Latency Minimization for Wireless Federated Learning with Heterogeneous Local Updates

Jingyang Zhu, *Graduate Student Member, IEEE*, Yuanming Shi, *Senior Member, IEEE*, Min Fu, *Member, IEEE*, Yong Zhou, *Member, IEEE*, Youlong Wu, *Member, IEEE*, and Liqun Fu, *Senior Member, IEEE*

Abstract

In this paper, we study the latency minimization problem for a wireless federated learning (FL) system with heterogeneous computation capability, where different edge devices perform different numbers of local updates in each communication round. We formulate a total latency minimization problem with probabilistic device selection, taking into account both the communication and computation latency in the whole FL procedure. However, it is highly challenging to optimally solve this problem due to the coupling issues of model convergence and latency minimization problem caused by the heterogeneity of local updates. Through convergence analysis, we reveal that decoupling the resource allocation variables from the model convergence is essential to reduce the problem to a single-round latency minimization problem. To solve this simplified problem, we propose an alternating optimization scheme to jointly consider communication and computation resource allocation and mitigate the straggler effect. We prove that the resulting sub-problems, i.e., bandwidth and computation capacity allocation, are both convex and can be optimally solved in closed form, respectively. Simulation results show that compared with the baseline scheme that allocates the communication and computation resources equally across edge devices, the proposed scheme can achieve up to 47.04% single-round latency reduction.

Index Terms

Federated learning, system heterogeneity, latency minimization, convergence analysis, and resource allocation.

J. Zhu, Y. Shi, Y. Zhou and Y. Wu are with the School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210, China (e-mail: {zhujy2, shiym, zhouyong, wuyl1}@shanghaitech.edu.cn).

M. Fu is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: e0684323@nus.edu.sg).

L. Fu is with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: liqun@xmu.edu.cn).

I. Introduction

As the storage and computing capabilities of Internet of Things (IoT) devices grow, big data has opened up bright avenues for machine learning (ML) applications. In conventional distributed ML, edge servers need to aggregate raw data from all edge devices, causing huge communication overhead and privacy concerns. Consequently, an emerging distributed ML paradigm called federated learning (FL) [1], [2] has been proposed, where all edge devices coordinated by an edge server collaboratively solve an ML problem without sharing their local data. FL has received increasing attention in many critical applications, ranging from 6G communication [3]–[6], IoT [7], to biomedical healthcare [8], [9] and drug discovery [10]. Nevertheless, how to implement FL over wireless networks has recently become a major concern due to the limited radio resources. Therein, the key challenges of implementing wireless FL include statistical heterogeneity, expensive communication overhead, and system heterogeneity [11].

Statistical heterogeneity, as a challenging problem for cross-device FL, usually refers to data heterogeneity (e.g., non-uniform local data sample sizes and non-IID data samples), and gradient heterogeneity among different edge devices. Existing methods for dealing with data heterogeneity include compression [12], personalized FL [13], which mainly consists of multi-task FL [14] and meta-learning [15], and algorithm design to handle non-IID data [16], [17]. From the perspective of wireless FL, the authors in [18], [19] adopted an analog model aggregation scheme in wireless FL, i.e., over-the-air computation (AirComp) [20], and considered the statistical heterogeneity of the gradients transmitted by different edge devices in different communication rounds and data heterogeneity, respectively. The authors in [21] re-designed the FL algorithm in [1] based on the upper bound of the expected weight divergence and proposed a data sharing scheme to mitigate the impact of non-IID data over wireless networks.

Communication heterogeneity, as one of the main system heterogeneity, mainly manifests as plenty of edge devices have diverse channel conditions, and the edge devices have heterogeneous communication resources. The heterogeneity of wireless links in wireless FL systems has detrimental influence on the training performance of FL. To tackle this issue, reconfigurable intelligent surface (RIS) has been proposed to reconfigure the propagation channels and thus support fast AirComp-based model aggregation [22]–[25]. In addition, to handle the heterogeneity of wireless links and avoid communication stragglers, considering limited wireless resources, the authors in [26]–[28] selected partial edge devices with good channel conditions to participate in the

training procedure [29]. To deal with heterogeneous communication resources (e.g. bandwidth and power) and alleviate communication overhead, resource allocation is essential to improve the performance of wireless FL [30], which typically consists of power control [18], [31], device selection [32]–[35], and resource management [36]–[39].

In practical scenarios, another manifestation of system heterogeneity is that different edge devices possess different computation capabilities (e.g., laptops have stronger CPUs and GPUs than smart phones). Specifically, given a time interval, faster edge devices that have abundant computation resources are capable of performing more local updates than the slower ones in one communication round. The slower devices are deficient in computation resources, and referred to as the *computation stragglers* [40], [41]. From the perspective of FL algorithm design, the authors in [42] first proposed a general framework that allows different edge devices to perform different rounds of local updates in each communication round with convergence guarantees, which is referred to as the *heterogeneity of local updates*. Moreover, the authors verified that the heterogeneous number of local updates may lead to offset in model aggregation. The algorithm proposed in [42] could eliminate this offset and achieve faster convergence rate than the state-of-art algorithms [1], [16]. Subsequently, the number of local updates was assumed to follow an specific distribution across edge devices in [43].

However, most recent works on wireless FL mainly involved the heterogeneity of communication resources and assumed that all edge devices have the same number of local updates in each communication round. Obviously, this assumption slows down the model convergence as faster edge devices can perform more local updates than the specified number E, i.e., all edge devices perform E local updates in each communication round. Furthermore, in order to catch up on local model updating, computation latency of these stragglers could be quite long, which prolongs the model aggregation, as the edge server has to wait until receiving the slowest updated local model with perfect synchronization considered. For wireless FL enabled applications, such as autonomous driving, instant feedback is required by the autonomous vehicles from the edge serve for objective detection. In this scenario, low-latency model training and communication is highly demanded. To reduce the total latency in wireless FL systems, previous works in [32]–[37] characterized the latency of FL over wireless networks. Therein, the authors in [32], [34], [37] minimized the latency of each global iteration to accelerate the training process due to the fact that the resource allocation is independent of the learning performance. The distribution of delay in wireless FL systems was fully studied in [44]. It is clear that for wireless, the total latency

not only relies on the communication time, but also depends on the local computation time in each communication round. However, these works did not consider the heterogeneity of local updates, which is a heterogeneity problem at the computational level. Motivated by these issues, it is worth investigating wireless FL with system heterogeneity, thereby fulfilling low-latency and communication-efficient FL and mitigating the computation stragglers' detrimental impact on the model aggregation.

In this paper, we consider a wireless FL system with heterogeneity of local updates, and aim to minimize the total latency by jointly allocating computation and communication resources. Our main contributions can be summarized as follows.

- This paper is one of the early attempts to study heterogeneous FL over wireless networks, where edge devices perform different number of local updates during each communication round. Moreover, we jointly allocate the communication and computation resources conditioning on the heterogeneity of local updates to minimize the total training-and-communicating latency of the wireless FL procedure.
- We formulate a total latency minimization problem in the presence of probabilistic device selection and the heterogeneity of local updates. By providing convergence analysis of the proposed FL algorithm, we reveal that the heterogeneous local updates has both effect on model convergence and the total latency minimization problem. If the number of local updates is unknown, the minimization problem can be hardly optimally solved while the convergence rate is unknown as well. To address this unique challenge, we propose to decouple the problem by fixing the number of local updates and reducing the problem to the single-round latency minimization where the resource allocation is independent of model convergence.
- Although the formulated problem is reduced to a single-round latency minimization problem, the resulting problem requires the joint allocation of the bandwidth and the local computation capacity. To support algorithm design, we adopt an alternating optimization scheme to decompose the resulting problem into two convex subproblems, which can be optimally solved. The proposed algorithm jointly allocates bandwidth and computation resources and mitigates the straggler effect, thereby considerably improving the performance compared with the state-of-art methods.

The simulation results are in line with our theoretical analysis. We show how the proposed

device selection scheme affects model convergence and demonstrate that the proposed solution to the minimization problem outperforms the baseline schemes in single-round latency reduction.

The rest of this paper is organized as follows. Section II describes the FL system model, scheduling policy and system latency model. Convergence analysis of device selection is given in Section III. The latency minimization problem and its simplification and solutions are provided in Section IV. Experimental results are analyzed in Section V. In Section VI, we draw our conclusions.

II. System Model

A. Heterogeneous Federated Learning Model

Consider a wireless FL system consisting of N single-antenna edge devices indexed by set $\mathcal{N} = \{1, 2, \ldots, N\}$ and a single-antenna edge server, as illustrated in Fig. 1. Each edge device $n \in \mathcal{N}$ has a local dataset $\mathcal{D}_n$ with m_n data samples. All edge devices collaboratively learn a shared global model by communicating with the edge server by minimizing the sum of the edge devices' local loss function. This can be formulated as the following optimization problem:

$$\mathscr{P}: \underset{\boldsymbol{\theta}, \{\boldsymbol{\theta}_n\}_{n=1}^N}{\text{minimize}} \; F(\boldsymbol{\theta}) \triangleq \sum_{n=1}^{N} \mu_n F_n(\boldsymbol{\theta}_n) \tag{1}$$

where $F(\boldsymbol{\theta})$ is the objective function defined by the learning task, $m = \sum_{n=1}^{N} m_n$ is the total number of data samples from all edge devices, $\mu_n = \frac{m_n}{m}$ is the relative sample size and $\boldsymbol{\theta} \in \mathbb{R}^d$ is the global model with dimension d. For each edge device n, $\boldsymbol{\theta}_n \in \mathbb{R}^d$ is the local model trained based on its local dataset $\mathcal{D}_n$ and F_n is the local loss function that can be written as

$$F_n(\boldsymbol{\theta}) = \frac{1}{m_n} \sum_{\xi \in \mathcal{D}_n} f_n(\boldsymbol{\theta}; \xi), \tag{2}$$

where ξ is a single data sample from local dataset $\mathcal{D}_n$ and f_n denotes the sample-wise loss function defined by the learning task.

To solve problem (1), the federated averaging (**FedAvg**) algorithm proposed in [1] can be applied, where each edge device takes $e \geq 1$ stochastic gradient steps and then sends its local updated model to the edge server. Then the edge server computes the average of the local models, and then sends the updated global model back to all edge devices.

During the training phase, multiple communication rounds are required before the global model converges. Let e_n denote the number of local updates performed by edge device n in a communication round. To reduce the expensive communication overhead, it is critical to reduce

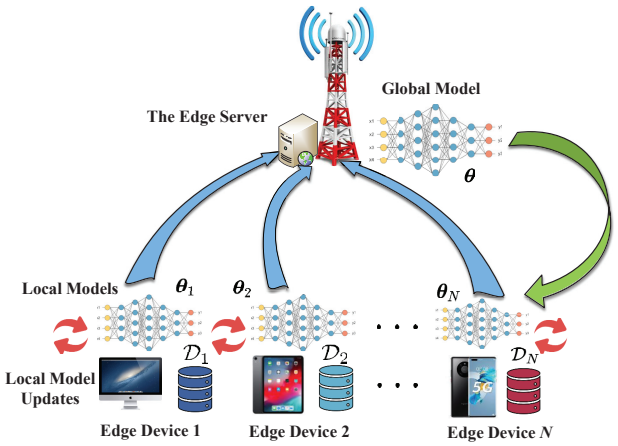


Fig. 1. Wireless FL system with N edge devices and an edge server.

the total number of communication rounds required by the whole training procedure. In order to achieve this goal, all edge devices need to fully leverage their computation resources to perform as more local updates as they can in each communication round. Most recent works assume that all edge devices compute the same number of local updates, i.e., $e_n = e$. However, in many practical scenarios, the edge devices involved in FL are highly heterogeneous in terms of the computation capacity and the size of local dataset $\mathcal{D}_n$. This causes heterogeneity in the number of local updates, e.g., faster edge devices, who are rich in computation resources, can perform more local iterations than the slower ones.

Simply averaging edge devices' local models under heterogeneous setting in e_n results in the model converging to the stationary point of a surrogate objective $\tilde{F}(\boldsymbol{\theta}) = \sum_{n=1}^{N} w_n F_n(\boldsymbol{\theta}_n)$, where w_n is the aggregation weight that is different from the original μ_n. This is referred to as the *Objective Inconsistency Problem* [42]. For example, when FedAvg algorithm is used for model aggregation, the aggregation weight for edge device n is $w_n = \frac{\mu_n e_n}{\sum_{n=1}^{N} \mu_n e_n}$ rather than μ_n. It is thus important to investigate the heterogeneous setting of local updates for practical implementations.

Let $\boldsymbol{\theta}^{(t,0)}$ denote the global model in communication round t and $\boldsymbol{\theta}_n^{(t,e)}$ denote the local model after e local updates in communication round t for edge device n. Fig. 2 illustrates this inconsistency phenomenon, where edge device 1 and 2 perform different numbers of local updates, i.e., $e_1 = 5$ for edge device 1 and $e_2 = 2$ for edge device 2. Obviously, the updated global model $\boldsymbol{\theta}^{(t+1,0)}$ is moving away from the global minimum $\boldsymbol{\theta}^*$, where

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} F(\boldsymbol{\theta}). \tag{3}$$

Here, $\boldsymbol{\theta}^{(t+1,0)}$ is pointing towards the local minimum of edge device 1, due to more local updates

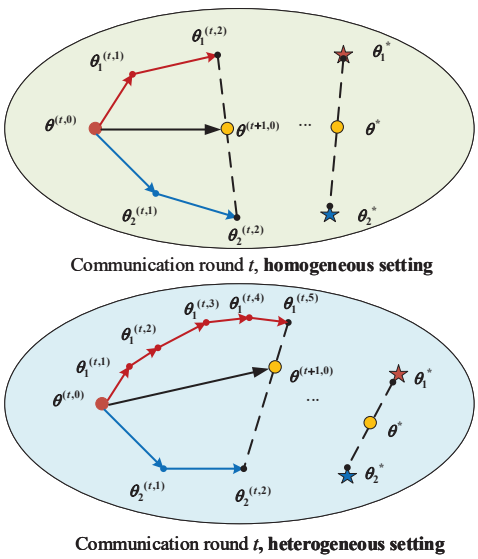


Fig. 2. Homogeneous and heterogeneous settings of local updates with two edge devices.

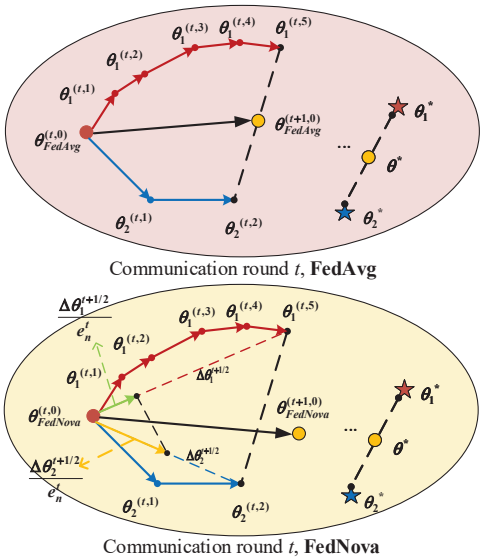


Fig. 3. Illustration of how FedNova deal with objective inconsistency problem in comparison to FedAvg with two edge devices, $e_1^t = 5$ for device 1 and $e_2^t = 2$ for device 2.

at edge device 1. This is different from the homogeneous setting where all edge devices compute the same number of local updates.

To address this challenging problem, [42] proposed a general framework which allows different e_n for different edge devices and different local optimizers such as GD, stochastic gradient descent (SGD), SGD with momentum, proximal updates, etc. Furthermore, [42] first provided the theoretical convergence rate under the heterogeneous computation setting, i.e., the heterogeneity in each edge device's local updates e_n. Finally, the proposed new algorithmic framework, called federated normalized averaging (**FedNova**), ensures objective consistency by providing correct method of weighted aggregation, which outperforms the state-of-art algorithms [1], [16] in terms of convergence rate. In FedNova, each local change is rescaled by a parameter called effective local steps τ_{eff}. We will present the principles of FedNova in the next subsection to illustrate its capability for mitigating the objective inconsistency problem.

B. Heterogeneous Federated Optimization Algorithm

In each communication round t, the computation and transmission procedure between the edge server and edge devices in the FL model can be summarized as follows:

a. The edge server broadcasts the global model $\boldsymbol{\theta}^{(t,0)}$ to all edge devices;

b. Each edge device $n \in \mathcal{N}$ receives the global model $\boldsymbol{\theta}^{(t,0)}$ and trains its local FL model using $\mathcal{D}_n$ by performing e_n^t local updates;

c. All edge devices send the trained local models to the edge server;

d. The edge server aggregates all the local models and updates the global model $\boldsymbol{\theta}^{(t+1,0)}$.

Note that the local SGD method is widely used to reduce the computation complexity for local updates. However, local SGD requires more global iterations to converge compared to local GD, i.e., more global communication rounds for wireless transmission, which causes high communication overheads. Meanwhile, if high accuracy is needed, local GD would be preferred [45]. In this case, we present the update rule of FedNova with local GD updates. Based on the received current global model $\boldsymbol{\theta}^{(t,0)}$ and the local dataset $\mathcal{D}_n$, edge device n updates the local model by using the local GD method.

Definition 1. *A single gradient update at communication round t given the index of local updates $e \in [0, e_n^t - 1]$ and a stepsize $s > 0$ for edge device n is*

$$\boldsymbol{\theta}_n^{(t,e+1)} \triangleq \boldsymbol{\theta}_n^{(t,e)} - s\nabla F_n(\boldsymbol{\theta}_n^{(t,e)}). \tag{4}$$

Applying Definition 1 from $e = 0$ to $e_n^t - 1$, the model progress from $\boldsymbol{\theta}_n^{(t,0)}$ to $\boldsymbol{\theta}_n^{(t,e_n^t)}$ can be expressed by the following definition:

Definition 2. *The global model progress after e_n^t local GD updates given a stepsize $s > 0$ for edge device n is*

$$\Delta\boldsymbol{\theta}_n^{t+1/2} \triangleq -s \sum_{e=0}^{e_n^t-1} \nabla F_n(\boldsymbol{\theta}_n^{(t,e)}). \tag{5}$$

Then $\Delta\boldsymbol{\theta}_n^{t+1/2}/e_n^t$ is uploaded by edge device n to the edge server as the local model. After receiving all local models in (5), the edge server immediately computes the global model update based on FedNova as follows:

$$\boldsymbol{\theta}^{(t+1,0)} = \boldsymbol{\theta}^{(t,0)} + \tau_{\text{eff}}^t \sum_{n=1}^{N} \mu_n \frac{\Delta\boldsymbol{\theta}_n^{t+1/2}}{e_n^t}, \tag{6}$$

where $\tau_{\text{eff}}^t = \sum_{n=1}^N \mu_n e_n^t$. Compared with the following global model aggregation method of FedAvg,

$$\boldsymbol{\theta}^{(t+1,0)} = \boldsymbol{\theta}^{(t,0)} + \sum_{n=1}^{N} \mu_n \Delta\boldsymbol{\theta}_n^{t+1/2}, \tag{7}$$

each local model progress in FedNova is multiplied by the coefficient $\tau_{\text{eff}}^t/e_n^t$ to eliminate objective inconsistency [42, Theorem 3]. To better show the advantages of the update rule of FedNova, we make a comparison between FedAvg and FedNova in Fig. 3. We can see that the standard FedAvg will clearly lead to a biased global model while FedNova makes the global

model back on track to the global minimum $\boldsymbol{\theta}^*$ by multiplying the scaling parameter $1/e_n^t$.

C. Probabilistic Selection Policy

Due to limited communication resources (e.g., bandwidth and power), it is generally impractical for all edge devices to upload their local models in each communication round t. This means that only a subset of edge devices in $\mathcal{N}$ can upload their local models to the edge server [29].

Let $S(t) \subset \mathcal{N}$ with size $|S(t)| = K$ denote the subset of edge devices involved in communication round t. If device n is selected in communication round t by the edge server, then we have $a_n^t = 1$. Otherwise, we set $a_n^t = 0$.

We present a probabilistic scheduling scheme *with replacement* for the sake of convenience for theoretical analysis. Specifically, we assume that K edge devices in subset $S(t)$ are randomly sampled from set $\mathcal{N}$ with replacement with probability μ_n [46]. The global model aggregation after receiving all the local models from edge devices in $S(t)$ can be written as

$$\boldsymbol{\theta}^{(t+1,0)} = \boldsymbol{\theta}^{(t,0)} + \tau_{\text{eff}}^t \sum_{n=1}^{K} \frac{1}{K} \frac{\Delta \boldsymbol{\theta}_n^{t+1/2}}{e_n^t}, \tag{8}$$

where $\tau_{\text{eff}}^t = \sum_{n=1}^{K} \mu_n e_n^t$.

In summary, the developed FedNova algorithm with device selection is presented in Algorithm 1.

D. System Latency Model

Latency is one of the key performance measures for implementing FL in wireless networks [44]. However, most recent studies on latency minimization for wireless FL failed to consider the heterogeneous local updates [32]–[37], [44]. We characterize the communication and computation latency of each FL communication round where different selected edge devices perform different numbers of local updates. Specifically, we consider the following three phases including local calculation, uplink transmission, and global dissemination.

1) Local Calculation: To perform e_n^t local updates in communication round t, the local calculation time of edge device n is defined as

$$T_l^{(n,t)} = \frac{e_n^t R m_n}{f_c^n}. \tag{9}$$

Here, f_n^c is the computation capacity of edge device n, quantified by the frequency of CPU of edge device n and R is the number of CPU cycles required for calculating a single data sample ξ.

Algorithm 1: FedNova with device selection

Input: Initial $\boldsymbol{\theta}^{(0,0)}$, max number of rounds τ;

Algorithm of the Edge Server:

Initialize $\boldsymbol{\theta}^{(0,0)}$ and broadcast it to the selected edge devices. Set $t = 0$, $n \in S(t)$;

repeat

  wait until receiving $\Delta\boldsymbol{\theta}_n^{t+1/2}/e_n^t$ and $\mu_n e_n^t$ from the selected edge devices ;

  update $\boldsymbol{\theta}^{(t+1,0)}$ via (8);

  broadcast global model $\boldsymbol{\theta}^{(t+1,0)}$ to all selected edge devices;

  set $t \leftarrow t + 1$;

until $t = \tau$;

Algorithm of the nth Selected Edge Device:

Initialization: $\boldsymbol{\theta}_n^{(0,0)} = \boldsymbol{\theta}^{(0,0)}, s, e_n^t, t = 0, n \in \mathcal{N}$;

repeat

  wait until receiving $\boldsymbol{\theta}^{(t,0)}$ from the edge server ;

  update e_n^t, $t = 0, 1, ..., \tau$, local model and local progress via (4) and (5) respectively;

  send $\Delta\boldsymbol{\theta}_n^{t+1/2}/e_n^t$ and $\mu_n e_n^t$ to the edge server;

  set $t \leftarrow t + 1$

until $t = \tau$;

2) Uplink Transmission: Each selected edge device n uploads its local model $\Delta\boldsymbol{\theta}_n^{t+1/2}$ to the edge server via frequency division multiple access (FDMA) [36]. The transmission rate in communication round t for edge device n is given by

$$r_u^{(n,t)} = B_u^{(n,t)} \log_2 \left(1 + \frac{h_n^t p_n}{B_u^{(n,t)} N_0}\right), \tag{10}$$

where $B_u^{(n,t)}$ is the uplink bandwidth assigned to edge device n, h_n^t is the channel gain between edge device n and the edge server in communication round t, p_n is the transmit power of edge device n and N_0 is the power spectral density of the additive Gaussian noise. We assume that the amount of information per local model parameter is I. Since the size of local model parameters for each edge device is a fixed constant d, the amount of information of a local model is Id. In this case, the uplink transmission time in communication round t for edge device n is

$$T_u^{(n,t)} = \frac{Id}{r_u^{(n,t)}} = \frac{Id}{B_u^{(n,t)} \log_2(1 + \frac{h_n^t p_n}{B_u^{(n,t)} N_0})}. \tag{11}$$

Due to limited bandwidth, we have $\sum_{n=1}^{N} B_u^{(n,t)} \leq B$, where B is the total system bandwidth.

3) Global Computation and Broadcast: After successfully receiving local models from the selected edge devices, the edge server generally has abundant computation resources to calculate the global model aggregation, and hence the global computation time can be neglected. As depicted in Fig. 1, in the t-th global communication round, the edge server broadcasts the

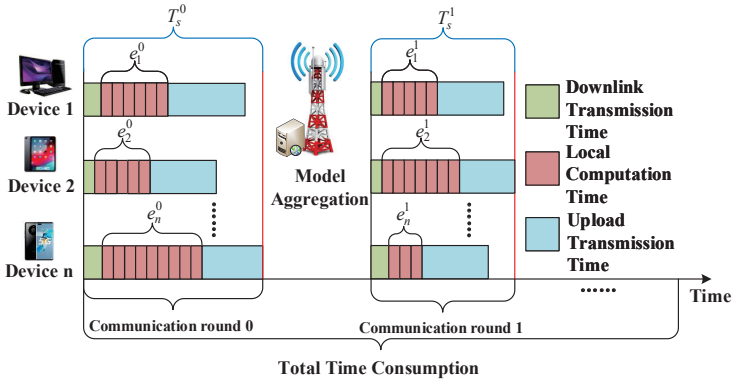


Fig. 4. Illustration of the transmission delay.

current global model $\boldsymbol{\theta}^{(t+1,0)}$ to all selected edge devices in the downlink, during which the downlink transmission time can be written as

$$T_D^{(n,t)} = \frac{Id}{B\log_2(1+\frac{h_n^t P_0}{BN_0})}, \tag{12}$$

where P_0 is the transmit power of the edge server and B is the downlink bandwidth assigned to the edge server, assuming that downlink bandwidth is equal to system bandwidth B [32], [33], [37]. Therefore, each edge device can be assumed to receive the current shared global model $\boldsymbol{\theta}^{(t+1,0)}$ without distortion [47]–[49].

4) Total Latency: As synchronized transmission scheme for global model aggregation is adopted, e.g., after broadcasting the global model to all the selected edge devices and the calculation locally, the edge server needs to wait until receiving the local model of the slowest edge device [2], as shown in Fig. 4. In particular, the single-round total latency is determined by the slowest selected edge device which consists of three parts

$$T_s^t = \max_{n\in S(t)} (T_D^{(n,t)} + T_l^{(n,t)} + T_u^{(n,t)}). \tag{13}$$

We assume that Algorithm 1 solves problem (1) with τ global iterations, where the communication rounds are indexed by set $\mathcal{T} = \{0, 1, 2, \ldots, \tau-1\}$. The total latency for solving the learning problem in τ communication rounds is thus given by

$$T(\boldsymbol{p}, \boldsymbol{f_c}, \boldsymbol{e^t}, \boldsymbol{B_u^t}, \boldsymbol{S}, \boldsymbol{\mathcal{T}}) = \sum_{t=0}^{\tau-1} T_s^t, \tag{14}$$

where $\boldsymbol{p} = [p_1, \ldots, p_K]^T$, $\boldsymbol{f_c} = [f_c^1, \ldots, f_c^K]^T$, $\boldsymbol{B_u^t} = [B_u^{(1,t)}, \ldots, B_u^{(K,t)}]^T$, $\boldsymbol{e^t} = [e_1^t, \ldots, e_K^t]^T$ and $\boldsymbol{S} = [S(0), \ldots, S(\tau-1)]^T$.

In this paper, we shall minimize the system latency by characterizing the convergence behavior of Algorithm 1 in Section III, followed by developing the system optimization algorithm in Section IV.

III. Convergence Analysis

In this section, we will provide convergence analysis of implementing device scheduling and local GD updates in FedNova algorithm. This helps us gain insights on dealing with the system total latency. The difficulty of convergence analysis of the heterogeneous FL with FedNova and device selection lies in the following two aspects. First, most recent works focused on providing convergence analysis for homogeneous local updates, which is unable to cope with new challenges posed by the heterogeneity of local updates. Second, in the presence of device selection, the methods used in [32], [33] mainly divide all edge devices into two subsets: the set of selected edge devices and the set of unselected edge devices, regarding device selection procedure as an error term. However, this method cannot be directly applied in FedNova. This is because that the local updates are rescaled in (8).

To address above challenges, we first make the following assumptions which are adopted in [16], [42], [46], [50], [51].

Assumption 1 (Smoothness)**.** *The loss functions $F_n : \mathbb{R}^d \to \mathbb{R}$ is L-smooth ($L > 0$), that is for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$\|\nabla F_n(\boldsymbol{x}) - \nabla F_n(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \tag{15}$$

which is equivalent to

$$F_n(\boldsymbol{y}) \leq F_n(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^T \nabla F_n(\boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{16}$$

Assumption 2 (Strongly Convex)**.** *The loss function $F : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex ($\mu > 0$), that is for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$F(\boldsymbol{y}) \geq F(\boldsymbol{x}) + (\boldsymbol{y} - \boldsymbol{x})^T \nabla F(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{17}$$

Lemma 1 (Polyak-Lojasiewicz Condition)**.** *For a given μ-strongly convex function $F(\boldsymbol{x})$, we have the following conclusion*

$$\|\nabla F(\boldsymbol{x}^t)\|^2 \geq 2\mu[F(\boldsymbol{x}^t) - F(\boldsymbol{x}^*)], \tag{18}$$

where $\boldsymbol{x}^*$ denotes the minimizer of function $F(\boldsymbol{x})$.

Assumption 3 (Bounded Gradient Dissimilarity)**.** *For all $\beta_1 \geq 1$ and $\beta_2 \geq 0$, we have*

$$\sum_{n=1}^{N} \mu_n \|\nabla F_n(\boldsymbol{x})\|^2 \leq \beta_2 + \beta_1 \|\sum_{n=1}^{N} \mu_n \nabla F_n(\boldsymbol{x})\|^2. \tag{19}$$

Second, in the presence of device selection, we derive an upper bound on the expected objective value and the convergence rate of FL model considered in communication round t as follows.

Theorem 1. *Based on Assumption 1, 2, 3 and Lemma 1, given a constraint learning rate s which satisfies $0 < s < \min_n \left\{1, \frac{4}{\tau_{\text{eff}}^t \mu}, \frac{1}{L\sqrt{2(2\beta_1+1)\max_n(e_n^t)}}, \frac{K}{(18K+12\beta_1)\tau_{\text{eff}}^t L}\right\}$, the upper bound of the optimality gap in each communication round t is*

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)})\right] - F(\boldsymbol{\theta}^*) \leq \rho_t \left\{\mathbb{E}\left[F(\boldsymbol{\theta}^{(t,0)})\right] - F(\boldsymbol{\theta}^*)\right\} + \delta_t, \tag{20}$$

where $\rho_t = 1 - \frac{\tau_{\text{eff}}^t s\mu}{4}$, $\delta_t = \frac{\tau_{\text{eff}}^t s\beta_2}{2}[\frac{(3\tau_{\text{eff}}^t sL+1)D}{1-D} + \frac{3\tau_{\text{eff}}^t sL}{K}]$ and $D = 2s^2L^2\max_n(e_n^t)$. What's more, the expectation is taken over the selected set $S(t)$ due to randomness of device selection.

Proof. Please refer to Appendix A for details. □

We can extend Theorem 1 to the upper bound on cumulative optimality gap in the whole FL training procedure, which is summarized in the following Corollary:

Corollary 1. *Based on Theorem 1, given initialized global model $\boldsymbol{\theta}^{(0,0)}$ together with a constraint learning rate $0 < s < \min_{n,t} \left\{1, \frac{4}{\tau_{\text{eff}}^t \mu}, \frac{1}{L\sqrt{2(2\beta_1+1)\max_n(e_n^t)}}, \frac{K}{(18K+12\beta_1)\tau_{\text{eff}}^t L}\right\}$, the upper bound on the cumulative gap after τ communication rounds is*

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(\tau,0)})\right] - F(\boldsymbol{\theta}^*) \leq \prod_{t=0}^{\tau-1} \rho_t \left\{\mathbb{E}\left[F(\boldsymbol{\theta}^{(0,0)})\right] - F(\boldsymbol{\theta}^*)\right\} + \sum_{t=0}^{\tau-1} \delta_t \prod_{i=t+1}^{\tau-1} \rho_i. \tag{21}$$

Proof. Applying (20) cumulatively for t from 0 to $\tau - 1$, we can obtain the cumulative upper bound after τ communication rounds shown above. □

Besides, from Theorem 1 and Corollary 1, two conclusions can be drawn in the following remarks.

Remark 1. *From Theorem 1, we observe that the convergence performance of FedNova with local GD updates depends on the size of selected set $S(t)$ and parameter τ_{eff}^t which is determined by the numbers of local updates and probabilities of the selected edge devices at communication round t.*

Remark 2. *When it comes to how K affects convergence performance, in the first place, we can see from* (20) *that the more edge devices we select, the smaller the gap δ_t will be, i.e. the gap δ_t decreases with K. When K reduces to N, i.e., full devices participation, the gap is minimized. Secondly, tracing back to* (39) *and* (40)*, ρ_t decreases with K, i.e., the convergence speed increases with K. When K increases to N, i.e., full devices participation, the convergence speed is maximized.*

As a result, it is clear that the number of local updates in each communication round e_n^t, and the number of selected edge devices at each communication round, are relevant to both the system latency and the model convergence.

IV. System Latency Minimization

In this section, we develop a system latency optimization approach to minimize the total latency by communication and computation resource allocation for all selected edge devices. Based on the convergence results in Section III, and given a scheme of local updates and the number of selected edge devices, we reduce the problem to single-round latency minimization where the resource allocation is independent of model convergence. We first derive a closed-form solution to the optimal bandwidth allocation, which is a convex problem and can be solved by bisection search. We then propose an alternating optimization scheme to find a tradeoff between communication and computation resources, thereby mitigating the straggler issue in the optimal bandwidth allocation.

A. Problem Formulation

We now formulate an optimization problem whose goal is to minimize the total latency among all selected edge devices. Specifically, all edge devices should have the same transmit power, i.e., $p_1 = p_2 = \cdots = p_N = p^{\max}$, which is always feasible without taking energy consumption into consideration as T_u^n decreasing with p_n. The latency minimization problem is given by

$$\underset{\boldsymbol{f_c},\boldsymbol{e^t},\boldsymbol{B_u^t},\boldsymbol{S},\boldsymbol{\mathcal{T}}}{\text{minimize}} \; T \tag{22}$$

$$\text{subject to } 0 \leq f_c^n \leq f^{\max}, \forall n \in \boldsymbol{S}, \tag{22a}$$

$$1 \leq e_n^t \leq e^{\max}, e_n^t \in \mathbb{Z}, \forall n \in S(t), \forall t, \tag{22b}$$

$$\sum_{n=1}^{K} B_u^{(n,t)} \leq B, B_u^{(n,t)} \geq 0, \forall n \in S(t), \forall t, \tag{22c}$$

where $e^{\max}$ and $f^{\max}$ are the maximum number of local steps and local computation capacity of all selected edge devices. In particular, (22a) represents the local computation capacity of all selected edge devices, while (22b) is the limit of the number of local updates in communication round t. The bandwidth limitation of selected edge device n in communication round t is given in (22c).

For resource allocation in wireless FL systems, the authors in [37] proved that distributed gradient calculation is equivalent to the centralized one. Hence the parameter allocation and

bandwidth allocation are independent of the convergence rate, reducing the problem to a one-round version. Furthermore, [32] showed that the total number of rounds for the FL algorithm to converge depends only on the device selection scheme and has nothing to do with the latency of each communication round, so that they can minimize the convergence time of each iteration. Similar to [34], the authors divided the joint device scheduling and bandwidth allocation problem into two sub-problems. In this case, given the scheduled edge devices in each communication round, the bandwidth allocation and latency are independent of model convergence, indicating that only optimizing latency in one round is sufficient. However in our work, notice that in problem (22) with the optimization variable set $\{\boldsymbol{f_c}, \boldsymbol{e^t}, \boldsymbol{B_u^t}, \boldsymbol{S}\}$, variables $\boldsymbol{e^t}$ and $\boldsymbol{S}$ are relevant to the model convergence according to Theorem 1 and Corollary 1, and thus are involved in the whole communication rounds during the learning procedure. In order to address this coupling issue, we first fix $\boldsymbol{e^t}$ for each communication round t. In particular, the proposed scheme of distribution of local updates for edge device n in communication round t is

$$(\hat{e}_n^t)^* = (\hat{e}_n)^* = \begin{cases} \text{round}(\frac{1}{\mu_n}) & \text{round}(\frac{1}{\mu_n}) < e^{\max}, \\ e^{\max} & \text{round}(\frac{1}{\mu_n}) \geq e^{\max}, \end{cases} \tag{23}$$

where the number of local updates of edge devices n is inversely proportional to the ratio of sample size, and is fixed for each communication round. This means that edge device with more local data samples performs fewer local updates, thus preserving heterogeneity. Moreover, the subsets of participated edge devices of all communication rounds $\boldsymbol{S}$ can be assumed to be known to the edge server at the beginning of each communication round, and the size of the selected edge devices, $K = |S(t)|$, is set to be fixed for all communication rounds. In this case, the total system latency becomes independent of the total communication rounds. Problem (22) is thus equivalent to minimizing single-round total latency denoted by $(T_s^t)^*$, which is different across communication rounds. The latency of all communication rounds is $T(\boldsymbol{f_c}, \boldsymbol{B_u}) = \sum_{t=0}^{\tau}(T_s^t)^*$.

In summary, the total latency minimization problem (22) can be further formulated as the following single-round latency minimization problem

$$\underset{\boldsymbol{f_c}, \boldsymbol{B_u}}{\text{minimize}} \; T_s^*(\boldsymbol{f_c}, \boldsymbol{B_u}) = \max_{n \in S} \; (T_d^n + T_l^n + T_u^n) \tag{24}$$

$$\text{subject to } 0 \leq f_c^n \leq f^{\max}, \forall n \in S, \tag{24a}$$

$$\sum_{n=1}^{K} B_u^n \leq B, B_u^n \geq 0, \forall n \in S, \tag{24b}$$

where $T_D^n = \frac{Id}{B^d \log_2(1+\frac{h_n P_0}{B^d N_0})}$, $T_l^n = \frac{(\hat{e}_n)^* R m_n}{f_c^n}$ and $T_u^n = \frac{Id}{B_u^n \log_2(1+\frac{h_n p^{\max}}{B_u^n N_0})}$.

This optimization problem aims at optimizing the single-round latency by jointly allocating the bandwidth and the local computation capacity. We shall develop an alternative optimization method to solve problem (24) by fixing one variable while optimizing another. For convenience, we omit the notation t of the bandwidth, the subset $S(t)$ and the latency.

B. Optimal Bandwidth Allocation

In order to solve problem (24), in this subsection, we first fix the local computation capacity. Assume that each local computation capacity f_c^n is fixed as $f_c^n = (\hat{f}_c^n)^*$, which satisfies (24b) and only depends on the computation capability of edge device. Then problem (24) reduces to the following joint single-round latency and bandwidth allocation problem:

$$\underset{\boldsymbol{B_u}}{\text{minimize}}\ T_s^*(\boldsymbol{B_u}) = \max_{n\in S}\ ((\hat{T}_l^n)^* + T_u^n + T_D^n) \tag{25}$$

$$\text{subject to}\ \sum_{n=1}^{K} B_u^n \le B, B_u^n \ge 0, \forall n \in S, \tag{25a}$$

where $(\hat{T}_l^n)^* = \frac{(\hat{e}_n)^* R m_n}{(\hat{f}_c^n)^*}$ and $T_u^n = \frac{Id}{B_u^n \log_2(1+\frac{h_n p^{\max}}{B_u^n N_0})}$. The following lemma can be applied to solve (25).

Lemma 2. *Problem* (25) *is a convex problem.*

By solving convex problem (25), we can numerically obtain the optimal bandwidth allocation and the minimal single-round latency of communication round t as presented in the following theorem.

Theorem 2. *Given fixed local updates $\hat{e}_n^*$ and the allocation of local computation capacities $(\hat{f}_c^n)^*$, by solving convex problem* (25)*, the proposed scheme of optimal bandwidth allocation is given by*

$$(B_u^n)^* = (r_u^n)^{-1}\left(\frac{Id}{T_s^* - (\hat{T}_l^n)^* - T_D^n}\right), \forall n \in S, \tag{26}$$

where $(r_u^n)^{-1}$ is the inverse function of $r_u^n(B_u^n)$. In fact, the explicit expression of the optimal solution to the bandwidth allocation (26) *is*

$$(B_u^n)^* = \frac{-\frac{h_n p^{\max}}{N_0}\alpha_n}{W\left(-\alpha_n e^{-\alpha_n}\right) + \alpha_n}, \forall n \in S, \tag{27}$$

where $\alpha_n = \frac{IdN_0 \ln 2}{h_n p^{\max}(T_s^* - (\hat{T}_l^n)^* - T_D^n)}$ and $W(\cdot)$ is Lambert-W function [52]. Furthermore, the optimal solution of the single-round latency minimization problem T_s^* can be attained using a bisection

Algorithm 2: Optimal Bandwidth Allocation

Input : $(\hat{f}_c^n)^*$, $\hat{e}_n^*$, m_n, $n \in S$, T_{low}, T_{up} and accuracy ϵ_0;
Initialize $B_s = \sum_{n=1}^{K} B_u^n > B + \epsilon_0$;
while $B_s - B > \epsilon_0$ **do**
    Set $T_{\text{middle}} = \frac{T_{\text{low}} + T_{\text{up}}}{2}$, and $T_s = T_{\text{middle}}$;
    For each device, compute the bandwidth B_u^n via (27) and calculate the summation of bandwidth allocation B_s;
    if $B_s - B < 0$ **then**
        $T_{\text{up}} \leftarrow T_{\text{middle}}$;
    else
        $T_{\text{low}} \leftarrow T_{\text{middle}}$;
    end
end
Output: T_s^* and $(B_n^u)^*, n \in S$.

Algorithm 3: Alternating Optimization Approach

Input : $(\hat{B}_n^u)^1$, $\hat{e}_n^*$, m_n, $n \in S$, iter;
for $i = 1$ *to iter* **do**
    $(\hat{f}_c^n)^i, n \in S \leftarrow$ Optimal local computation capacity from Lemma 3 with $(\hat{B}_n^u)^i$, $\hat{e}_n^*$, $n \in S$;
    T_s^i and $(B_n^u)^{i+1}, n \in S \leftarrow$ Optimal Bandwidth Allocation from Algorithm 2 with $(\hat{f}_c^n)^i, \hat{e}_n^*, n \in S$;
end
Output: T_s^* and $(B_n^u)^*, (f_c^n)^*, n \in S$.

search approach,

$$\sum_{n=1}^{K} \frac{-\frac{h_n p^{\max}}{N_0}\alpha_n}{W\left(-\alpha_n e^{-\alpha_n}\right) + \alpha_n} = B. \tag{28}$$

It indicates that all edge devices have the same latency T_s^* in one communication round.

Proof. Please refer to Appendix B for details. □

The scheme for optimal bandwidth allocation and bisection search is presented in Algorithm 2 and the complexity of this algorithm is in the order of $\mathcal{O}\left(K \log_2\left(\frac{T_{\text{up}}}{\epsilon_0}\right)\right)$.

Remark 3. *To analyze Theorem 2, we first observe, from* (26)*, that when the predefined local computation time of edge device n, $(\hat{T}_l^n)^*$, together with downlink transmission time T_D^n, are small, then the bandwidth allocated to edge device n will be small as well. That is to say, we should allocate the highest bandwidth to the slowest edge device who receives the global model with high latency and slow calculation, i.e., the straggler, while the smallest bandwidth*

should be allocated to the fastest edge device whose $T_D + T_l$ is the smallest among all the edge devices. Secondly, if some selected edge devices have finished their local updates earlier than other edge devices, we can re-allocate some bandwidth to other slower edge devices. Hence, the optimal bandwidth can be attained if and only if all bandwidth is allocated and all the selected edge devices have the same latency T_s^* in a single communication round. Thirdly, the minimal single-round latency T_s^* decreases with system bandwidth B. Finally, due to the heterogeneity of local computation time, the optimal bandwidth allocation may suffer from the stragglers.

C. Optimal Local Computation Capacity

In this subsection, given fixed bandwidth allocation for each communication round, we optimize the local computation capacity. Similar to the last subsection, problem (24) reduces to the following joint single-round latency and computation capacity allocation problem given fixed bandwidth allocation $(\hat{B}_u^n)^*$:

$$\underset{\boldsymbol{f_c}}{\text{minimize}}\ T_s^*(\boldsymbol{f_c}) = \max_{n \in S}\ (T_d^n + T_l^n + (\hat{T}_u^n)^*) \tag{29}$$

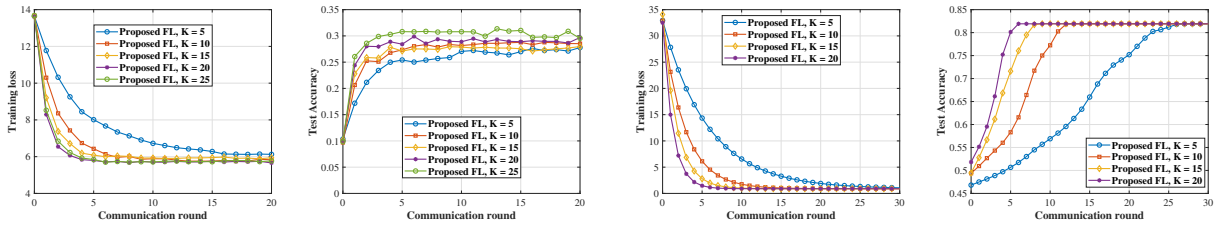
$$\text{subject to } 0 \leq f_c^n \leq f^{\max}, \forall n \in S, \tag{29a}$$

where $(\hat{T}_u^n)^* = \frac{Id}{(\hat{B}_u^n)^* \log_2(1+\frac{h_n p^{\max}}{(\hat{B}_u^n)^* N_0})}$. We can obtain the optimal solution from the following lemma.

Lemma 3. *Problem* (29) *is a convex problem and can be solve optimally by the off-the-shelf solvers incorporated in the disciplined convex program modeling framework CVX [53].*

Based on the above two results, we thus propose a scheme to jointly optimize the bandwidth allocation and local computation capability as presented in Algorithm 3.

From Remark 3, we know that the highest bandwidth should be allocated to the stragglers, which may become infeasible to the single-round latency minimization problem. To analyze the features of Algorithm 3, all the selected edge devices have the same latency T_s^* in a single communication round. Next, given the initialized bandwidth allocation $(\hat{B}_n^u)^1$, it manages to find an optimal local computation capacity $(\hat{f}_c^n)^i, n \in S$ for selected edge devices under initialized bandwidth allocation $(\hat{B}_n^u)^1$ and then uses the obtained optimal local computation capacity to find the optimal bandwidth allocation, which will be served as the initial bandwidth allocation in the next iteration. As a result, the proposed joint optimization method in Algorithm 3 aims at



(a) Training Loss of CIFAR-10 versus Communication Round

(b) Test Accuracy of CIFAR-10 versus Communication Round

(c) Training Loss of Mental Stress Detection versus Communication Round

(d) Test Accuracy of Mental Stress Detection versus Communication Round

Fig. 5. Simulation results for SVM on the CIFAR-10 and the mental stress detection dataset under proposed FL algorithm, revealing influence of the number of selected edge devices in each communication round.

further matching the bandwidth allocation with local computation time, thereby mitigating the stragglers effect and reducing the single-round latency.

V. Numerical Experiments

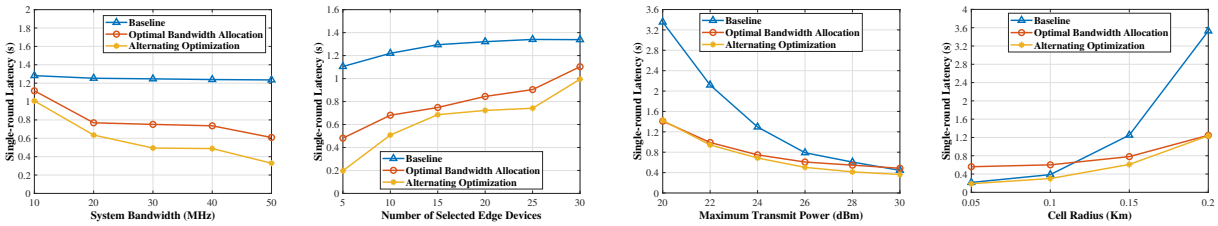
In this section, we provide numerical experiments to evaluate the performance of the proposed FL algorithms to solve problem (1) and the system optimization schemes for latency minimization problem (24).

A. Experiment Settings

For the setting of wireless network environment, we consider $N = 30$ edge devices uniformly distributed in a disk sized 0.15km $\times$ 0.15km with the edge server located at the center. Unless specified otherwise, we choose a system bandwidth $B = 20$MHz, maximum local updates $e^{\max} = 50$, maximum transmit power $p^{\max} = 24$dBm, broadcast power of the edge server $P_0 = 46$dBm, and select $K = 15$ edge devices in each communication round. The path loss model between edge devices and the edge server is $128.1+37.6 \log L$ with distance L in kilometer and the standard deviation of shadow fading is 10 dB. What's more, the power spectral density is $N_0 = -174$dBm/Hz. Parameter R is set as 1×10^4 cycles/sample, while the parameter I is 4 bits/dimension. The number of the local data samples of device n is randomly generated. All the experiments are averaged upon 100 independent runs.

B. Learning Performance of the FL Algorithm

We evaluate the influence of the number of selected edge devices K under two datasets.



(a) Influence of System Bandwidth (b) Influence of Number of Selected Edge Devices (c) Influence of Maximum Transmit Power (d) Influence of Cell Radius

Fig. 6. Simulation results for single-round latency minimization problem under the CIFAR-10 dataset with three proposed schemes, plotting optimal latency versus system bandwidth, number of selected edge devices, maximum transmit power and cell radius, respectively.

*1) **The CIFAR-10 Dataset [54] for Image Classification**:* The CIFAR-10 dataset has 50,000 training images with 10,000 images for tests, consisting of 10 categories of images. Each image sample has $32 \times 32 \times 3 = 3072$ features, i.e., $d = 3072$. For the experiments conducted under the CIFAR-10 dataset, we assume that different edge devices have random numbers of image samples in their local datasets and data samples follow the non-IID distribution.

*2) **The Mental Stress Detection Dataset [55] for Binary Classification**:* The raw mental stress detection dataset can be biomedically collected by means of the function of photoplethysmogram (PPG) based heart activity monitoring on the wearable smart watches. There were 32 people participating in a specific real-life experiment in Turkey, whose scenario is promising and perfectly implemented in the FL due to the privacy-preserving feature. The dataset was pre-processed by the authors in [55], and the detailed information on the pre-processing procedure can be found in [55, Section 3]. This dataset consists of 1798 data samples in total which are allocated in 26 edge devices, and 16 features of stressed and relaxed people. In addition, $N = 20$ edge devices are considered in the task of mental stress detection, i.e., binary classification. The remaining datasets collected by another six edge devices are served for tests.

We train a support vector machine (SVM) classifier via **FedNova** algorithm with local GD updates and select $K \in \{5, 10, 15, 20, 25\}$ edge devices for image classification and $K \in \{5, 10, 15, 20\}$ edge devices for stress detection by probabilistic scheduling policy in each communication round. Each selected edge device performs $(\hat{e}_n)^*$ local updates detailed in (23). We use a learning rate of 1×10^{-6} for image classification and stress detection. The training loss and test accuracy of the two aforementioned datasets versus communication round t are plotted in Fig. 5(a), 5(b) and Fig. 5(c), 5(d), respectively.

Fig. 5(a) and Fig. 5(c) show the training loss of different numbers of selected edge devices in each round versus the communication round t. We notice that, the more edge devices we select

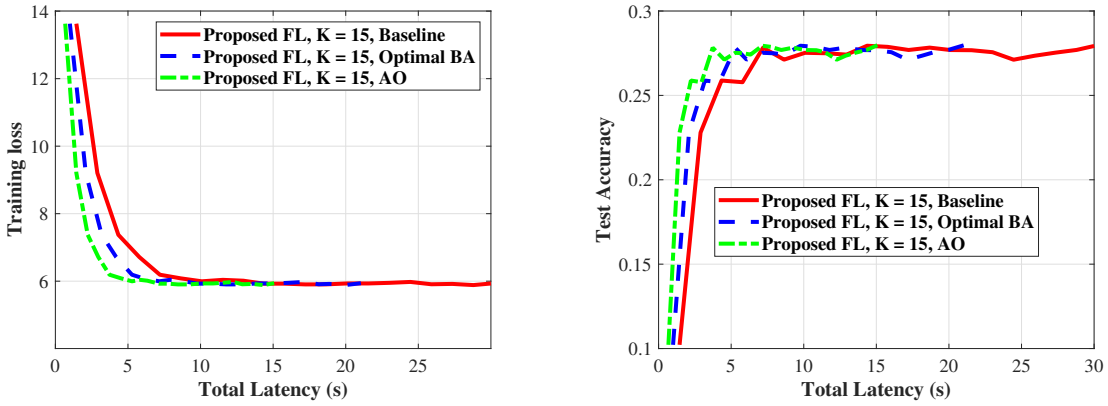
at each communication round, the faster the global model will converge. Moreover, as can be seen in Fig. 5(b) and Fig. 5(d), if we select more edge devices in each communication round, the test accuracy will be higher at the same communication round compared to that of selecting fewer edge devices. As the amount of the mental stress detection dataset is relatively small, the highest accuracy is only 81.92%. In conclusion, the simulation results verify our observations in Remark 2.

C. System Latency Minimization over the CIFAR-10 Dataset

We consider the following schemes for comparison.

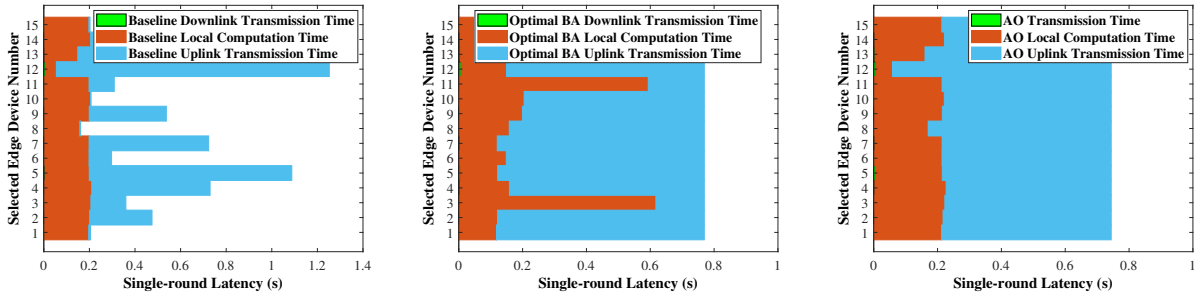
- **Baseline:** The local bandwidth are equally allocated by $(\hat{B}_u^1)^* = \cdots = (\hat{B}_u^K)^* = \frac{B}{K}$. Furthermore, the computation capacity of edge device n is assumed to be equal across edge devices, i.e., $(\hat{f}_c^n)^* = 3$GHz.
- **Optimal Bandwidth Allocation (Optimal BA):** The edge devices' computation capacities are uniformly and randomly selected from $[1, 2, \ldots, 6]\times$GHz followed by optimizing the bandwidth allocation by the method in Section IV-B.
- **Alternating Optimization (AO):** The input local bandwidth are equally allocated as $(\hat{B}_n^u)^1 = \frac{B}{K}$. This scheme aims at jointly optimizing bandwidth allocation and local computation capacity by Algorithm 3.

Consider the image classification task on the CIFAR-10 dataset, which is more complicated than mental stress detection, the performance of solving the single-round latency minimization problem of the three aforementioned algorithms are demonstrated in Fig. 6. First we analyze the performance for a varying system bandwidth B. The results clearly show that, when system bandwidth increases, the single-round latency of Optimal BA and AO will decrease, which verifies our theoretical analysis in Appendix B. However, there is no indication that it would have significant influence on the baseline scheme due to complete averaged bandwidth allocation. Next, we find that for all three schemes, the more edge devices selected in each communication round, the higer latency of each round will be. It is also observed that larger maximum transmit power leads to smaller single-round latency for all three schemes. However, Optimal BA and AO are less sensitive to the change of maximum transmit power. Last but not least, it is intuitively observed from Fig. 6(d) that, as the radius of the cell increases, both the distance between the edge device and the edge server increases, and the communication time increases. This causes the single-round latency to increase with distance. When the radius is very small, the performance of



(a) Training Loss versus Total Latency (b) Test Accuracy versus Total Latency

Fig. 7. Simulation results for SVM on the CIFAR-10 dataset under proposed FL algorithm with $K = 15$ edge devices selected in each communication round, revealing the performance of three proposed schemes.



(a) Single-round Latency of Baseline (b) Single-round Latency of Optimal BA (c) Single-round Latency of AO

Fig. 8. Simulation results for single-round latency minimization problem under three proposed schemes, plotting an experimental version of Fig. 4, to illustrate the practical distribution of local calculation time and uplink/downlink transmission time.

Optimal BA is even the worst, but Optimal BA and AO are less sensitive to the radius, showing a linear growth trend, maintaining robustness. In contrast, baseline scheme shows an exponential growth trend with respect to the radius.

Practically speaking, with respect to the system optimization schemes for latency minimization, the AO scheme is the most ideal for baseline improvement, reaching 47.04%, followed by the Optimal BA scheme, reaching 42.23% under the default parameter settings. To show how the FL model training performance changes with time, we plot Fig. 7 to illustrate the learning performance versus time (latency) in seconds achieved by the three latency minimization schemes with $K = 15$. As we can see, without optimizing the resource allocation, i.e., baseline scheme, it would take more time for the whole training-and-communicating procedure to achieve the same training loss or test accuracy.

In the end, we plot an experimental version of Fig. 4 in Fig. 8 to illustrate the practical distribution of local calculation time and uplink/downlink transmission time. It is observed

from Fig. 8(a), 8(b) that in Optimal BA and AO, all edge devices share the same single-round latency, while the single-round latency of baseline maintains heterogeneity among $K = 15$ selected edge devices. The Optimal BA scheme just matches the bandwidth allocation, i.e., uplink transmission time, to local computation time, while AO scheme takes both communication and computation resources into consideration, further reducing the single-round latency and alleviating the stragglers effect, as depicted in Fig. 8(b). Besides, to the best of our knowledge, downlink transmission time can be neglected compared with uplink transmission [34], [36]. Therefore in Fig. 8, our experimental results confirm this assumption as downlink transmission time is quite small.

VI. Conclusion

In this paper, we investigated the problem of latency minimization for training a federated learning algorithm over wireless networks, where the edge devices perform different local updates in one communication round. Due to resource limitation in wireless networks, a probabilistic device selection policy selecting a part of edge devices in each round was considered. We established a total latency minimization problem and provided the convergence analysis of the considered federated learning algorithm. Observations from the convergence results inspired us to decouple the model convergence and the resource allocation. As a result, given the scheme of local updates and the number of selected edge devices, we reformulated the problem as the single-round latency minimization whose purpose is to minimize the latency among selected edge devices in one communication round and thereby achieving the goal to allocate both communication and computation resources. To solve this problem, we proposed an alternating optimization scheme to balance communication and computation resources. Our simulation results were consistent with our theoretical analysis for the impact of device selection and the properties of proposed solutions to latency minimization and resource allocation.

Appendix

A. *Proof of Theorem 1*

First reformulate (8) as

$$\boldsymbol{\theta}^{(t+1,0)} - \boldsymbol{\theta}^{(t,0)} = -\tau_{\text{eff}}^{t} s \sum_{n=1}^{K} \frac{1}{K} G_n^t, \tag{30}$$

where $G_n^t = \frac{1}{e_n^t} \sum_{e=0}^{e_n^t-1} \nabla F_n(\boldsymbol{\theta}_n^{(t,e)})$. According to (16), given $\boldsymbol{y} = \boldsymbol{\theta}^{(t+1,0)}$ and $\boldsymbol{x} = \boldsymbol{\theta}^{(t,0)}$, it is expressed as

$$F(\boldsymbol{\theta}^{(t+1,0)}) - F(\boldsymbol{\theta}^{(t,0)}) \leq -\tau_{\text{eff}}^t s \left\langle \bar{G}^t, \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\rangle + \frac{L(\tau_{\text{eff}}^t)^2 s^2}{2} \left\| \bar{G}^t \right\|^2, \tag{31}$$

where $\bar{G}^t = \frac{1}{K} \sum_{n=1}^{K} G_n^t$. Take expectation on the elements in the selected set $S(t)$, we have

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)}) - F(\boldsymbol{\theta}^{(t,0)})\right] \\
\leq& -\tau_{\text{eff}}^t s \mathbb{E}\left[\left\langle \bar{G}^t, \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\rangle\right] + \frac{L(\tau_{\text{eff}}^t)^2 s^2}{2} \mathbb{E}\left[\left\| \bar{G}^t \right\|^2\right] \\
=& -\tau_{\text{eff}}^t s \left\langle \sum_{n=1}^{N} \mu_n G_n^t, \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\rangle + \frac{L(\tau_{\text{eff}}^t)^2 s^2}{2} \mathbb{E}\left[\left\| \bar{G}^t \right\|^2\right] \\
=& -\frac{\tau_{\text{eff}}^t s}{2} (\left\| \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\|^2 + \left\| \sum_{n=1}^{N} \mu_n G_n^t \right\|^2 - \underbrace{\left\| \nabla F(\boldsymbol{\theta}^{(t,0)}) - \sum_{n=1}^{N} \mu_n G_n^t \right\|^2}_{P_1}) + \frac{L(\tau_{\text{eff}}^t)^2 s^2}{2} \mathbb{E}\left[\left\| \bar{G}^t \right\|^2\right],
\end{aligned} \tag{32}$$

where the first equation holds by [46, Lemma 4] and the last equation follows the fact that $2\langle a, b \rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$.

In order to achieve the upper bound on the right-hand side of the inequality (32), the main barrier is to bound P_1 due to heterogeneity of local updates. To bound P_1, we have

$$\begin{aligned}
P_1 &= \left\| \nabla \sum_{n=1}^{N} \mu_n F_n(\boldsymbol{\theta}^{(t,0)}) - \sum_{n=1}^{N} \mu_n G_n^t \right\|^2 = \left\| \sum_{n=1}^{N} \mu_n \left[\nabla F_n(\boldsymbol{\theta}^{(t,0)}) - G_n^t \right] \right\|^2 \\
&\leq \sum_{n=1}^{N} \mu_n \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) - G_n^t \right\|^2 = \sum_{n=1}^{N} \mu_n \left\| \frac{1}{e_n^t} \sum_{e=0}^{e_n^t-1} \left[\nabla F_n(\boldsymbol{\theta}^{(t,0)}) - \nabla F_n(\boldsymbol{\theta}_n^{(t,e)}) \right] \right\|^2 \\
&\leq \sum_{n=1}^{N} \mu_n \left[\frac{1}{e_n^t} \sum_{e=0}^{e_n^t-1} \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) - \nabla F_n(\boldsymbol{\theta}_n^{(t,e)}) \right\|^2 \right] \leq \sum_{n=1}^{N} \mu_n \left[\frac{L^2}{e_n^t} \sum_{e=0}^{e_n^t-1} \underbrace{\left\| \boldsymbol{\theta}^{(t,0)} - \boldsymbol{\theta}_n^{(t,e)} \right\|^2}_{P_2} \right],
\end{aligned} \tag{33}$$

where the first and the second inequality follow from Jensen's Inequality: $\left\| \sum_{n=1}^{N} a_n z_n \right\|^2 \leq \sum_{n=1}^{N} a_n \left\| z_n \right\|^2$, and the last inequality uses Assumption 1.

To further give a bound of P_2, according to (4), we have

$$\begin{aligned}
P_2 &= s^2 \left\| \sum_{k=0}^{e-1} \nabla F_n(\boldsymbol{\theta}_n^{(t,k)}) \right\|^2 \leq s^2 \sum_{k=0}^{e-1} \left\| \nabla F_n(\boldsymbol{\theta}_n^{(t,k)}) \right\|^2 \leq s^2 \sum_{k=0}^{e_n^t-1} \left\| \nabla F_n(\boldsymbol{\theta}_n^{(t,k)}) \right\|^2 \\
&= s^2 \sum_{k=0}^{e_n^t-1} \left\| \nabla F_n(\boldsymbol{\theta}_n^{(t,k)}) - \nabla F_n(\boldsymbol{\theta}^{(t,0)}) + \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 \\
&\leq 2s^2 \sum_{k=0}^{e_n^t-1} \left\| \nabla F_n(\boldsymbol{\theta}_n^{(t,k)}) - \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 + 2s^2 \sum_{k=0}^{e_n^t-1} \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 \\
&\leq 2s^2 L^2 \sum_{k=0}^{e_n^t-1} \left\| \boldsymbol{\theta}_n^{(t,k)} - \boldsymbol{\theta}^{(t,0)} \right\|^2 + 2s^2 e_n^t \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2, \qquad (34)
\end{aligned}$$

where the first inequality follows from Jensen's Inequality, the second inequality uses the fact that $e_n^t \geq e$, the fact $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ accounts for the third inequality, and the last one uses Assumption 1. Substituting the conclusion into (33), we can obtain

$$\begin{aligned}
\frac{L^2}{e_n^t} \sum_{e=0}^{e_n^t-1} P_2 \leq & \frac{L^2}{e_n^t} \sum_{e=0}^{e_n^t-1} \left(2s^2 L^2 \sum_{k=0}^{e_n^t-1} \left\| \boldsymbol{\theta}_n^{(t,k)} - \boldsymbol{\theta}^{(t,0)} \right\|^2 + 2s^2 e_n^t \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 \right) \\
= & \frac{L^2}{e_n^t} \left(2s^2 L^2 e_n^t \sum_{e=0}^{e_n^t-1} P_2 \right) + 2s^2 e_n^t L^2 \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2. \qquad (35)
\end{aligned}$$

Rearrange and we have

$$\frac{L^2}{e_n^t} \sum_{e=0}^{e_n^t-1} P_2 \leq \frac{2s^2 e_n^t L^2}{1 - 2s^2 e_n^t L^2} \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 \leq \frac{D}{1-D} \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2. \qquad (36)$$

where $D = 2s^2 L^2 \max_n(e_n^t) < 1$. Substituting the conclusion into (33), we obtain the upper bound of P_1.

$$P_1 \leq \sum_{n=1}^{N} \mu_n \left(\frac{L^2}{e_n^t} \sum_{e=0}^{e_n^t-1} P_2 \right) \leq \frac{D}{1-D} \sum_{n=1}^{N} \mu_n \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) \right\|^2 \leq \frac{D}{1-D} (\beta_2 + \beta_1 \|\nabla F(\boldsymbol{\theta}^{(t,0)})\|^2), \qquad (37)$$

where the second inequality uses Assumption 3.

Next we show an upper bound of term $\mathbb{E}\left[\left\|\bar{G}^t\right\|^2\right]$. The proof here directly follows the conclusion in [42, Lemma 5], we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{G}^t\right\|^2\right] &\leq 3 \sum_{n=1}^{N} \mu_n \left\| \nabla F_n(\boldsymbol{\theta}^{(t,0)}) - G_n^t \right\|^2 + 3 \left\| \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\|^2 + \frac{3}{K} (\beta_2 + \beta_1 \|\nabla F(\boldsymbol{\theta}^{(t,0)})\|^2) \\
&\leq \frac{3D}{1-D} (\beta_2 + \beta_1 \|\nabla F(\boldsymbol{\theta}^{(t,0)})\|^2) + 3 \left\| \nabla F(\boldsymbol{\theta}^{(t,0)}) \right\|^2 + \frac{3}{K} (\beta_2 + \beta_1 \|\nabla F(\boldsymbol{\theta}^{(t,0)})\|^2) \\
&= 3 \left(\frac{D\beta_1}{1-D} + \frac{\beta_1}{K} + 1 \right) \|\nabla F(\boldsymbol{\theta}^{(t,0)})\|^2 + 3\beta_2 \left(\frac{D}{1-D} + \frac{1}{K} \right), \qquad (38)
\end{aligned}$$

where the second inequality holds due to (33) and (37). By substituting (37) and (38) into (32), we have

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)}) - F(\boldsymbol{\theta}^{(t,0)})\right] \leq -\frac{\tau_{\text{eff}}^t s\alpha^t}{2}\left\|\nabla F(\boldsymbol{\theta}^{(t,0)})\right\|^2 + \delta_t, \tag{39}$$

where $\alpha_t = 1 - \frac{D\beta_1}{1-D} - 3\tau_{\text{eff}}^t sL(\frac{D\beta_1}{1-D} + \frac{\beta_1}{K} + 1)$ and $\delta_t = \frac{\tau_{\text{eff}}^t s\beta_2}{2}[\frac{(3\tau_{\text{eff}}^t sL+1)D}{1-D} + \frac{3\tau_{\text{eff}}^t sL}{K}]$. Applying Lemma 1 into (39), we have

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)})\right] - F(\boldsymbol{\theta}^*) \leq \rho_t\left\{\mathbb{E}\left[F(\boldsymbol{\theta}^{(t,0)})\right] - F(\boldsymbol{\theta}^*)\right\} + \delta_t, \tag{40}$$

where $\rho_t = 1 - \tau_{\text{eff}}^t s\alpha^t$.

In order to simplify the coefficient α_t, assuming that $D = 2s^2L^2\max_n(e_n^t) \leq \frac{1}{2\beta_1+1} < 1$, then we have $\frac{D\beta_1}{1-D} \leq \frac{1}{2}$, i.e., $\alpha_t \geq \frac{1}{2} - 3\tau_{\text{eff}}^t sL(\frac{3}{2} + \frac{\beta_1}{K})$.

Furthermore, we assume that $3\tau_{\text{eff}}^t sL(\frac{3}{2} + \frac{\beta_1}{K}) \leq \frac{1}{4}$, which means $\alpha_t \geq \frac{1}{4}$, thus rewriting the above inequality (39) as

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)}) - F(\boldsymbol{\theta}^{(t,0)})\right] \leq -\frac{\tau_{\text{eff}}^t s}{8}\left\|\nabla F(\boldsymbol{\theta}^{(t,0)})\right\|^2 + \delta_t. \tag{41}$$

Applying Lemma 1 into (41), we have

$$\mathbb{E}\left[F(\boldsymbol{\theta}^{(t+1,0)})\right] - F(\boldsymbol{\theta}^*) \leq \rho_t\left\{\mathbb{E}\left[F(\boldsymbol{\theta}^{(t,0)})\right] - F(\boldsymbol{\theta}^*)\right\} + \delta_t, \tag{42}$$

where $\rho_t = 1 - \frac{\tau_{\text{eff}}^t s\mu}{4} \in (0,1)$.

With regard to the constraint on the learning rate s, we have

$$0 < s < \min_n\left\{1, \frac{4}{\tau_{\text{eff}}^t\mu}, \frac{1}{L\sqrt{2(2\beta_1+1)\max_n(e_n^t)}}, \frac{K}{(18K+12\beta_1)\tau_{\text{eff}}^t L}\right\}$$

due to the assumptions we have made.

B. Proof of Theorem 2

It is clear that $g'(x)$ is a decreasing function of $x > 0$. By $\lim_{x\to+\infty} g'(x) = 0$, we have $g'(x) > 0$ for all $x > 0$, determining that $g(x)$ is an increasing function with regard to x when $x > 0$. In this case, from (10), we can draw a conclusion that $r_u^n(B_u^n)$ is a strictly increasing function of B_u^n.

As a result, the inverse function of $g(x)$, symboled as g^{-1}, which satisfies $g^{-1}(g(x)) = x$, is a strict monotone increment function as well, which means we can solve for a unique x in terms of a given value $g(x)$. Under this circumstance, the optimal bandwidth allocation B_u^n can be derived if the value of $T_u^n(B_u^n)$ is determined, i.e.,

$$B_u^n = (r_u^n)^{-1}\left(\frac{Id}{T_u^n(B_u^n)}\right), \tag{43}$$

where $T_u^n(B_u^n)$ monotonically decreases with B_u^n.

Then we reformulate problem (25) by taking $T^n = (\hat{T}_l^n)^* + T_u^n(B_u^n) + T_D^n$ into consideration by

$$\underset{\boldsymbol{B_u}}{\text{minimize}}\ T_s^*(\boldsymbol{B_u}) = \max_{n\in S}\ (T^n) \tag{44}$$

$$\text{subject to } T^n \leq T_s^*(\boldsymbol{B_u}), \forall n \in S, \tag{44a}$$

$$\sum_{n=1}^{N} B_u^n \leq B, B_u^n \geq 0, \forall n \in S. \tag{44b}$$

Using Karush-Kuhn-Tucker (KKT) conditions to solve problem (44) is feasible and the augment Lagrange function of the problem is

$$\mathcal{L}_1 = T_s + \mu\left(\sum_{n=1}^{K} B_u^n - B\right) + \sum_{n=1}^{K} \lambda_n \left(T^n - T_s^*\right). \tag{45}$$

Subsequently, the KKT conditions can be written as

$$\begin{cases} \frac{\partial \mathcal{L}_1}{\partial B_n^u} = \mu - \lambda_n IdA_n = 0, \forall n \in S, \\ \frac{\partial \mathcal{L}_1}{\partial T_s} = 1 - \sum\limits_{n=1}^{k} \lambda_n = 0, \\ \lambda_n \left(T^n - T_s^*\right) = 0, \forall n \in S, \\ \sum\limits_{n=1}^{K} B_n^u - B = 0, \forall n \in S, \end{cases} \tag{46}$$

where

$$A_n = \frac{\log_2(1 + \frac{h_n p^{\max}}{B_u^n N_0}) - \frac{1}{\ln 2(B_u^n + \frac{h_n p^{\max}}{N_0})}}{[B_u^n \log_2(1 + \frac{h_n p^{\max}}{B_u^n N_0}]^2}. \tag{47}$$

To analyze the KKT conditions in (46), first observe from the second condition that $\exists n \in S, \lambda_n \neq 0$. In order to prove that $\lambda_n \neq 0, \forall n \in S$, we need to start with the first condition by proving $A_n \neq 0, \forall n \in S$.

As a matter of fact, we have proved a proposition that $g'(x) = \log_2(1+\frac{1}{x}) - \frac{1}{\ln 2(x+1)} > 0$, so it is obvious that $A_n > 0$. We can conclude that $\mu \neq 0$, thus we have $\lambda_n \neq 0, \forall n \in S$. Substituting this conclusion into the third condition of (46), it can be derived that $T^n - T_s^* = 0, \forall n \in S$. Applying (43) and (11) into the last equation, then we obtain (28). In the end, comparing the last condition with (28), the optimal bandwidth allocation can be attained as (26) and (27) by bisection search because T_s increases as system bandwidth B decreases.

To obtain (27) which is an explicit version of (26) and mathematically express the explicit inverse function of r_u^n, we use the fact that the inverse function of $r(x) = x\ln(1+\frac{1}{x})$ is

$$x = \frac{-r}{W(-re^{-r}) + r}, \tag{48}$$

where $W(\cdot)$ is Lambert-W function which satisfies $z = xe^x$, and $W(z) = x$ according to [52].

We can easily prove this by multiplying $-(1+\frac{1}{x})$ on both sides of r,

$$-(1+\frac{1}{x})r = -x(1+\frac{1}{x})\ln(1+\frac{1}{x}), \tag{49}$$

then we have

$$\begin{aligned}-(1+\frac{1}{x})re^{-(1+\frac{1}{x})r} &= -x(1+\frac{1}{x})\ln(1+\frac{1}{x})e^{\ln(1+\frac{1}{x})^{-(x+1)}} = -x(1+\frac{1}{x})\ln(1+\frac{1}{x})(1+\frac{1}{x})^{-(x+1)} \\ &= -x(1+\frac{1}{x})^{-x}\ln(1+\frac{1}{x}) = -x\ln(1+\frac{1}{x})e^{-x\ln(1+\frac{1}{x})} = -re^{-r},\end{aligned} \tag{50}$$

which implies that $W(-re^{-r}) = -(1+\frac{1}{x})r$. We can arrive at (48) by rearranging (50).

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, 2019.

[4] 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, 2020.

[5] Y. Shi, K. Yang, Z. Yang, and Y. Zhou, "Mobile edge artificial intelligence: Opportunities and challenges," 2021.

[6] K. B. Letaief, 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, 2022.

[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. 9, no. 1, pp. 1–24, 2022.

[8] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[9] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, 2020.

[10] S. Chen, D. Xue, G. Chuai, Q. Yang, and Q. Liu, "Fl-qsar: a federated learning-based qsar prototype for collaborative drug discovery," *Bioinformatics*, vol. 36, no. 22-23, pp. 5492–5498, 2020.

[11] 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, 2020.

[12] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, 2019.

[13] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.

[14] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. 37th Int. Conf. Mach. Learn.*, 2021, pp. 6357–6368.

[15] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[17] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "Fedpd: A federated learning framework with adaptivity to non-iid data," *IEEE Trans. Signal Process.*, vol. 69, pp. 6055–6070, 2021.

[18] N. Zhang and M. Tao, "Gradient statistics aware power control for over-the-air federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5115–5128, 2021.

[19] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, 2021.

[20] 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, 2020.

[21] Z. Zhao, C. Feng, W. Hong, J. Jiang, C. Jia, T. Q. S. Quek, and M. Peng, "Federated learning with non-iid data in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1927–1942, 2022.

[22] Z. Wang, Y. Shi, Y. Zhou, H. Zhou, and N. Zhang, "Wireless-powered over-the-air computation in intelligent reflecting surface-aided IoT networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1585–1598, 2020.

[23] K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu, and W. Chen, "Federated machine learning for intelligent IoT via reconfigurable intelligent surface," *IEEE Netw.*, vol. 34, no. 5, pp. 16–22, 2020.

[24] H. Liu, X. Yuan, and Y.-J. A. Zhang, "Reconfigurable intelligent surface enabled federated learning: A unified communication-learning design approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7595–7609, 2021.

[25] Z. Wang, J. Qiu, Y. Zhou, Y. Shi, L. Fu, W. Chen, and K. B. Letaief, "Federated learning via intelligent reflecting surface," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 808–822, 2022.

[26] S. Xia, J. Zhu, Y. Yang, Y. Zhou, Y. Shi, and W. Chen, "Fast convergence algorithm for analog federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6.

[27] M. M. Amiria, D. Gündüzb, S. R. Kulkarni, and H. Vincent 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, 2021.

[28] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–7.

[29] 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, 2020.

[30] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated learning: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 14–41, 2022.

[31] 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, 2020.

[32] M. Chen, H. Vincent 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, 2021.

[33] 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, 2021.

[34] W. Shi, S. 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, 2021.

[35] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3606–3621, 2021.

[36] 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, 2021.

[37] D. Wen, M. Bennis, and K. Huang, "Joint parameter-and-bandwidth allocation for improving the efficiency of partitioned edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8272–8286, 2020.

[38] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[39] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[40] M. M. Amiri and D. Gündüz, "Computation scheduling for distributed machine learning with straggling workers," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6270–6284, 2019.

[41] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640–3653, 2021.

[42] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.

[43] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning," in *Proc. Int. Conf. Artificial Intell. Stat. (AISTATS)*, 2021, pp. 3403–3411.

[44] L. Li, L. Yang, X. Guo, Y. Shi, H. Wang, W. Chen, and K. B. Letaief, "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, 2021.

[45] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Rev*, vol. 60, no. 2, pp. 223–311, 2018.

[46] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. Int. Conf. Learn. Representations*, 2020.

[47] T. Sery and K. Cohen, "On analog gradient descent learning over multiple access fading channels," *IEEE Trans. Signal Process.*, vol. 68, pp. 2897–2911, 2020.

[48] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[49] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[50] M. P. Friedlander and M. Schmidt, "Hybrid Deterministic-Stochastic Methods for Data Fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, Jan. 2012.

[51] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv preprint arXiv:1910.14425*, 2019.

[52] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, 1996.

[53] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[54] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[55] Y. S. Can and C. Ersoy, "Privacy-preserving federated deep learning for wearable IoT-based biomedical monitoring," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–17, 2021.