Exploring Representativity in Device Scheduling for Wireless Federated Learning

Zhixiong Chen, *Student Member, IEEE*, Wenqiang Yi, *Member, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

Abstract

Existing device scheduling works in wireless federated learning (FL) mainly focused on selecting the devices with maximum gradient norm or loss function and require all devices to perform local training in each round. This may produce extra training costs and schedule devices with similar data statistics, thus degrading learning performance. To mitigate these problems, we first theoretically characterize the convergence behaviour of the considered FL system, finding that the learning performance is degraded by the difference between the aggregated gradient of scheduled devices and the full participation gradient. Inspired by this, we propose to find a subset of representative devices and the corresponding pre-device stepsizes to approximate the full participation aggregated gradient. Considering the limited wireless bandwidth, we formulate a problem to capture the trade-off between representativity and latency by optimizing device scheduling and bandwidth allocation policies. Our analysis reveals optimal bandwidth allocation is achieved when all scheduled devices have the same latency. Then, by proving the non-monotone submodularity of the problem, we develop a double greedy algorithm to solve the device scheduling policy. To avoid the local training of unscheduled devices, we utilize the historical gradient information of devices to estimate the current gradient for device scheduling design. Compared to existing scheduling algorithms, the proposed representativity-aware device scheduling algorithm improves 6.7% and 4.02% accuracies on two typical datasets under heterogeneous local data distributions, i.e., MNIST and CIFAR-10, respectively. In addition, the proposed latency- and representativity-aware scheduling algorithm saves over 16% and 12% training time for MNIST and CIFAR-10 datasets than the scheduling algorithms based on either latency and representativity individually.

Index Terms

Device scheduling, wireless federated Learning, resource allocation, submodular optimization

Zhixiong Chen, Wenqiang Yi, and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. (emails: {zhixiong.chen, w.yi, a.nallanathan}@qmul.ac.uk).

This article was presented in part by the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (IEEE ICECCME 2022) [1].

I. Introduction

With the explosive growth of data at wireless network edges, machine learning (ML) approaches have attracted significant attention to serve diverse new applications in sixth-generation (6G) wireless networks, such as autonomous driving, intelligent industry, and metaverse [2]. The conventional centralized ML requires transmitting massive raw user data to the edge server, which is inapplicable to support those emerging 6G applications due to high latency and privacy concerns [3]. In addition, it is unaffordable to support centralized ML approaches in 6G with limited spectrum resources. In this context, wireless federated learning (FL) becomes a promising solution since it enables 6G devices to learn a global shared model while preserving data locally [4]. In wireless FL, a parameter server orchestrates multiple devices via wireless channels to engage in the training process that repeatedly performs the alternative optimization process of device-local training and server-model aggregation [5]. Instead of transmitting raw user data, wireless FL only shares the local model parameters of users. This unique property reduces the wireless communication load and simultaneously protects the users' privacy.

A. Related Works

In wireless FL, the main challenge is that the limited communication resource and stringent training latency only allow a small proportion of devices to upload their local models in each round for aggregation. Due to the few participant devices, the learning performance of wireless FL may be drastically degraded [6]. From the perspective of wireless communication systems, existing works focused on different aspects to alleviate this problem, such as *over-the-air* model aggregation [7]–[9], *asynchronous communication* [10]–[12], and *device scheduling* [13]. Over-the-air aggregation leverages the superposition property of wireless waveforms to compute the desired function of the distributed local gradients, so it efficiently mitigates the pressure of limited bandwidth and reduces the model aggregation latency [7]. The device selection and beamforming design in [8] is able to accelerate the convergence speed for over-the-air FL. Considering devices' communication and computation energy limitations, an online device scheduling policy for over-the-air FL has been developed in [9] to maximize the training performance. While demonstrably effective, the over-the-air aggregation has strict requirements on synchronization and channel state information, which slows down the speed of its implementation [14]. Moreover, the parameter server has to wait for the slowest devices in this synchronous communication protocol, leading to significant waiting time due to edge heterogeneity. To relax the straggler

effect in FL, asynchronous communication is adopted to allow the central server conducts global aggregation immediately when it collects a few local models, even if a training round is still in progress [10]. By adaptively aggregating a certain number of local models by their arrival order in each round and adjusting their learning rates, the asynchronous FL proposed in [11] effectively reduces the training completion time. The centralized fusion algorithm in [12] that determines the fusion weight of local models in asynchronous aggregation is able to achieve fast and smooth convergence for FL. However, asynchronous FL approaches suffer from the problem of delayed gradients, resulting in unexpected turbulence of the training trajectory [15]. Besides over-the-air model aggregation and asynchronous communication, device scheduling is able to handle the straggler effect and asynchronous issues by selecting a subset of devices to participate in the per-round training process [13].

Existing device scheduling works in FL mainly focused on channel-condition-aware scheduling [16], [17], parameter-importance-aware scheduling [18]–[20], as well as their joint scheduling [21]–[24]. Specifically, the joint device scheduling and bandwidth allocation scheme in [16] that maximizes the scheduled data samples is efficient in improving learning performance. A probabilistic device scheduling policy has been developed in [17] to minimize communication time. Although channel-condition-aware scheduling algorithms increase the number of participating devices in the learning process, they may degrade the learning performance due to the significant variance introduced in the device selection procedure. In the parameter-importance-aware scheduling schemes, the selected probability of each device is determined proportionally to its importance measured by the norm of gradients [18], loss function values [19], or test accuracy [20]. Allocating larger scheduling probabilities to devices with higher gradient norms has been proposed in [18], which is capable of accelerating the convergence for FL. It has been revealed in [19] that scheduling the devices with higher local loss achieves faster convergence. Maximizing the scheduling probability of clients with higher test accuracy in [20] has proven effective in stateful FL. However, it is ineffective in stateless FL and requires pre-known testing accuracies of devices. Although the above parameter-importance-aware scheduling schemes improve the learning performance, they need all devices to perform local training in each round. To accelerate the learning convergence of FL in practical wireless networks, several device scheduling works considered both channel conditions and parameter importance. The scheduling policy based on both devices' channel conditions and gradient norms in [21] provides a better learning performance than scheduling policies based on a single metric. The resource allocation and

device selection problem is investigated in [22] to capture the trade-offs among the convergence speed of FL, wall clock time, and energy consumption of devices in each round. A probabilistic user selection scheme is proposed in [23] to minimize the training time, which allocates high probabilities to the devices whose local models significantly affect the global FL model. The diversity of local datasets is adopted in [24] to characterize the device scheduling priorities and minimize completion time through resource optimization to speed up the learning process.

B. Motivations and Contributions

The motivations for this work mainly come from: Firstly, traditional device scheduling methods do not utilize computing resources efficiently. Although the approaches in [21]–[24] consider both device importance and communication conditions for device scheduling policy design, they require all devices to perform local training in each round and upload corresponding indicators, e.g., the gradient norm. This may produce extra training costs. To avoid the extra local training of unscheduled devices, this work utilizes devices' *past gradient information* to estimate the current one for guiding the device scheduling design. Secondly, the previous device-importance-aware scheduling policies may select a subset of devices that drift from the global data distribution. The scheduling policies that measure device importance based on gradient norm [21], [22], inner product [23], or the diversity of local dataset [24] trend to schedule devices with similar gradient information in each round. This may exacerbate the global model bias toward the scheduled devices and further degrade the learning performance in heterogeneous data distribution scenarios. Particularly, our convergence analysis of the considered FL in this work reveals that device scheduling policy affects the convergence through the difference between the aggregated gradient of scheduled devices and the full participation gradient. This work uses that gradient difference to characterize the representativity of the scheduled device set. Inspired by the success of accelerating centralized ML algorithms by selecting a weighted subset of training data points to approximate the full gradient of the whole dataset [25]. This work attempts to select a subset of devices to approximate the full devices' aggregated gradient for accelerating the FL. Finally, the previous works only consider heterogeneous channel conditions when minimizing the latency of FL. In this work, we propose a novel latency- and representativity-aware device scheduling algorithm to accelerate the learning process of FL in bandwidth-limited wireless networks, in which the heterogeneous communication, computation, and representativity among devices are all taken into account. The main contributions of this paper are summarized as follows:

- To enable effective FL in bandwidth-limited wireless networks, we theoretically characterize the convergence bound of the considered FL system under the general non-convex loss function setting, finding a new metric, i.e., the difference between the aggregated gradient of scheduled devices and the full participation gradient, which negatively affect the convergence. By minimizing this metric, the convergence speed of FL can be improved.
- To minimize the difference between the scheduled devices' gradient and the full participation gradient, we aim to find a subset of devices and the corresponding pre-device step sizes to approximate the full participation aggregated gradient. To this end, we characterize the representativity of a device set as the approximation error of its aggregated gradient for the full participation gradient. The small approximation error contributes to strong representative. In addition, we utilize the past gradient information of devices to determine the scheduling policy in each round, avoiding the unscheduled devices to perform local training.
- To balance the representativity and latency for the device scheduling policy, we formulate a problem to minimize the weighted sum of gradient approximation error and latency through jointly optimizing the device scheduling and bandwidth allocation policy, which is intractable to solve. Our analysis reveals that the optimal bandwidth allocation policy is optimal when all scheduled devices have the same latency. Furthermore, by proving the submodularity of the problem, we develop a double-greedy algorithm to obtain a sub-optimal device scheduling policy.
- Experiments show that the proposed scheduling algorithm achieves faster convergence speed and higher model accuracy than the benchmarks. Specifically, compared to other benchmark algorithms, the proposed device representativity-aware schedule algorithm is able to boost 6.7% and 4.02% accuracies on MNIST and CIFAR-10 datasets[1], respectively. The proposed latency- and representativity-aware scheduling algorithm saves over 16% and 12% training time for MNIST and CIFAR-10 datasets than the scheduling algorithms based on either latency and representativity individually.

C. Organization and Notations

The rest of this paper is organized as follows: Section II introduces the FL system and the training latency model. The convergence analysis of the considered FL algorithm and the problem

[1]MNIST: http://yann.lecun.com/exdb/mnist/, CIFAR-10: https://www.cs.toronto.edu/ kriz/cifar.html

TABLE I
NOTATION SUMMARY

Notation	Definition		
$\mathcal{K}$; K;	Set of devices; size of $\mathcal{K}$	$\mathcal{D}_k$; D_k	Local dataset of device k; size of $\mathcal{D}_k$
$\mathcal{D}$; D	Overall dataset in the system; size of $\mathcal{D}$	η; τ	Learning rate; local iteration number
$\boldsymbol{S}_t$; $\alpha_{k,t}$	Scheduling policy in round t, i.e., the set of scheduled devices; scheduling indicator of device k in round t	$\widetilde{\boldsymbol{g}}_{k,t}$; $\widetilde{\boldsymbol{g}}_t$	local gradient of device k in round t; aggregated gradient of devices in $\boldsymbol{S}_t$
$\boldsymbol{g}_t$; L_b	aggregated gradient of all devices in $\mathcal{K}$; local batch size	f_k; p_k	CPU frequency of device k; transmit power of device k
Q; q	Number of elements of each local gradient; quantized bits of each gradient element	B; $\boldsymbol{\theta}_t$	Wireless transmission bandwidth; the proportion of B allocated to devices in round t
$T^{\mathrm{L}}_{k,t}$	Computation time for device k in round t	$T^{\mathrm{C}}_{k,t}$	Communication time for device k in round t

formulation are illustrated In Section II. In Section IV, we develop three device scheduling algorithms for FL. Section V verifies the effectiveness of the proposed device scheduling algorithms by simulation. The conclusion is drawn in Section VI. For convenience, we use "$\triangleq$" to denote "is defined to be equal to", $|\cdot|$ denote the size operation of a set, $\nabla(\cdot)$ denote gradient operator, $\langle\cdot,\cdot\rangle$ denote inner product operator, and "$\|\cdot\|$" denote the ℓ_2 norm throughout this paper. The main notations used in this paper are summarized in Table I.

II. SYSTEM MODEL

We consider a typical wireless federated learning (FL) system, in which one edge server undertakes the role of the parameter server to coordinate K devices for training a machine learning model. The server and all devices communicate via bandwidth-limited wireless channels. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. Each device k ($k \in \mathcal{K}$) has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples. Without loss of generality, we assume there is no overlapping for datasets from different devices, i.e., $\mathcal{D}_k \cap \mathcal{D}_h = \emptyset, (\forall k, h \in \mathcal{K})$. Thus, the entire dataset is denoted by $\mathcal{D} = \cup \{\mathcal{D}_k\}_{k=1}^{K}$ with the total number of samples $D = \sum_{k=1}^{K} D_k$.

Let $\zeta = (\boldsymbol{x}, y)$ denote a data sample in $\mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input feature vector of the sample, and $y \in \mathbb{R}$ is the corresponding ground-truth label. For a machine learning model $\boldsymbol{w}$, we use $f(\boldsymbol{w}; \zeta)$ to denote its sample-wise loss function on the data sample ζ, which quantifies the error between the ground-truth label y and its predicted output of $\boldsymbol{x}$. Thus, the local loss function of device k that measures the model error on its local dataset $\mathcal{D}_k$ can be defined as

$$F_k(\boldsymbol{w}) \triangleq \frac{1}{D_k} \sum_{\zeta \in \mathcal{D}_k} f(\boldsymbol{w}; \zeta). \tag{1}$$

Accordingly, the global loss function associated with all distributed local datasets is given by

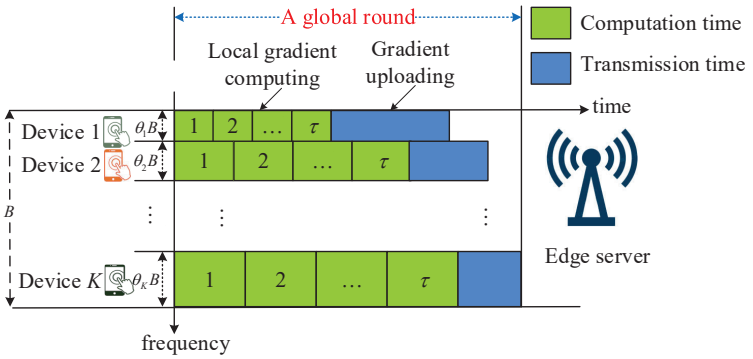


Fig. 1. An implementation of FL via FDMA, where the scheduled devices perform τ local iterations and upload gradients to the edge server.

$$F(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} p_k F_k(\boldsymbol{w}), \tag{2}$$

where p_k is the weight of device k such that $\sum_{k=1}^{K} p_k = 1$. Similar to many existing works, e.g., [17], [26], [27], we set $p_k = \frac{1}{K}$.

A. Federated Learning Algorithm

The goal of FL is to train a model $\boldsymbol{w}$ by leveraging the devices' local datasets. To preserve the data privacy of devices, the devices collaboratively learn $\boldsymbol{w}$ by only uploading local gradients to the edge server for periodical aggregation, instead of transmitting the raw training data. The edge server orchestrates the training process, by repeating the following steps until the model converge:

1) The edge server selects a subset of devices from $\mathcal{K}$ to participate the training in the current communication round, denoted by $\boldsymbol{S}_t$. Let $\alpha_{k,t}$ denote the schedule indicator of device k in round t, we have $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$.
2) The edge server broadcasts the latest global model to the scheduled devices for local training. It is worth mentioning that only the scheduled devices perform local training and upload their local gradients to the edge server for the global model update. Thus, in the FL process, the edge server only broadcasts the latest global model to scheduled devices instead of all devices. After the FL process, the edge server broadcasts the trained global model to all devices for serving them.
3) After receiving the global model, each selected device computes the local gradient $\boldsymbol{g}_{k,t}$ by running τ steps stochastic gradient descent (SGD) on its local dataset, according to

$$\widetilde{\boldsymbol{g}}_{k,t} = \sum_{l=0}^{\tau-1} \nabla F_k(\boldsymbol{w}_{k,t,l}; \mathcal{B}_{k,t,l}), \tag{3}$$

where

$$\nabla F_k(\boldsymbol{w}_{k,t,l}; \mathcal{B}_{k,t,l}) = \frac{1}{L_b} \sum\nolimits_{\zeta \in \mathcal{B}_{k,t,l}} \nabla f(\boldsymbol{w}_{k,t,l}; \zeta) \tag{4}$$

is the gradient in iteration l ($0 \leq l \leq \tau - 1$), $\mathcal{B}_{k,t,l}$ is a local mini-batch data uniformly sampled from $\mathcal{D}_k$ with $L_b = |\mathcal{B}_{k,t,l}|$ data samples.

4) After all selected devices accomplish local gradients computing, they upload their gradients to the edge server for aggregation as follows:

$$\widetilde{\boldsymbol{g}}_t = \frac{1}{|\boldsymbol{S}_t|} \sum_{k \in \boldsymbol{S}_t} \widetilde{\boldsymbol{g}}_{k,t}. \tag{5}$$

Then, the edge server updates the global model as $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \widetilde{\boldsymbol{g}}_t$, where η denotes the learning rate. For ease of comparison in the following discussion, we use $\boldsymbol{g}_t = \frac{1}{K} \sum_{k \in \mathcal{K}} \widetilde{\boldsymbol{g}}_{k,t}$ to denote the aggregated gradient for all devices, namely the full-participation stochastic gradient (FP-SG).

B. Latency Model

In the following, we analyze the one-round latency for the FL process.

1) Computation latency: Denote C_k as the number of float-point operations (FLOPs) required to process one data sample at device k. Let f_k denote the computational capability (float-point operations per second) of device k. Thus, the local gradient calculation latency of device k can be expressed as

$$T_{k,t}^{\mathrm{L}} = \frac{\tau L_b C_k}{f_k}, \forall t. \tag{6}$$

2) Communication latency: Let Q denote the number of elements in each local gradient. Each element is quantized by q bits. In this work, we consider that the frequency division multiple access (FDMA) technique is deployed in the system with total B Hz wireless bandwidth for devices to upload their local gradients. Let p_k denote the transmit power of device k. We assume that the channel gain, including both small-scale fading and path loss, between device k and the edge server, i.e., $h_{k,t}$, remains unchanged within one round but varies independently and identically over rounds. Let $\theta_{k,t} \in [0, 1]$ denote the fraction of the overall bandwidth allocated to device k in round t, and $\boldsymbol{\theta}_t = (\theta_{1,t}, \theta_{2,t}, \cdots, \theta_{K,t})$. The uplink rate of device k can be characterized by $r_{k,t} = \theta_{k,t} B \log(1 + \frac{p_k h_{k,t}}{\sigma^2})$, where σ^2 is the variance of Gaussian additive noise. Thus, the local gradient uploading latency of device k is

$$T_{k,t}^{\mathrm{C}} = \frac{Qq}{r_{k,t}} = \frac{Qq}{\theta_{k,t} B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)}. \tag{7}$$

According to the above models, the completion time of each participating device $k(k \in \mathcal{K})$ includes the local computation time $T_{k,t}^{\mathrm{L}}$ and communication time $T_{k,t}^{\mathrm{C}}$, as shown in Fig. 1. The one round latency determined by the slowest device is given by

$$\mathcal{T}_t(\boldsymbol{S}_t) = \max_{k \in \boldsymbol{S}_t} \left\{ T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} \right\}. \tag{8}$$

Note that the above discussion ignored the global model broadcasting and updating latency, because the broadcasting process occupies the entire bandwidth and the edge server has large transmit power, the broadcasting latency is negligible. Moreover, the edge server is usually computational powerful, and the global model update latency can be ignored compared to the communication and computation latencies.

III. Convergence Analysis and Problem Formulation

This section starts with the convergence analysis of the considered FL system under the general non-convex loss function setting, finding a metric, i.e., device representativity, to guide the device scheduling policy design. Then, we formulate an optimization problem for device scheduling which balance the latency and representative ability in each round.

A. Convergence Analysis

To develop a concrete metric to evaluate the representativity of each local gradient, we first analyze the convergence behavior of the FL system. To this end, we make the following assumptions to the local loss function $F_k(\cdot)$:

Assumption 1. (Lipschitz gradient continuity): Each local loss functions $F_k(\cdot)(k \in \mathcal{K})$ is continously differentiable, and its gradient $\nabla F_k(\boldsymbol{w})$ is L-Lipschitz continuous, that is

$$\|\nabla F_k(\boldsymbol{w}) - \nabla F_k(\boldsymbol{v})\| \leq L \|\boldsymbol{w} - \boldsymbol{v}\|. \tag{9}$$

Assumption 2. (Unbiased stochastic gradient): For the mini-batch data samples $\mathcal{B}_{k,t}$ that uniformly sampled from $\mathcal{D}_k$ on device k ($k \in \mathcal{K}$), the resulting stochastic gradient is unbiased and variance bounded, that is

$$\mathbb{E}\left[\nabla F_k(\boldsymbol{w}_{k,t}; \mathcal{B}_{k,t})\right] = \nabla F_k(\boldsymbol{w}_{k,t}), \tag{10}$$

and

$$\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t}; \mathcal{B}_{k,t}) - \nabla F_k(\boldsymbol{w}_{k,t})\right\|^2 \leq G^2. \tag{11}$$

Assumption 3. (Bounded stochastic gradient): The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\left\|\nabla F_k(\boldsymbol{w}_{k,t};\mathcal{B}_{k,t})\right\|^2 \leq \chi^2$.

Assumption 1, 2, and 3 are widely used in the convergence analysis of FL systems and satisfied by loss functions for widely used learning models, e.g., support vector machines (SVM), Logistic regression, and most neural networks [28]. In particular, according to [29], a deep neural network defined by a composition of functions is a Lipschitz neural network if the functions in all layers are Lipschitz. It has been proved in [30] and [31] that the convolution layer, linear layer, some nonlinear activation functions (e.g., Sigmoid, tanh, Leaky ReLU, and SoftPlus), and the widely used cross-entropy function have Lipschitz smooth gradients. That is, the loss functions of most neural networks that are consisted of Lipschitz layers and loss functions are Lipschitz continuous. Based on this, we provide the one round convergence bound of the considered FL system in Theorem 1, proved in Appendix A.

Theorem 1. *Let Assumption* 1, 2, *and* 3 *hold, and the learning rate satisfy* $\eta \leq \frac{1}{L}$, *we have*

$$\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right] \leq -\frac{L}{2}\eta^2\mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2(\tau-1)^2\chi^2 + L\eta^2(2\tau^2-2\tau+1)G^2 + L\eta^2\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2. \quad (12)$$

According to Theorem 1, the expected gap of the loss function values between two global round is bounded by four terms: 1) the squared norm of the ground-truth global gradient $\|\nabla F(\boldsymbol{w}_t)\|^2$; 2) the expected squared norm of stochastic gradients χ^2, 3) the variance of stochastic gradient G^2, 4) the difference between the aggregated gradient of the scheduled devices and the FP-SG that aggregates all devices' stochastic gradients, i.e., $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$. The first three terms are independent with the device scheduling decision. The last term is an explicit form related to the device scheduling policy. Thus, the learning performance can be improved by minimizing the $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$. Based on Theorem 1, we further characterize the convergence bound of the considered FL system after T rounds training in Corollary 1, proved in Appendix B.

Corollary 1. *Let Assumption* 1, 2, *and* 3 *hold, and the learning rate satisfy* $\eta \leq \frac{1}{L}$, *the* T*-round convergence is upper-bounded by*

$$\mathbb{E}[F(\boldsymbol{w}_T)-F(\boldsymbol{w}^*)] \leq (1-L^2\eta^2)^{T-1}\mathbb{E}[F(\boldsymbol{w}_0)-F(\boldsymbol{w}^*)] + \frac{1-L^2\eta^2-(1-L^2\eta^2)^T}{L}\left((2\tau^2-2\tau+1)G^2+(\tau-1)^2\chi^2\right) + \sum\nolimits_{t=1}^{T-1}(1-L^2\eta^2)^t L\eta^2\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2. \quad (13)$$

Corollary 1 presents the expected gap between the global loss after T rounds and the optimal loss training, which is bounded by the expected gap between the initial global loss and the optimal one, the variance of SGD, the bounded norm of stochastic gradient, and the cumulative difference of gradient between full participation and partial participation. By minimizing the difference in gradient between full and partial participation in each round, the learning performance can be improved.

B. Device Representativity Measurement

The previous works, e.g., [21], [32], prone to select the devices with maximum gradient norm to minimize $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$. To further minimize the $\|-\boldsymbol{g}_t+\widetilde{\boldsymbol{g}}_t\|^2$ and accelerate the learning convergence, we aim to find a subset of devices (i.e., $\boldsymbol{S}_t \subseteq \mathcal{K}$) and the corresponding pre-device stepsizes γ_k ($\forall k \in \boldsymbol{S}_t$) in each global round t, such that the aggregated gradient approximate the FP-SG (i.e., $\boldsymbol{g}_t$) that aggregated by all the K devices. Toward this end, we define a mapping function $\varphi : \mathcal{K} \to \boldsymbol{S}_t$, which maps each device $k \in \mathcal{K}$ to a scheduled device $\varphi(k) \in \boldsymbol{S}_t$ such that the gradient $\nabla F_k(\boldsymbol{w})$ from device k is approximated by the gradient from $\varphi(k)$. For each device $h \in \boldsymbol{S}_t$, let $\mathcal{C}_h = \{k : k \in \mathcal{K}, \varphi(k) = h\}$ denote the set of devices approximated by device h, and $\gamma_h = |\mathcal{C}_h|$. Thus, we have

$$\sum\nolimits_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} = \sum\nolimits_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t} + \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right) = \sum\nolimits_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right) + \sum\nolimits_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}. \quad (14)$$

By rearranging (14) and then taking the norm of both sides, the approximation error of the gradient aggregated from $\boldsymbol{S}_t$ (i.e., $\sum_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}$) on the FP-SG satisfies

$$\left\|\sum\nolimits_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} - \sum\nolimits_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}\right\| = \left\|\sum\nolimits_{k=1}^{K} \left(\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right)\right\| \le \sum\nolimits_{k=1}^{K} \left\|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{\varphi(k),t}\right\|, \quad (15)$$

where the inequality follows from the triangle inequality. The upper-bound in (15) is minimized when φ maps each $k \in \mathcal{K}$ to an device in $\boldsymbol{S}_t$ with minimum Euclidean distance between their gradient. That is, $\varphi(k) = \arg\min_{h\in\boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$. Hence, the approximation error in (15) satisfies

$$\left\|\sum\nolimits_{k=1}^{K} \widetilde{\boldsymbol{g}}_{k,t} - \sum\nolimits_{k\in\boldsymbol{S}_t} \gamma_k \widetilde{\boldsymbol{g}}_{k,t}\right\| \le \sum\nolimits_{k=1}^{K} \min_{h\in\boldsymbol{S}_t} \left\|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\right\|. \quad (16)$$

Thus, the approximation error can be minimized by minimizing the right-hand side of (16). Substituting (16) into (12), the one-round convergence bound can be expressed as:

$$\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \le -\frac{L}{2}\eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2(\tau-1)^2\chi^2$$

$$+ L\eta^2(2\tau^2 - 2\tau + 1)G^2 + L\eta^2 \left(\sum\nolimits_{k=1}^{K} \min_{h \in \boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\| \right)^2. \quad (17)$$

The convergence bound in (17) shows how the device scheduling policy affects the convergence bound. According to (17), the learning performance can be improved by minimizing the upper bound of the approximation error of the gradient aggregated from $\boldsymbol{S}_t$ on the FP-SG, i.e., $\sum_{k=1}^{K} \min_{h \in \boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$. We define $\mathcal{H}(\boldsymbol{S}_t) = \sum_{k=1}^{K} \min_{h \in \boldsymbol{S}_t} \|\widetilde{\boldsymbol{g}}_{k,t} - \widetilde{\boldsymbol{g}}_{h,t}\|$ to quantify the approximate error of a device scheduling decision $\boldsymbol{S}_t \subseteq \mathcal{K}$.

C. Problem Formulation

To accelerate the learning convergence, one should schedule the devices with the lowest latency (implemented by good channel conditions and powerful computing capability) as well as the smallest FP-SG approximation error. However, it rarely happens that a device always has the lowest latency and smallest FP-SG approximation error simultaneously in a practical system. Similar to many existing works, e.g., [22], [33], we aim to capture the trade-offs between device representativity and latency for improving the learning performance of FL. Towards this end, we define two weight factors $\rho_1 \geq 0$ and $\rho_2 \geq 0$ to capture the Pareto-optimal trade-offs among the device representativity and latency, the values of which depend on specific scenarios. A large ρ_1 and small ρ_2 emphasis more on device representativity, while a small ρ_1 and large ρ_2 pay more attention to devices' latency. In addition, similar to many existing works, e.g., [21]–[24], we optimize the FL performance in each round since the available bandwidth and devices are independent among rounds, instead of optimizing the FL performance over all rounds under long-term resource constraints as the existing paper [5], [9], [16]. Thus, we formulate the problem as follows:

$$\mathcal{P}: \quad \min_{\boldsymbol{S}_t, \boldsymbol{\theta}_t} \quad \rho_1 \mathcal{H}(\boldsymbol{S}_t) + \rho_2 \mathcal{T}(\boldsymbol{S}_t) \quad (18)$$

$$\text{s. t.} \quad \alpha_{k,t} \in \{0, 1\}, \quad (18a)$$

$$\sum\nolimits_{k \in \boldsymbol{S}_t} \theta_{k,t} \leq 1, \quad (18b)$$

$$0 \leq \theta_{k,t} \leq 1. \quad (18c)$$

In problem $\mathcal{P}$, (18a) indicates which devices are scheduled in each round. (18b) assures that the wireless bandwidth resource allocated to all devices would not exceed the total available bandwidth resource. (18c) imposes restrictions on the wireless bandwidth resource allocated to each device. Notably, similar to [22], we can adapt to the problem with hard constraints

on latency via setting "virtual devices". According to Lemma 1 in Section IV-A, the optimal bandwidth allocation policy is achieved when all scheduled devices have the same latency. Thus, by setting a virtual device whose latency is the delay constraint into the scheduled device set, the latency of devices can satisfy the delay constraint by adjusting the bandwidth allocation policy. There are two major challenges in solving problem $\mathcal{P}$:

1) **Unknown gradient information of devices**: Problem $\mathcal{P}$ requires devices' gradient information that can only be acquired after local gradient computing and uploading. However, the device scheduling decision should be made before gradient computation.

2) **Non-deterministic polynomial-time hard (NP-Hard)**: Problem $\mathcal{P}$ involves a combinatorial optimization over the multi-dimensional discrete and continuous space, which is challenging to solve. In the following analysis, we show that two special cases of problem $\mathcal{P}$, i.e., latency-aware device scheduling problem and representativity-aware device scheduling problem are both submodular maximization problem, which has been proven to be NP-Hard. Thus, Problem $\mathcal{P}$ is NP-Hard in fact.

IV. Device Scheduling Policies for Federate Learning

In this section, we develop an efficient algorithm to solve the problem $\mathcal{P}$ within polynomial time complexity. To facilitate the algorithm design, we first focus on analyzing two special cases of problem $\mathcal{P}$: 1) $\rho_1 = 0$ and $\rho_2 = 1$ for the latency aware device scheduling problem, 2) $\rho_1 = 1$ and $\rho_2 = 0$ for the device representativity aware scheduling problem. Then, based on the obtained properties of these two special-case problems, we prove that problem $\mathcal{P}$ is a non-monotone submodular minimization problem. Finally, we develop an efficient double greedy algorithm to solve problem $\mathcal{P}$ and obtain the joint latency and device representativity aware device scheduling policy.

A. Optimal Wireless Bandwidth Allocation

In this subsection, we solve the optimal bandwidth allocation policy for any given device scheduling policy $\boldsymbol{S}_t$. Given the scheduled device set $\boldsymbol{S}_t$, the optimal bandwidth allocation problem can be decomposed from $\mathcal{P}$ as follows:

$$\mathcal{P}_1: \quad \min_{\boldsymbol{\theta}_t} \quad \max_{k \in \boldsymbol{S}_t} \left\{ T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} \right\} \tag{19}$$

$$\text{s. t.} \quad (18\text{b}), (18\text{c}).$$

Problem $\mathcal{P}_1$ is a typical convex optimization problem [34], we obtain its optimal solution by using Lemma 1, proved in Appendix C.

Lemma 1. *The optimal wireless bandwidth allocation solution for problem $\mathcal{P}_1$ satisfies the following condition:*

$$\theta_{k,t}=\frac{Qq}{\left(\mathcal{T}_t^*(\boldsymbol{S}_t)-\frac{\tau L_b C_k}{f_k}\right) B\log\left(1+\frac{p_k h_{k,t}}{\sigma^2}\right)}, \forall k \in \boldsymbol{S}_t, \tag{20}$$

where $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is the optimal latency for device scheduling decision $\boldsymbol{S}_t$ in round t, its value is determined by the equation $\sum_{k\in\boldsymbol{S}_t}\theta_{k,t}=1$.

In Lemma 1, there is still an unknown variable $\mathcal{T}_t^*(\boldsymbol{S}_t)$ in the optimal expression of bandwidth allocation policy. Since $\theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t))$ is a monotonically decreasing function with respect to $\mathcal{T}_t(\boldsymbol{S}_t)$, the bisection method can be deployed to obtain the optimal bandwidth allocation policy. To this end, we derive the lower bound and upper bound of $\mathcal{T}(\boldsymbol{S}_t)$ in the following. To derive the lower bound of $\mathcal{T}_t(\boldsymbol{S}_t)$, we have the minimal fraction of bandwidth allocated to devices in $\boldsymbol{S}_t$ should less than $\frac{1}{|\boldsymbol{S}_t|}$, i.e., $\min_{k\in\boldsymbol{S}_t}\theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t))\leq\frac{1}{|\boldsymbol{S}_t|}$. Hence,

$$\frac{\min\limits_{k\in\boldsymbol{S}_t}\frac{Qq}{B\log\left(1+\frac{p_k h_{k,t}}{\sigma^2}\right)}}{\max\left(\mathcal{T}_t(\boldsymbol{S}_t)-\frac{\tau L_b C_k}{f_k}\right)}\leq\frac{1}{|S_t|}. \tag{21}$$

Thus, the lower bound of $\mathcal{T}_t(\boldsymbol{S}_t)$ is

$$\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)=\min_{k\in\boldsymbol{S}_t}\frac{|\boldsymbol{S}_t|\,Qq}{B\log\left(1+\frac{p_k h_{k,t}}{\sigma^2}\right)}+\min_{k\in\boldsymbol{S}_t}\frac{\tau L_b C_k}{f_k}. \tag{22}$$

Then, to derive the upper bound of $\mathcal{T}_t(\boldsymbol{S}_t)$, we use $\max_{k\in\boldsymbol{S}_t}\theta_{k,t}(\mathcal{T}_t(\boldsymbol{S}_t))\geq\frac{1}{|\boldsymbol{S}_t|}$. The derivation of the upper bound is similar to that of lower bound, and thus omitted for brevity. The upper bound of $\mathcal{T}_t(\boldsymbol{S}_t)$ is

$$\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t)=\max_{k\in\boldsymbol{S}_t}\frac{|\boldsymbol{S}_t|\,Qq}{B\log\left(1+\frac{p_k h_{k,t}}{\sigma^2}\right)}+\max_{k\in\boldsymbol{S}_t}\frac{\tau L_b C_k}{f_k}. \tag{23}$$

According to the lower and upper bounds above, the bisection method is deployed to solve the optimal $\mathcal{T}_t^*(\boldsymbol{S}_t)$. For clarity, we summarize the detailed steps for solving the optimal bandwidth allocation policy in Algorithm 1. The bisection process will halve the searching region in every iteration and terminate when the given precision requirement (i.e., ε) is satisfied. Thus, the time complexity of this bisection method is $\mathcal{O}\left(\log_2\frac{\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t)-\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)}{\varepsilon}\right)$. Based on above analysis, we

Algorithm 1 Optimal Wireless Bandwidth Allocation

Initialize $\boldsymbol{S}_t$, the precision requirement $\varepsilon > 0$.
Initialize the lower bound ($\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)$) and upper bound ($\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t)$) of the latency based on (22) and (23), respectively.
repeat
  Set $\mathcal{T} = (\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t) + \mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t))/2$.
  For each device $k \in \boldsymbol{S}_t$, compute the required bandwidth allocation ratio $\theta_{k,t}(\mathcal{T})$ based on (20).
  Compute the summation of required bandwidth allocation ratio $\sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mathcal{T})$.
  if $\sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}) > 1$ **then**
    Halve the searching region by setting $\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t) = \mathcal{T}$ and $\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) = \mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t)$.
  else if $0 < \sum_{k\in\boldsymbol{S}_t} \theta_{k,t}(\mathcal{T}) < 1 - \varepsilon$ **then**
    Halve the searching region by setting $\mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t) = \mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)$ and $\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) = \mathcal{T}$.
  else
    Break the circulation.
  end if
until $|\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)| < \varepsilon$
return The optimal latency $\mathcal{T}_t^*(\boldsymbol{S}_t) = \mathcal{T}$ and the optimal bandwith allocation policy $\boldsymbol{\theta}_t$

have the following remark.

Remark 1. From (20), the proportion of the wireless bandwidth allocated to device k ($k \in \mathcal{K}$), i.e., $\theta_{k,t}$, is monotonically decreasing with its CPU frequency f_k and its channel gain $h_{k,t}$. That is, more bandwidth should be allocated to the devices with low computation capability and weak channel conditions.

B. Latency-aware Device Scheduling Policy

In this subsection, we investigate a special case of problem $\mathcal{P}$, i.e., the latency-aware device scheduling problem. By setting $\rho_1 = 0$ and $\rho_2 = 1$ in problem $\mathcal{P}$, we formulate the latency-aware device scheduling problem as follows:

$$\mathcal{P}_2: \quad \min_{\boldsymbol{S}_t, \boldsymbol{\theta}_t} \quad \mathcal{T}(\boldsymbol{S}_t) \tag{24}$$

$$\text{s. t.} \quad |\boldsymbol{S}_t| = N, \tag{24a}$$

$$(18a), (18b), (18c).$$

Note that, we add a constraint (24a) into $\mathcal{P}_2$ since the objective function is monotone with respect to device set size (as shown in the following Lemma 2). Without constraint (24a), the solution of problem $\mathcal{P}_2$ is trivial simply taking the empty device scheduling set (i.e., $\boldsymbol{S}_t = \emptyset$) as the solution. However, by adding constraint (24a), the device scheduling problem $\mathcal{P}_2$ is non-trivial.

Problem $\mathcal{P}_2$ involving wireless bandwidth allocation and device scheduling is a typical mixed-integer non-linear programming that is generally difficult to solve in polynomial time. Based on the above analysis, the optimal bandwidth allocation policy for any device scheduling set $\boldsymbol{S}_t$

can be obtained by using Algorithm 1, the corresponding optimal latency is denoted as $\mathcal{T}_t^*(\boldsymbol{S}_t)$. Substituting $\mathcal{T}_t^*(\boldsymbol{S}_t)$ into problem $\mathcal{P}_2$, we transform $\mathcal{P}_2$ into the following equivalent problem:

$$\widetilde{\mathcal{P}_2}: \quad \min_{\boldsymbol{S}_t} \quad \mathcal{T}_t^*(\boldsymbol{S}_t) \tag{25}$$
$$\text{s. t.} \quad (18\text{c}), (24\text{a}).$$

For problem $\widetilde{\mathcal{P}_2}$, an intuitive method to obtain the optimal device scheduling policy is to compute the optimal latency for all the possible device scheduling policies and then select the one with minimal latency. However, there are total C_K^N possible device scheduling policies. In the practical systems, the overall number of devices (i.e., K) is large while the participating device number (i.e., N) in each round is small, inducing a large number of possible scheduling device set. Thus, computing the latency for all possible device scheduling policies is impractical due to the high time complexity. In the following, we prove that problem $\widetilde{\mathcal{P}_2}$ is a submodular set cover problem. Based on this, we find a near-optimal solution for problem $\widetilde{\mathcal{P}_2}$ by using greedy algorithm with polynomial time complexity. To this end, we first introduce the definition of submodular function as follows:

Definition 1. (Submodular function) [35]: A function $\phi : 2^{\mathcal{K}} \to \mathbb{R}$ is submodular if for every $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$ and $h \in \mathcal{K} \backslash \boldsymbol{S}_2$, it holds $\Delta(h \,|\boldsymbol{S}_1) \geq \Delta(h \,|\boldsymbol{S}_2)$, where $\Delta(h \,|\boldsymbol{S}_1) = \phi(\boldsymbol{S}_1 \cup \{h\}) - \phi(\boldsymbol{S}_1)$ is the discrete derivative of ϕ at $\boldsymbol{S}_1$ with respect to h, also named as marginal gain.

According to Definition 1, we have the following lemma for the optimal latency function $\mathcal{T}_t^*(\boldsymbol{S}_t)$, proved in Appendix D.

Lemma 2. *The optimal latency function $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is monotonically increasing with respect to the device set $\boldsymbol{S}_t$, i.e., for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$, we have $\mathcal{T}_t^*(\boldsymbol{S}_1) < \mathcal{T}_t^*(\boldsymbol{S}_2)$. Moreover, the negative of $\mathcal{T}_t^*(\boldsymbol{S}_t)$, i.e., $-\mathcal{T}_t^*(\boldsymbol{S}_t)$, is a monotonically decreasing submodular function with respect to the device set $\boldsymbol{S}_t$. That is, for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$ and $h \in \mathcal{K} \backslash \boldsymbol{S}_2$, we have*

$$\mathcal{T}_t^*(\{h\} \cup \boldsymbol{S}_1) - \mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*(\{h\} \cup \boldsymbol{S}_2) - \mathcal{T}_t^*(\boldsymbol{S}_2). \tag{26}$$

According to Lemma 2, problem $\widetilde{\mathcal{P}_2}$ is a cardinality constraint submodular maximization problem, which is general NP-Hard. Below we find a near-optimal solution of problem $\widetilde{\mathcal{P}_2}$ by using greedy algorithm [36], which starts from $\boldsymbol{S}_t = \emptyset$, and adds one client $k \in \mathcal{K} \backslash \boldsymbol{S}_t$ with the greatest marginal gain to $\boldsymbol{S}_t$ in every step, i.e. $k = \arg\min_{k \in \mathcal{K} \backslash \boldsymbol{S}_t} (\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\}) - \mathcal{T}_t^*(\boldsymbol{S}_t))$. For clarity, we summarize the detail steps for latency-aware device scheduling algorithm in

Algorithm 2 Greedy Algorithm for Latency-aware Device Scheduling

1: Initialize $\boldsymbol{S}_t \leftarrow \emptyset$ and $\mathcal{T}_t^*(\boldsymbol{S}_t) = 0$, the number of selected devices N.
2: **while** $|\boldsymbol{S}_t| < N$ **do**
3: **for** $k \in \mathcal{K} \backslash \boldsymbol{S}_t$ **do**
4: Compute the optimal latency for device set $\boldsymbol{S}_t \cup \{k\}$ as $\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\})$ by using Algorithm 1.
5: **end for**
6: $k^* = \arg\min_{k \in \mathcal{K} \backslash \boldsymbol{S}_t} (\mathcal{T}_t^*(\boldsymbol{S}_t \cup \{k\}) - \mathcal{T}_t^*(\boldsymbol{S}_t))$.
7: $\boldsymbol{S}_t \leftarrow \boldsymbol{S}_t \cup \{k^*\}$.
8: **end while**
9: **return** The device scheduling set $\boldsymbol{S}_t$.

Algorithm 2. Note that Algorithm 2 performs optimal bandwidth allocation (i.e., Algorithm 1) at most KN times for select N devices. Thus, the time complexity of Algorithm 2 is $\mathcal{O}(KN \log_2 \frac{\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)}{\varepsilon})$. Based on the performance analysis in [36], the greedy device scheduling algorithm is able to achieve a worst-case approximation factor of $1 - \frac{1}{e}$ for the optimal solution, where e is the Euler's number.

C. Device Representativity-aware Scheduling Policy

In this subsection, we investigate another special case of problem $\mathcal{P}$, i.e., the device representativity-aware scheduling problem which aims to find a subset of devices and the corresponding pre-device stepsizes to approximate the FP-SG. By setting $\rho_1 = 1$ and $\rho_2 = 0$ in problem $\mathcal{P}$, the device representativity aware scheduling problem can be formulated as follows:

$$\mathcal{P}_3: \quad \min_{\boldsymbol{S}_t} \quad \mathcal{H}(\boldsymbol{S}_t) \tag{27}$$

$$\text{s. t.} \quad |\boldsymbol{S}_t| = N. \tag{27a}$$

Similar to the formulation of problem $\mathcal{P}_2$, we also add a scheduled device number constraint in problem $\mathcal{P}_3$ since its objective function is monotone with respect to device set size. Without constraint (27a), problem $\mathcal{P}_3$ is trivial simply taking all devices (i.e., $\boldsymbol{S}_t = \mathcal{K}$) to the solution. However, problem $\mathcal{P}_3$ is still difficult to solve since the edge server requires the gradient information of all devices. The gradient information can only be obtained after local gradient computing and uploading by devices. If the edge server collected all the gradient information for devices, it can directly aggregate all local gradients to minimize the convergence bound in (12), and the device scheduling is meaningless. To tackle this challenge, there are two heuristics in the following to estimate the gradient information at the start of each global round.

1) *Estimating by mini-batch gradient (E-MBG)*: Compute the gradient of devices with a smaller mini-batch data (the batch size is less than L_b), and upload all local gradients to the edge server. This method can only reduce part of the computation cost compared to the method of uploading complete gradient information computed by L_b data samples at each device.

2) *Estimate by past gradient information (E-PG)*: The edge server straightforwardly uses the most recently received gradients from devices to approximate their current gradients for solving the problem $\mathcal{P}_3$ for device scheduling.

In addition to the above two heuristics, there are some neural-network-based methods, e.g., [23], [37], to predict devices' local gradients, which require collecting devices' gradient information to train extra machine learning models. This may produce extra training time and energy consumption for the FL system. However, the two heuristics are convenient to implement. In particular, the E-PG method simply uses the past gradients of devices to approximate the current one and does not require extra computation and communication costs compared to the E-MBG and neural network-based methods. In addition, the experimental results in Section V verify that the use of past gradients can effectively approximate devices' current gradients.

To evaluate the effectiveness of these two methods, we show in Fig. 2 in Section V the difference between the recently received gradients at the edge server and the current one of an arbitrary device, under the considered datasets. It is observed that E-MBG ($L_b \in \{4, 8, 16\}$) performs not well due to the high variance of the stochastic gradients, while E-PG has a more accurate estimation of the current one. Note that, similar to many existing works in [21]–[24], the E-MBG method requires all devices to compute their gradient with mini-bath data samples and upload their gradient to the edge server. This produces extra computing and transmission costs since the estimated gradients of devices are not used for the global model aggregation. In contrast, the E-PG method only requires the edge server to save the past gradients information for devices and does not require extra computation and transmission. Thus, E-PG is computation and transmission-free compared to E-MBG.

In fact, the past gradient information has been successfully used in FL to estimate the current gradient of devices. For example, replacing the lost gradient (induced by transmission error) in decentralized SGD with the past gradients is able to achieve the same asymptotic convergence rate as the decentralized SGD with no transmission error [38]. Using the most recent ℓ_2-norm of the local gradient to estimate the current one at each device to decide the transmit power has proved to be effective in the over-the-air FL system [9]. Motivated by this, we apply the

most recent gradient information of devices uploaded to the edge server to compute the device scheduling policy in the problem $\mathcal{P}_3$.

For problem $\mathcal{P}_3$, we have the following lemma, proved in appendix E.

Lemma 3. *The objective function of problem $\mathcal{P}_3$, i.e., $\mathcal{H}(\boldsymbol{S}_t)$ is monotonically decreasing with respect to the device set $\boldsymbol{S}_t$, i.e., for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$, we have $\mathcal{H}(\boldsymbol{S}_1) \geq \mathcal{H}(\boldsymbol{S}_2)$. The negative of $\mathcal{H}(\boldsymbol{S}_t)$, i.e., $-\mathcal{H}(\boldsymbol{S}_t)$, is a monotonically increasing submodular function with respect to the device set $\boldsymbol{S}_t$, i.e., for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$, and $h \in \mathcal{K} \backslash \boldsymbol{S}_2$, we have*

$$\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) \leq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_2) \tag{28}$$

According to Lemma 3, problem $\mathcal{P}_3$ is also a cardinality constraint submodular maximization problem. Thus, the greedy algorithm [36] is deployed to obtain a suboptimal solution in polynomial time complexity. Similarly, the greedy algorithm starts from $\boldsymbol{S}_t \leftarrow \emptyset$, and adds one device k with the maximum marginal gain, i.e., $k = \arg\min_{h \in \mathcal{K} \backslash \boldsymbol{S}_t} \left(\mathcal{H}(\boldsymbol{S}_t \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_t)\right)$ in every iteration, until $|\boldsymbol{S}_t| = N$. The detailed steps for finding the representativity-aware device scheduling policy are similar to Algorithm 2, and thus omitted for brevity.

D. Latency and Representativity-aware Scheduling Policy

In the above subsections, we develop the latency-aware and representativity-aware device scheduling policies. However, devices usually have different computing capabilities and channel conditions in the practical system, as well as different representativity in the different global rounds. Thus, the device scheduling policy should simultaneously consider the devices' latency and gradient representativity for accelerating the learning convergence. In this subsection, by utilizing the properties of latency and device representativity obtained in the above discussions, we develop an efficient algorithm to solve problem $\mathcal{P}$, which balances devices' latency and gradient representativity.

According to Lemma 1, the optimal latency for any device scheduling set $\boldsymbol{S}_t \subseteq \mathcal{K}$ can be obtained by using Algorithm 1, denoted as $\mathcal{T}_t^*(\boldsymbol{S}_t)$. Substituting $\mathcal{T}_t^*(\boldsymbol{S}_t)$ into problem $\mathcal{P}$, we transform $\mathcal{P}$ into the following equivalent problem:

$$\begin{aligned} \widetilde{\mathcal{P}}: \quad & \min_{\boldsymbol{S}_t} \quad \mathcal{R}(\boldsymbol{S}_t) = \rho_1 \mathcal{H}(\boldsymbol{S}_t) + \rho_2 \mathcal{T}_t^*(\boldsymbol{S}_t) \\ & \text{s. t.} \quad \text{(18a)}. \end{aligned} \tag{29}$$

For problem $\widetilde{\mathcal{P}}$, we have the following remark:

Algorithm 3 Double Greedy Algorithm for Latency and Representativity-aware Device Scheduling

Initialize $\boldsymbol{S}_1 \leftarrow \emptyset$ and $\boldsymbol{S}_2 \leftarrow \mathcal{K}$
for $k \in \mathcal{K}$ **do**
  Let $a_k \leftarrow (\max \mathcal{R}(\boldsymbol{S}_1) - \mathcal{R}(\boldsymbol{S}_1 \cup \{k\}), 0)$
  Let $b_k \leftarrow (\max \mathcal{R}(\boldsymbol{S}_2) - \mathcal{R}(\boldsymbol{S}_2 \backslash \{k\}), 0)$
  If $a_k = b_k = 0$, let $\frac{a_k}{a_k + b_k} = 1$
  With probability $\frac{a_k}{a_k + b_k}$ do $\boldsymbol{S}_1 \leftarrow \boldsymbol{S}_1 \cup \{k\}$ and $\boldsymbol{S}_2 \leftarrow \boldsymbol{S}_2$
  Otherwise $\boldsymbol{S}_1 \leftarrow \boldsymbol{S}_1$ and $\boldsymbol{S}_2 \leftarrow \boldsymbol{S}_2 \backslash \{k\}$
end for
Let $\boldsymbol{S}_t = \boldsymbol{S}_1$ (or $\boldsymbol{S}_t = \boldsymbol{S}_2$).
return The device scheduling set $\boldsymbol{S}_t$.

Remark 2. According to Lemma 2 and Lemma 3, $-\mathcal{T}_t^*(\boldsymbol{S}_t)$ is a monotonically decreasing submodular function with respect to the device set $\boldsymbol{S}_t$, while $-\mathcal{H}(\boldsymbol{S}_t)$ is a monotonically increasing submodular function. Consequently, the negative of the objective function of problem $\widetilde{\mathcal{P}}$, i.e., $-\rho_1 \mathcal{T}_t^*(\boldsymbol{S}_t) - \rho_2 \mathcal{H}(\boldsymbol{S}_t)$ is a non-monotone submodular function. Thus, problem $\widetilde{\mathcal{P}}$ is an unconstrained non-monotone submodular maximization problem, which is NP-Hard in general.

Base on Remark 2, we use the double greedy algorithm [39] to find a suboptimal solution for problem $\widetilde{\mathcal{P}}$. With regards to the implementation of the proposed algorithm, the edge server requires to collect devices channel information for computing their optimal bandwidth allocation policies and latency. After that, the edge server starts by initializing two device sets, i.e., $\boldsymbol{S}_1 = \emptyset$ and $\boldsymbol{S}_2 = \mathcal{K}$, and then serially passes through the devices in $\mathcal{K}$. When the algorithm passes device k ($k \in \mathcal{K}$), it determines online whether to add k into $\boldsymbol{S}_1$ or remove k from $\boldsymbol{S}_2$. This decision is based on a probability that trades off the gains of adding device k to $\boldsymbol{S}_1$ and removing k from $\boldsymbol{S}_2$. For clarity, we summarize the detailed steps of the double greedy algorithm for solving problem $\widetilde{\mathcal{P}}$ in Algorithm 3, which requires solving $2K$ times bandwidth allocation problem for finding the device scheduling set. Thus, the time complexity of Algorithm 3 is $\mathcal{O}(2K \log_2 \frac{\mathcal{T}_{t,\mathrm{ub}}(\boldsymbol{S}_t) - \mathcal{T}_{t,\mathrm{lb}}(\boldsymbol{S}_t)}{\varepsilon})$. In addition, for any device ordering, many existing works, e.g., [39], [40], have proved that the double greedy algorithm can achieve a tight 1/2 approximation of the optimal solution. Note that, the achieved approximation ratio of the double greedy algorithm is lower than the approximation ratio of Algorithm 2 (i.e., $1 - \frac{1}{e}$) for the optimal solution of the two special-case problems, i.e., latency-aware device scheduling problem and representativity-aware scheduling problem.

V. Numerical Results

In this section, we evaluate the proposed device scheduling algorithms for image classification tasks. All codes are implemented in python 3.8 and Pytorch, running on a Linux server.

A. Experiment Setting

Wireless setting: Unless specified, the default system settings are given as follows. We consider that $K = 100$ devices are randomly distributed within a 500m $\times$ 500m cell, and the edge server is located at the centre of this cell. The transmit power of devices is set to $p_k = 10$ dBm ($\forall k \in \mathcal{K}$). The system bandwidth and the variance of the complex white Gaussian channel noise are set to $B = 10$ MHz and $\sigma^2 = 10^{-12}$ W. The quantitative bit number for each gradient element is $q = 16$ bits. The channel gain is modelled as $h_{k,t} = h_0 \rho_k(t) d_k^{-2}$ [41], where $h_0 = -30$ dBm is the path loss constant; d_k is the distance between device k and the PS; $\rho_k(t) \sim \text{Exp}(1)$ is exponentially distributed with unit mean, which represents the small-scale fading channel power gain from the device k to the PS in round t. The CPU frequency for all devices are random selected from $\{0.8, 1.0, 1.2, 1.4, 1.6\}$ GHz.

Datasets and Models: For the exposition, we evaluate the proposed device scheduling policies under two classification learning tasks, i.e., the handwritten digits classification task on the MNIST dataset and the image classification task on the CIFAR-10 dataset. For the MNIST dataset, we train a multi-layer perceptron (MLP) model with a 784-units input layer, three hidden layers with 512, 256, and 64 units, and a 10-unit softmax output layer. The input layer and three hidden layers are all activated by the ReLU function. The MLP possesses 550346 parameters, which equals the number of FLOPs required for one data sample for gradient calculation. For the CIFAR-10 dataset, we train a convolutional neural network (CNN) with the following structure: two 5×5 convolution layers each with 64 channels and followed by a 2×2 max-pooling layer; three fully connected layers with 1600, 120, and 64 units, respectively; and a 10-unit softmax output layer. Each convolution or fully connected layer is activated by the ReLU function. The CNN possesses 307842 parameters, and the number of FLOPs required for dealing one data sample is 28206904. For both MLP and CNN, the learning rate η is set to 0.05, a momentum of 0.9 is adopted, the number of local iterations is set to $\tau = 8$, the batch size is set to 64, and cross entropy is adopted as the loss function. Besides, we first classify the training data samples according to their labels, then randomly split each class of data samples into $mK/10$ shards, finally randomly distribute m shards of data samples to each device.

B. Gradient Continuity

In Fig. 2, we evaluate the E-MBG and E-PG methods proposed in Section IV-C that estimate the gradient information of arbitrary device k ($k \in \mathcal{K}$). Fig. 2 provides the squared norm of the

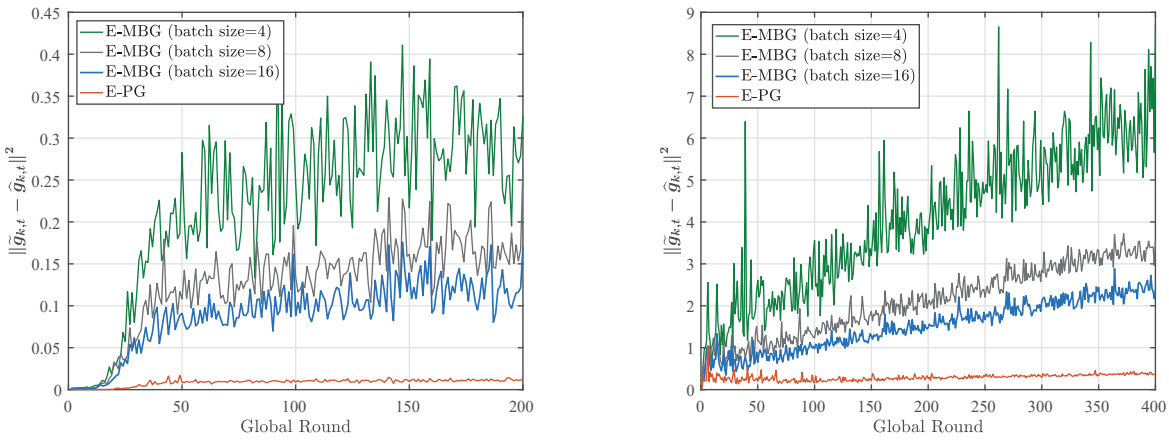


Fig. 2. The ℓ_2 norm of the difference between the estimated gradient and the true gradient of device k: (a) on MNIST dataset; (b) on CIFAR-10 dataset.

difference between the estimated gradient ($\widehat{\boldsymbol{g}}_{k,t}$) by E-MBG/E-PG and the true gradient of device k on MNIST and CIFAR-10 datasets, respectively. The batch size used to compute the gradient for device k is $L_b = 64$. In each round, device k further computes and records its local gradient with smaller batch sizes $L_b = 4$, 8, and 16, which is used for E-MBG to estimate its local gradient that is computed by $L_b = 64$. The E-PG method adapts the most recently received gradient at the edge server from device k to estimate the current gradient information of device k. For both MNIST and CIFAR-10 datasets, it is observed that E-PG outperforms the E-MBG method. In addition, the gradient estimation errors of E-MBG with different batch sizes are highly varying, and a smaller batch size produces a larger estimation error. Compared to E-MBG, the E-PG is able to achieve more accurate estimation, as well as no extra computation and communication cost. Thus, the E-PG method is embedded in the device scheduling algorithms in this work.

C. Performance of Representativity-aware Device Scheduling

To verify the effectiveness of the device representativity-aware scheduling policy proposed in Section IV-C, we compare its performance with the following three benchmark device scheduling schemes. Note that, we do not consider the computation latency and communication latency in this subsection.

1) *Random scheduling* (RS): The edge server uniformly selects a subset of devices from all devices to participate in the training in each round.

2) *Power-of-Choice scheduling* (PC) [19]: The edge server schedules a subset of devices with larger local losses each round. Note that this scheme requires devices to compute the local loss

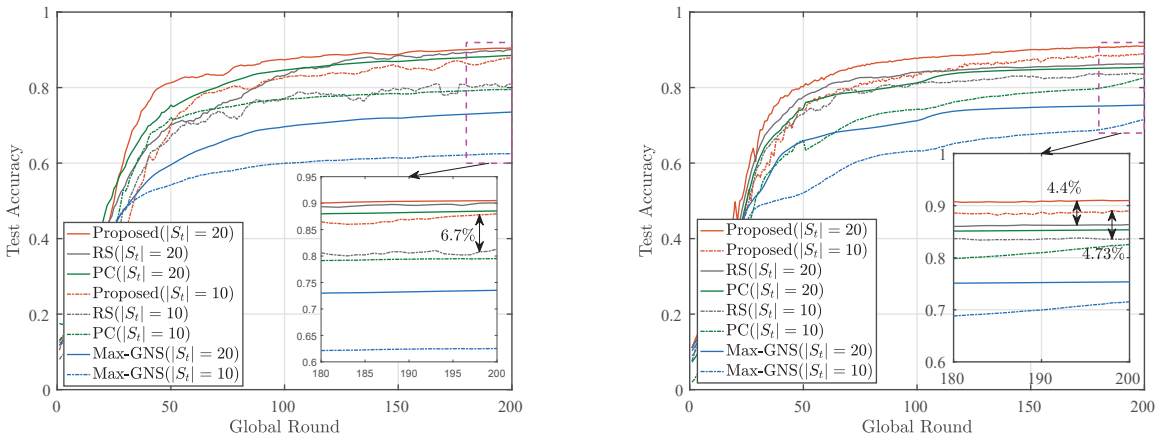
functions and upload them to the edge server in each round, thus may result in extra computation and transmission costs.

3) *Maximum gradient norm scheduling* (Max-GNS): The edge server schedules a subset of devices with the maximum gradient norm in each round. The ℓ_2-norm of the gradients have been widely used in existing works, e.g., [21], [32], to represent the significance of local gradients. However, the existing works require all devices to perform local training and then upload their gradient norm to the edge server for device scheduling. This may result in the unnecessary energy consumption of the unscheduled devices. Based on the above analysis, we use the past gradient norms of devices in this baseline to decide which devices are scheduled.

Based on the MNIST dataset, Fig. 3 compares the learning performance of the proposed algorithm with the above-listed three scheduling schemes under different data heterogeneity and scheduling ratios. In Fig. 3(a), we distribute at most two classes of data samples to each device. The results show that our proposed algorithm outperforms the three benchmarks, converges faster, and obtains higher accuracy. Specifically, when 10 devices participate in the learning process in each round, the proposed algorithm achieves a 6.7% accuracy improvement compared with the random scheduling policy. Although the proposed algorithm obtains a similar accuracy to the random scheduling policy when 20 devices participate in each round, it has a faster convergence speed.

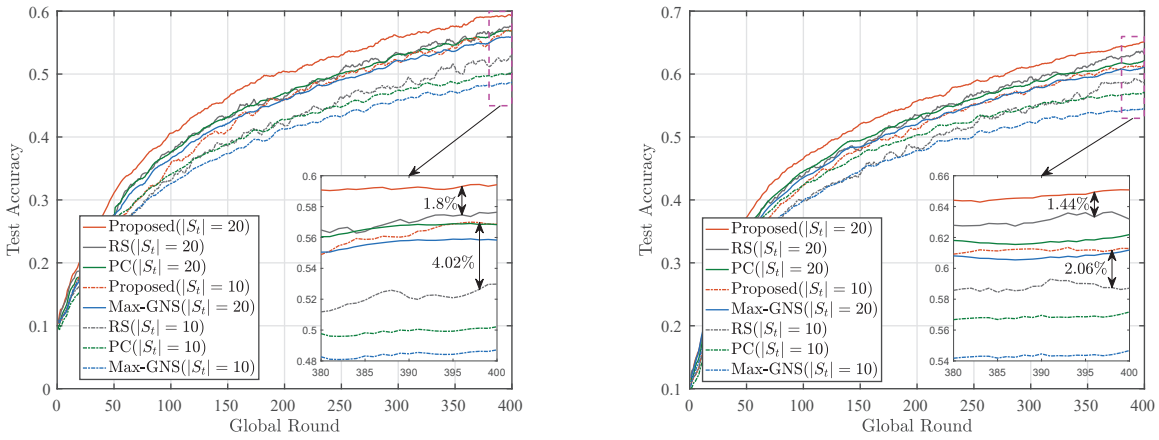
Fig. 3(b) distributes at most three classes of data samples to each device, in which the data heterogeneity between devices is lower than Fig. 3(a). It is observed that the learning accuracies of all the scheduling schemes improved compared with Fig. 3(a). This is because high data heterogeneity can weak the generalisation ability of the learned global model, further resulting in poor learning performance. In addition, it is also observed that the proposed algorithm obtains high accuracies than the three benchmarks. Compared with the random scheduling policy, the proposed algorithm obtains a 4.73% accuracy improvement when $|\boldsymbol{S}_t| = 10$ and a 4.4% accuracy improvement when $|\boldsymbol{S}_t| = 20$.

A similar evaluation is conducted on the CIFAR-10 dataset, as shown in Fig. 4. In Fig. 4(a), we set the data heterogeneity related control parameter as $m = 2$. Compared with the random scheduling policy, the proposed algorithm boosts 4.02% accuracy when $|\boldsymbol{S}_t| = 10$ and improves 1.8% accuracy when $|\boldsymbol{S}_t| = 20$. In Fig. 4(b), we set $m = 3$. A distinct accuracy improvement for all scheduling schemes is observed compared with $m = 2$ on this more complicated dataset. In addition, the proposed algorithm performs well compared to the three benchmarks, obtaining



(a) (b)

Fig. 3. Learning performance of different device scheduling algorithms on MNIST dataset: (a) $m = 2$; (b) $m = 3$.



(a) (b)

Fig. 4. Learning performance of different device scheduling algorithms on CIFAR-10 dataset: (a) $m = 2$; (b) $m = 3$.

2.06% and 1.44% accuracy improvement when $|\boldsymbol{S}_t| = 10$ and $|\boldsymbol{S}_t| = 20$, respectively.

D. Performance of Latency and Representativity-aware Device Scheduling

In this subsection, we evaluate the performance of the proposed device scheduling policies, i.e., 1) latency-aware device scheduling (in Section IV-B), 2) representativity-aware device scheduling (in Section IV-C), 3) latency- and representativity-aware device scheduling (L&R-aware) (in Section IV-D). Note that for both latency-aware scheduling and representativity-aware scheduling policies, we test their performance on $|\boldsymbol{S}_t| = 0.1K, 0.2K, \cdots, 1.0K$ and then plot the best two results. For the latency- and representativity-aware device scheduling scheme, the number of participants is automatically decided by the algorithm to adapt the parameters ρ_1 and ρ_2. When ρ_1 is large and ρ_2 is relatively small, $|S_t|$ will increase to reduce the gradient approximation

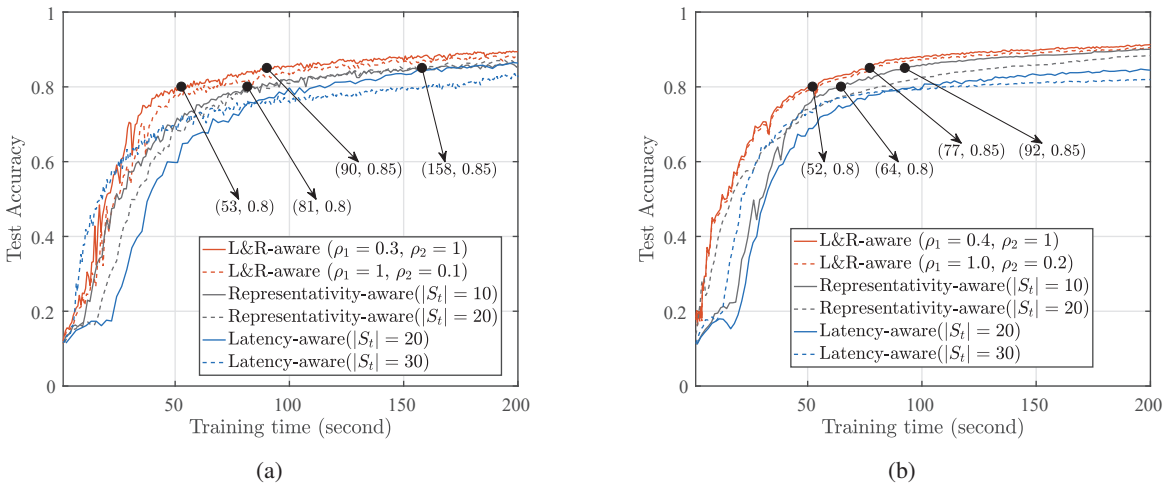


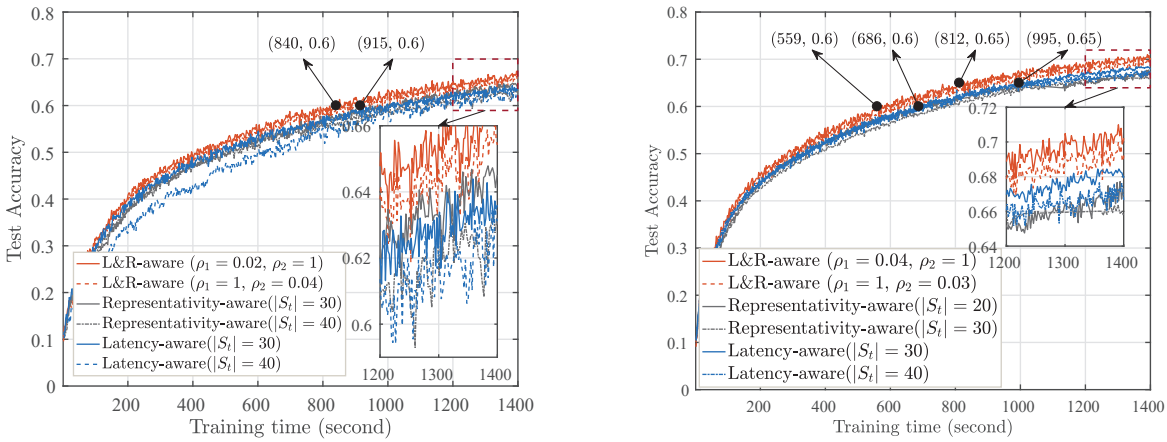
Fig. 5. Learning performance of different device scheduling algorithms on MNIST dataset: (a) $m = 2$; (b) $m = 3$.

error ($\mathcal{H}(S_t)$) as much as possible. In contrast, when ρ_1 is small while ρ_2 is large, $|S_t|$ would decrease to reduce the latency ($\mathcal{T}(S_t)$).

Fig. 5 shows the performance of the proposed three device scheduling algorithms on the MNIST dataset. For both $m = 2$ and $m = 3$, we serially select ρ_1 and ρ_2 from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and plot the best and worst results. It is observed that the proposed latency- and representativity-aware device algorithm always performs better than the other two device scheduling algorithms. In addition, the proposed latency- and representativity-aware device algorithm achieves better performance by setting $\rho_1 < \rho_2$.

Fig. 5 shows the performance of the proposed three device scheduling algorithms on the MNIST dataset. For both $m = 2$ and $m = 3$, we set $\rho_1 = 0.3$ and $\rho_2 = 1$. In Fig. 5(a), we evaluate the test accuracy with $m = 2$ which indicates that each device possesses at most two classes of the data samples. Specifically, given the target accuracy is 80%, the latency- and representativity-aware device scheduling algorithm only spends 53 seconds for achieving the target, while the representativity-aware scheduling algorithm takes 81 seconds. That is, compared with the representativity-aware scheduling algorithm, the latency- and representativity-aware algorithm is able to save 34.5% training time to obtain 80% test accuracy. In addition, when the target accuracy is 85%, the latency- and representativity-aware algorithm is able to save at least 43% training time in comparison with other device scheduling algorithms.

Fig. 5(b) evaluates the performance of the proposed device scheduling algorithms in a less heterogeneous scenario, i.e., $m = 3$. It is observed that all the algorithms perform well in this situation compared to that in $m = 2$. Similar to the evaluation in $m = 2$, the latency-



(a) (b)

Fig. 6. Learning performance of different device scheduling algorithms on CIFAR-10 dataset: (a) $m = 2$; (b) $m = 3$.

and representativity-aware algorithm obtains the best learning performance. Compared to other device scheduling algorithms, the latency- and representativity-aware algorithm saves 18.8% and 16.3% training time when the target accuracy is 80% and 85%, respectively.

Fig. 6 presents the learning performance of the proposed three device scheduling algorithms on the CIFAR-10 dataset. For the latency- and representativity-aware scheduling algorithm, we evaluate its performance by selecting ρ_1 from $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$ and ρ_2 from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, then plot the best and worst results. Similar to the results on the MNIST dataset, the latency- and representativity-aware scheduling algorithm provides a better learning performance than the other two scheduling policies based only on either of the two metrics individually. Fig. 6(a) presents the learning performance of the device scheduling algorithms with $m = 2$. Specifically, when the target accuracy is 60%, the latency- and representativity-aware scheduling algorithm require at most 88% training time of the other two scheduling schemes. In Fig. 6(b), the data heterogeneity parameter is set to $m = 3$. It is observed that all the scheduling algorithms converge faster in this situation than that in $m = 2$. When the target accuracy is 60%, the latency- and representativity-aware scheduling algorithm is able to reduce 18.5% of the training time compared to the other two benchmarks and save 18.4% time when the target accuracy is 65%.

VI. Conclusion

In this work, we proposed a novel latency- and representativity-aware device scheduling algorithm to accelerate the learning process for FL. We first revealed that the device scheduling policies affect learning convergence through the error between the scheduled devices' aggregated

gradient and full participation aggregated gradient. Then, by proving the submodularity of both latency and representativity of the scheduled device set, we developed a double greedy algorithm to capture the trade-off between latency and representativity in each round. To mitigate the extra costs produced by local training of unscheduled devices, we utilized the past gradient information to guide the device scheduling policy design in each round. The experiments verified the effectiveness of the proposed device scheduling algorithm and the use of past gradient information to schedule devices.

APPENDIX

A. *Proof of Theorem* 1

According to the L-Lipschitz continuous of loss gradients $\nabla F_k(\boldsymbol{w})$ in Assumption 1, we have

$$F_k(\boldsymbol{w}) \leq F_k(\boldsymbol{v}) - \langle \nabla F_k(\boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{L}{2} \|\boldsymbol{w} - \boldsymbol{v}\|^2 \tag{30}$$

For ease of proof, we define $\bar{\boldsymbol{g}}_t = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=0}^{\tau-1} \nabla F_k(\boldsymbol{w}_{k,t,l})$. Thus,

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}_t)\right] \\
&\leq \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \boldsymbol{w}_{t+1} - \boldsymbol{w}_t \rangle\right] + \frac{L}{2} \mathbb{E}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|^2 \\
&= -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E}\|\widetilde{\boldsymbol{g}}_t\|^2 \\
&= -\eta \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_t) + \widetilde{\boldsymbol{g}}_t\|^2 \\
&= -\frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + (L\eta^2 - \eta) \mathbb{E}\left[\langle \nabla F(\boldsymbol{w}_t), \widetilde{\boldsymbol{g}}_t \rangle\right] + \frac{L}{2} \eta^2 \mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \widetilde{\boldsymbol{g}}_t\|^2 \\
&\overset{(a)}{\leq} -\frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + \frac{L}{2} \eta^2 \mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \boldsymbol{g}_t - \boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 \\
&\overset{(b)}{\leq} -\frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E}\|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \boldsymbol{g}_t\|^2 \\
&= -\frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E}\|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t - \bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2 \\
&\overset{(c)}{=} -\frac{L}{2} \eta^2 \mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 + L\eta^2 \mathbb{E}\|-\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2 + L\eta^2 \mathbb{E}\|-\bar{\boldsymbol{g}}_t + \boldsymbol{g}_t\|^2,
\end{aligned} \tag{31}$$

where (a) derived by Cauchy-Schwarz inequality and $\eta \leq \frac{1}{L}$, (b) follows the triangle-inequality, (c) is due to the unbiased stochastic gradient in Assumption 2.

Below we first bound $\mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2$.

$$\begin{aligned}
&\mathbb{E}\|-\nabla F(\boldsymbol{w}_t) + \bar{\boldsymbol{g}}_t\|^2 \\
&= \mathbb{E}\left\| \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{\tau-1} \nabla F(\boldsymbol{w}_{k,t,l}) \right\|^2
\end{aligned}$$

$$\begin{aligned}
&\overset{(a)}{\leq} \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l})\|^2 \\
&= \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_{k,t,l}) - \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}) + \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\right\|^2 \\
&\overset{(b)}{=} \frac{1}{K}\sum_{k=1}^{K}(\tau-1)\sum_{l=1}^{\tau-1}\left\{\mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l}) - \nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\|^2 + \mathbb{E}\|\nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l})\|^2\right\} \\
&\overset{(c)}{\leq} (\tau-1)^2(G^2+\chi^2)
\end{aligned} \tag{32}$$

where (a) follows Jensen's inequality, (b) is due to the unbiased gradient in Assumption 2, (c) follows the Assumption 2 and Assumption 3. In the following, we bound $\mathbb{E}\|-\bar{\boldsymbol{g}}_t+\boldsymbol{g}_t\|^2$ as follows:

$$\begin{aligned}
\mathbb{E}\|-\bar{\boldsymbol{g}}_t+\boldsymbol{g}_t\|^2 &= \mathbb{E}\left\|\frac{1}{K}\sum\nolimits_{k=1}^{K}\sum\nolimits_{l=0}^{\tau-1}\left(\nabla F(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}) - \nabla F(\boldsymbol{w}_{t,l})\right)\right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{K}\sum_{k=1}^{K}\tau\sum_{l=0}^{\tau-1}\mathbb{E}\|\left(\nabla F\left(\boldsymbol{w}_{k,t,l},\mathcal{B}_{k,t,l}\right) - \nabla F(\boldsymbol{w}_{t,l})\right)\|^2 \\
&\overset{(b)}{\leq} \tau^2 G^2
\end{aligned} \tag{33}$$

where (a) follows Jensen's inequality, (b) is due to Assumption 2. Substituting (32) and (33) into (31), the proof is completed.

B. Proof of Corollary 1

For the sake of proof, we define an auxiliary function $L_{\boldsymbol{w}}(\boldsymbol{v}) = F(\boldsymbol{v}) - \langle\nabla F(\boldsymbol{w}),\boldsymbol{v}\rangle$, which has a minimizer $\boldsymbol{v}^* = \boldsymbol{w}$. In addition, let $\Xi(\boldsymbol{v}) = \frac{L}{2}\|\boldsymbol{v}\|^2 - L_{\boldsymbol{w}}(\boldsymbol{v}) = \frac{L}{2}\|\boldsymbol{v}\|^2 - F(\boldsymbol{v}) + \langle\nabla F(\boldsymbol{w}),\boldsymbol{v}\rangle$. According to the proof of Theorem 1 in Appendix A, $\frac{L}{2}\|\boldsymbol{v}\|^2 - F(\boldsymbol{v})$ is a convex function with respect to $\boldsymbol{v}$. Thus, $\Xi(\boldsymbol{v})$ is a convex function with respect to $\boldsymbol{v}$ due to the convexity of $\langle\nabla F(\boldsymbol{w}),\boldsymbol{v}\rangle$. Utilizing the first-order condition of convex function, i.e., $\Xi(\boldsymbol{v}) \geq \Xi(\boldsymbol{u}) + \langle\nabla\Xi(\boldsymbol{u}),\boldsymbol{v}-\boldsymbol{u}\rangle$, we have

$$L_{\boldsymbol{w}}(\boldsymbol{v}) \leq L_{\boldsymbol{w}}(\boldsymbol{u}) + \langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}),\boldsymbol{v}-\boldsymbol{u}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2. \tag{34}$$

Taking minimization with $\boldsymbol{v}$ on the both sides of the above inequation,

$$\begin{aligned}
\inf_{\boldsymbol{v}} L_{\boldsymbol{w}}(\boldsymbol{v}) = L_{\boldsymbol{w}}(\boldsymbol{w}) &\leq \inf_{\boldsymbol{v}}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) + \langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}),\boldsymbol{v}-\boldsymbol{u}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2\right\} \\
&= \inf_{\|\boldsymbol{x}\|=1}\inf_{\boldsymbol{y}}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) + \boldsymbol{y}\langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}),\boldsymbol{x}\rangle + \frac{L}{2}\boldsymbol{y}^2\right\} \\
&= \inf_{\|\boldsymbol{x}\|=1}\left\{L_{\boldsymbol{w}}(\boldsymbol{u}) - \frac{\|\langle\nabla L_{\boldsymbol{w}}(\boldsymbol{u}),\boldsymbol{x}\rangle\|^2}{2L}\right\} \\
&= L_{\boldsymbol{w}}(\boldsymbol{u}) - \frac{\|\nabla L_{\boldsymbol{w}}(\boldsymbol{u})\|^2}{2L}.
\end{aligned} \tag{35}$$

Substituting $L_{\boldsymbol{w}}(\boldsymbol{w})$ and $L_{\boldsymbol{w}}(\boldsymbol{u})$ into (35), we have

$$F(\boldsymbol{u}) \geq F(\boldsymbol{w}) + \langle \nabla F(\boldsymbol{w}), \boldsymbol{u} - \boldsymbol{w} \rangle + \frac{1}{2L} \left\| \nabla F(\boldsymbol{u}) - \nabla F(\boldsymbol{w}) \right\|^2. \tag{36}$$

Since $\nabla F(\boldsymbol{w}^*) = 0$, we have

$$\left\| \nabla F(\boldsymbol{w}_t) \right\|^2 \leq 2L \left(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*) \right). \tag{37}$$

By substracting $F(\boldsymbol{w}^*)$ for both $F(\boldsymbol{w}_{t+1})$ and $F(\boldsymbol{w}_t)$ in the one-round convergence bound in (12),

$$\begin{aligned} \mathbb{E}(F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*)) \leq \mathbb{E}(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)) - \frac{L}{2}\eta^2 \mathbb{E} \left\| \nabla F(\boldsymbol{w}_t) \right\|^2 \\ + L\eta^2 \left(2\tau^2 - 2\tau + 1 \right) G^2 + L\eta^2 (\tau - 1)^2 \chi^2 + L\eta^2 \left\| -\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t \right\|^2. \end{aligned} \tag{38}$$

Substituting (37) into (38), we have

$$\begin{aligned} F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*) \leq \left(1 - L^2\eta^2 \right) \left(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*) \right) \\ + L\eta^2 \left(2\tau^2 - 2\tau + 1 \right) G^2 + L\eta^2 (\tau - 1)^2 \varepsilon^2 + L\eta^2 \left\| -\boldsymbol{g}_t + \widetilde{\boldsymbol{g}}_t \right\|^2. \end{aligned} \tag{39}$$

Telescoping the above equation, the convergence bound after T rounds can be derived as Corollary 1. The proof is completed.

C. Proof of Lemma 1

From the definition of $T_{k,t}^{\mathrm{C}}$ in (7), it is straightforward to see that $T_{k,t}^{\mathrm{C}}$ is a monotonically decreasing function with respect to $\theta_{k,t}$. For any device that finished the local gradient computing process earlier than other devices, we can reallocate some of its bandwidth to other slower devices. As a result, the one-round latency determined by the slowest device can be reduced. The bandwidth reallocation process will be performed until all devices simultaneously finish the local gradient computing and uploading. Consequently, the optimal solution of $\mathcal{P}_1$ is achieved when the entire bandwidth is allocated to all scheduled devices to have the same finishing time. Thus, the optimal bandwidth allocation policy satisfies

$$\begin{cases} T_{k,t}^{\mathrm{L}} + T_{k,t}^{\mathrm{C}} = \mathcal{T}_t^*(\boldsymbol{S}_t), \forall k \in \boldsymbol{S}_t \\ \sum_{k \in \boldsymbol{S}_t} \theta_k = 1, \end{cases} \tag{40}$$

where $\mathcal{T}_t^*(\boldsymbol{S}_t)$ is the optimal latency in round t. By solving (40), the proof is completed.

D. Proof of Lemma 2

For ease of presentation, we first denote the minimal gradient uploading latency for device k ($k \in \mathcal{K}$) as $T_{k,t}^{\mathrm{C,min}} = \frac{Qq}{B \log\left(1 + \frac{p_k h_{k,t}}{\sigma^2}\right)}$, which is derived when device k occupies the entire bandwidth to upload its gradient. Below we first prove that the optimal latency function $\mathcal{T}_t^*(\boldsymbol{S})$

is a monotonically increasing function with the device set $\boldsymbol{S}$. Based on Lemma 1, for device set $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$, we have

$$\sum\nolimits_{k\in\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} = \sum\nolimits_{k\in\boldsymbol{S}_2} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} = 1, \tag{41}$$

which equivalent to

$$\sum\nolimits_{k\in\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} = \sum\nolimits_{k\in\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} + \sum\nolimits_{k\in\boldsymbol{S}_2\backslash\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}}. \tag{42}$$

By rearranging the above equation, we have

$$\sum_{k\in\boldsymbol{S}_1} \left(\frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} - \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} \right) = \sum_{k\in\boldsymbol{S}_2\backslash\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_2) - T_{k,t}^{\mathrm{L}}} > 0 \tag{43}$$

Thus, we have $\mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*(\boldsymbol{S}_2)$. That is, $\mathcal{T}_t^*(\boldsymbol{S})$ is a monotonically increasing function with the device set $\boldsymbol{S}$. Similarly, for device $h \in \mathcal{K}\backslash\boldsymbol{S}_2$, we have

$$\sum_{k\in\boldsymbol{S}_1} \left(\frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - T_{k,t}^{\mathrm{L}}} - \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} \right) + \frac{T_{h,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - T_{h,t}^{\mathrm{L}}} = 0 \tag{44}$$

By rearranging the above equation, we have

$$\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_1) = \frac{T_{h,t}^{\mathrm{C,min}}}{1 + \sum\limits_{k\in\boldsymbol{S}_1} \frac{T_{k,t}^{\mathrm{C,min}}}{\mathcal{T}_t^*(\boldsymbol{S}_1) - T_{k,t}^{\mathrm{L}}} \frac{T_{k,t}^{\mathrm{L}} - T_{h,t}^{\mathrm{L}}}{\mathcal{T}_t^*(\boldsymbol{S}_1 + h) - T_{k,t}^{\mathrm{L}}}} \tag{45}$$

Since $\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) < \mathcal{T}_t^*\left(\boldsymbol{S}_2 + h\right)$, based on (41), we have $\mathcal{T}_t^*\left(\boldsymbol{S}_1 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_1) \leq \mathcal{T}_t^*\left(\boldsymbol{S}_2 + h\right) - \mathcal{T}_t^*(\boldsymbol{S}_2)$, the proof is completed.

E. Proof of Lemma 3

For ease of presentation, we define two device sets, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, such that $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2 \subseteq \mathcal{K}$, and a device $h \in \mathcal{K}\backslash\boldsymbol{S}_2$. Based on the definition of $H(\boldsymbol{S}_t)$, we have $\mathcal{H}(\boldsymbol{S}_1) \geq \mathcal{H}(\boldsymbol{S}_2)$ and $\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) \geq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\})$. Moreover, we have

$$\begin{aligned}
\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) &= \sum_{k=1}^{K} \min_{e\in\boldsymbol{S}_1\cup\{h\}} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_e(\boldsymbol{w}_t)\| - \sum_{k=1}^{K} \min_{e\in\boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_e(\boldsymbol{w}_t)\| \\
&= \sum_{k=1}^{K} \min\Big(0, \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_n(\boldsymbol{w}_t)\| - \min_{h\in\boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\|\Big).
\end{aligned} \tag{46}$$

Since $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$, we have $\min\limits_{h\in\boldsymbol{S}_1} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\| \geq \min\limits_{h\in\boldsymbol{S}_2} \|\nabla F_k(\boldsymbol{w}_t) - \nabla F_h(\boldsymbol{w}_t)\|$. Thus, $\mathcal{H}(\boldsymbol{S}_1 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_1) \leq \mathcal{H}(\boldsymbol{S}_2 \cup \{h\}) - \mathcal{H}(\boldsymbol{S}_2)$, the proof is completed.

REFERENCES

[1] Z. Chen, W. Yi, Y. Deng, and A. Nallanathan, "Device scheduling for wireless federated learning with latency and representativity," in *Proc. International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–6.

[2] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[3] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 2021.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 20–22, Apr. 2017.

[5] Z. Chen, W. Yi, A. Nallanathan, and G. Y. Li, "Federated learning for energy-limited wireless networks: A partial model aggregation approach," *arXiv preprint arXiv:2204.09746*, 2022.

[6] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Knowledge-aided federated learning for energy-limited wireless networks," *IEEE Trans. Commun.*, pp. 1–1, 2023.

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

[8] 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.

[9] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas in Commun.*, vol. 40, no. 1, pp. 227–242, 2022.

[10] M. R. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, and M. Kopp, "Asynchronous federated learning for geospatial applications," in *Proc Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018.

[11] 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 in Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[12] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, H. Shan, W. Wang, and T. Q. Quek, "Asynchronous federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2022.

[13] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, 2021.

[14] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (wfl) for 6g networkspart i: Research challenges and future trends," *IEEE Commun. Letters*, vol. 26, no. 1, pp. 3–7, 2022.

[15] S. Zheng, Q. Meng, T. Wang, W. Chen, N. Yu, Z.-M. Ma, and T.-Y. Liu, "Asynchronous stochastic gradient descent with delay compensation," in *Proc International Conference on Machine Learning (ICML)*, 06–11 Aug 2017.

[16] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, 2021.

[17] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, and S. Cui, "Communication-efficient federated edge learning via optimal probabilistic device scheduling," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2022.

[18] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," in *Proc. IEEE ICASSP*, 2021.

[19] Y. Jee Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proc Artificial Intelligence and Statistics (AISTATS)*, 28–30 Mar 2022.

[20] I. Mohammed, S. Tabatabai, A. Al-Fuqaha, F. E. Bouanani, J. Qadir, B. Qolomany, and M. Guizani, "Budgeted online selection of candidate iot clients to participate in federated learning," *IEEE Internet of Things J.*, vol. 8, no. 7, pp. 5938–5952, 2021.

[21] 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, 2021.

[22] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, and W. Zhuang, "Efficient federated meta-learning over multi-access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1556–1570, 2022.

[23] 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, 2021.

[24] A. Tak, Z. Mlika, and S. Cherkaoui, "Data-aware device scheduling for federated edge learning," *IEEE Trans. Cognitive Commun. and Netw.*, vol. 8, no. 1, pp. 408–421, 2022.

[25] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *Proc International Conference on Machine Learning (ICML)*, 13–18 Jul 2020.

[26] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, 2021.

[27] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, 2021.

[28] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, 2012.

[29] E. Abbasnejad, J. Shi, and A. van den Hengel, "Deep Lipschitz networks and dudley GANs," 2018. [Online]. Available: https://openreview.net/forum?id=rkw-jlb0W

[30] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2018.

[31] A. Dedieu, "Improved error rates for sparse (group) learning with lipschitz loss functions," *arXiv preprint arXiv:1910.08880*, 2019.

[32] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[33] 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, 2020.

[34] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[35] A. Rafiey and Y. Yoshida, "Fast and private submodular and k-submodular functions maximization with matroid constraints," in *Proc International Conference on Machine Learning (ICML)*, 13–18 Jul 2020.

[36] A. Krause and D. Golovin, "Submodular function maximization." *Tractability*, vol. 3, pp. 71–104, 2014.

[37] S. Wang, M. Chen, C. Shen, C. Yin, and C. G. Brinton, "Cross-layer federated learning optimization in mimo networks," *arXiv preprint arXiv:2302.14648*, 2023.

[38] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE J. Sel. Topics in Signal Processing*, pp. 1–1, 2022.

[39] X. Pan, S. Jegelka, J. E. Gonzalez, J. K. Bradley, and M. I. Jordan, "Parallel double greedy submodular maximization," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2014.

[40] N. Buchbinder and M. Feldman, "Submodular functions maximization problems," in *Handbook of Approximation Algorithms and Metaheuristics, Second Edition*. Chapman and Hall/CRC, 2018, pp. 753–788.

[41] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, 2022.