000 rounds, the percentage of participating devices per round is equal to 0, which means that no device is involved in FL training during this time interval. This explains why the UAV stays at the initial position in this interval, where the communication straggler issue is most severe.

- **Stage II**: From about 1000 to 1700 rounds, the percentage of participating devices is dynamically changing, which means our proposed design can dynamically adjust device selection when the channels change.

- **Stage III**: From about 1700 rounds to 4000 rounds, the percentage of participating devices equals 100% in each round, which means that thanks to the UAV trajectory design, all devices can participate in the FL training process and contribute their data exploitation, thereby reducing model aggregation loss and accelerating FL convergence. This explains why in this interval the UAV needs to hover over a certain location which strikes an optimal balance enhancing all the devices’ channels to shorten the flight time of the UAV.

### B. Performance Comparison for Completion Time

Fig. 6 shows the effect of the convergence accuracy of FL on the completion time when \( E_k = 10 \) J, \( \forall k \). Since the full scheduling scheme does not result in the model aggregation loss, its performance is invariant to the convergence accuracy \( \epsilon \). However, this scheme costs prohibitive completion time without device scheduling. Note that this scheme may be forbidden in general since UAVs usually have a limited endurance due to practical physical constraints (e.g., 30 minutes for the typical rotary-wing UAV [26]). For the other three schemes with device scheduling, the completion time decreases with \( \epsilon \), which is consistent with Theorem 2. In addition, the Static UAV scheme performs better than the Static UAV w/ HS scheme since the former avoids the aggregation error explosion by excluding the weak devices. However, in the case of sever straggler issues, the performance gain brought by device scheduling optimization alone is marginal. Fortunately, the proposed joint design scheme incorporated trajectory design with device scheduling significantly reduce the completion time compared to all benchmarks. This is because, the mobile
UA V can shorten communication distances between the scheduled devices and always achieve a high transmission rate for local model uploading, thereby accelerating the model aggregation process. These results demonstrate in UA V-enabled systems, joint design of trajectory and device scheduling is critical to mitigating the communication stragglers’ effect on completion time.

Fig. 7 shows the total mission completion time versus the device energy budget when $\epsilon = 0$. It is observed that, for all schemes, the total completion time decreases as $E_k$ increases. This is because, as $E_k$ increases, the scheduled devices can afford more energy to upload local models with higher transmission rates. We also find that the proposed joint design achieves the smallest completion time compared to all benchmarks. This observation further verifies the superiority of the proposed scheme to mitigate the stragglers’ effect.

C. Performance Comparison for FL

To demonstrate the performance of our proposed joint design for dealing with FL tasks, we train image classifier models on the widely-used CIFAR-10 dataset.

Fig. 8 shows training accuracy over communication rounds with different schemes and different $\epsilon$ when $E_k = 10$ J, $\forall k$. We can observe that the convergence accuracy $\epsilon$ heavily affects the learning performance of the Static UAV w/ HS since a larger value of $\epsilon$ implies that the data are less exploited in this scheme due to the fact of device exclusion. In contrast, the proposed joint design still achieves a training accuracy that is close to the ideal benchmark even as $\epsilon$ increases. For example, we can see from Fig. 5 with $\epsilon = 0.2$, the percentage of participating devices is 100% from about 1400 rounds to 4000 rounds. This is because the proposed joint design scheme overcomes the straggler issue and maximizes data exploitation. Specifically, the UAV’s trajectory tends to be close to the devices to establish strong communication links, thus maximizing data exploitation. In contrast, at some UAV locations (e.g., the initial position), where the communication straggler issue is severe, the device schedule is optimized to let no device be involved in FL training. This is the reason why the curve of the proposed joint design scheme is invariant during the initial communication rounds. On the other hand, by properly designing the trajectory of the UAV, short-distance LoS links can be proactively and dynamically established for any device, which ensures all devices have equal opportunity to participate in FL training instead of device exclusion, thereby increasing the data exploitation and training accuracy.

Fig. 9 shows the test accuracy versus convergence accuracy of FL training with different schemes when $E_k = 10$ J, $\forall k$. It is observed that the test accuracy achieved by the full-scheduling case is highest and close to 40%, which is consistent with the previous FL works on classification task with CIFAR-10 datasets [6], [47]. Furthermore, this significantly improves the learning performance compared to random classification with test accuracy $\frac{1}{\# \text{ of classes}} \times 100\% = 10\%$. Observed from Fig. 6 and Fig. 9, it shows that there exists a fundamental trade-off between the mission completion time and the learning performance of FL. However, by exploiting the mobility of the UAV and flexibility of device scheduling, our proposed joint design significantly improves the trade-off compared to the benchmarks. For example, when $\epsilon = 0.2$, the completion time of FL achieved by our proposed joint design, static UAV w/HS, and the full scheduling scheme are respectively 930 s, and 1150 s, 1500 s. Meanwhile, when $\epsilon = 0.2$, the test accuracy achieved by our proposed joint design, static UAV w/HS, and the full scheduling scheme are respectively 38.99% s, and 32.52% s, and 39.74%. For static UAV w/HS, compared to the full scheduling scheme, the completion time reduces 23.33% while the test accuracy reduces 7.22%. Fortunately, for our proposed joint design,
compared to the full scheduling scheme, the completion time reduces 38\% while the test accuracy only reduces 0.71\%.

VI. CONCLUSION

In this paper, we studied the UAV-assisted FL system to effectively address the straggler issue by exploiting the UAV’s high altitude and mobility when the terrestrial PS was unavailable. Specifically, a mobile UAV was deployed as a flying PS to exchange model parameters from distributed ground devices and train a shared FL model. We focused on the joint consideration of device scheduling, UAV trajectory, and time allocation to minimize the completion time required for FL to converge to the desired accuracy level. Before proceeding with problem formulation, we first theoretically analyzed the effect of device scheduling on the convergence accuracy of FL training without the assumptions of convexity. We also provided a convergence bound for the average norm of the global gradient. Based on such convergence results, we formulated the completion time minimization problem, taking into account the practical device’s energy budget, UAV’s mobility, and convergence accuracy constraints. Although the formulated problem was non-convex, we exploited its structures to decompose it into two sub-problems, followed by deriving closed-form solutions via the Lagrange dual ascent method. Simulation results demonstrated that our proposed joint design can significantly reduce completion time and enhance the tradeoff between completion time and prediction accuracy compared to benchmarks.

APPENDIX

A. Proof of Theorem 1

Let $g_n \triangleq (\sum_{k=1}^{K} a_k[n+1]D_k\nabla F_k(w_n)) / (\sum_{k=1}^{K} a_k[n+1]D_k)$. Combining the update rules at devices in (4) and the update rule at the server in (5), the update of the global FL model at round $n+1$ is given by $w_{n+1} = w_n - \eta \sum_{k=1}^{K} a_k[n+1]D_k\nabla F_k(w_n) / \sum_{k=1}^{K} a_k[n+1]D_k = w_n - \eta g_n$. From the assumption in (16), we have that

$$F(w_{n+1}) - F(w_n) \leq \langle \nabla F(w_n), w_{n+1} - w_n \rangle + (L/2)\|w_{n+1} - w_n\|^2,$$

$$= \langle \nabla F(w_n), -\eta g_n \rangle + (\eta^2 L/2)\|g_n\|^2. \quad (50)$$

Using $g_n = \nabla F(w_n) - (\nabla F(w_n) - g_n)$, we derive that

$$F(w_{n+1}) - F(w_n) \leq \langle \eta L/2 - 1\rangle \eta \|\nabla F(w_n)\|^2 + (\eta^2 L/2)\|\nabla F(w_n) - g_n\|^2 + (1 - \eta L)\eta \|\nabla F(w_n)\|\|\nabla F(w_n) - g_n\|^2. \quad (51)$$

The third term at the right-hand side in (51) can be upper bounded as

$$\langle \nabla F(w_n), \nabla F(w_n) - g_n \rangle \leq \frac{1}{2} \|\nabla F(w_n)\|^2 + \|\nabla F(w_n) - g_n\|^2. \quad (52)$$

Given $0 < \eta \leq \frac{1}{L}$, we substitute (52) into (51) and rearrange the result, yielding

$$\|\nabla F(w_n)\|^2 \leq \frac{2}{\eta} (F(w_n) - F(w_{n+1})) + \|\nabla F(w_n) - g_n\|^2. \quad (53)$$

Therein, the second term, namely the global gradient deviation error in expression (53) can be rewritten as

$$\|\nabla F(w_n) - g_n\|^2 = \left\| \sum_{k=1}^{K} a_k[n+1]D_k \nabla F_k(w_n) - \frac{1}{D} \sum_{k=1}^{K} D_k \nabla F_k(w_n) \right\|^2$$

$$= \left\| \sum_{k=1}^{K} D_k(1 - a_k[n+1]) \frac{1}{D} \sum_{k=1}^{K} a_k[n+1]D_k \nabla F_k(w_n) - \frac{1}{D} \sum_{k=1}^{K} (1 - a_k[n+1]) D_k \nabla F_k(w_n) \right\|^2. \quad (54)$$

Thus, based on (54) and triangle inequality, the global gradient deviation error is bounded as

$$\|\nabla F(w_n) - g_n\|^2 \leq \left( \frac{\sum_{k=1}^{K} D_k(1 - a_k[n+1])}{D \sum_{k=1}^{K} a_k[n+1]D_k} \right) \sum_{k=1}^{K} a_k[n+1]D_k \|\nabla f(w_n; x_{ki}, y_{ki})\|^2$$

$$+ \frac{1}{D} \sum_{k=1}^{K} (1 - a_k[n+1]) \sum_{i=1}^{D_k} \|\nabla f(w_n; x_{ki}, y_{ki})\|^2 \leq 4D^2 \left( \frac{\sum_{k=1}^{K} (1 - a_k[n+1]) D_k^2}{D \sum_{k=1}^{K} a_k[n+1]D_k} \right) \sum_{k=1}^{K} a_k[n+1]D_k^2. \quad (55)$$

Therefore, based on (53) and (55), it follows that

$$\frac{1}{N} \sum_{n=0}^{N-1} \|\nabla F(w_n)\|^2 \leq \frac{2}{\eta N} (F(w_0) - F(w_N)) + \frac{1}{N} \sum_{n=0}^{N-1} \|\nabla F(w_n) - g_n\|^2$$

$$\leq \frac{2}{\eta N} (F(w_0) - F^*) + \frac{1}{N} \sum_{n=0}^{N-1} \|\nabla F(w_n) - g_n\|^2 \leq \frac{2}{\eta N} (F(w_0) - F^*) + \frac{4K\kappa}{ND^2} \sum_{n=0}^{N-1} \sum_{k=1}^{K} (1 - a_k[n+1]) D_k^2. \quad (56)$$

This completes the proof.

B. Proof of Proposition 1

Given an arbitrary convergence threshold $\epsilon$, to make constraint (20d) in problem (20) feasible, if and only if the minimum communication rounds satisfies $N \geq \left\lceil \frac{2(F(w_0) - F^*)}{\epsilon \eta} \right\rceil$, we set $N$, with $\epsilon$ and $\eta$ as per given conditions for FL training.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( a_k[n] = 1, \forall k, \forall N \). In addition, we have that
\[
\frac{4K\kappa}{N^2D^2} \sum_{n=0}^{N-1} \sum_{k=1}^{K} (1 - a_k[n + 1])D_k^2 \leq \frac{4K\kappa}{N^2D^2} \max_k D_k^2 \sum_{n=0}^{N-1} \sum_{k=1}^{K} (1 - a_k[n + 1]).
\]

(57)

By instituting the right term of inequality (57) into constraint (20d), if (20d) is satisfied, then
\[
\sum_{n=0}^{N-1} \sum_{k=1}^{K} a_k[n + 1] \geq |S|,
\]
where \(|S| = [KN - (\varepsilon - \frac{2\kappa}{N\eta}(F(w_0) - F^*))\frac{N^2D^2}{4K\kappa \max_k D_k^2}]\). For device \( k \), the minimum energy consumption for local model uploading is given by
\[
\lim_{\tau_k[n] \to +\infty} E_k^{\text{comm}}[n] = \lim_{\tau_k[n] \to +\infty} \frac{\tau_k[n]\sigma^2}{h_k[n]} \left( 2\frac{a_k[n]}{\sigma^2} - 1 \right) = \frac{a_k[n]\sigma^2 \ln 2}{Bh_k[n]} \geq \frac{a_k[n]\sigma^2 H_k \ln 2}{B\beta_0}.
\]

(59)

Besides, if constraints (20b) are satisfied, we have that
\[
\frac{\sigma^2H_k \ln 2}{B\beta_0} \sum_{k=1}^{K} \sum_{n=1}^{N} a_k[n] \leq \sum_{k=1}^{K} \sum_{n=1}^{N} \left( E_k^{\text{comm}}[n] + E_k^{\text{comp}}[n] \right) \leq \sum_{k=1}^{K} E_k.
\]

(60)

Combining (58) with (60), to make constraints (20b) and constraint (20d) feasible, we have that
\[
\frac{\sigma^2\gamma^2 \ln 2}{B\beta_0} |S| \leq \sum_{k=1}^{K} E_k, N \geq \left[ \frac{2(F(w_0) - F^*)}{\epsilon \eta} \right].
\]

(61)

Thus, this completes the proof.

C. Proof of Theorem 2

Denote the optimal solutions of problem (20) with \( \epsilon^* \) and \( \hat{\epsilon} \) by with \( \mathcal{E}^* = \{ A^*, \Gamma^*, Q^*, \delta^* \} \) and \( \hat{\mathcal{E}} = \{ \hat{A}, \hat{\Gamma}, \hat{Q}, \hat{\delta} \} \), respectively. To prove Theorem 2, we only need to show that \( \sum \delta^*[n] \geq \sum \hat{\delta}[n] \) holds when \( \epsilon^* \leq \hat{\epsilon} \). Note that in problem (20), the convergence threshold is only involved in constraint (20d). Thus, we have the following inequalities
\[
\frac{2}{\eta} \left( F(w_0) - F^* \right) + \frac{4K\kappa}{N^2D^2} \sum_{n=0}^{N-1} \sum_{k=1}^{K} (1 - a_k[n + 1])D_k^2 \leq \epsilon^* \leq \hat{\epsilon}.
\]

(62)

which implies that \( \mathcal{E}^* \) is also a feasible solution of problem (20) with \( \hat{\epsilon} \). Since \( \sum \delta[n] \) is the minimum objective value of problem (20) with \( \hat{\epsilon} \), it follows that \( \sum \delta^*[n] \geq \sum \hat{\delta}[n] \), which thus completes the proof of Theorem 2.

D. Proof of Theorem 3

We take the first derivative of the objective function of problem (30) w.r.t. \( \tau_k[n] \), i.e.,
\[
\frac{\partial}{\partial \tau_k[n]} \sum_{k=1}^{K} \frac{\mu_k[n]}{h_k[n]} + \frac{\lambda_k}{h_k[n]} \left( 2\frac{a_k[n]}{\sigma^2} \ln 2 \right) = \frac{a_k[n]s \ln 2}{B\tau_k[n]} - 1.
\]

(63)

By setting (63) to zero and solving this equation, we have
\[
\left( \frac{a_k[n]s \ln 2}{B\tau_k[n]} - 1 \right) = \frac{h_k[n]}{\lambda_k \sigma^2} \sum_{k=1}^{K} \mu_k[n] - 1.
\]

(64)

By solving the above equation, one can have \( a_k[n]s \ln 2 - 1 = \frac{h_k[n]}{\lambda_k \sigma^2} \sum_{k=1}^{K} \mu_k[n] - 1 \). By simplifying the above equation, we can have that
\[
\tau_k[n] = \frac{a_k[n]s \ln 2}{B\left( \frac{h_k[n]}{\lambda_k \sigma^2} \sum_{k=1}^{K} \mu_k[n] - 1 \right)}.
\]

(65)

Thus, this completes the proof.

E. Proof of Proposition 2

We denote \( T^*(A^*, \Gamma^*, \delta^*, Q^*) = \sum_{n=1}^{N} \delta^*[n] \) as the objective value of problem (20) with solution \( \{ A^*, \Gamma^*, \delta^*, Q^* \} \). As shown in step 5 of Algorithm 3, a feasible solution of problem (41) (i.e., \( \{ A^*, \Gamma^*, \delta^*, Q^* \} \)) is also feasible to problem (25). We denote \( \{ A^*, \Gamma^*, \delta^*, Q^* \} \) and \( \{ A^r+1, \Gamma^r+1, \delta^r+1, Q^r+1 \} \) as a feasible solution of problem (20) at the \( r \)-th and \( (r+1) \)-th iterations, respectively.

Because, given \( Q^r \) as shown in step 5 of Algorithm 3, \( \{ A^r+1, \Gamma^r+1, \delta^r+1 \} \) is the optimal solution to problem (25), we have
\[
T(A^r, \Gamma^r, \delta^r, Q^r) \geq T(A^r+1, \Gamma^r+1, \delta^r+1, Q^r),
\]

(66)

Similarly, given \( A^r+1, \Gamma^r+1, \delta^r+1 \) as shown in step 5 of Algorithm 3, \( Q^r+1 \) is the optimal solution to problem (42), it follows that
\[
T(A^r+1, \Gamma^r+1, \delta^r+1, Q^r) \leq T(A^r+1, \Gamma^r+1, \delta^r+1, Q^r+1).
\]

(67)

This holds because the original objective function \( T \) is independent of \( Q \) but depends on \( \delta \). Based on (66) and (67), we further obtain
\[
T(A^r+1, \Gamma^r+1, \delta^r+1, Q^r+1) \leq T(A^r, \Gamma^r, \delta^r, Q^r),
\]

(68)

which shows that the objective value of problem (20) is always decreasing over iterations. This thus completes the proof.

REFERENCES

[1] W. Y. B. Lim et al., “Federated learning in mobile edge networks: A comprehensive survey,” IEEE Commun. Surveys Tuts., vol. 22, no. 3, pp. 2031–2063, 3rd Quart, 2020.

[2] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, “Communication-efficient edge AI: Algorithms and systems,” IEEE Commun. Surveys Tuts., vol. 22, no. 4, pp. 2167–2191, 4th Quart, 2020.
[1] K. B. Letaiif, Y. Shi, J. Lu, and J. Lu, “Edge artificial intelligence for 6G: Vision, enabling technologies, and applications,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[3] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[4] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[5] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and L. Barbieri, “Opportunities of federated learning in connected, cooperative, and automated industrial systems,” *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 16–21, Feb. 2021.

[6] L. Li et al., “Delay analysis of wireless federated learning based on saddle point approximation and large deviation theory,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3772–3789, Dec. 2021.

[7] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, “A survey on federated learning for resource-constrained IoT devices,” *IEEE Internet Things J.*, vol. 8, no. 1, pp. 398–413, Feb. 2021.

[8] G. Zhu, Y. Wang, and K. Huang, “Broadband analog aggregation for low-latency federated edge learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[9] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Convergence time optimization for federated learning over wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[10] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, “Scheduling for cellular federated edge learning with importance and channel awareness,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[11] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, “Convergence of update aware device scheduling for federated learning at the wireless edge,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, June 2021.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[13] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, “Edge learning for B5G networks with distributed signal processing: Static communication, edge computing, and wireless sensing,” *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 9–39, Jan. 2023.

[14] Z. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, “Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.

[15] W. Y. B. Lim et al., “Towards federated learning in UAV-enabled Internet of Vehicles: A multi-dimensional contract-matching approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5140–5154, Aug. 2021.

[16] J. S. Ng et al., “Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled Internet of Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2326–2344, Apr. 2021.

[17] H. Yang, J. Zhao, X. Zong, K.-Y. Lam, S. Sun, and L. Xiao, “Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.

[18] C. T. Dinh et al., “Federated learning over wireless networks: Convergence analysis and resource allocation,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–413, Feb. 2021.

[19] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy efficient federated learning over wireless communication networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[20] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, “Multiarmed bandit-based client scheduling for federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[21] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, “FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, Dec. 2021.

[22] C. Zhan and Y. Zeng, “Completion time minimization for multi-UAV-enabled data collection,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4858–4872, Oct. 2019.

[23] 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.

[24] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, “Scheduling policies for federated learning in wireless networks,” *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[25] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, Jan. 2018.

[26] S. Wang et al., “Adaptive federated learning in resource constrained edge computing systems,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[27] Z. Allen-Zhu, “Natala 2: Faster non-convex optimization than SGD,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2018, pp. 1–12.

[28] Q. Zeng, Y. Du, and K. Huang, “Wirelessly powered federated edge learning: Optimal tradeoffs between convergence and power transfer,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 680–695, Jan. 2022.

[29] H. Yu, S. Yang, and S. Zhu, “Parallel restarted SGD with faster convergence,” in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 5693–5700.

[30] D. Liu and O. Simeone, “Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 170–185, Jan. 2021.

[31] A. Reisszadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, “FedFPQ: A communication-efficient federated learning method with periodic averaging and quantization,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2021–2031.

[32] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, Jan. 2013.

[33] A. Krithevsky and G. Hinton, “Learning multiple layers of features from unsupervised images,” *Univ. Toronto, Toronto, ON, Canada, Tech. Rep.*, 2009.

[34] W. Shi, Z. Zhou, Z. Niu, M. Jiang, and L. Geng, “Joint device scheduling and resource allocation for latency constrained wireless federated learning,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.
Min Fu (Member, IEEE) received the B.S. degree in smart grid from the Nanjing University of Science and Technology, Nanjing, China, in 2017, and the Ph.D. degree in communication and information systems from the University of Chinese Academy of Sciences, Beijing, China, in 2022. She is currently a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. Her research interests include optimization, wireless communications, and their applications to 6G.

Yuanming Shi (Senior Member, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology (HKUST) in 2015. Since September 2015, he has been with the School of Information Science and Technology, ShanghaiTech University, where he is currently a tenured Associate Professor. He visited the University of California at Berkeley, Berkeley, CA, USA, from October 2016 to February 2017. His research areas include federated edge learning, edge AI, and task-oriented communications. He was a recipient of the IEEE Marconi Prize Paper Award in Wireless Communications in 2016, the Young Author Best Paper Award by the IEEE Signal Processing Society in 2016, the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2021, and the Chinese Institute of Electronics First Prize in Natural Science in 2022. He is an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and Journal of Communications and Information Networks.

Yong Zhou (Senior Member, IEEE) received the B.Sc. and M.Eng. degrees from Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2015. From November 2015 to January 2018, he was a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. He is currently an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests include 6G communications, edge intelligence, and the Internet of Things. He was the Track Co-Chair of IEEE VTC 2020 Fall and IEEE VTC 2023 Spring, and the General Co-Chair of IEEE ICC 2022 Workshop on Edge Artificial Intelligence for 6G. He serves as an Associate Editor for IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY.