Federated Learning at the Edge: An Interplay of Mini-batch Size and Aggregation Frequency

Weijie Liu[1], Xiaoxi Zhang[1], Jingpu Duan[2], Carlee Joe-Wong[3], Zhi Zhou[1], Xu Chen[1]
[1]*Sun Yat-sen University,* [2]*Southern University of Science and Technology,* [3]*Carnegie Mellon University*
Email: liuwj55@mail2.sysu.edu.cn, {zhangxx89, zhouzhi9, chenxu35}@mail.sysu.edu.cn
duanjp@sustech.edu.cn, cjoewong@andrew.cmu.edu

***Abstract*—Federated Learning (FL) is a distributed learning paradigm that can coordinate heterogeneous edge devices to perform model training without sharing private raw data. Prior works on the convergence analysis of FL have focused on mini-batch size and aggregation frequency separately. However, increasing the batch size and the number of local updates can differently affect model performance and system overhead. This paper proposes a novel model in quantifying the interplay of FL mini-batch size and aggregation frequency to navigate the unique trade-offs among convergence, completion time, and resource cost. We obtain a new convergence bound for synchronous FL with respect to these decision variables under heterogeneous training datasets at different devices. Based on this bound, we derive closed-form solutions for co-optimized mini-batch size and aggregation frequency, uniformly among devices. We then design an efficient exact algorithm to optimize heterogeneous mini-batch configurations, further improving the model accuracy. An adaptive control algorithm is also proposed to dynamically adjust the batch sizes and the number of local updates per round. Extensive experiments demonstrate the superiority of our offline optimized solutions and online adaptive algorithm.**

I. Introduction

Federated Learning (FL) [1]–[3] has gained much attention as it enables distributed model training through multiple collaborative devices without exposing their raw data. In the meanwhile, with the increasing amount of data generated from different geographical locations and the proliferation of edge computing technologies [4], [5], deploying FL at edge devices has become a promising computation paradigm to facilitate data-driven applications while preserving data privacy. Unlike traditional distributed machine learning (DML) [6], [7], FL allows each training device (a.k.a. worker) to perform multiple local updates before uploading their model parameters to the central server in each aggregation round and does not require partitioning a central pool of data across distributed workers.

Despite its advantages, FL still faces two major challenges: 1) skewed distributions and unbalanced sizes of training data at different devices (*statistical challenge*), and 2) heterogeneous and limited edge resources (*system challenge*). The former is also referred to as non-independent-and-identical (non-i.i.d.) data, which has been analyzed for representative FL algorithms, especially FedAvg [3]. Studies to address the system challenge have mainly focused on mitigating the impact of slow "straggler" devices on the wall-clock training time [5], [8]. Besides, the cost due to either the energy consumption [9] or operational cost paid to incentivize participating clients [10], [11] can be prohibitive for FL at the edge. Thus, taking both time and cost into consideration is of vital importance for performing FL on heterogeneous devices. Recent works have analyzed the model convergence when varying different controls, e.g., balancing the number of local updates and aggregation rounds [8], or adjusting workers' mini-batch sizes under a time budget [5], but these metrics are generally considered separately. To address these challenges simultaneously, we call for a full-fledged FL algorithm that can capture the three-way trade-off between convergence, training time, and cost expenditure. We then aim to jointly optimize the aggregation frequency and mini-batch sizes, as they are the hyperparameters that determine the amount of data processed per round and thus most affect these performance metrics.

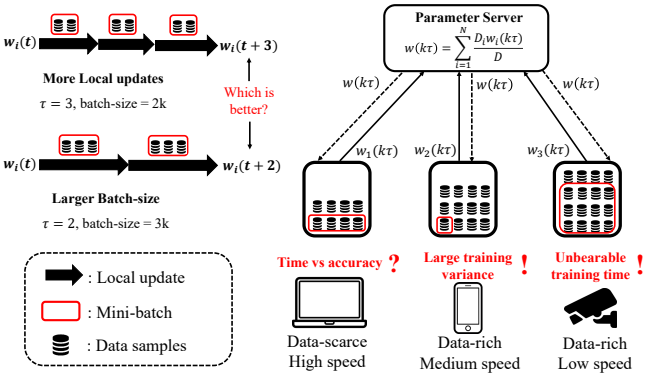


Fig. 1. Left: Interplay of mini-batch size and aggregation frequency; Right: Heterogeneous mini-batch sizes among clients

Further, we have the following intuitions. As illustrated in Figure 1-Left, increasing either the mini-batch size or the number of local updates can lead to more training samples processed and thus improve the local model accuracy. However, doing so can also increase the consumed cost and training time. Moreover, a larger number of local updates (lower

Xiaoxi Zhang is the corresponding author.

This work was supported by Key-Area Research and Development Program of Guangdong Province (2021B0101400001), Guangdong Basic and Applied Basic Research Foundation (2021B151520008), Guangzhou Basic and Applied Basic Research Project (202201011392) under the Guangzhou Science and Technology Plan Project, NSFC grant 62102460, and NSF grants CNS-2106891 and CNS-1751075.

aggregation frequency) may result in a larger gap between the local and global models [3], though this effect may depend on the batch size of each device. Therefore, we ask: *what is the best way to improve the FL model training when we can control both of these variables?* To the best of our knowledge, this is the first work that co-optimizes batch size and global aggregation frequency, considering performance metrics of model accuracy, training time, and resource cost.

This work also reveals that strategically choosing different mini-batch sizes among clients is crucial to improve model performance and system overhead. As illustrated in Figure 1-Right, in this scenario, the generally accepted "no-straggler" principle [5], which assigns the batch sizes of different FL devices for ensuring a uniform time per aggregation round [5], [12], is not always effective. Specifically, the laptop with high training speed but relatively few data samples will have a large mini-batch while other data-rich devices such as the smartphone can only have a small mini-batch due to the relatively slow training speed. This could severely impede the convergence rate, as a small mini-batch size could introduce a high variance to the stochastic gradients [13]. On the other hand, if we neglect the clients' heterogeneous computing capacities by simply setting a uniform batch size, the straggler effects can be severe. Batch sizes, however, cannot help to limit battery usage and communication latency during model synchronization. Therefore, jointly choosing the aggregation frequency in the meanwhile is also important for balancing the energy cost, training time, and model accuracy. To achieve this, we make the following **technical contributions**:

1) *New convergence bound with respect to batch size and global aggregation frequency (Section IV).* We extend FedAvg [3] by allowing clients to use different batch sizes. We derive a novel convergence upper bound for the global model training under non-i.i.d. datasets, with respect to the aggregation frequency and batch sizes. Prior theoretical works usually assume a *full-batch* training setting to achieve bounded convergence rates, but practical FL deployments generally adopt the mini-batch approach. Our error bound can help bridge this inconsistency by quantifying the impacts of batch sizes considering clients' heterogeneous data characteristics.

2) *Novel closed-form results and co-optimization algorithm (Section V).* We propose an optimization model to capture the complex trade-offs among accuracy, completion time, and cost. Driven by our derived convergence bound, we provide closed-form solutions that co-optimize the batch size and aggregation frequency uniformly across clients. These results capture the interplay between these two control variables and can be easily adopted by FL developers. An efficient algorithm is also designed to optimize heterogeneous batch sizes for different clients, which further increases the model accuracy.

3) *Online adaptive joint optimization algorithm (Sections V-C and VI).* We adapt our offline algorithm to the online setting by dynamically choosing the number of local updates and heterogeneous batch sizes among clients, accommodating the online estimates of the computation and communication capabilities in the edge network. Extensive experiments under different testbed settings demonstrate the superiority of our algorithms in terms of the accuracy, cost, and training time.

II. Related Work

Improving the FL Efficiency has been studied in several directions, such as gradient compression [7], [14] and hyper-parameter selection [5], [8], [15]. This work is orthogonal to the former and falls in the latter regime. To optimize the learning speed, most studies choose hyperparameters to mitigate the effect of "straggler" devices. Prior works have proposed various methods to address the straggler problem, like device sampling [16]–[18], client selection [10], [11] or staleness control [4], [19]. Alternatively, recent works [5], [12], [20] also jointly optimize batch sizes and local epochs to improve FL efficiency by equalizing the epoch time for each device. However, their works either lack theoretical analysis or neglect data heterogeneity and resource constraints across clients, which are important characteristics in edge systems.

Controlling FL under resource constraints has risen as the main challenge for edge-enabled FL training. Many studies have been proposed to improve FL accuracy under resource budgets, accounting for either completion time [21]–[23] or operational cost [9], [24]. Luo et al. [25] propose a cost-effective FL design to choose the number of participants and local updates for total training cost minimization. Wang et al. [8] derive a tractable convergence bound with an arbitrary number of local updates and design an algorithm for dynamically adjusting the aggregation frequency. A few recent works also consider optimizing the mini-batch size. E.g, Zhao et al. [13] jointly optimize the batch size and client selection to minimize training cost, and Liu et al. [14] jointly optimize the batch size, gradient compression ratio, and spectrum allocation for wireless FL. However, our work not only additionally analyzes the joint effect of mini-batch size on both convergence, time, and cost metrics, but also provides *both closed-form optimal solutions and efficient algorithm* for jointly selecting the aggregation frequency and batch sizes.

III. Preliminaries and Problem Formulation

A. FL with arbitrary batch size and local update steps

We first consider a parameter-server (PS) architecture, which consists of a set (defined as $\mathcal{N}$) of clients with N distributed edge devices (clients) and a centralized PS for global aggregation. Each device $i \in \mathcal{N}$ has a local data set $\mathcal{D}_i$ with D_i data samples $\mathbf{x}_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,D_i}]$, and $\mathcal{D}_i$ is non-i.i.d. across i. We define the loss function for each sample $\mathbf{x}_{i,j}$ as $f(\mathbf{w}, \mathbf{x}_{i,j})$ and the local loss function of device i as:

$$F_i(\mathbf{w}) = \frac{1}{D_i} \sum_{j \in \mathcal{D}_i} f(\mathbf{w}, \mathbf{x}_{i,j}). \tag{1}$$

The ultimate goal is to train a shared (global) model $\mathbf{w}$ that minimizes the global loss function, defined as:

$$F(\mathbf{w}) = \sum_{i \in \mathcal{N}} \frac{D_i}{D} F_i(\mathbf{w}), \tag{2}$$

where D is defined as $D = \sum_{i \in \mathcal{N}} D_i$.

To capture different batch sizes across clients, we define the loss function $F_{i,\mathcal{S}_i}(\mathbf{w})$ under a mini-batch for each device i:

$$F_{i,\mathcal{S}_i}(\mathbf{w}) = \frac{1}{s_i} \sum_{j \in \mathcal{S}_i} f(\mathbf{w}, \mathbf{x}_{i,j}), \tag{3}$$

where $\mathcal{S}_i$ denotes a mini-batch randomly selected from $\mathcal{D}_i$ and s_i represents the size of $\mathcal{S}_i$. With a learning rate $\eta > 0$, the local update rule can be expressed as:

$$\mathbf{w}_i(t) = \mathbf{w}_i(t-1) - \eta g_i(\mathbf{w}_i(t-1)), \tag{4}$$

where the batch gradient is $g_i(\mathbf{w}_i(t-1)) \triangleq \nabla F_{i,\mathcal{S}_i}(\mathbf{w}_i(t-1))$. The model update at each global aggregation step is:

$$\mathbf{w}(t) = \frac{\sum_{i=1}^{N} D_i \mathbf{w}_i(t)}{D},\ t = k\tau, \tag{5}$$

where τ is the number of local updates in each aggregation round, meaning that the PS only performs (5) and sends the global model $\mathbf{w}(t)$ to the clients at $t = k\tau, k = 1, 2, ..., K$.

B. Accuracy-time-and-cost joint optimization model

Compared to data centers, edge devices usually bear high bandwidth costs and have limited computing resources. It is therefore necessary to consider both computation and communication costs. Formally, we suppose that a units of computation cost are incurred for processing a single sample, and b units of bandwidth cost are consumed in each global aggregation step. Let $s_{tot} = \sum_{i \in \mathcal{N}} s_i$ represent the sum of batch sizes per iteration over all clients. We consider that the total cost incurred by the entire training process cannot exceed R, i.e., $K(a\tau s_{tot} + b) \leq R$, which conforms to the definition of model training cost in [26]. Besides, different edge devices can have heterogeneous computation and communication capacities, and the training time in each round is determined by the slowest device. Let p_i denote the computation speed of device i. We then define t_{ci} as the computation time of i for a single local update and assume that it is proportional to the batch size, i.e., $t_{ci} = s_i/p_i$. Further, t_{ui} is the communication time of each device i incurred by synchronizing her local model with the PS. These definitions are consistent with practical system modelings for FL training [5], [11], [12]. Suppose that the FL task owner has an expected completion time θ, and thus we have the constraint on the completion time, $\max_{i \in \mathcal{N}} K(\tau t_{ci} + t_{ui}) \leq \theta$. Our goal is to find the optimal batch sizes $\mathbf{s}^* = [s_1, s_2, ..., s_N]$ and the number of local update steps τ^* to minimize the gap between the expected global loss function $\mathbb{E}[F(\mathbf{w}(K\tau))]$ and the optimum F^* after performing K communication rounds, while satisfying the cost and completion time constraints. We define $[X] \triangleq \{1, \cdots, X\}$ and formulate the optimization problem as follows:

$$\textbf{Minimize}_{\mathbf{s},\tau} \quad \mathbb{E}[F(\mathbf{w}(K\tau))] - F^* \quad \textbf{(Training error)} \tag{6}$$

$$\textbf{s.t.} \quad \max_{i \in \mathcal{N}} K(\tau t_{ci} + t_{ui}) \leq \theta \quad \textbf{(Completion time)} \tag{7}$$

$$K(a\tau s_{tot} + b) \leq R \quad \textbf{(Cost)} \tag{8}$$

$$s_i \in [D_i], \forall i, \tau \in [\tau_{max}] \quad \textbf{(Feasibility)} \tag{9}$$

To solve the above optimization problem, we need to first navigate the complex trade-offs among the expected error, completion time, and total cost incurred by the training process, via controlling our decision variables $\mathbf{s}$ (mini-batch size) and τ (the number of local updates). Our first challenge is then to simultaneously quantify the effects of $\mathbf{s}$ and τ in the training error, formalized in our next section.

IV. Training Error Bound Analysis

In this section, we derive a new convergence bound to approximate (6), considering the effects of mini-batch sizes s_i and the number of local updates τ.

Assumption 1. *ρ-quadratic-continuous: For each client $i \in \mathcal{N}$, the batch loss function $F_{i,\mathcal{S}_i}$ satisfies: $\|F_{i,\mathcal{S}_i}(\mathbf{w}_1) - F_{i,\mathcal{S}_i}(\mathbf{w}_2)\| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|_2^2$ for all $\mathbf{w}_1, \mathbf{w}_2$.*

Assumption 2. *(First and Second Moment Limits) For some scalars $\mu_G \geq \mu > 0$ and $M_i > 0$, under any given model $\mathbf{w}$ and batch of data samples ξ_t randomly selected from $\cup_i \mathcal{D}_i$, the global batch-gradient $g(\mathbf{w}, \xi_t)$ and global gradient under any single data $s \in \xi_t$, denoted as $g_s(\mathbf{v}_{[k]}(t))$, satisfy:*

$$\nabla F(\mathbf{w})^T \mathbb{E}_{\xi_t}[g(\mathbf{w}, \xi_t)] \geq \mu \|\nabla F(\mathbf{w})\|_2^2,$$
$$\|\mathbb{E}_{\xi_t}[g(\mathbf{w}, \xi_t)\|_2 \leq \mu_G \|\nabla F(\mathbf{w})\|_2,$$
$$\mathbb{V}[g_s(\mathbf{w}, \xi_t)] \leq M_i,\ \forall i \in \mathcal{N}.$$

Theorem 1 (Error bound with heterogeneous batch sizes s_i)**.** *Suppose that $F_{i,\mathcal{S}_i}$ is c-strongly convex and β-smooth [3] and satisfies Assumptions 1-2 . Assuming $F^* \geq 0$, given a fixed learning rate $0 \leq \eta \leq \frac{\mu}{\beta \mu_G^2}$ and the initial global parameter $\mathbf{w}(0)$, the expected error after K aggregation rounds with τ local updates per round is:*

$$\mathbb{E}[F(\mathbf{w}(K\tau))] - F^* \leq q^{K\tau} [F(\mathbf{w}(0)) - F^*] + \frac{1 - q^K}{1 - q} \left(\frac{\beta \eta^2 (1 - q^\tau)}{2D^2(1-q)} \sum_{i \in \mathcal{N}} \frac{M_i D_i^2}{s_i} + \rho h(\tau)^2 \right), \tag{10}$$

where $q = 1 - \eta c \mu$, $h(\tau) = \frac{\delta}{\beta}((\eta\beta + 1)^\tau - 1) - \eta\delta\tau$ and $\delta = \sum_{i \in \mathcal{N}} \frac{D_i \delta_i}{D}$. δ_i upper bounds the gradient divergence between global data $\cup_i \mathcal{D}_i$ and local data $\mathcal{D}_i$, i.e. $\|g_i(\mathbf{w}) - g(\mathbf{w})\|$, describing the non-iid degree of the data in client i.

All the proofs in this paper are provided in our online technical report [27] due to space limitation.

Our bound (10) has a richer structure than those in [5], [8], [14] to show the effects of s_i, τ, and data distributions. The first term is determined by the initial global loss which continuously decreases during the training. The term associated with M_i can be interpreted as the "gradient variance loss" resulting from the error of using a randomly selected batch to estimate the loss gradient under the entire local dataset. The last term $\rho h(\tau)^2$ can be regarded as the "local bias" which monotonically increases with τ, since a larger τ means less frequent communications between clients and the server and thus a larger gap between the global and local models.

V. Co-Optimization: Theory and Algorithm

In this section, we first provide optimal solutions and an efficient algorithm for co-optimized batch sizes and the number of local updates in two offline settings (Sections V-A and V-B), where the parameters related to the model (in (10)) and the system (in the optimization constraints) are pre-obtained. We then adapt the solutions to the online setting with parameters estimated online in Section V-C.

A. Case 1: Co-optimizing τ and uniform s

We first consider the most common scenario used by FL developers in practice [1], [2] where every device has the same batch size s and number of local updates per round τ. Based on our bound (10), we derive closed-form solutions of s and τ in Theorem 2, by solving (6)–(9) with $s_i = s_{i'},\ \forall i \neq i'$.

Theorem 2 (Interplay of uniform s and τ)**.** *Given the number of aggregation rounds K, and a feasible deadline ($\theta > Kt_{ui}$) and cost budget ($R > Kb$), the optimal uniform batch size s^* and number of local updates τ^* satisfy:*

$$s^*(\tau) = \mathbf{min}\left\{\frac{R-Kb}{a\tau n}, \min_{i\in\mathcal{N}}\left\{\frac{p_i(\theta - Kt_{ui})}{K\tau}\right\}\right\}, \quad (11)$$

$$\tau_1 = \lfloor\hat{\tau}\rfloor,\ \tau_2 = \lceil\hat{\tau}\rceil,\ \frac{\partial f(\hat{\tau})}{\partial\tau} = 0, \quad (12)$$

$$\tau^* = \underset{\tau\in\{\tau_1,\tau_2\}}{\arg\min}\ f(\tau),\ s^* = \lfloor s^*(\tau^*)\rfloor, \quad (13)$$

where $h(\tau)$ is defined in (10), $f(\tau) = q^{K\tau}\left[F(\mathbf{w}(0)) - F^*\right] + \frac{1-q^K}{1-q}\left(\frac{\beta\eta^2(1-q^\tau)}{2D^2(1-q)}\sum_{i\in\mathcal{N}}\frac{M_iD_i^2}{s^*(\tau)} + \rho h(\tau)^2\right)$.

B. Case 2: Co-optimizing τ and heterogeneous s_i (offline)

In this case, we generalize Case 1 by enabling different batch sizes assigned for different clients since edge devices can have different computation and communication capacities. However, directly applying integer programming optimizers [28] or using a brute force algorithm to solve (6)–(9) may incur a high time complexity with at least $O(\kappa^N\tau_{max})$, where $\kappa = \frac{s_{tot}}{N} >> 1$. Instead, we design an efficient exact algorithm **CoOptFL** with at most $O(N^2\tau_{max})$ time complexity, as we state in Algorithm 1 and the proof provided in tech report [27].

Other batch size assignment schemes (e.g., [5], [12]), focus on eliminating straggler effects brought by the system heterogeneity. They choose clients' batch sizes according to their computational capacity in order to minimize the average waiting time. However, these strategies are sub-optimal since they neglect data heterogeneity and cost constraints, and clients with higher computation capacities but low data value, i.e. a smaller $D_i\sqrt{M_i}$, will have more training resources, which could significantly undermine the model accuracy. Our Algorithm 1 instead captures both the data heterogeneity and system heterogeneity, as well as navigating the trade-off between the completion time and resource consumption.

C. Case 3: Co-optimizing τ and heterogeneous s_i (online)

Case 2 provides optimal solutions of batch sizes and the number of local updates, but does not consider how to adapt them online with unknown parameters to be estimated such as the computation speed c_i, communication time t_{ui}, and the parameters associated with the model. Thus, in this case, we present a marginal error bound to adjust $\mathbf{s}$ and τ based on our online parameter estimation, realizing a more practical online FL training at the edge under fluctuating network characteristics.

Algorithm 1: An exact offline algorithm to Co-Optimize batch sizes and the number of local updates for FL training (**CoOptFL**)

Input : $\mathbf{G}, M_i, D_i, K, \tau_{max}, a, b, R, \theta, p_i, t_{ui}, \forall i$
Output: $\tau^*, \mathbf{s}^* = [s_1, s_2, ..., s_N]$
foreach $\tau \in [1, \tau_{max}]$ **do**
  Set $C = \mathcal{N}$, $s_{tot} = \frac{R-Kb}{a\tau}$, $s_r = s_{tot}$;
  foreach *node* $i \in \mathcal{N}$ **do**
    $s_i(\theta) = \lfloor p_i\left(\frac{\theta}{K\tau} - \frac{t_{ui}}{\tau}\right)\rfloor$
  repeat
    $flag = 0$;
    foreach *node* $i \in C$ **do**
      $s_i = \lfloor\frac{s_r\sqrt{M_i}D_i}{\sum_{i\in C}\sqrt{M_i}D_i}\rfloor$;
      if $s_i \geq s_i(\theta)$ **then**
        $s_i = s_i(\theta), s_r = s_r - s_i(\theta)$;
        Remove node i from set C, $flag = 1$;
  until $flag = 0$ *or* $C = \emptyset$;
  repeat
    Find $i^{'} = \text{argmax}_{i\in C}\ \frac{D_i^2}{s_i(s_i+1)}$, $s_{i'} = s_{i'} + 1$;
    if $s_{i'} = s_{i'}(\theta)$ **then**
      Remove node i from set C;
  until $\sum_{i\in\mathcal{N}} s_i = s_{tot}$ *or* $C = \emptyset$;
Find the optimum $(\tau^*, \mathbf{s}^*) = \text{argmin}_{(\tau,\mathbf{s})}\mathbf{G}$;
/* Offline: $\mathbf{G} \triangleq$ (10) Online: $\mathbf{G} \triangleq$ (16) */

1) Marginal Error bound: Revisiting our offline optimization problem (6)–(9), the objective function derived in (10) with static parameters and decision variables is no longer suitable for our online setting. Instead, we propose a marginal upper bound, which is defined as the gap between the optimum F^* and $\mathbb{E}[F(\mathbf{w}^{(k)})]$ that denotes the expected loss under the model that will be updated **in aggregation round** k. We derive this in Lemma 1.

Lemma 1 (Marginal bound with heterogeneous batch size s_i)**.** *With the same assumptions in Theorem 1, for a fixed learning rate $0 \leq \eta \leq \frac{\mu}{\beta\mu_G^2}$, the expected loss after k global rounds with the number of updates τ_k and batch sizes s_{ik} for round k, defined as $\mathbb{E}[F(\mathbf{w}^{(k)})] - F^*$, is at most*

$$q^{\tau_k}[\mathbb{E}[F(\mathbf{w}^{(k-1)})] - F^*] + \frac{\beta\eta^2(1-q^{\tau_k})}{2D^2(1-q)}\sum_{i\in\mathcal{N}}\frac{M_iD_i^2}{s_{ik}} + \rho h(\tau_k)^2, \quad (14)$$

where $q = 1 - \eta c\mu$, $h(\tau_k) = \frac{\delta}{\beta}\left((\eta\beta+1)^{\tau_k} - 1\right) - \eta\delta\tau_k$, $F(\mathbf{w}^{(k-1)}) \triangleq F(\mathbf{w}(\sum_{i=1}^{k-1}\tau_i))$.

Compared to Theorem 1, Lemma 1 is defined for the setting where we can obtain better estimates of the unknown model and system parameters in each new aggregation round. Our optimization problem (6)–(9) can be adapted to the following to solve for τ_k and $\mathbf{s}_k = [s_{1k}, s_{2k}, ..., s_{Nk}]$ used for each aggregation round $k \in [K]$.

$$\underset{\mathbf{s}_k, \tau_k}{\textbf{Minimize}} \quad \mathbb{E}[F(\mathbf{w}^{(k)})] - F^* \quad \text{(Approximated by (14))}$$

$$\textbf{S.t.} \ \max_{i \in \mathcal{N}} \sum_{k=1}^{K} (\tau_k t_{ci} + t_{ui}) \leq \theta, \ t_{ci} = s_{ik}/p_i \quad (15)$$

$$\sum_{k=1}^{K} (a\tau_k \sum_{i \in \mathcal{N}} s_{ik} + b) \leq R, s_{ik} \leq D_i, \tau > 0$$

To solve (15), the remaining thing is to estimate the unknown parameters in (14), as elaborated in the next section.

2) Online Parameter Estimation: To simplify the problem (15), we first set $F^* = 0$ as it is impossible to accurately evaluate it for model training. We approximate the expected global loss $\mathbb{E}[F(\mathbf{w}^{(k-1)})] \approx \frac{\sum_{i=1}^{N} D_i F_{i,\mathcal{S}_i}(\mathbf{w}^{(k-1)})}{D} \triangleq \hat{F}(\mathbf{w}^{(k-1)})$ by replacing the local loss $F_i(\cdot)$ with the batch loss $F_{i,\mathcal{S}_i}(\cdot)$, since it can be quite time-consuming to calculate the exact value of $F_i(\mathbf{w}^{(k)})$, especially for large number of data samples.

For ρ, β, c, and δ, we evaluate them in two steps. First, each client estimates these parameters ρ_i, β_i, c_i, and $g_i(\mathbf{w}(t))$ using the global model $\mathbf{w}(t)$ just received at the beginning of every round k before synchronizing their local model $\mathbf{w}_i(t)$ with the global model. Then they will send these results back to the PS to calculate ρ, β, c, and δ as a weighted average of ρ_i, β_i, c_i, and δ_i. Finally our objective function (14), can be approximated by the following error bound:

$$q^{\tau_k} \hat{F}(\mathbf{w}^{(k-1)}) + \frac{\beta\eta^2(1-q^{\tau_k})}{2D^2(1-q)} \sum_{i \in \mathcal{N}} \frac{M_i D_i^2}{s_{ik}} + \rho h(\tau_k)^2. \quad (16)$$

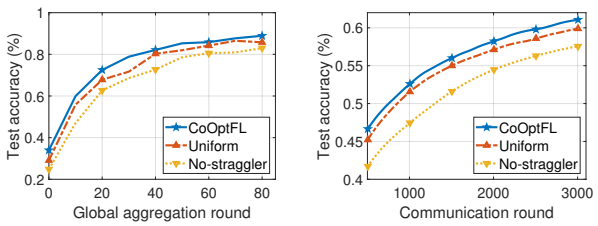
We can leverage the above marginal error bound and online parameter estimation by integrating them into **CoOptFL** to solve our refined co-optimization problem shown in (15). Complete pseudo codes of **CoOptFL** in online setting will be refined and provided in our future work.

VI. Experimental Validation

In this section, we validate our theories and the performance of **CoOptFL** (Algorithm 1) in: 1) Optimal batch size assignment; 2) Co-optimization of heterogeneous batch sizes and aggregation frequency presented in Sections V-B and V-C.

A. Experiment setup

1) Testbed: To simulate the system heterogeneity, we first conduct our experiments in a small-scale testbed with 1 laptop PC, 1 desktop PC, and 3 docker containers launched from a workstation. We manually assign different numbers of CPU cores (3, 6, 12) to each container. We further conduct two larger scale experiments: 1) 100 clients simulated in a lab server cluster; and 2) 20 clients deployed at 20 geo-distributed cloud VM instances from Hetzner [29], including six 1-vCPU instances (2GB RAM), seven 2-vCPU instances (4GB RAM), and seven 4-vCPU instances (8GB RAM) to simulate computational heterogeneity among clients. The PS instance is always deployed on the device with the most CPU cores.



(a) MNIST (5-client) (b) CIFAR (20-client)

Fig. 2. Our batch size assignment in algorithm **CoOptFL** achieves the highest accuracy for both datasets

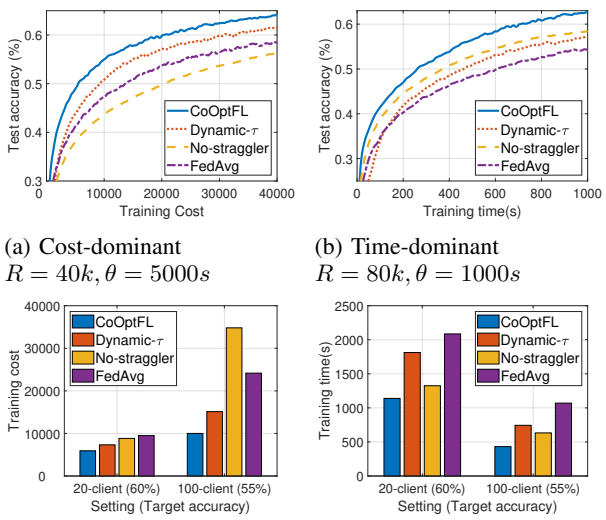
2) Models and datasets: We use MNIST and CIFAR-10 datasets to train a convex SVM model and a non-convex CNN model. We adopt a similar non-i.i.d. data distribution setting in [8] to simulate data heterogeneity among clients.

3) Baselines: To demonstrate the effectiveness of our careful batch size configuration across different clients using **CoOptFL**, we compare with **Uniform**, a widely-adopted method with uniform batch size [8], and with **No-straggler**, a time-efficient batch size distribution proposed in [5]. To evaluate the co-optimization performance of **CoOptFL** in the online setting, we compare with **FedAvg**, which maintains τ and batch size unchanged after their initialization, an adaptive algorithm **Dynamic-τ** proposed by [8], and the **No-straggler** algorithm in [5]. Parameters and settings are provided in the technical report [27].

B. Experimental results and interpretation

1) **Optimal heterogeneous batch sizes s across clients:** We compare our offline algorithm **CoOptFL** to **No-straggler** [5] and **Uniform**. Fig. 2 shows that **CoOptFL** can converge faster and achieve better final testing accuracy compared to the two baselines in both 5-client and 20-client settings. Note that **No-straggler** always tends to assign bigger batch sizes to devices with higher computing capacities regardless of their non-i.i.d. data properties, which leads to a lower resource utilization than **Uniform**, especially when devices with higher computing capacity have less and similar data samples.

2) **Co-optimization of heterogeneous batch sizes and aggregation frequency:** We further compare our **CoOptFL** with three benchmarks for CIFAR-10 FL training: **FedAvg** [3], **No-straggler** [5] and **Dynamic-τ** [8] in the online setting. We compare the strategies in two different scenarios of our optimization problem, where the cost constraint (R) and time constraint (θ) dominates, respectively. We set different values of R and θ to simulate these two different scenarios. Fig.3 shows that **CoOptFL** can outperform the baselines in both scenarios under different settings. **CoOptFL** can achieve a 2.7%–7.9% higher final test accuracy than **FedAvg** while reducing the cost by 37.6%–58% with the same accuracy in the cost-dominant scenario; and a 3.8%–8.4% higher final test



(a) Cost-dominant $R = 40k, \theta = 5000s$

(b) Time-dominant $R = 80k, \theta = 1000s$

(c) Cost-dominant scenario

(d) Time-dominant scenario

Fig. 3. **CoOptFL** achieves the highest final model accuracy and consumes minimal cost and time to achieve the target accuracy under CIFAR-10 in both cost-dominant and time-dominant scenarios

accuracy with 45.4%–59.6% less completion time to achieve the same accuracy in the time-dominant scenario, showing the great adaptability of **CoOptFL**.

VII. Conclusion

This work proposes a novel framework to quantify and optimize the interplay of aggregation frequency and heterogeneous batch sizes across clients for synchronous federated learning performed at distributed edge devices. Technically, we derive a novel convergence bound with respect to those control variables and analyze the performance metrics of training cost expenditure and completion time. We then provide closed-form solutions for our joint optimization , handling both heterogeneous system characteristics and non-i.i.d. data. We finally verify the advantages of our solutions in extensive experiments with several performance metrics considered.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of Artificial intelligence and statistics*, 2017.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[3] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. of International Conference on Learning Representations*, 2019.

[4] J. Liu, H. Xu, L. Wang, Y. Xu, C. Qian, J. Huang, and H. Huang, "Adaptive asynchronous federated learning in resource-constrained edge computing," *IEEE Transactions on Mobile Computing*, 2021.

[5] Z. Ma, Y. Xu, H. Xu, Z. Meng, L. Huang, and Y. Xue, "Adaptive batch size for federated learning in resource-constrained edge computing," *IEEE Transactions on Mobile Computing*, early access, 2021.

[6] X. Zhang, J. Wang, G. Joshi, and C. Joe-Wong, "Machine learning on volatile instances," in *Proc. of IEEE INFOCOM*, 2020.

[7] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons, and O. Mutlu, "Gaia:{Geo-Distributed} machine learning approaching {LAN} speeds," in *Proc. of USENIX NSDI*, 2017.

[8] 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.

[9] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110–124, 2021.

[10] Y. Ruan, X. Zhang, and C. Joe-Wong, "How valuable is your data? optimizing client recruitment in federated learning," in *Proc. of WiOpt*, 2021.

[11] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *Proc. of USENIX OSDI*, 2021.

[12] J. Park, D. Yoon, S. Yeo, and S. Oh, "Amble: Adjusting mini-batch and local epoch for federated learning with heterogeneous devices," *Journal of Parallel and Distributed Computing*, vol. 170, pp. 13–23, 2022.

[13] Y. Zhao and X. Gong, "Quality-aware distributed computation and user selection for cost-effective federated learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2021, pp. 1–6.

[14] S. Liu, G. Yu, R. Yin, J. Yuan, and F. Qu, "Adaptive batchsize selection and gradient compression for wireless federated learning," in *Proc. of IEEE GLOBECOM*, 2020.

[15] J. Zhang, S. Guo, Z. Qu, D. Zeng, Y. Zhan, Q. Liu, and R. A. Akerkar, "Adaptive federated learning on non-iid data with resource constraint," *IEEE Transactions on Computers*, 2021.

[16] W. Xia, T. Q. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7108–7123, 2020.

[17] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning," in *Proc. of AISTATS*, 2021.

[18] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," *arXiv preprint arXiv:2112.11256*, 2021.

[19] J. Cipar, Q. Ho, J. K. Kim, S. Lee, G. R. Ganger, G. Gibson, K. Keeton, and E. Xing, "Solving the straggler problem with bounded staleness," in *Proc. of 14th Workshop on Hot Topics in Operating Systems*, 2013.

[20] D. Shi, L. Li, M. Wu, M. Shu, R. Yu, M. Pan, and Z. Han, "To talk or to work: Dynamic batch sizes assisted time efficient federated learning over future mobile edge devices," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 11 038–11 050, 2022.

[21] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

[22] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. of IEEE INFOCOM*, 2020.

[23] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *Proc. of IEEE ICC*, 2020.

[24] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with cpu-gpu heterogeneous computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7947–7962, 2021.

[25] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. of IEEE INFOCOM*, 2021.

[26] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[27] "Federated learning at the edge: An interplay of mini-batch size and aggregation frequency," https://www.dropbox.com/s/44dq7d61fpppkk0/AdaCoOpt-TR.pdf?dl=0.

[28] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com

[29] Hetzner Online GmbH. [Online]. Available: https://www.hetzner.com/cloud