Resource-Efficient and Convergence-Preserving Online Participant Selection in Federated Learning

Yibo Jin[1], Lei Jiao[2], Zhuzhong Qian[1], Sheng Zhang[1], Sanglu Lu[1], Xiaoliang Wang[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China [2]University of Oregon, USA

***Abstract*—Federated learning achieves the privacy-preserving training of models on mobile devices by iteratively aggregating model updates instead of raw training data to the server. Since excessive training iterations and model transferences incur heavy usage of computation and communication resources, selecting appropriate devices and excluding unnecessary model updates can help save the resource usage. We formulate an online time-varying non-linear integer program to minimize the cumulative resource usage over time while achieving the desired long-term convergence of the model being trained. We design an online learning algorithm to make fractional control decisions based on both previous system dynamics and previous training results, and also design an online randomized rounding algorithm to convert the fractional decisions into integers without violating any constraints. We rigorously prove that our online approach only incurs sub-linear dynamic regret for the optimality loss and sub-linear dynamic fit for the long-term convergence violation. We conduct extensive trace-driven evaluations and confirm the empirical superiority of our approach over alternative algorithms in terms of up to 27% reduction on the resource usage while sacrificing only 4% reduction on accuracy.**

I. Introduction

Federated Learning enables on-device training of machine learning models using local data [1, 2]. This is in stark contrast to traditional approaches, where data that are used for model training are often sent to a central location, such as a remote data center, incurring users' increasing concerns on their data privacy [3]. Federated learning can thus protect data privacy, finding extensively potential applications to the mobile devices (e.g., smart phones, tablets) as they often continuously produce abundant and diverse data (e.g., website click logs [4], GPS trajectories [5]) via various applications [6, 7].

However, federated learning could consume excessive computation and communication resources. First, as shown in previous research [8, 9], model training often leads to thousands of computing iterations. In each iteration, each participant (i.e., the mobile device that joins the current federated learning process) uses its own raw data to update the local model iteratively through Stochastic Gradient Descent (SGD), batch SGD, etc., which is both time- and resource-consuming [10, 11]. Second, despite no raw data need to be sent in the federated learning scheme, each participant needs to upload the updated local model to a (logically) centralized server for aggregation and download the aggregated model for further updating. Given the often asymmetric and scare [12] wireless bandwidth for mobile devices, it can incur heavy workload over the network if many devices simultaneously participate in federated learning.

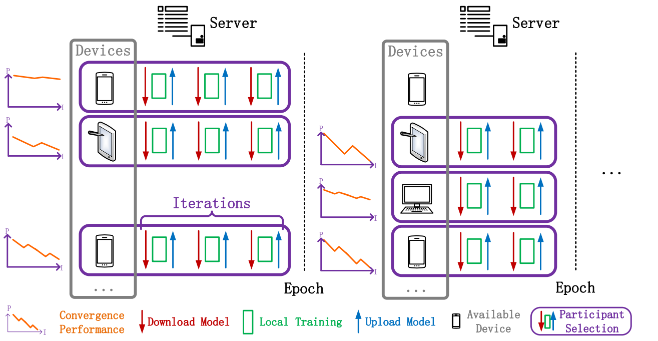


Fig. 1: A federated learning system with participant selection

To reduce the resource usage of a federated learning system, one may strategically select a subset of all the possibly considered participants (with their data) to join the training process [1, 12, 13]; however, participant selection turns out to be a non-trivial problem, especially in an online setting. First, the mobile devices' availabilities [6], the amount of training data each device generates, and the wireless networks' bandwidth are all highly dynamic and can be unpredictable. It is thus difficult to select suitable participants beforehand or keep adapting the selection on the fly. Second, reducing the resource usage should not sacrifice the quality of the model that the system can train. Particularly, one may have convergence requirements [8, 14, 15] for each device's local model being trained and also for the global aggregated model eventually. How to preserve such convergence by using only the subsets of all the available data over time is hard, which also involves the control of the number of training iterations set for each device. The fundamental challenge and obstacle of the online setting is that, in each epoch, one needs to select the participants and set the number of iterations *before* getting to know the resource usage and the training results incurred by the decisions being made. Fig. 1 illustrates a federated learning system with device availabilities and participant selection.

Existing research falls insufficient for addressing the aforementioned challenges. Despite some works [11, 13, 16–18] have considered aggregation control or participant selection, they were largely for an offline setting and could hardly adapt to the unpredictably time-varying inputs with guaranteed model convergence. Others [9, 19–21] focused on analyzing the convergence performance of the federated learning procedure itself, and did not investigate it from a system perspective considering resource usage. The rest literatures [22–27] widely studied online resource management for cloud/edge systems, but could not be applied here directly, due to the lack of the consideration of the computation and communication patterns and model convergence in federated learning systems.

In this paper, we firstly model the participant selection problem of federated learning as an online time-varying non-linear integer program that minimizes the total cumulative usage of the computation and communication resources, subject to the server capacity and the long-term convergence requirements for both local and the aggregated models on each device and on the sever, respectively. Our problem formulation leverages the relationship between the desired model convergence and the needed maximal number of training iterations, as well as captures the system dynamics of mobile device availabilities, training data volumes, and wireless network bandwidth.

Afterwards, we design and present an online learning algorithm to jointly control, in an online manner, the participant selection and the number of training iterations that needs to be done for each selected participant. Our algorithm features two algorithmic components: an online learning component that makes fractional decisions based on previous resource usage and training results (without worrying about not seeing the possible consequence of the current decisions being made), and an online rounding component that converts the fractional decisions into integral selection decisions without violating our problem's constraints. In particular, the online learning component, using no a priori knowledge of the inputs, solves the problem through a series of carefully-designed alternating descent-ascent iterations based only on the observable inputs so far, while the randomized rounding component keeps the expectation of randomized integers equal to the corresponding fractions based on a compensation strategy to maintain the constraints. We consider the performance metrics of the dynamic regret [28, 29], which quantifies the optimality loss of our approach relative to a sequence of instantaneous optimizers with assumed access to the inputs beforehand, and the dynamic fit [24, 30], which characterizes the cumulative long-term constraint violation of our approach. We highlight that, through rigorous formal analysis, we prove that both of these two metrics only grow sub-linearly along with time.

Finally, we conduct extensive evaluations using real-world training traces from Google [8, 12, 14] to validate the practical performance of our proposed online schema. We find that our approach performs the best for a variety of system dynamics. In general, our approach saves at least 27% resource usage on average compared to state-of-the-art algorithms, with only 4% reduction on global convergence, which is very moderate in realistic scenarios [8, 9]. We also evaluate our approach under various workloads and scenarios, confirming the empirical superiority of proposed online schema over other algorithms in terms of up to 41.6% reduction on the resource usage.

The remainder of this paper proceeds as follows. Section II presents the system model and the problem formulation. Sections III and IV design our online algorithm and present the theoretical analysis, respectively. Section V evaluates the practical performance through trace-driven simulations. Section VI summarizes and reviews the related works, and finally Section VII concludes.

TABLE I: Summary of Major Notations

Symbol	Description
$\mathcal{S}_t$	Set of available devices in epoch t
$D_{t,i}$	Set of raw data generated on device i in epoch t
$\mathcal{F}_{t,i}$	Loss on the data of device i in epoch t
$\tilde{\mathcal{F}}_t, \mathcal{F}_t$	Loss on the data of all participants and all available devices
$\boldsymbol{w_{t,i}^{(j)}}$	Model[1] trained by device i after iteration j in epoch t
$\theta_{t,i}^{(j)}, \varepsilon_0, \varepsilon$	Local and global convergence parameters
κ_t	Number of training iterations in epoch t
$\alpha_{t,i}$	Resource used for training one data sample on device i
b_t	Available bandwidth for model updates in epoch t
$f_t(\cdot), g_t(\cdot)$	Abstract objective and long-term constraint in epoch t
$\alpha, \mu, \boldsymbol{\lambda}_t$	Non-negative algorithmic parameters
Decision	Description
$\phi_{t,i}$	Whether device i is selected for participation in epoch t
ϑ_t	Maximum local convergence accuracy in epoch t

1. All of the model refers to the model parameters (vector form).

II. System Model and Problem Formulation

A. System Model

We model our target federated learning system as follows. We summarize all the major notations in Table I.

Mobile Devices and Server: We consider a set of mobile devices $\mathcal{N} = \{1, ..., N\}$. Without loss of generality, all devices connect to a (logically) centralized server, e.g., the micro server co-located with a nearby cellular base station or the server hosted in a remote cloud. This server or the network has the transmission capacity or the bandwidth constraint m, i.e., at most m mobile devices can transfer data concurrently to the server during the model training process.

Federated Learning: We consider a series of consecutive epochs $\mathcal{T} = \{1, ..., T\}$, and in each epoch we train a model via federated learning. We adopt the federated learning algorithm [15, 17, 31] as follows. Each epoch can be further divided into a number of "training iterations", and each training iteration consists of three steps: every participating device downloading the latest aggregated model and the corresponding gradient from the server, then updating the model using all the raw data of the current epoch on the local device, and finally uploading the updated model and the corresponding gradient to the server for aggregation. While not every device in the system may be available in epoch t, the participants are only chosen from the set of the available devices denoted as $\mathcal{S}_t$, $\forall t \in \mathcal{T}$. We denote by $D_{t,i}, \forall i \in \mathcal{S}_t, t \in \mathcal{T}$ the set of the raw data samples generated on device i in epoch t [15]. We also denote by κ_t the number of the training iterations conducted in epoch t.

0) ***Control Decisions***: We introduce our control decisions first, because our models rely on these decisions. We use $\phi_{t,i} \in \{1, 0\}, \forall i \in \mathcal{S}_t, \forall t \in \mathcal{T}$ to denote whether, out of the set S_t of the available devices, we select device i for participating in the model training process in epoch t. We use $\vartheta_t \in [0, 1), \forall t \in \mathcal{T}$ to denote the maximum local "convergence accuracy" across all training iterations on all devices in epoch t, which will be further elaborated later. We use ϑ_t to control the number of training iterations conducted in epoch t.

1) ***Loss***: We start to describe our federated learning models. For each sample $< \boldsymbol{x}_d, y_d >, \forall d \in D_{t,i}$, a convex loss [10], without loss of generality, $\mathcal{L}(\mathcal{H}(\boldsymbol{x}_d; \boldsymbol{w}), y_d)$ is used to measure the performance of a predictor $\mathcal{H}(\boldsymbol{x}_d; \boldsymbol{w})$, where $\boldsymbol{w}$ is the parameter of the predictor $\mathcal{H}$, or in other words, $\boldsymbol{w}$ is the "model" to be trained. With the functions $\mathcal{L}(\cdot)$ and $\mathcal{H}(\cdot)$, the average loss of the data on device i in epoch t is

$$\mathcal{F}_{t,i}(\boldsymbol{w}) = \tfrac{1}{|D_{t,i}|} \textstyle\sum_{d \in D_{t,i}} \mathcal{L}(\mathcal{H}(\boldsymbol{x}_d; \boldsymbol{w}), y_d).$$

The average loss of the data on all devices [9] in epoch t is

$$\mathcal{F}_t(\boldsymbol{w}) = \textstyle\sum_{i \in \mathcal{S}_t} \{ \tfrac{|D_{t,i}|}{|D_t|} \cdot \mathcal{F}_{t,i}(\boldsymbol{w}) \},$$

where $|D_t| = \sum_i |D_{t,i}|$ is the size of the dataset in epoch t.

2) ***Training on Device***: For each iteration $j \in [1, \kappa_t]$, the participant device i updates the local model as follows [32]:

$$\boldsymbol{w}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}} = \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}} + \boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}} = \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}} + \arg\min_{\boldsymbol{\rho}} \mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}),$$

where $\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}}$ is the aggregated model downloaded from the server; $\boldsymbol{w}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}$ is the local model on device i after the update; and $\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\cdot)$ is a dedicated function.[1] In order to construct $\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\cdot)$, $\mathcal{J}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}})$ also needs to be downloaded from the server. Note that we have $\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(0)}} = \boldsymbol{w}_{\boldsymbol{t-1}}^{\boldsymbol{(\kappa_t)}}$, $\forall t$.

To solve for $\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}$, one can choose to use a gradient-based process. For example:

$$\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j),(k)}} = \boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j),(k-1)}} - \delta \nabla \mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j),(k-1)}}),$$

where δ can be a constant step size and $\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j),(0)}}$ can be set an initial value $\mathbf{0}$. Updating $\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}$ as the above for tens of times [7, 14] could make the performance acceptable. Here, we assume the maximum value of k is a pre-determined constant.

3) ***Aggregation on Server***: At the end of each iteration $j \in [1, \kappa_t]$, both $\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}$ and $\nabla \mathcal{F}_{t,i}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}})$ need to be uploaded to the server for aggregation [15]:

$$\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}} = \boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}} + \tfrac{1}{|\mathcal{S}_t|} \textstyle\sum_i \{ \phi_{t,i} \cdot \boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}} \},$$
$$\mathcal{J}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}}) = \tfrac{1}{|\mathcal{S}_t|} \textstyle\sum_i \nabla \mathcal{F}_{t,i}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}}).$$

Meanwhile, the average loss on all the data of the participants is measured by

$$\tilde{\mathcal{F}}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}}) = \textstyle\sum_{i \in \mathcal{S}_t} \{ \tfrac{\phi_{t,i} \cdot |D_{t,i}|}{|D_t|} \cdot \mathcal{F}_{t,i}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j)}}) \}.$$

Model Convergence: We then consider the convergence of both the local model on the device and the aggregated model on the server in each epoch. Analogous to previous research [14, 17, 33], assuming $\mathcal{F}_{t,i}$ is Lipschitz continuous and strongly convex, for each epoch t, the convergence can be represented as

$$\mathcal{G}_{t,i}^{\boldsymbol{(\kappa_t)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(\kappa_t)}}) - \mathcal{G}_{t,i}^{\boldsymbol{(\kappa_t)}*} \leq \theta_{t,i}^{(\kappa_t)} [\mathcal{G}_{t,i}^{\boldsymbol{(\kappa_t)}}(\mathbf{0}) - \mathcal{G}_{t,i}^{\boldsymbol{(\kappa_t)}*}], \tag{1}$$
$$\tilde{\mathcal{F}}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(\kappa_t)}}) \leq \tilde{\mathcal{F}}_t^* + \varepsilon_0 [\tilde{\mathcal{F}}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(0)}}) - \tilde{\mathcal{F}}_t^*] = \varepsilon, \tag{2}$$

where $\theta_{t,i}^{(\kappa_t)}$ is the "convergence accuracy" of the local training on device i, and ε_0 is the "convergence accuracy" of the aggregated global loss. $\mathcal{G}_{t,i}^{\boldsymbol{(j)}*}$ and $\tilde{\mathcal{F}}_t^*$ are the local optimum and the global optimum, respectively. Here, we also introduce ε as the desired upper bound of the global loss [8].

We connect the convergence accuracies to the number of gradient steps in each training iteration and the number of training iterations. Recall that in each training iteration j we run a pre-specified number of gradient steps. We let $\theta_{t,i}^{(j)} = \mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}) / \mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\mathbf{0})$ after finishing the j^{th} iteration. With $\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}) \leq \mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\mathbf{0})$, we can write

$$\forall t, i, j : \tfrac{\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}}) - \mathcal{G}_{t,i}^{\boldsymbol{(j)}*}}{\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\mathbf{0}) - \mathcal{G}_{t,i}^{\boldsymbol{(j)}*}} \leq \tfrac{\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}_{\boldsymbol{t,i}}^{\boldsymbol{(j)}})}{\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\mathbf{0})} = \theta_{t,i}^{(j)} \leq \vartheta_t,$$
$$\vartheta_t = \max\nolimits_i \theta_{t,i}, \quad \theta_{t,i} = \max\nolimits_j \theta_{t,i}^{(j)}, \quad \vartheta_t, \theta_{t,i}, \theta_{t,i}^{(j)} \in [0, 1).$$

Thus, ϑ_t is the maximum local convergence accuracy across all training iterations on all devices in epoch t. Also, similar to [17, 34], in order to reach the global accuracy ε_0, we need to conduct at least $\kappa_t(\vartheta_t, \varepsilon_0)$ training iterations, where

$$\kappa_t(\vartheta_t, \varepsilon_0) = \tfrac{\mathcal{O}(log(1/\varepsilon_0))}{1 - \vartheta_t}$$

Given ε_0, we can just control ϑ_t in order to control the number of training iterations conducted.

Resource Consumption: In each training iteration of each epoch, each device downloads the model and the gradient once and also uploads the updated model and the corresponding gradient for another time. Thus, the resource consumption for model transferences is $2 \cdot \frac{\mathscr{D}}{b_t}$ per training iteration, where $\mathscr{D}$ is the sum of the model size and the gradient size, and b_t is the available network bandwidth in epoch t. We also denote by $\alpha_{t,i}$ the computational resource consumption by training one data sample on device i in epoch t. Note that, as each training iteration takes a pre-specified number of gradient steps, $\alpha_{t,i}$ thus represents the computational resource consumption by training one data sample for the pre-specified number of times. The resource consumption for computation is then $\alpha_{t,i}|D_{t,i}|$, as training goes through the entire dataset $D_{t,i}$ in each training iteration. Overall, we have the total resource consumption of device i in each training iteration of epoch t as

$$\phi_{t,i} \cdot (\frac{2\mathscr{D}}{b_t} + \alpha_{t,i} \cdot |D_{t,i}|).$$

B. Problem Formulation

With the system models, we formulate the following optimization problem for minimizing the overall resource used:

$$\begin{array}{lll}
\min & \displaystyle\sum_{t \in \mathcal{T}} \big\{ \kappa_t(\vartheta_t, \varepsilon_0) \cdot \sum_{i \in \mathcal{S}_t} \{ \phi_{t,i} \cdot (\frac{2\mathscr{D}}{b_t} + \alpha_{t,i} \cdot |D_{t,i}|) \} \big\} & \\
s.t. & \displaystyle\sum_{i \in \mathcal{S}_t} \phi_{t,i} \leq m, \quad \forall t \in \mathcal{T}, & (3) \\
 & \theta_{t,i} \cdot \phi_{t,i} \leq \vartheta_t, \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{S}_t, & (4) \\
 & \displaystyle\sum_{t \in \mathcal{T}} \big\{ \mathcal{F}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(\kappa_t)}}) - \varepsilon \big\} \leq 0, & (5) \\
var. & \phi_{t,i} \in \{0, 1\}, \vartheta_t \in [0, 1), \forall t \in \mathcal{T}, \forall i \in \mathcal{S}_t. & (6)
\end{array}$$

[1] We have $\mathcal{G}_{t,i}^{\boldsymbol{(j)}}(\boldsymbol{\rho}) = \mathcal{F}_{t,i}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}} + \boldsymbol{\rho}) - (\nabla \mathcal{F}_{t,i}(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}}) - \xi_1 \mathcal{J}_t(\boldsymbol{w}_{\boldsymbol{t}}^{\boldsymbol{(j-1)}}))^\top \boldsymbol{\rho}$ $+ \frac{\xi_2}{2} ||\boldsymbol{\rho}||^2$, where ξ_1 and ξ_2 are non-negative constants, and $\mathcal{J}_t(\cdot)$ is produced by the aggregation on the server, as defined later.

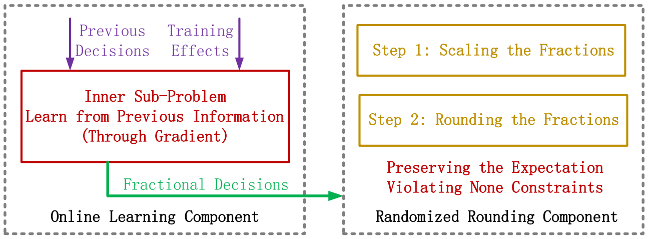

Previous Decisions
Training Effects
Inner Sub-Problem
Learn from Previous Information
(Through Gradient)
Fractional Decisions
Online Learning Component
Step 1: Scaling the Fractions
Step 2: Rounding the Fractions
Preserving the Expectation
Violating None Constraints
Randomized Rounding Component

Fig. 2: Structure of proposed online schema

```
Algorithm 1 Online Learning Algorithm
Input: Initial decision $\tilde{\mathbf{I}}_1$; Initial update parameter $\boldsymbol{\lambda_1}=\mathbf{0}$;
       Proper step sizes $\alpha$ and $\mu$.
for $t=1,2,...,T$ do
    Obtain $\mathbf{I}_t$ by invoking Algorithm 2 on $\tilde{\mathbf{I}}_t$.
    Conduct federated learning based on $\mathbf{I}_t$.
    Observe current cost $f_t(\mathbf{I}_t)$ and constraint $\mathbf{g}_t(\mathbf{I}_t)$.
    Update $\boldsymbol{\lambda_{t+1}}$ according to (18).
    Update $\tilde{\mathbf{I}}_{t+1}$ according to (16).
end for
```

```
Algorithm 2 Randomized Rounding Algorithm
Input: Fractional decision $\tilde{\mathbf{I}}_t\in\tilde{\mathcal{X}}$.
    // Step 1 separates $\{\tilde{\phi}_{t,i}\}$ and $\sigma_t$, as well as ensures
           the sum of all columns in $\mathbf{U}_t$ is an integer.
$\mathbf{U}_t=[\tilde{\mathbf{I}}_{1,t},...,\tilde{\mathbf{I}}_{N,t}]^\top$
$k=\mathbf{1}^\top\mathbf{U}_t, \gamma_1=1-(k-\lfloor k\rfloor)/k, \gamma_2=1+(\lceil k\rceil-k)/k.$
$\mathbf{V}_t^\top=\begin{cases}[\gamma_1 U_{1,t},...,\gamma_1 U_{N,t}]\ \textit{with prob.}\ \lceil k\rceil-k;\\ [\gamma_2 U_{1,t},...,\gamma_2 U_{N,t}]\ \textit{with prob.}\ k-\lfloor k\rfloor.\end{cases}$
    // Step 2 ensures each column of $\mathbf{V}_t$ is an integer.
while $V_{i,t}\in(0,1)\wedge V_{j,t}\in(0,1)$ do
    $\theta_1=\min\{1-V_{i,t},V_{j,t}\}$, $\theta_2=\min\{V_{i,t},1-V_{j,t}\}$.
    $(V_{i,t},V_{j,t})=\begin{cases}(V_{i,t}+\theta_1,V_{j,t}-\theta_1)\ \textit{with prob.}\ \frac{\theta_2}{\theta_1+\theta_2};\\ (V_{i,t}-\theta_2,V_{j,t}+\theta_2)\ \textit{with prob.}\ \frac{\theta_1}{\theta_1+\theta_2}.\end{cases}$
end while
Return $\mathbf{I}_t=[V_{1,t},...,V_{N,t},\sigma_t]^\top$.
```

Constraint (3) ensures that the number of concurrent model transference obeys the network capacity. Constraints (4) and (5) guarantee the convergence of both the local models and the aggregated model. Note that we train the model based on the loss $\tilde{\mathcal{F}}_t$ of the selected devices, but we ensure that the trained model, when applied to all the data in the system, the loss $\mathcal{F}_t$ is bounded. Constraint (6) enforces the variables' domains.

To simplify the problem presentation, we aggregate the decision variables as $\mathbf{I}_t=[\phi_{t,1},...,\phi_{t,N},\sigma_t]^\top$, where $\sigma_t=\frac{1}{1-\vartheta_t}$. In fact, we introduce some new notations:

$$f_t(\mathbf{I}_t)=\sigma_t\cdot\textstyle\sum_{i\in\mathcal{S}_t}\{\phi_{t,i}^2\cdot(\frac{2\mathscr{D}}{b_t}+\alpha_{t,i}\cdot|D_{t,i}|)\},\tag{7}$$

$$g_t^0=\mathcal{F}_t(\boldsymbol{w}_t^{(\boldsymbol{\kappa_t-1})}+\tfrac{1}{|\mathcal{S}_t|}\textstyle\sum_i\{\phi_{t,i}\cdot\boldsymbol{\rho}_{t,i}^{(\boldsymbol{\kappa_t})}\})-\varepsilon,\tag{8}$$

$$g_t^i=\theta_{t,i}\cdot\phi_{t,i}\cdot\sigma_t-\sigma_t+1,\quad\forall i\in\mathcal{S}_t,\tag{9}$$

$$\mathbf{g}_t(\mathbf{I}_t)=[g_t^0,g_t^1,...,g_t^N]^\top,\tag{10}$$

$$h(\mathbf{I}_t)=\textstyle\sum_{i\in\mathcal{S}_t}\phi_{t,i}-m,\tag{11}$$

where f_t corresponds to the objective function, h corresponds to Constraint (3). Then, the problem can be reformulated:

$$\begin{aligned}\min\quad&\sum_{t\in\mathcal{T}}f_t(\mathbf{I}_t)\\ s.t.\quad&\sum_{t\in\mathcal{T}}\mathbf{g}_t(\mathbf{I}_t)\preceq\mathbf{0}, &(12)\\ &h(\mathbf{I}_t)\leq 0, &(13)\\ var.\quad&\mathbf{I}_t\in\mathcal{X}=\{\mathbf{I}_t|\phi_{t,i}\in\{0,1\},\sigma_t\geq 1\}. &(14)\end{aligned}$$

We highlight that we aim to solve our problem in an *online* manner, i.e., making decisions on the fly as the inputs reveal themselves as time goes. In particular, $\{\boldsymbol{w}_t^{(\boldsymbol{\kappa_t-1})},\boldsymbol{\rho}_{t,i}^{(\boldsymbol{\kappa_t})},\theta_{t,i}\}$ disclose themselves to us only after we make the decision $\mathbf{I}_t$ at t, which is essentially the setting of "online learning". Due to such obliviousness to the inputs, any online algorithm can potentially excessively violate the constraints, especially (4) and (5), which are supposed to be satisfied for each single epoch. We will need to overcome this algorithmic challenge when designing effective online algorithms.

III. Online Algorithm Design

We design a novel online schema with two algorithmic components, as exhibited in Fig. 2. The online learning component explores rectified descent-ascent steps alternately to make fractional decisions based on previous information only, which essentially incorporates the previous learning effects. The randomized rounding component explores a compensation strategy to convert fractional decisions into integers in a randomized manner while preserving the expectation and violating none of our problem's constraints.

A. Online Learning Algorithm

We design our online algorithm through updating the primal variables and the dual variables alternately using carefully-designed, *rectified* descent-ascent steps. Note that we need to ensure that such descent-ascent iterations do not reply on unknown or future information, but only exploit observable information so far as inputs.

First, we know that solving the convex problem of

$$\min\textstyle\sum_t f_t(\tilde{\mathbf{I}}_t)\,,s.t.\,\sum_t\mathbf{g}_t(\tilde{\mathbf{I}}_t)\preceq\mathbf{0},h(\tilde{\mathbf{I}}_t)\leq 0,\tilde{\mathbf{I}}_t\in\tilde{\mathcal{X}},$$

where $\tilde{\mathbf{I}}_t$ represents the fractional variable[2], is equivalent to solving the corresponding convex-concave problem of

$$\min\nolimits_{\tilde{\mathbf{I}}_t}\max\nolimits_{\boldsymbol{\lambda_t}}\textstyle\sum_t\Big(f_t(\tilde{\mathbf{I}}_t)+\boldsymbol{\lambda_t}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_t)\Big),\,s.t.\,h(\tilde{\mathbf{I}}_t)\leq 0,\tilde{\mathbf{I}}_t\in\tilde{\mathcal{X}},$$

where $\boldsymbol{\lambda_t}\succeq 0$ is the Lagrange multiplier or dual variable. At t, we can consider the following, in an online manner:

$$\mathcal{L}_t(\tilde{\mathbf{I}},\boldsymbol{\lambda}):=f_t(\tilde{\mathbf{I}})+\boldsymbol{\lambda}^\top\mathbf{g}_t(\tilde{\mathbf{I}}).\tag{15}$$

Second, we design our online learning algorithm as follows. We alternate between minimizing $\mathcal{L}_t(\tilde{\mathbf{I}},\boldsymbol{\lambda_{t+1}})$ with respect

[2] $\tilde{\mathbf{I}}_t$ and its corresponding domain $\tilde{\mathcal{X}}$ are defined in real domain while decision $\mathbf{I}_t$ and $\mathcal{X}$ are defined in integral domain.

to the primal variable $\tilde{\mathbf{I}}$ via a *rectified* descent step and maximizing $\mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda})$ with respect to the dual variable $\boldsymbol{\lambda}$ via a standard dual ascent step. Specifically, in epoch $t+1$, we solve the following problem of

$$\min_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{\boldsymbol{t+1}}^\top \mathbf{g}_t(\tilde{\mathbf{I}}) + \frac{||\tilde{\mathbf{I}} - \tilde{\mathbf{I}}_t||^2}{2\alpha}, \quad (16)$$

$$s.t. \;\; h(\tilde{\mathbf{I}}) \leq 0, \quad (17)$$

to get $\tilde{\mathbf{I}}_{t+1}$ by using the previous fractional decision $\tilde{\mathbf{I}}_t$, where $\nabla f_t(\tilde{\mathbf{I}}_t)$ is the gradient of primal objective $f_t(\cdot)$ at $\tilde{\mathbf{I}} = \tilde{\mathbf{I}}_t$, and α is a positive step size. We highlight that the first two terms form an approximation to $\mathcal{L}_t(\tilde{\mathbf{I}}, \boldsymbol{\lambda}_{\boldsymbol{t+1}})$, and the last term is a proximal term. We also update the dual variable as

$$\boldsymbol{\lambda}_{\boldsymbol{t+1}} = [\boldsymbol{\lambda}_{\boldsymbol{t}} + \mu \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda}_{\boldsymbol{t}})]^+ = [\boldsymbol{\lambda}_{\boldsymbol{t}} + \mu \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+, \quad (18)$$

where μ is also a positive step size, and $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda}_{\boldsymbol{t}}) = \mathbf{g}_t(\tilde{\mathbf{I}}_t)$ is the gradient of $\mathcal{L}_t(\tilde{\mathbf{I}}_t, \cdot)$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{\boldsymbol{t}}$. Obviously, at $t+1$, updating $\boldsymbol{\lambda}_{\boldsymbol{t+1}}$ as in (18) and updating $\tilde{\mathbf{I}}_{t+1}$ as in (16) only requires information from t, which is the key feature of our algorithm. Note that (16) is not a standard but a *rectified* descent step that directly penalizes the constraint violation, which facilitates our performance analysis as shown later.

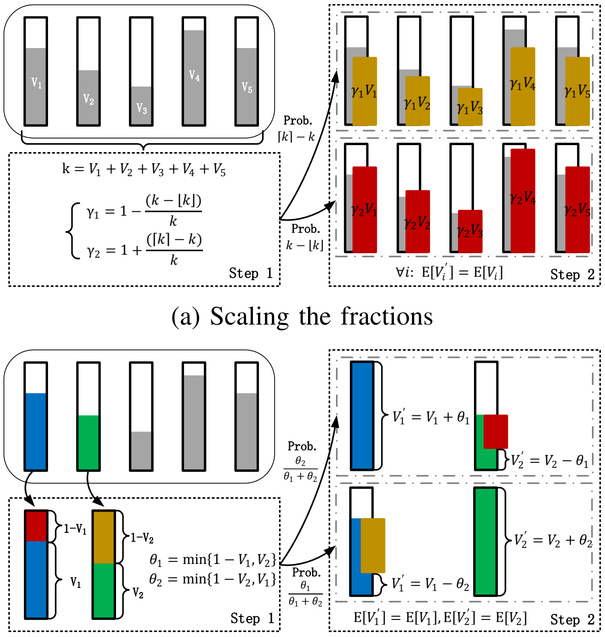
Our online component is exhibited as Algorithm 1. The dual update of $\boldsymbol{\lambda}_{\boldsymbol{t+1}}$ and the primal update of $\tilde{\mathbf{I}}_{t+1}$ are in Lines 5 and 6, respectively. In order to convert the fractional decisions $\tilde{\mathbf{I}}_t$, $\forall t$ into integers, we propose a rounding component as Algorithm 2, which is described next.

B. Randomized Rounding Algorithm

Our randomized rounding algorithm proceeds in two steps as shown in Algorithm 2 and Fig. 3. The first step separates $\tilde{\mathbf{I}}_t$ into two parts: $\{\tilde{\phi}_{t,i}\}$ and $\sigma_t = \frac{1}{1-\vartheta_t}$, because the domain of σ_t has already been reals, and there is no need to round σ_t. We point out that both of these two steps keep the expectation of the decisions unchanged without violating $h(\cdot) \leq 0$.

Scaling the Fractions: The first step further adjusts $\{\tilde{\phi}_{t,i}\}$, i.e., $\mathbf{U}_t$, in a randomized manner, ensuring that the sum of the adjusted values equals an integer and that the expectation of each adjusted value equals its corresponding value before such adjustment, as shown in Lines 2 to 3, and Fig 3(a). This step either decreases each column of $\mathbf{U}_t$ by multiplying $\gamma_1 < 1$ so that the sum of all columns in $\mathbf{V}_t$ is $\lfloor \mathbf{1}^\top \mathbf{U}_t \rfloor$, or increases each column of $\mathbf{U}_t$ by multiplying $\gamma_2 > 1$ so that the sum of all columns in $\mathbf{V}_t$ is $\lceil \mathbf{1}^\top \mathbf{U}_t \rceil$. The probabilities of taking these two choices are $\lceil k \rceil - k$ and $k - \lfloor k \rfloor$, respectively, which can thus ensure $E_{\tilde{\phi}}[\mathbf{V}_t] = \mathbf{U}_t$. We should mention here that the sum of all columns increases to at most $\lceil k \rceil$ such that $\sum_{i \in \mathcal{S}_t} \phi_{t,i} \leq m$ would not be violated, i.e., $h(\cdot) \leq 0$.

Rounding the Fractions: The second step rounds $\mathbf{V}_t$ into integers, also in a randomized manner, while guaranteeing that 1) the sum of all the values stay unchanged after rounding, 2) each value is an integer after rounding, and 3) the expectation of each randomized integer equals its corresponding fractional value before rounding, as shown in Lines 4 through 7, and Fig 3(b). We use the values of $\mathbf{V}_t$ as the probability to round the columns in pairs into integers, while letting the two fractions compensate each other. Since the sum of all columns is an integer beforehand as a result of the previous step, $\mathbf{V}_t$ can be guaranteed as a vector that only contains 0 and 1 after the loop. The complexity of the loop reaches $O(N^2)$ according to [35]. Consequently, for the final control decisions $\mathbf{I}_t$, we have $E_{\tilde{\phi}}[\mathbf{I}_t] = \tilde{\mathbf{I}}_t$, which is necessary for our performance analysis as described next.


V1
V2
V3
V4
V5
k = $V_1 + V_2 + V_3 + V_4 + V_5$
$\gamma_1 = 1 - \frac{(k - \lfloor k \rfloor)}{k}$
$\gamma_2 = 1 + \frac{(\lceil k \rceil - k)}{k}$
Step 1
Prob. $\lceil k \rceil - k$
Prob. $k - \lfloor k \rfloor$
$\gamma_1 V_1$
$\gamma_1 V_2$
$\gamma_1 V_3$
$\gamma_1 V_4$
$\gamma_1 V_5$
$\gamma_2 V_1$
$\gamma_2 V_2$
$\gamma_2 V_3$
$\gamma_2 V_4$
$\gamma_2 V_5$
$\forall i$: $E[V_i^{'}] = E[V_i]$
Step 2
1−V1
V1
1−V2
V2
$\theta_1 = \min\{1 - V_1, V_2\}$
$\theta_2 = \min\{1 - V_2, V_1\}$
Step 1
Prob. $\frac{\theta_2}{\theta_1 + \theta_2}$
Prob. $\frac{\theta_1}{\theta_1 + \theta_2}$
$V_1^{'} = V_1 + \theta_1$
$V_2^{'} = V_2 - \theta_1$
$V_2^{'} = V_2 + \theta_2$
$V_1^{'} = V_1 - \theta_2$
$E[V_1^{'}] = E[V_1], E[V_2^{'}] = E[V_2]$
Step 2

(a) Scaling the fractions

(b) Rounding the fractions

Fig. 3: Illustration of randomized rounding

IV. Performance Analysis

A. Performance Metrics

We focus on two metrics that measure the performance of an online algorithm: *dynamic regret* [28, 29] and *dynamic fit* [24, 30]. The ultimate goal of our performance analysis is to exhibit that both of the dynamic regret and the dynamic fit for our online algorithms only grow *sub-linearly* along with time.

Dynamic Regret: The dynamic regret is defined as the difference between the long-term objective function value of the online decisions $\{\mathbf{I}_t\}$ that are made without knowing the inputs in each epoch and the long-term objective function value of the optimal decisions $\{\mathbf{I}_t^*\}$ that optimize the objective function in each epoch by observing the corresponding inputs. Both integral and real domains are considered, namely:

$$Reg_T^d := E[\textstyle\sum_{t=1}^T f_t(\mathbf{I}_t)] - \sum_{t=1}^T f_t(\mathbf{I}_t^*), \quad (19a)$$

$$\mathbf{I}_t^* \in arg\min_{\mathbf{I} \in \mathcal{X}} f_t(\mathbf{I}), \quad s.t. \; \mathbf{g}_t(\mathbf{I}_t^*) \preceq \mathbf{0}, h(\mathbf{I}_t^*) \leq 0,$$

$$\widetilde{Reg}_T^d := \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*), \quad (19b)$$

$$\tilde{\mathbf{I}}_t^* \in arg\min_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} f_t(\tilde{\mathbf{I}}), \quad s.t. \; \mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \preceq \mathbf{0}, h(\tilde{\mathbf{I}}_t^*) \leq 0,$$

where the expectation is introduced due to the randomized rounding component of our proposed online schema.

Dynamic Fit: The dynamic fit is defined as the norm of the cumulative violation of the long-term constraints, incurred by the online decisions $\{\mathbf{I}_t\}$. In order to measure the deviation from 0 incurred by the long-term constraints, according to Constraint (12), we use the function of $[\cdot]^+ = max\{\cdot, 0\}$ to capture such violation, similar to [24]. Also, both of the integral and real domains are considered as follows:

$$Fit_T^d := ||\big[E[\textstyle\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]\big]^+||, \forall t : \mathbf{I}_t \in \mathcal{X}, \tag{20a}$$

$$\widetilde{Fit}_T^d := ||[\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||, \forall t : \tilde{\mathbf{I}}_t \in \tilde{\mathcal{X}}. \tag{20b}$$

B. Regret and Fit Analysis

Roadmap: We firstly present Lemmas 1 and 2, based on which we connect the dynamic regret and the dynamic fit in the integral domain to those in the real domain. Afterwards, we bound the dynamic fit in Theorem 1 and bound the dynamic regret in Theorem 2. Finally, we show in Corollary 1 that by properly choosing the step sizes we can concretize these bounds into sub-linear functions of time.

Lemma 1. *The relationship on dynamic regret and dynamic fit in the domain of integers and reals can be illustrated as*

$$Reg_T^d \leq \widetilde{Reg}_T^d, \quad Fit_T^d \leq \widetilde{Fit}_T^d + M\sigma_\beta^\beta, \tag{21}$$

where M and σ_β are constants from Jensen Gap [36].

Proof. See Appendix A. □

Assumptions: Before proceeding further, we introduce the following assumptions to facilitate our analysis. These assumptions are very common [37–40], and easy to be satisfied.

Assumption 1: $\forall t$, $f_t(\tilde{\mathbf{I}})$ has bounded gradients in $\tilde{\mathcal{X}}$, i.e., $||\nabla f_t(\tilde{\mathbf{I}})|| \leq F, \forall \tilde{\mathbf{I}} \in \tilde{\mathcal{X}}$; and $\mathbf{g}_t(\tilde{\mathbf{I}})$ is bounded in $\tilde{\mathcal{X}}$, i.e., $||\mathbf{g}_t(\tilde{\mathbf{I}})|| \leq G, \forall \tilde{\mathbf{I}} \in \tilde{\mathcal{X}}$.

Assumption 2: There exists a constant $\varepsilon > 0$, and an interior point $\hat{\mathbf{I}}_t \in \tilde{\mathcal{X}}$ such that $\forall t, \mathbf{g}_t(\hat{\mathbf{I}}_t) \preceq -\varepsilon \mathbf{1}$.

Assumption 3: The slack constant ε in Assumption 2 satisfies $\varepsilon > \overline{V}(\mathbf{g})$, where the point-wise maximal variation of the consecutive constraints is defined as

$$\overline{V}(g) := \max_t \max_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||. \tag{22}$$

Assumption 1 bounds both primal and dual gradients per slot, which is a very common assumption [37, 38]. Assumption 2 is the Slater's condition [39], which guarantees the existence of a bounded optimal Lagrange multiplier. Assumption 3 implies that the slack constant ε is larger than the maximal variation of the constraints, requiring $\min_t \max_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} [-\mathbf{g}_t(\tilde{\mathbf{I}})]^+ > \max_t \max_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||$, which is valid when the feasible region defined by $\mathbf{g}_t(\tilde{\mathbf{I}}) \preceq \mathbf{0}$ is large enough, or the trajectory of $g_t(\tilde{\mathbf{I}})$ is smooth enough across time [38, 40].

Lemma 2. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda_1} = \mathbf{0}$, we have the following:*

$$\tfrac{(||\boldsymbol{\lambda_{t+1}}||^2 - ||\boldsymbol{\lambda_t}||^2)}{2} \leq \mu \boldsymbol{\lambda_t}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t) + \tfrac{\mu^2}{2} ||\mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2, \tag{23a}$$

$$\forall t, ||\boldsymbol{\lambda_t}|| \leq ||\bar{\boldsymbol{\lambda}}|| := \mu G + \tfrac{2FR + R^2/(2\alpha) + (\mu G^2)/2}{\varepsilon - \overline{V}(\mathbf{g})}. \tag{23b}$$

Proof. See Appendix B. □

TABLE II: Summary of Traces Used

Content	Description
Available devices	Numbers measured every quarter from [12]
Global convergence	Global convergence changes with iterations [8]
Local convergence	Local convergence changes with iterations [14]
Participant numbers	Convergence changes with participants [8]
Network traffic	Available bandwidth for model updates [12]

Theorem 1. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda_1} = \mathbf{0}$, the integral dynamic fit in (20a) is upper-bounded:*

$$\widetilde{Fit}_T^d \leq \tfrac{\boldsymbol{\lambda_{T+1}}}{\mu} \leq \tfrac{||\bar{\boldsymbol{\lambda}}||}{\mu}. \tag{24}$$

Proof. See Appendix C. □

Theorem 2. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda_1} = \mathbf{0}$, the integral dynamic regret in (19a) is upper-bounded:*

$$Reg_T^d \leq \widetilde{Reg}_T^d \leq \mathcal{R}_T,$$

where

$$\mathcal{R}_T = \frac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \frac{\alpha F^2 T}{2} + \frac{\mu G^2 (T+1)}{2} + \frac{R^2}{2\alpha} + ||\bar{\boldsymbol{\lambda}}|| V(\{\mathbf{g}_t\}_{t=1}^T), \tag{25a}$$

$$V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T) := \textstyle\sum_{t=1}^T \underbrace{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*||}_{V(\tilde{\mathbf{I}}_t^*)}, \tag{25b}$$

$$V(\{\mathbf{g}_t\}_{t=1}^T) := \textstyle\sum_{t=1}^T \underbrace{\max_{\tilde{\mathbf{I}} \in \tilde{\boldsymbol{\mathcal{X}}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||}_{V(\mathbf{g}_t)}. \tag{25c}$$

Proof. See Appendix D. □

Corollary 1. *Under previous assumptions and initialization, dynamic regret and fit are bounded by controlling step sizes:*

$$\alpha = \mu = \max\{\sqrt{\tfrac{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{T}}, \sqrt{\tfrac{V(\{\mathbf{g}_t\}_{t=1}^T)}{T}}\},$$

$$Reg_T^d = \mathcal{O}(\max\{\sqrt{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)T}, \sqrt{V(\{\mathbf{g}_t\}_{t=1}^T)T}\}),$$

$$Fit_T^d \leq \tfrac{||\bar{\boldsymbol{\lambda}}||}{\mu} + M\sigma_\beta^\beta = \mathcal{O}(\max\{\tfrac{T}{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}, \tfrac{T}{V(\{\mathbf{g}_t\}_{t=1}^T)}\}) + M\sigma_\beta^\beta.$$

Following this corollary, if we set the step sizes as

$$\alpha = \mu = \mathcal{O}(T^{-\frac{1}{3}}), \tag{26}$$

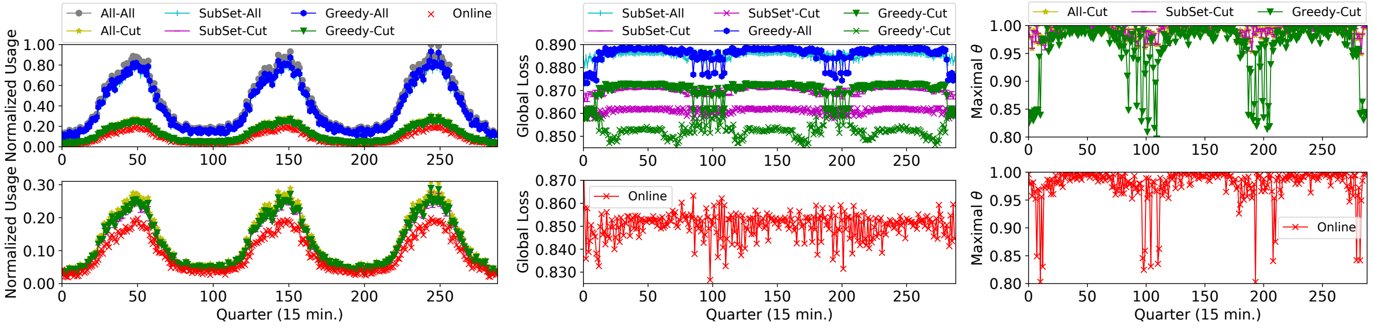
then the dynamic regret and the dynamic fit can be bounded, respectively, by

$$Reg_T^d = \mathcal{O}(\max\{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)T^{\frac{1}{3}}, V(\{\mathbf{g}_t\}_{t=1}^T)T^{\frac{1}{3}}, T^{\frac{2}{3}}\}),$$
$$Fit_T^d = \mathcal{O}(T^{\frac{2}{3}}) + M\sigma_\beta^\beta. \tag{27}$$

V. Evaluation

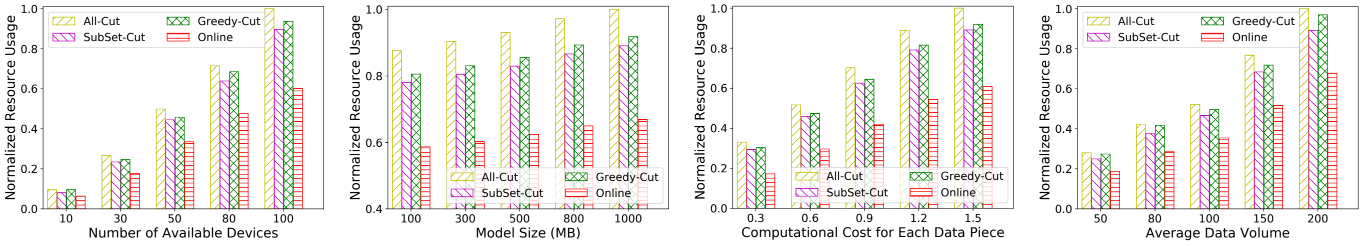
A. Data and Settings

Federated Learning Workload: We use a series of traces to mimic the online training process of federated leaning, as shown in Table II. More specifically, we first obtain the dynamic numbers of available devices over a two days period for a US-centric population [12] as the workload trace, which is measured for every quarter (i.e., every 15 minutes). After



(a) Time-varying resource usage (b) Time-varying global convergence (c) Time-varying local convergence

Fig. 4: Effectiveness of proposed online schema



(a) Various workloads (b) Various model sizes (c) Various computational costs (d) Various data volumes

Fig. 5: Scalability of proposed online schema

that, we obtain the changes on the global convergence for a set of participants during the iterative training of federated learning [8]. Then, we obtain the changes on the local convergence for a single participant during the iterative training of federated learning [14]. Next, we obtain the changes on the network traffic from [12]. At last, we obtain the changes on the global convergence from [8] when the numbers of participants are different. Besides, the model size used in our simulation is 500MB according to the typical size of machine learning models as mentioned in [41].

Algorithms and Metrics: Except for the online schema we proposed with the desired step sizes $\alpha = \mu = 0.15$ based on the corollary mentioned before, we also evaluate our schema under various scenarios and compare it with other algorithms:

- **All-All** selects all of the available devices into training for federated learning and iteratively trains the model until the number of iterations within each epoch reaches to the maximum, i.e., 2500 according to the trace [8].
- **All-Cut** selects all of the available devices into training. However, it terminates the training until the global convergence reaches to a realistic threshold within each epoch, i.e., 0.88, also according to the Google trace.
- **SubSet-All** chooses a subset of available devices based on a threshold so as to ensure that the participant ratio is larger than the threshold. Besides, it terminates the training until the iteration number reaches to the maximum.
- **SubSet-Cut** chooses a subset of available devices so as to ensure that the participant ratio is larger than the given threshold, until the desired global convergence is reached.
- **Greedy-All** selects those participants, whose local convergence performances within previous epoch are less than a given threshold. It continuously trains models until the iterations reach to the maximum.
- **Greedy-Cut** selects participants according to the observable local convergence from previous epoch and a given threshold. Furthermore, it continuously trains the models until the threshold of the global convergence is reached.

We should mention here that all of these algorithms compared would not obtain the actual set of available devices as well as the actual local convergence performances beforehand, since the training scenario is online. Our primary metric contains both resource usage and the global convergence for all of these algorithms. Note that the key step of our online schema, i.e., solving subproblem (16), is conducted by using mature optimization tools: AMPL [42, 43] and IPOPT [44].

B. Evaluation Results

Time-Varying Resource Usage: Fig. 4(a) illustrates the time-varying resource usage during federated learning. Those strategies spend much on training in federated learning, which iteratively train the models until the iteration number reaches to the maximum, i.e., those *-All ones. In contrast, those strategies with reduced number of iterations in each epoch, i.e., those *-Cut ones as well as our online schema, save the resource usage from the training iterations and scarify only a little on the global convergence, since they terminate the training immediately if the global convergence has been reached. Those strategies with reduced number of iterations spend at most 30% resource usage compared with others, which train the models to the maximal iteration number, i.e., nearly 2500 iterations for each epoch. Moreover, the online schema always performs the best compared with others, decreasing at least 25% resource usage on average, because it chooses suitable participants dynamically according to previous local convergence performances. As shown in Fig. 4(c), our online schema dynamically changes the maximum local convergence accuracy guided by previous decisions and training results, and is willing to select those devices with better local convergence performances and lower resource usage. Unlike the Greedy one choosing devices based on a fixed threshold heuristically, our online schema adjusts the maximum local convergence accuracy for better usage of resources.

Convergence of Federated Learning: As illustrated in Fig. 4(b), although our online schema actually decreases a little on the global convergence, the average reduction is only 4%. Except for the maximal global convergence achieved by All-All strategy, i.e., 90%, other strategies more or less decrease the global convergence. First, SubSet-* randomly chooses 90% available devices as participants while SubSet′ chooses 80%. As a result, those devices with poor local convergence performances are more likely to be chosen when the SubSet threshold is high. Second, Greedy-* chooses devices based on previous results and the threshold 0.98 while the threshold for Greedy′ is 0.9. A lower threshold leads to much reduction on the global convergence since more devices are excluded. In contrast, our proposed online schema dynamically chooses the subset of devices guided by observing results from previous local convergence. Since adjusting both subset of devices and the local convergence threshold for online scenario needs more information beforehand, the little sacrifice on the global convergence in our online schema is acceptable, which earns much more reduction on the resource usage.

Performance under Various Workloads and Settings: We also evaluate our online schema under various scenarios and diverse settings compared with other strategies. As shown in Fig. 5(a), with the growth of available device number, our proposed online schema improves much more than that of other strategies, decreasing 21.5% resource usage when the number is small and decreasing 32.9% when number is large. The higher number of devices refers to the heavier workload in the training of federated learning, which leaves more opportunities for our online schema to choose participants from candidates. As shown in Fig. 5(b) and Fig. 5(c), we vary both the model sizes from 100MB to 1000MB as well as the computational costs for each data piece from 0.3 to 1.5. Compared with the model sizes, i.e., the cost for transmissions, the average cost for computation affects much more on the overall resource usage, since each participant needs to iteratively calculate the latest model based on its own raw data by using SGD, batch SGD, etc. The average decrease on the resource usage is 24.9% when we vary the model sizes while the average decrease on the resource usage is 33.1% when we vary the computational costs for each data piece. Further shown in Fig. 5(d), we vary the data volumes for each participant from 50 to 200. Although excluding those devices with higher data volumes may help to decrease the resource usage, it may also affect the global convergence. Therefore, the online schema considers both resource and convergence, decreasing 24.4% resource usage with only a little sacrifice on the global loss.

We evaluate the designed online schema under various step sizes. With the growth of step sizes, the resource usage decreases. Although tuning the step sizes may help improve the performance of online schema, the performance of our proposed online schema under the step sizes used according to the corollary is acceptable. Furthermore, the calculation cost for our proposed online schema is also reasonable, which only takes several hundreds of milliseconds for calculation.

VI. Related Work

We summarize prior research in three categories, and highlight their drawbacks compared to our work, respectively.

Federated Learning System Optimization: Nguyen *et al.* [17] optimized the training time and the energy consumption of mobile devices. Wang *et al.* [16] controlled model aggregation to minimize the loss function under a given resource budget. Mehdi *et al.* [11] designed a novel federated learning scheme via hierarchical aggregation in heterogeneous edges. Chen *et al.* [13] minimized the loss function under limited wireless bandwidth through user selection. Yang *et al.* [18] studied scheduling federated learning over wireless networks.

These works have considered model training and resource usage; however, they mainly consider an offline setting, and are inapplicable to the time-varying and unpredictable device availabilities, data volumes, and other inputs. The blindness to the training results before making resource decisions further prohibits the applicability of such works to an online setting.

Federated Learning Convergence Analysis: Li *et al.* [9] analyzed the convergence performance of federated learning over Non-IID data. Mohri *et al.* [19] analyzed the connection between convergence and the data-dependent complexity. Xie *et al.* [20] analyzed the asynchronous federated optimization. Haddadpour *et al.* [21] analyzed the convergence generalized to nonconvex problems in federated learning.

These works focus on the convergence performance of the federated learning algorithms themselves, and largely ignore the heavy resource usage of federated learning from a system perspective. Besides, they do not often analyze the relationship between model convergence and resource usage.

Cloud/Edge Online Resource Provisioning: Jiao *et al.* [25–27] proposed a series of online algorithms to provision cloud/edge resources under multi-layer, multi-granularity, and multi-timescale settings, respectively. Xu *et al.* [23] proposed online service caching and offloading for stochastic inputs. Gao *et al.* [22] proposed an online iterative algorithm for access network selection and service placement. The long-term effect of instantaneous violation was also studied in [24, 30, 45], where online algorithms with sub-linear regret and accumulated constraint violation were developed, but they failed to consider the integral decisions.

These works focus on online service provisioning, but rarely consider the specific computation and communication patterns and characteristics of federated learning. Few of them are suitable for federated learning, due to lack of the consideration and incorporation of model convergence in their optimization.

VII. Conclusion

Federated learning often incurs heavy usage of computation and communication resources. To reduce such resource usage, in this paper, we propose to select suitable participants and exclude unnecessary model updates during the model training process. We build a time-varying non-linear integer program to minimize the cumulative resource usage, subject to the long-term model convergence. We design an online algorithm that consists of an online learning component and a randomized rounding component to solve the problem through rectified alternating decent-ascent steps. We also rigorously prove the sub-linear dynamic regret and dynamic fit. Our trace-driven simulations confirm the advantages of our proposed online approach over multiple alternative algorithms in practice.

Appendix

A. Proof of Lemma 1

Proof. Dynamic fit Fit_T^d can be also treated as follows:

$$\begin{aligned}
&||[E[\textstyle\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]]^+|| \overset{(28a)}{\leq} ||E[\textstyle\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]|| \\
&\overset{(28b)}{\leq} ||\textstyle\sum_{t=1}^T \mathbf{g}_t(E[\mathbf{I}_t]) + M\sigma_\beta^\beta|| \overset{(28c)}{=} ||\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t) + M\sigma_\beta^\beta|| \\
&\overset{(28d)}{\leq} ||\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)|| + ||M\sigma_\beta^\beta|| = \widetilde{Fit}_T^d + M\sigma_\beta^\beta, \qquad (28)
\end{aligned}$$

where the inequality (28a) holds because applying $[]^+$ on each dimension would only decrease the absolute value, e.g., a negative value is converted to be 0 after applying $[]^+$. Thus, the value of 2-Norm increases after we omit $[]^+$. Equation (28b) holds due to the Jensen Gap [36] in terms of the item $\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)$, where M and σ_β are constants introduced by Jensen Gap. Equation (28c) holds also due to the unchanged expectation property of our randomized rounding algorithm, and (28d) is due to the triangle inequality of norms [46].

Dynamic regret Reg_T^d can be treated as

$$\begin{aligned}
&E[\textstyle\sum_{t=1}^T f_t(\mathbf{I}_t)] - \sum_{t=1}^T f_t(\mathbf{I}_t^*) \overset{(29a)}{=} \sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\mathbf{I}_t^*) \\
&\overset{(29b)}{=} \textstyle\sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\mathbf{I}_t^*) + (\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t)) \\
&\overset{(29c)}{=} \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\mathbf{I}_t^*) + \sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) \\
&\overset{(29d)}{\leq} \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) = \widetilde{Reg}_T^d, \qquad (29)
\end{aligned}$$

where the equation (29a) holds due to the linear form. Note that ϕ means the same as ϕ^2 for binary variables; the equation (29b) holds since we add two temporary items whose sum is 0; the equation (29c) holds since we re-arrange the terms, and the inequality (29d) holds since the optimum in reals is lower than the optimum in integers for minimization, and $E_\phi[\mathbf{I}_t] = \tilde{\mathbf{I}}_t$ is guaranteed by our delicate designed randomized rounding. □

B. Proof of Lemma 2

Proof. Updating $\boldsymbol{\lambda}$ by using the equation in (18), we have

$$\begin{aligned}
||\boldsymbol{\lambda}_{t+1}||^2 &= ||[\boldsymbol{\lambda}_t + \mu\mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||^2 \overset{(30a)}{\leq} ||\boldsymbol{\lambda}_t + \mu\mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2 \\
&= ||\boldsymbol{\lambda}_t||^2 + 2\mu\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t) + \mu^2||\mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2, \qquad (30)
\end{aligned}$$

where inequality (30a) holds with the same reason as inequality (28a). After re-arranging terms in (30), we obtain (23a). Since $\tilde{\mathbf{I}}_{t+1}$ is the optimum for objective in (16), by using the interior point $\widehat{\mathbf{I}}_t$ mentioned in Assumption 2, we have

$$\begin{aligned}
&\nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2}{2\alpha} \\
&\leq \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\widehat{\mathbf{I}}_t) + \tfrac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha} \\
&\overset{(31a)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \varepsilon\boldsymbol{\lambda}_{t+1}^\top \mathbf{1} + \tfrac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha} \\
&\overset{(31b)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha}, \qquad (31)
\end{aligned}$$

where inequality (31a) holds due to Assumption 2, and inequality (31b) holds because $||\boldsymbol{\lambda}_{t+1}||$ is less or equal to $\boldsymbol{\lambda}_{t+1}^\top \mathbf{1}$ for any non-negative vector $\boldsymbol{\lambda}_{t+1}$. Then, we re-arrange the terms in (31) as follows:

$$\begin{aligned}
&\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \leq \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) \\
&\qquad -\varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{(||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2)}{2\alpha} \\
&\overset{(32a)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \\
&\overset{(32b)}{\leq} ||\nabla f_t(\tilde{\mathbf{I}}_t)||\,(||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t|| + ||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||) - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \\
&\overset{(32c)}{\leq} 2FR - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \overset{def}{=} \Phi_{t+1}, \qquad (32)
\end{aligned}$$

where inequality (32a) holds since the bounded radius on the domain mentioned in footnote, and $||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2 \geq 0$; inequality (32b) holds by using Cauchy-Schwartz inequality twice on the first two terms; and inequality (32c) holds by using the bounded gradient in Assumption 1 and bounded domain. After plugging inequality in (32) into inequality (23a), we have

$$\begin{aligned}
&\triangle(\boldsymbol{\lambda}_{t+1}) := \tfrac{(||\boldsymbol{\lambda}_{t+1}||^2 - ||\boldsymbol{\lambda}_t||^2)}{2} \\
&\leq \mu\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu^2}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\
&\overset{(33a)}{\leq} \mu\boldsymbol{\lambda}_{t+1}^\top(\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})) + \tfrac{\mu^2 G^2}{2} + \Phi_{t+1} \\
&\overset{(33b)}{\leq} \mu\boldsymbol{\lambda}_{t+1}^\top[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})]^+ + \tfrac{\mu^2 G^2}{2} + \Phi_{t+1} \\
&\overset{(33c)}{\leq} \mu\overline{V}(\mathbf{g})||\boldsymbol{\lambda}_{t+1}|| + \tfrac{\mu^2 G^2}{2} + 2FR - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha}, \qquad (33)
\end{aligned}$$

where inequality (33a) holds by adding two complementary terms to the right side, i.e., $\pm\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})$, as well as by using the upper-bound of $\mathbf{g}$; inequality (33b) holds due to the non-negative property of $\boldsymbol{\lambda}_{t+1}$ and the property of $[]^+$; and inequality (33c) holds due to Assumption 3.

Next, we show the correctness of inequality (23b) by contradiction. Without loss of generality, we suppose that $t+2$ is the first time index that breaks inequality (23b), namely:

$$||\boldsymbol{\lambda}_{t+1}|| \leq ||\bar{\boldsymbol{\lambda}}|| < ||\boldsymbol{\lambda}_{t+2}||. \qquad (34)$$

However, by using the equation in (18), the relationship can be obtained on $\boldsymbol{\lambda}$ between consecutive epochs as follows:

$$\begin{aligned}
||\boldsymbol{\lambda}_{t+1}|| &\overset{(35a)}{\geq} ||\boldsymbol{\lambda}_{t+2}|| - ||\boldsymbol{\lambda}_{t+2} - \boldsymbol{\lambda}_{t+1}|| \\
&= ||\boldsymbol{\lambda}_{t+2}|| - ||[\boldsymbol{\lambda}_{t+1} + \mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1})]^+ - \boldsymbol{\lambda}_{t+1}|| \\
&\overset{(35b)}{\geq} ||\boldsymbol{\lambda}_{t+2}|| - ||\boldsymbol{\lambda}_{t+1} + \mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1}) - \boldsymbol{\lambda}_{t+1}|| \\
&= ||\boldsymbol{\lambda}_{t+2}|| - ||\mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1})|| \overset{(35c)}{>} ||\bar{\boldsymbol{\lambda}}|| - \mu G, \qquad (35)
\end{aligned}$$

where inequality (35a) holds due to the triangle inequality; inequality (35b) holds because of the non-expansive property of the projection, i.e., $[]^+$; and inequality (35c) holds by using the hypothesis on $||\boldsymbol{\lambda}_{t+2}||$ from (34). Then, by plugging (35) into (33), we obtain that $\triangle(\boldsymbol{\lambda}_{t+1})<0$, leading to $||\boldsymbol{\lambda}_{t+2}||<||\boldsymbol{\lambda}_{t+1}||$, which contradicts (34). Thus, $\forall t$, inequality (23b) holds. □

C. Proof of Theorem 1

Proof. $\boldsymbol{\lambda}$ is updated by using equation in (18), namely:

$$[\boldsymbol{\lambda}_T + \mu \mathbf{g}_T(\tilde{\mathbf{I}}_T)]^+ \geq ... \geq \boldsymbol{\lambda}_1 + \textstyle\sum_{t=1}^{T} \mu \mathbf{g}_t(\tilde{\mathbf{I}}_t). \tag{36}$$

Since $\boldsymbol{\lambda}_1 = \mathbf{0}$, by re-arranging the terms in (36), we obtain

$$\textstyle\sum_{t=1}^{T} \mathbf{g}_t(\tilde{\mathbf{I}}_t) \leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu} - \frac{\boldsymbol{\lambda}_1}{\mu} \leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu}. \tag{37}$$

Therefore, $\widetilde{Fit}_T^d = ||[\sum_{t=1}^{T} \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||$ can be treated as

$$\widetilde{Fit}_T^d \overset{(38a)}{\leq} ||\textstyle\sum_{t=1}^{T} \mathbf{g}_t(\tilde{\mathbf{I}}_t)|| \leq ||\frac{\boldsymbol{\lambda}_{T+1}}{\mu}|| \leq \frac{||\bar{\boldsymbol{\lambda}}||}{\mu}, \tag{38}$$

where inequality (38a) holds due to the same reason for (28a). By using (28) again, we complete the proof. □

D. Proof of Theorem 2

Proof. The objective in (16) implies that it is $1/\alpha$-strongly convex with respect to $\tilde{\mathbf{I}}$, denoted by $J_t(\tilde{\mathbf{I}})$, i.e., $\forall \mathbf{a}, \mathbf{b} \in \tilde{\mathcal{X}}$:

$$J_t(\mathbf{b}) \geq J_t(\mathbf{a}) + \nabla J_t(\mathbf{a})^\top (\mathbf{b} - \mathbf{a}) + \tfrac{||\mathbf{b}-\mathbf{a}||^2}{2\alpha}. \tag{39}$$

Since $\tilde{\mathbf{I}}_{t+1}$ is the optimum for $\min_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} J_t(\tilde{\mathbf{I}})$, it holds the optimality condition [39], namely:

$$\nabla J_t(\tilde{\mathbf{I}}_{t+1})^\top (\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}) \geq 0. \tag{40}$$

Thus, by setting $\mathbf{a} = \tilde{\mathbf{I}}_{t+1}, \mathbf{b} = \tilde{\mathbf{I}}_t^*$, as well as plugging inequality (40) into inequality (39), we have

$$J_t(\tilde{\mathbf{I}}_t^*) \geq J_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{1}{2\alpha}||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2. \tag{41}$$

After adding $f_t(\tilde{\mathbf{I}}_t)$ on both two sides, expanding $J_t(\cdot)$ according to its definition, i.e., the objective in (16), as well as using the property of convex function on $f_t(\cdot)$, i.e., $f_t(\tilde{\mathbf{I}}_t^*) \geq f_t(\tilde{\mathbf{I}}_t) + \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t)$, we have

$$\begin{aligned} &f_t(\tilde{\mathbf{I}}_t) + \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2}{2\alpha} \\ &\leq f_t(\tilde{\mathbf{I}}_t^*) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t^*) + \tfrac{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2}{2\alpha} - \tfrac{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2}{2\alpha} \\ &\overset{(42a)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2}{2\alpha} - \tfrac{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2}{2\alpha}, \end{aligned} \tag{42}$$

where inequality (42a) comes from the fact that $\boldsymbol{\lambda}_{t+1} \succeq \mathbf{0}$ and the per-slot optimal solution $\tilde{\mathbf{I}}_t^*$ is feasible, i.e., $\mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \preceq \mathbf{0}$, such that $\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \leq 0$. Then, we analyze the gradient term as

$$\begin{aligned} &-\nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) \overset{(43a)}{\leq} ||\nabla f_t(\tilde{\mathbf{I}}_t)||\, ||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t|| \\ &\overset{(43b)}{\leq} \tfrac{||\nabla f_t(\tilde{\mathbf{I}}_t)||^2}{2\eta} + \tfrac{\eta}{2}||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2 \overset{(43c)}{\leq} \tfrac{F^2}{2\eta} + \tfrac{\eta}{2}||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2, \end{aligned} \tag{43}$$

where η is an arbitrary positive constant. Inequality (43a) holds because of the property of norms; inequality (43b) holds because $a^2 + b^2 \geq 2ab$; and inequality (43c) holds due to the bounded gradient of f_t. After that, we plug inequality (43) into inequality (42) and re-arrange the terms as

$$\begin{aligned} &f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \leq f_t(\tilde{\mathbf{I}}_t^*) + (\tfrac{\eta}{2} - \tfrac{1}{2\alpha})||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2 \\ &\quad + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{F^2}{2\eta} \\ &\overset{(44a)}{=} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2}, \end{aligned} \tag{44}$$

where inequality (44a) holds because η is chosen, i.e., $\eta = 1/\alpha$, such that $(\frac{\eta}{2} - \frac{1}{2\alpha}) = 0$. By applying (44) into (23a), we have

$$\begin{aligned} &\tfrac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + f_t(\tilde{\mathbf{I}}_t) \overset{(45a)}{\leq} \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\ &\quad + f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) - \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \\ &\overset{(45b)}{=} f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\ &\quad + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \\ &\overset{(45c)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2} \\ &\quad + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 + \boldsymbol{\lambda}_{t+1}^\top (\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})) \\ &\overset{(45d)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2} \\ &\quad + \tfrac{\mu G^2}{2} + \boldsymbol{\lambda}_{t+1}^\top [\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})]^+ \\ &\overset{(45e)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2} \\ &\quad + \tfrac{\mu G^2}{2} + ||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t), \end{aligned} \tag{45}$$

where inequality (45a) holds because we add the term $f_t(\tilde{\mathbf{I}}_t)$ on both two sides based on (23a) as well as two complementary terms, i.e., $\pm\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})$; equation (45b) holds because we re-arrange the terms; inequality (45c) holds due to the application of inequality (44); inequality (45d) holds due to the bounded value of $\mathbf{g}_{t+1}$ as well as the property of $[]^+$; and inequality (45e) holds based on Assumption 3. Next, we consider the intermediate terms as follows:

$$\begin{aligned} &||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 \overset{(46a)}{=} ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 \\ &\overset{(46b)}{=} ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*||\, ||\tilde{\mathbf{I}}_t^* - 2\tilde{\mathbf{I}}_t + \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 \\ &\overset{(46c)}{\leq} 2R||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2, \end{aligned} \tag{46}$$

where equation (46a) holds because we add two complementary terms; equation (46b) holds because we apply difference of two squares on the first two terms; and inequality (46c) holds due to the triangle inequality for vectors and the bounded radius on domain. Applying inequality (46) to (45), we have

$$\begin{aligned} &\tfrac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + f_t(\tilde{\mathbf{I}}_t) \leq f_t(\tilde{\mathbf{I}}_t^*) + ||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t) + \tfrac{\alpha F^2}{2} + \tfrac{\mu G^2}{2} \\ &+ \tfrac{1}{2\alpha}(2R||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2). \end{aligned}$$

Summing up previous inequality over $t = 1$ to T, we have

$$\begin{aligned} &\textstyle\sum_{t=1}^{T} \tfrac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + \sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t) \leq \sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{\alpha F^2 T}{2} + \tfrac{\mu G^2 T}{2} \\ &\quad + \tfrac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \textstyle\sum_{t=1}^{T} \{||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t)\} \\ &\quad + \tfrac{1}{2\alpha} \textstyle\sum_{t=1}^{T} (||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) \\ &\overset{(47a)}{\leq} \textstyle\sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{\alpha F^2 T}{2} + \tfrac{\mu G^2 T}{2} + \tfrac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \\ &\quad + ||\bar{\boldsymbol{\lambda}}|| \textstyle\sum_{t=1}^{T} V(\mathbf{g}_t) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2 - ||\tilde{\mathbf{I}}_T^* - \tilde{\mathbf{I}}_{T+1}||^2) \\ &\overset{(47b)}{\leq} \textstyle\sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{\alpha F^2 T}{2} + \tfrac{\mu G^2 T}{2} + \tfrac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \\ &\quad + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2), \end{aligned} \tag{47}$$

where inequality (47a) holds due to the definition of $||\bar{\boldsymbol{\lambda}}||$ and (25c), and inequality (47b) holds also due to (25c). Then,

$$\begin{aligned} &\widetilde{Reg}_T^d = \textstyle\sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^{T} f_t(\tilde{\mathbf{I}}_t^*) \leq \tfrac{\alpha F^2 T}{2} + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) \\ &\quad + \tfrac{\mu G^2 T}{2} + \tfrac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \tfrac{||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2}{2\alpha} - \textstyle\sum_{t=1}^{T} \tfrac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} \\ &= \frac{\alpha F^2 T}{2} + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) + \frac{\mu G^2 T}{2} + \frac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \end{aligned}$$

$$+\frac{||\tilde{\mathbf{I}}_1-\tilde{\mathbf{I}}_0^*||^2}{2\alpha}-\frac{||\boldsymbol{\lambda}_{T+2}||^2}{2\mu}+\frac{||\boldsymbol{\lambda}_2||^2}{2\mu}\overset{(48a)}{\leq}\mathcal{R}_T, \tag{48}$$

where inequality (48a) holds because $||\tilde{\mathbf{I}}_1-\tilde{\mathbf{I}}_0^*||^2$ has been bounded by R according to bounded radius of domain, $||\boldsymbol{\lambda}_{T+2}||^2 \geq 0$, as well as $||\boldsymbol{\lambda}_2||^2 \leq \mu^2G^2$ if $\boldsymbol{\lambda}_1 = \mathbf{0}$. □

Acknowledgment

This work is partially supported by the National Key R&D Program of China under Grant No. 2017YFB1001801, NSFC under Grant No. 61872175, and the Natural Science Foundation of Jiangsu Province under Grant No. BK20181252. This work is also partially supported by the Fundamental Research Funds for the Central Universities under Grant No. 14380060, Nanjing University Innovation and Creative Program for PhD Candidate under Grant No. CXCY19-25, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE ICC*, 2019.

[2] K. L. Lueth *et al.*, "State of the iot 2018: Number of iot devices now at 7b–market accelerating," *IoT Analytics*, 2018.

[3] W. House, "Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy," *White House, Washington, DC*, pp. 1–62, 2012.

[4] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints," in *USENIX NSDI*, 2015.

[5] K. Chen and G. Tan, "Bikegps: Localizing shared bikes in street canyons with low-level gps cooperation," *ACM Transactions on Sensor Networks*, vol. 15, no. 4, pp. 1–28, 2019.

[6] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *arXiv preprint arXiv:1909.11875*, 2019.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[8] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS*, 2016.

[9] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2020.

[10] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM*, 2018.

[11] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," *arXiv preprint arXiv:1909.02362*, 2019.

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

[13] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv preprint arXiv:1909.07972*, 2019.

[14] C. Ma, J. Konečnỳ, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[15] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

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

[17] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM*, 2019.

[18] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2020.

[19] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *IEEE ICML*, 2019.

[20] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

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

[22] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *IEEE INFOCOM*, 2019.

[23] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM*, 2018.

[24] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "An online learning approach to network application optimization with guarantee," in *IEEE INFOCOM*, 2018.

[25] L. Jiao, A. M. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed online resource allocation in multi-tier distributed cloud networks," *IEEE/ACM Transactions On Networking*, vol. 25, no. 4, pp. 2556–2570, 2017.

[26] L. Jiao, L. Pu, L. Wang, X. Lin, and J. Li, "Multiple granularity online control of cloudlet networks for edge computing," in *IEEE SECON*, 2018.

[27] W. You, L. Jiao, S. Bhattacharya, and Y. Zhang, "Dynamic distributed edge resource provisioning via online learning across timescales," in *IEEE SECON*, 2020.

[28] L. Zhang, T.-Y. Liu, and Z.-H. Zhou, "Adaptive regret of convex and smooth functions," in *IEEE ICML*, 2019.

[29] Z. Wang, B. Kim, and L. P. Kaelbling, "Regret bounds for meta bayesian optimization with an unknown gaussian process prior," in *NIPS*, 2018.

[30] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350–6364, 2017.

[31] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *arXiv preprint arXiv:1911.02417*, 2019.

[32] S.-S. Shai and Z. Tong, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," in *IEEE ICML*, 2014.

[33] C. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *arXiv preprint arXiv:1910.13067*, 2019.

[34] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, 2017.

[35] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[36] X. Gao, M. Sitharam, and A. E. Roitberg, "Bounds on the jensen gap, and implications for mean-concentrated distributions," *The Australian Journal of Mathematical Analysis and Applications*, vol. 16, no. 2, pp. 1–16, 2019.

[37] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[38] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[39] D. P. Bertsekas, "Nonlinear programming," in *Taylor & Francis JORS*, vol. 48, no. 3, 1997, pp. 334–334.

[40] H. Wang and A. Banerjee, "Online alternating direction method," in *IEEE ICML*, 2013.

[41] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," in *ICLR*, 2015.

[42] "AMPL," https://ampl.com/, 2020.

[43] "AMPL API," https://ampl.com/products/api/, 2020.

[44] "IPOPT," https://github.com/coin-or/Ipopt/, 2020.

[45] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2017.

[46] "List of Triangle Inequalities," https://en.wikipedia.org/wiki/List_of_triangle_inequalities/, 2019.