2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)

Quality-Aware Distributed Computation and Communication Scheduling for Fast Convergent Wireless Federated Learning

Dongsheng Li, Yuxi Zhao, Xiaowen Gong
Department of Electrical and Computer Engineering
Auburn University, Auburn, AL 36849
Email: {dzl0093,yzz0171,xgong}@auburn.edu

***Abstract*—In wireless federated learning (WFL), machine learning (ML) models are trained distributively on wireless edge devices without the need of collecting data from the devices. In such a setting, the quality of a local model update heavily depends on the variance of the local stochastic gradient, determined by the mini-batch data size used to compute the update. In this paper, we explore quality-aware distributed computation for WFL where user devices share limited communication resources, using mini-batch size as a "knob" to control the quality of users' local updates. In particular, we study joint mini-batch size design and communication scheduling, with the goal of minimizing the training loss as well as the training time of the FL algorithm. For the case of IID data, we first characterize the optimal communication scheduling and the optimal mini-batch sizes. Then we develop a greedy algorithm that finds the optimal set of participating users with an approximation ratio. For the case of non-IID data, we first characterize the optimal communication structure and the optimal mini-batch sizes. Then we develop algorithms that find the optimal communication order for some special cases. Our findings provide useful insights for the computation-communication co-design for WFL. We evaluate the proposed mini-batch size design and communication scheduling using simulations, which corroborate improved learning accuracy and learning time.**

I. Introduction

The confluence of two transformative global trends – the accelerating penetration of machine learning (ML) and AI in a variety of domains and the explosive growth of wireless applications – is creating both significant challenges and rich opportunities. It is therefore of great interest to build a synergy between ML/AI and wireless applications. Notably, in federated learning (FL) which is an emerging ML paradigm, the model training is carried out in a distributed manner [1]. One significant advantage of using FL is to preserve the privacy of individual users' data. Moreover, since only local ML model parameters, instead of the local data, are sent to the server, the communication costs can be greatly reduced. Furthermore, FL can exploit ubiquitous smart devices with substantial computing capabilities, which are often under-utilized. In particular, when FL is used in a wireless edge network, the data samples generated at individual wireless devices can be exploited via local computation and global aggregation based on distributed ML. As a result, *wireless federated learning* (WFL) can achieve collaborative intelligence in wireless edge networks. A general consensus is that WFL can support intelligent control and management of wireless communications and networks (such as in [2], [3]), and can enable many AI applications based on wireless networked systems.

As is standard, learning accuracy is a key performance metric for FL. The accuracy of the trained machine learning model in FL depends heavily on the *quality* of participating users' local model updates. Specifically, when distributed stochastic gradient descent (SGD) is used for FL, the quality of a local stochastic gradient in each iteration can be measured by the variance of the gradient, which depends on the *mini-batch size* used to compute the gradient. It is important to observe that the quality of local updates (determined by the mini-batch size) can be treated as a design parameter and used as a *control "knob"* to be adapted across users and over time. Such quality-aware distributed computation can substantially improve the learning accuracy of WFL.

Besides learning accuracy, another important performance metric for FL is learning time, which plays a critical role in real-time applications. The *wall-clock* learning time of a distributed learning algorithm depends on users' computation and communication times of local model updates. Note that the computations and communications of local updates need to be carried out in a coordinated manner, within each round and across different rounds of the learning process. As a result, there is non-trivial interdependence between communication scheduling and mini-batch sizes used in computations, indicating that they should be designed jointly in a judicious and coordinated manner so that they work in concert.

In this paper, we will explore quality-aware distributed computation for wireless federated learning (WFL) for achieving collaborative intelligence in wireless edge networks. We treat mini-batch size as a key "knob" to control the quality of users' local stochastic gradient updates, which has substantial impacts on the learning accuracy of FL. In particular, we study how to *jointly* design users' mini-batch sizes and schedule their communications to reduce the wall-clock learning time of FL, in a wireless edge network where users share limited wireless communication resources. To this end, two significant challenges need to be addressed: 1) The quality (quantified by

This work was supported by the startup fund of Xiaowen Gong, Intramural Grants Program 190599 of Auburn University, and U.S. NSF grant ECCS-2121215.

ISBN 978-3-903176-37-9 © 2021 IFIP

the variance and determined by the mini-batch size) of local stochastic gradient updates and thus its impacts on the training loss can be heterogeneous across users and time-varying over the training process. 2) Due to wireless interference and limited communication resources, there is non-trivial coupling between mini-batch size design and communication scheduling across users. Therefore, it is desirable to take a holistic approach to address these issues in a joint manner.

The main contributions of this paper are summarized as follows:

- We propose quality-aware distribute computation for FL in wireless edge networks, which controls the *quality* of users' local model updates via the mini-batch sizes used to compute the updates. We focus on the joint optimization of mini-batch sizes and communication scheduling, with the goal of minimizing the training loss (quantified by an upper bound that depends on users' mini-batch sizes) as well as the training time of the FL algorithm.
- For the case of IID data, we first characterize the optimal communication scheduling, which are non-preemptive, non-idle, and in the non-increasing order of the ratio of a user's computation rate and communication time. Then we characterize individual users' optimal mini-batch sizes and the optimal total mini-batch size. Next we develop a greedy algorithm that selects participating users, which achieves an approximation ratio by exploiting the non-monotone submodular property of the problem.
- For the case of non-IID data, we first show that the optimal communication structure is non-preemptive and non-idle. Then we find users' optimal mini-batch sizes based on the bisection method. We next develop algorithms that finds the optimal communication order when users have the same communication time or computation rate, which is in the non-decreasing order of the ratio of a user's local data weight and her communication time or computation rate (except the first communicating user).
- We evaluate the proposed joint mini-batch size design and communication scheduling using simulations. The results demonstrate that our proposed algorithms and schemes outperform existing methods in terms of learning accuracy and learning time.

The remainder of this paper is organized as follows. Section II reviews related work. In Section III, we describe quality-aware distributed computation for wireless federated learning. In Section IV and Section V, we study the optimal communication scheduling and mini-batch size design for the cases of IID data and non-IID data, respectively. Simulation results are provided in Section VI.

II. Related work

Wireless Federated Learning. Most prior works on distributed ML have focused on the algorithm design for distributed learning [4], [5], including communication-efficient distributed learning [5], [6]. Less attention has been paid to joint optimization of computation and communication for carrying out distributed learning algorithms. Since it was introduced in 2017, FL has been mostly studied in the setting where nodes are orchestrated by a cloud server. Some more recent works studied FL in *wireless networks* [7]–[9], where nodes are connected wirelessly to each other such that they share limited wireless resources. A few of these works studied both algorithm design as well as computation and wireless resource allocation for WFL. For example, Tran et al [7] studied the joint design of local learning accuracy, computation rates, and communication times. However, all these works have not exploited mini-batch sizes to *control the quality* of users' local model updates. A very recent work [10] has studied mini-batch size design for minimizing users' total training cost in WFL. However, it has not considered *joint mini-batch size design and communication scheduling* for minimizing the training time in WFL.

Wireless Network Scheduling. Wireless network scheduling has been studied extensively for more than a decade. Most of the works focused on maximizing the throughput of wireless networks [11], including those on deadline-constrained throughput [12] and on distributed scheduling [13]. Many other works considered the total utility of data flows in the network [14] which depends on the throughput. Much fewer works studied the *delay* performance of wireless network scheduling [15]. On the other hand, some works studied the cross-layer design of scheduling, routing, and/or congestion control for the objective of improving the throughput [16], delay, or utility. However, these works have not considered the *joint design of wireless network scheduling and distributed computation* for accelerating the convergence of distributed learning.

III. System Model and Problem Formulation

In this section, we present the framework of quality-aware distributed computation and communication scheduling of FL in wireless edge networks, and formulate the joint optimization problem of mini-batch size design and communication scheduling.

A. Quality-Aware Distributed Computation for FL

Consider a FL system with an edge sever and N available users who collaboratively train a ML model with distributed local data in a synchronous manner. One goal of the FL system is to minimize the training loss, which is given by the following optimization problem:

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \triangleq \sum_{i=1}^{N} \frac{D^i}{D} F_i(\boldsymbol{w}),$$

where $F(\boldsymbol{w})$ is the global loss function, $\boldsymbol{w}$ is the model parameter, $F_i(\boldsymbol{w})$ is the local loss function determined by user i's local dataset, D^i is the size of user i's local dataset, and $D \triangleq \sum_{i=1}^{N} D^i$. We make common assumptions that $F(\boldsymbol{w})$ is L-smooth and μ-strongly convex. The local loss function is defined by

$$F_i(\boldsymbol{w}) \triangleq \frac{1}{D^i} \sum_{m=1}^{D^i} l_i(\boldsymbol{w}^i, \delta_m^i),$$

where $l_i(\cdot)$ is the per-sample loss function and $\{\delta_1^i, \delta_2^i, ..., \delta_{D^i}^i\}$ is user i's local dataset.

ISBN 978-3-903176-37-9 © 2021 IFIP

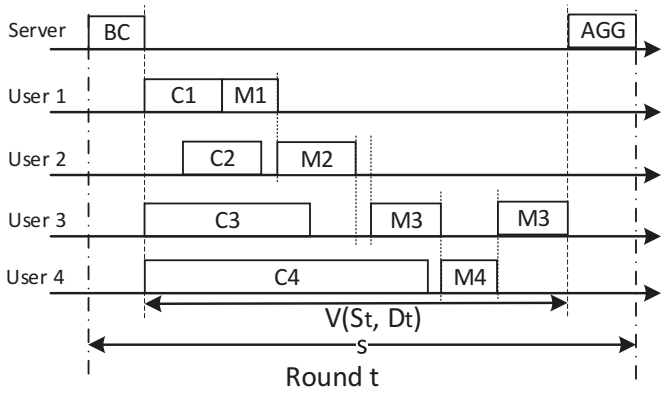

Fig. 1. Schedule of computations (C) and communications (M) for WFL where users share limited wireless communication resources. BC is the global model broadcast, AGG is the local update aggregation, Ci and Mi are user i's local update computation and communication, respectively.

In each round t of the FL algorithm, the sever first broadcasts the current global model $\boldsymbol{w}_{t-1}$ to a set of users S_t selected to participate in round t. Then each participating user i computes the gradient g_t^i of her local loss function using a set of data samples randomly drawn from her local dataset, based on the global model $\boldsymbol{w}_{t-1}$, and updates her local model as $\boldsymbol{w}_t^i = \boldsymbol{w}_{t-1} - \eta g_t^i$, where

$$g_t^i \triangleq \frac{1}{D_t^i} \sum_{m=1}^{D_t^i} l_i(\boldsymbol{w}_{t-1}, \delta_{m,t}^i),$$

D_t^i is user i's mini-batch size in round t, η is the stepsize, $\delta_{m,t}^i$ is the mth data sample randomly drawn from user i's local dataset. Next, each participating user i communicates her local model update to the server. Finally, the server updates the global model with the aggregated local updates received from the users as $\boldsymbol{w}_t = \sum_{i \in S_t} \frac{D_t^i}{D_t} \boldsymbol{w}_t^i$, where S_t is the set of users selected to participate in round t, and $D_t \triangleq \sum_{i \in S_t} D_t^i$.

The *quality* of a user's local update is quantified by the variance of the local stochastic gradient, given by

$$q_i \triangleq E\left[\left\|g_t^i - \bar{g}_t\right\|^2\right], \tag{1}$$

where $\bar{g}_t \triangleq E[g_t^i]$. Assume that the loss function f satisfies $E\left\|\nabla l_i\left(\mathbf{w}_t, \delta_m^i\right) - E[\nabla l_i\left(\mathbf{w}_t, \delta\right)]\right\|^2 \leq \sigma^2, \forall t$. It can be shown that $E\left[\left\|g_t^i - \bar{g}_t\right\|^2\right] \leq \sigma^2 / D_t^i$. Note that a user's quality is determined by the *mini-batch size* D_t^i used to update her local model. Thus, a local update computed with a larger mini-batch size has higher quality.

B. FL in Wireless Edge Network

We focus on the situation where the users are connected to the edge server in a wireless edge network where they share limited wireless communication resources. Due to interference among the wireless users, they need to communicate in a time-division manner to avoid mutual interference, i.e., users' communications cannot overlap in time.

We notice that over-the-air computation (e.g., [17]–[19]) has been recently studied in some works for WFL, which allows users to transmit their local updates to the server simultaneously by taking advantage of the superposition property of wireless signals. However, this scheme requires substantial modification to the physical layer communication protocol (e.g., scaling transmitted data based on wireless channel states), which is difficult to achieve on existing end user devices. Therefore, in this paper, we focus on the time-sharing based wireless multi-access, which is easy to implement on off-the-shelf devices.

The completion time $V(S_t, D_t)$ of round t is the total time it takes to complete all computations and communications of participating users S_t in the round, i.e., the time span from when the first user starts computation to when the last user ends communication (as illustrated in Fig. 1). Note that the completion time depends on users' mini-batch sizes for computing their local updates (which determine their computation workloads), and also on the schedule of users' communications of their local updates. Let C_t^i and M_t^i be user i's computation rate (i.e., computation workload completed per unit time) and communication time, respectively, which generally vary across users and over rounds.

C. Problem Formulation

We aim to minimize the training loss as well as the training time of the FL algorithm, by jointly optimizing user selection, communication scheduling and users' mini-batch sizes. The mathematical formulation of the optimal communication scheduling problem is very complex (due to the very large space of possible communication scheduling policies), and thus is omitted here for brevity. Given the optimal communication scheduling, the problem of user selection and mini-batch size design is formulated as

$$\min_{\{S_t\},\{D_t^i\}} \xi E[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] + (1-\xi) \sum_{t=1}^{T} V(S_t, \{D_t^i\}) \tag{2}$$

where $\xi \in [0,1]$ is the weight that balances the training loss and the training time.

Note that we treat mini-batch sizes as continuous-valued variables in our problem formulation, which can be converted back to the nearest integer values when used in practice. Also note that problem (2) involves multi-objective optimization of the training loss and training time: By controlling the weight ξ, any Pareto-optimal solution of these two objectives can be reached by solving problem (2). A variant formulation of problem (2) is a constrained optimization problem, where the objective function is the training loss while the training time is subject to a constraint (or vice versa). The solution of this variant problem can be derived from that of problem (2).

IV. Optimal Mini-Batch Size Design and Communication Scheduling for IID Data

In this section, we study the mini-batch size design and communication scheduling problem when users have IID data. According to Theorem 1 in [10], the training loss is upper bounded by

$$E[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \leq \frac{L}{2}(1-\mu\eta)^T \left\|\boldsymbol{w}_0 - \boldsymbol{w}^*\right\|^2 + \frac{L}{2} \sum_{t=1}^{T} (1-\mu\eta)^{T-t} \eta^2 \frac{\sigma^2}{\sum_{i \in S_t} D_t^i}.$$

ISBN 978-3-903176-37-9 © 2021 IFIP

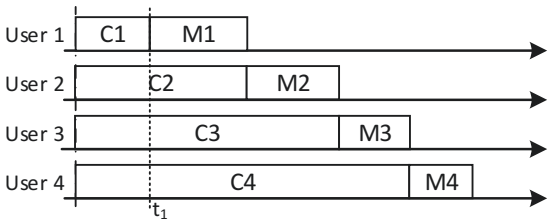

Fig. 2. Schedule of the optimal communication scheduling for the IID data case: user 1 has the largest communication time, user 4 has the largest computation rate.

Since the first term of the upper bound does not depend on S_t and D_t^i, it suffices to consider the second term only. Thus, the problem (2) is equivalent to

$$\min_{\{S_t\},\{D_t^i\}} \sum_{t=1}^{T} f(S_t,\{D_t^i\}) \triangleq \sum_{t=1}^{T} \Big(\xi\frac{L}{2}(1-\mu\eta)^{T-t}\eta^2 \frac{\sigma^2}{\sum_{i\in S_t} D_t^i} + (1-\xi)V(S_t,\{D_t^i\})\Big). \tag{3}$$

Note that problem (3) can be decomposed into T independent problems, each for one of the T rounds. Thus, we focus on finding the optimal communication scheduling, user selection S_t, and mini-batch size design $\{D_t^i\}$ for a single round t. Moreover, we decompose the problem in round t into three subproblems: 1) we first study the optimal communication scheduling given any total mini-batch size design and any user selection; 2) then we study the optimal mini-batch size design given the optimal communication scheduling and any user selection; 3) last we study the optimal user selection given the optimal communication scheduling and the optimal mini-batch size design.

A. Optimal Communication Scheduling

We first present the optimal communication scheduling, which consists of the optimal communication structure and the optimal communication order.

Theorem 1: Given any total mini-batch size design and any user selection, the optimal communication scheduling is non-preemptive and non-idle; based on this structure, the optimal communication order of users is in the *non-decreasing* order of the ratio of a user's computation rate and communication time.

The proof of the optimal communication structure is similar to that in [20], which has studied the optimal joint computation workload allocation and communication scheduling for distributed computation offloading. The proof of the optimal communication order and the proofs of other main results in this paper are provided in our online technical report [21] due to space limitation.

Remark 1: Preemptive communication scheduling means a user's communication can be interrupted by another user's communication (such as M3 interrupted by M4 in Fig. 1). Non-idle communication scheduling means there is no idle communication period between any two consecutive communications (such as between M1 and M2 in Fig. 1). The optimal communication order shows that a user with a larger communication time should communicate earlier (such as user 1 in Fig. 2), since it allows for more computation time for other users. Moreover, it shows that a user with a higher computation rate should communicate later (such as user 4 in Fig. 2), since it allows for more computation time for this user.

B. Optimal Mini-Batch Size Design

Then we study the optimal mini-batch sizes. We decompose this problem into two subproblems: 1) the optimal mini-batch size of each selected user and 2) the optimal total mini-batch size of selected users. In the rest of this section, let users in S_t be indexed according to the optimal communication order (i.e., user 1 communicates first, user 2 communicates second, etc). Let

$$D_r^m \triangleq \sum_{i=2}^{|S_t|}(C_t^i \sum_{j=1}^{i-1} M_t^j)$$

be the maximum total mini-batch size for which the computation workload can be completed after the first communication starts (i.e., after time t_1 in Fig. 2).

Theorem 2: Given the optimal communication scheduling, any total mini-batch size D_t, and any selected users S_t, user 1's optimal mini-batch size is

$$D_t^{1*} = \max\{0,\ \frac{D_t - D_r^m}{\sum_{i\in S_t} C_t^i} C_t^1\},$$

and the other users' optimal mini-batch sizes are given by

$$D_t^{i*} = \begin{cases} C_t^i(\frac{D_t^{1*}}{C_t^1} + \sum_{j=1}^{i-1} M_t^j), & \text{if } D_t > D_r^m \\ \min\{D_t - \sum_{j=1}^{i-1} D_t^{j*},\ C_t^i \sum_{j=1}^{i-1} M_t^j\}, & \text{if } D_t \le D_r^m. \end{cases}$$

Remark 2: Theorem 2 shows that D_r^m is a threshold such that if the total mini-batch size D_t is smaller than D_r^m, then no computation of any user is needed before the first communication starts. In this case, the completion time $V(S_t,\{D_t^i\})$ is fixed and equal to the sum of communication times of selected users S_t. If $D_t > D_r^m$, then users start computations at the same time, and each user keeps computing until her communication starts (as illustrated in Fig. 2). Moreover, each user i's optimal mini-batch size increases with her computation rate C_t^i, as she can perform more computation per unit time; it also increases with the total communication time before user i's communication, since she has more time for computation.

Theorem 3: Given the optimal communication scheduling and any selected users S_t, the optimal total mini-batch size D_t^* is given by $\max\{D_r^m, {D_t}'\}$, where

$${D_t}' \triangleq \eta\sigma\sqrt{\frac{\xi L(1-\mu\eta)^{T-t}\sum_{i\in S_t} C_t^i}{2(1-\xi)}}.$$

Remark 3: Theorem 3 shows that the optimal total mini-batch size D_t^* is affected by several factors. When $\sum_{i\in S_t} C_t^i$ is small, i.e., the number of selected users is small or the users' computation rates are small, then $D_t^* = D_r^m$. When $D_t^* = D_t' > D_r^m$, D_t^* increases with the number of selected users and their computation rates. Also observe that D_t^* (and thus each user i's optimal mini-batch size D_t^{i*}) is larger in a later round, because the weight $(1-\mu\eta)^{T-t}$ of local updates on the training loss bound increases with the round number t.

ISBN 978-3-903176-37-9 © 2021 IFIP

C. Optimal User Selection

Last we study the optimal user selection. From Theorems 2 and 3, there must exist an optimal total mini-batch size D_t^* such that $D_t^* \geq D_r^m$. In this case, we have $D_t^* = D_t'$ where D_t' is given in Theorem 3. Then substituting each D_t^i in (3) as D_t^{i*} in Theorems 2 and 3, we have

$$\min_{S_t} f(S_t) = \frac{\sqrt{\xi L(1-\mu\eta)^{T-t}}}{\sqrt{2}\sum_{i\in S_t} C_t^i}\eta\sigma\sqrt{(1-\xi)} + (1-\xi)\sum_{i\in S_t} M_t^i + (1-\xi)\frac{D_t^* - \sum_{i=2}^{|S_t|} C_t^i \sum_{j=1}^{i-1} M_t^j}{\sum_{i\in S_t} C_t^i}. \quad (4)$$

Next we study problem (4) first in two special cases and then in the general case.

1) Case of Homogeneous Computation Rate or Homogeneous Communication Time: To obtain some useful insights, we first consider the optimal user selection when users have the same computation rate or communication time.

Proposition 1: When users have the same computation rate (or communication time, respectively), the optimal user selection can be found by a greedy algorithm, which selects users incrementally in the non-decreasing order (or non-increasing order, respectively) of their communication times (or computation rates, respectively), until the objective value $f(S_t)$ does not decrease.

Remark 4: Proposition 1 shows that for the case of homogeneous computation rate, it is optimal to select users with smaller communication times. This is because it can reduce the total communication time, and this reduction is more than the increase of the computation time before the first communication starts. For the case of homogeneous communication time, we should select users with larger computation rates, as they can perform more computation per unit time. Note that finding the optimal set of selected users is non-trivial: selecting more users increases the total communication time while reducing the computation time before the first communication starts, so that we should strike the optimal balance between these two effects.

2) Case of Heterogeneous Computation Rates and Heterogeneous Communication Times: We then study the optimal user selection in the general case where users have heterogeneous computation rates and communication times. We first give an important and useful property of the problem. Let S be the set of all N available users.

Lemma 1: $h(S_t) \triangleq f(S) - f(S_t)$ is a non-negative, non-monotone submodular function of the set of selected users.

Based on the submodular property given in Lemma 1, we propose Algorithm 1 to select a set of users. This algorithm is largely based on Algorithm DeterministicUSM developed in [22]. It can provide performance guarantee as below.

Theorem 4 ([22]): Algorithm 1 finds a set of selected users with an approximation ratio of $1/3$, i.e., $f(X_N) \geq \frac{1}{3} f(S^*)$.

Remark 5: Note that it is difficult to solve a negative, non-monotone submodular minimization problem with performance guarantee. To address this challenge, we transform the original user selection problem into a non-negative non-monotone submodular maximization problem, with a new objective function $h(S_t)$. Then we can leverage the non-monotone submodular property to find a solution with an approximation ratio.

Algorithm 1 Approximate optimal user selection

```
index all users in S according to the optimal communication order
X_0 ← ∅, Y_0 ← S.
for i = 1 to N do
    a_i ← f(Y_0) − f(X_{i−1} ∪ {i}) − (f(Y_0) − f(X_{i−1}));
    b_i ← f(Y_0) − f(Y_{i−1} \ {i}) − (f(Y_0) − f(Y_{i−1}));
    if a_i ≥ b_i then
        X_i ← X_{i−1} ∪ {i}, Y_i ← Y_{i−1};
    else
        X_i ← X_{i−1}, Y_i ← Y_{i−1} \ {i};
    end if
end for
Return set of selected users X_N.
```

V. Optimal Mini-Batch Size Design and Communication Scheduling for Non-IID Data

In this section, we study the mini-batch size design and communication scheduling problem when users have non-IID data. According to Theorem 2 in [10], the training loss is upper bounded by

$$E[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \leq \frac{L}{2}(1-\mu\eta)^T \|\boldsymbol{w}_0 - \boldsymbol{w}^*\|^2 + \frac{L\eta^2}{2}\sum_{t=1}^{T}(1-\mu\eta)^{T-t}\sum_{i\in S_t}(p_t^{i\,2}\frac{\sigma^2}{D_t^i} + 2Lp_t^i\tau^i),$$

where $p_t^i \triangleq \frac{p^i}{\sum_{i\in S_t} p^i}$ with p^i being user i's local data's weight (e.g., it can be the size of user i's local dataset), and τ^i quantifies the heterogeneity degree of user i's local data compared to other users' local data, where $\tau^i = E[F(\boldsymbol{w}^*)] - E[F_i(\boldsymbol{w}_i^*)]$ is a constant over the rounds. Since the first term of the upper bound does not depend on S_t and $\{D_t^i\}$, it suffices to consider the second term only. Thus, problem (2) is equivalent to

$$\min_{\{S_t\},\{D_t^i\}} \sum_{t=1}^{T} f(S_t, \{D_t^i\}) \triangleq \frac{L\eta^2}{2}\sum_{t=1}^{T}(1-\mu\eta)^{T-t} \sum_{i\in S_t}(p_t^{i\,2}\frac{\sigma^2}{D_t^i} + 2Lp_t^i\tau^i) + (1-\xi)\sum_{t=1}^{T} V(S_t, \{D_t^i\}). \quad (5)$$

Compared to the training loss bound for the IID data case in (3) which depends on only the total mini-batch size of participating users $\sum_{i\in S_t} D_t^i$, the training loss bound for the non-IID data case in (5) depends on the mini-batch size of each individual participating user (as captured by the term $\sum_{i\in S_t} p_t^{i\,2}\frac{\sigma^2}{D_t^i}$). This key difference substantially complicates the joint design of mini-batch size and communication scheduling (as will be shown later).

Like in the IID data case, we first show the optimal communication structure as below, and the proof is similar to that of Theorem 1.

Proposition 2: The optimal communication structure is non-preemptive and non-idle.

Similar to the IID data case, we decompose problem (5) into T independent problems, each for one of the T rounds.

ISBN 978-3-903176-37-9 © 2021 IFIP

For the problem in a single round t, we decompose it into 2 subproblems: 1) we first study the optimal mini-batch size design given any communication order and any user selection; 2) then we study the optimal communication order given the optimal mini-batch size design and any user selection. The optimal user selection problem is very challenging, and will be studied in our future work.

A. Optimal Mini-Batch Size Design

We first study the optimal mini-batch size design. Let users in S_t be indexed according to their communication order. We decompose this problem into two subproblems: 1) the optimal mini-batch sizes of users 2, 3, $\cdots$, $|S_t|$ given the first communicating user's (i.e., user 1's) mini-batch size and 2) the first communicating user's optimal mini-batch size.

Theorem 5: Given the optimal communication structure, any communication order, any mini-batch size D_t^1 of the first communicating user, and any selected users S_t, the optimal mini-batch size for the remaining users are given by

$$D_t^{i*} = C_t^i \left(\frac{D_t^1}{C_t^1} + \sum_{j=1}^{i-1} M_t^j \right).$$

Remark 6: Theorem 5 shows that the optimal mini-batch sizes have the same structure as for the IID data case: users start computations at the same time and keep computing until their respective communications start. This is because any idle computation time before the communication starts reduces the user's mini-batch size which increases the training loss, without increasing the completion time.

Theorem 6: Given the optimal communication structure, any communication order, and any selected users S_t, the first communicating user's optimal mini-batch size is the solution of D_t^1 to the equation $(1-\xi) - z \sum_{i=2}^{|S_t|} \frac{{p_t^i}^2}{C_t^i \left(D_t^1/C_t^1 + \sum_{j=1}^{i-1} M_t^j \right)^2} = 0$, where $z \triangleq \frac{L}{2}(1-\mu\eta)^{T-t}\eta^2\sigma^2$, which can be solved using the bisection method.

The computational complexity of solving the equation above using the bisection method is $\mathcal{O}(\log_2 n)$.

B. Optimal Communication Order

Next we study the optimal communication order. Substituting the optimal mini-batch sizes in Theorem 5 and 6 into (5), we can find $f(S_t, \{D_t^{i*}\})$ for any communication order. So our problem is to find the optimal communication order such that $f(S_t, \{D_t^{i*}\})$ is minimized.

In the following, to obtain some insights, we will focus on the optimal communication order in two special cases. The general case where users have heterogeneous communication times and heterogeneous computation rates is a very challenging problem, and will be studied in our future work.

1) Case of Homogeneous Communication Time: To find the optimal communication order in this case, we develop Algorithm 2. A key idea of this algorithm is to order the communications of users (except the first communicating user) in the non-decreasing order of ${p_t^i}^2/C_t^i$. We show that this algorithm is optimal as below.

Algorithm 2 Optimal communication order for the case of homogeneous communication time

index users in S_t in the non-decreasing order of ${p_t^i}^2/C_t^i$;
for $i = 1$ to n **do**
  $y1 = C_t^i$, $y2 = p_t^i$
  for $j = i$ to 1 **do**
    $C_t^j = C_t^{j-1}$, $p_t^j = p_t^{j-1}$, $j = j-1$;
  end for
  $C_t^1 = y1$, $p_t^1 = y2$
  if $D_t - \sum_{j=2}^{|S_t|} M C_t^j (j-1) \leq 0$ **then**
    $k(i_a) = \frac{z}{M} \sum_{b=2}^{|S_t|} \frac{{p^b}^2}{C_t^b} \frac{1}{(b-1)}$;
  else
    $t_1 = \frac{D_t - \sum_{j=2}^{|S_t|} M_t C_t^j (j-1)}{\sum_{b=1}^{|S_t|} C_t^b}$;
    $k(i_a) = t_1 + \frac{z}{M\, t} \sum_{b=2}^{|S_t|} \frac{{p_t^b}^2}{C_t^b} \frac{1}{(b-1)+t_1}$;
  end if
end for
find $k_{min} = \min\{\{k(i_a)\}_{i=1}^{|S_t|}\}$ and determine the first communication user according to k_{min}, then the remaining users are ordered in the non-decreasing order of ${p_t^i}^2/C_t^i$;
Return optimal communication order

Proposition 3: When users have the same communication time, Algorithm 2 finds the optimal communication order.

Remark 7: The rationale of Algorithm 2 is as follows. We can show that given the first communicating user, the objective $f(S_t, \{D_t^{i*}\})$ is minimized when the remaining $|S_t|-1$ users communicate in the non-decreasing order of ${p_t^i}^2/C_t^i$. Therefore, it suffices to compare the minimum $f(S_t, \{D_t^{i*}\})$ for each possible first communicating user, for which there are $|S_t|$ number of possible choices. So the computational complexity of the algorithm is $\mathcal{O}(n^2 \log n)$, which is substantially lower than that of the exhaustive search.

2) Case of Homogeneous Computation Rate: To find the optimal communication order in this case, we use an algorithm which is similar to Algorithm 2, but with two differences. The first difference is that we order the communications of users (except the first communicating user) in the non-decreasing order of ${p_t^i}^2/M_t^i$ rather than ${p_t^i}^2/C_t^i$. The second difference is that we need to change the comparing equation $k(i) = z/M_t \sum_{i=2}^{n} \frac{{p_t^i}^2}{C_t^i} \frac{1}{(i-1)}$ to $k(i) = z/C_t \sum_{i=2}^{n} \frac{{p_t^i}^2}{\sum_{j=2}^{n} M_t^j J(i,j)}$, where $J(i,j) = 1$ if user j communicates before user i and $J(i,j) = 0$ otherwise. This algorithm results in a larger mini-batch size for users with larger ${p_t^i}^2/M_t^i$.

Proposition 4: When users have the same computation rate, a variant of Algorithm 2 finds the optimal communication order.

Remark 8: The main idea and rationale of the algorithm above are similar to that of Algorithm 2. This is because the computation rate and communication time play the same role in (5). The computational complexity of the algorithm here is also $\mathcal{O}(n^2 \log n)$.

VI. Performance Evaluation

In this section, we conduct simulations to validate the theoretical findings and evaluate the mini-batch size design and

ISBN 978-3-903176-37-9 © 2021 IFIP

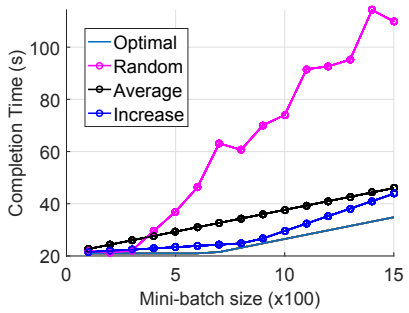

Fig. 3. Impact of individual mini-batch size in IID case

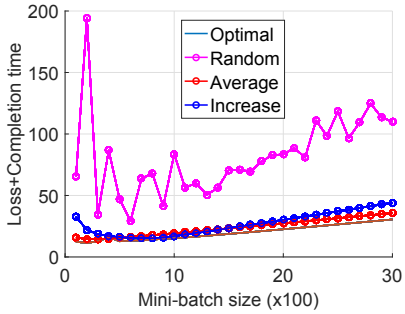

Fig. 4. Impact of individual mini-batch size vs loss and completion time in non-IID case

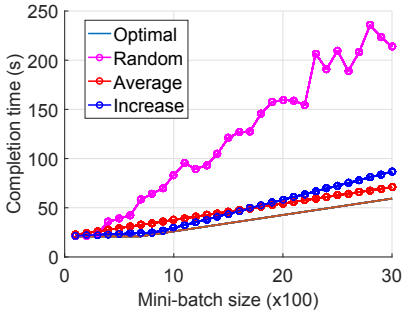

Fig. 5. The completion time with mini-batch size in non-IID case

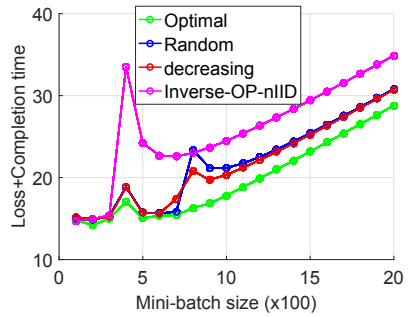

Fig. 6. Impact of communication order in non-IID case

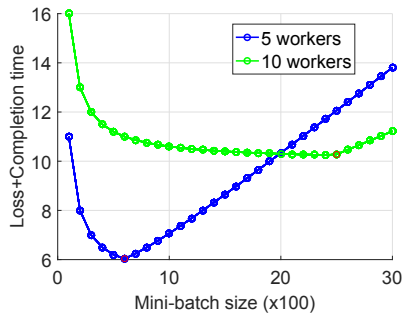

Fig. 7. total mini-batch size in IID case

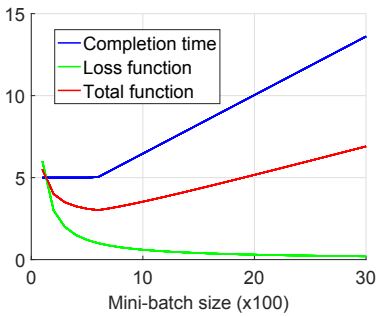

Fig. 8. The changes of 3 function with mini-batch size in IID case

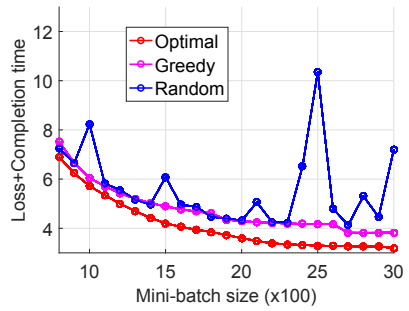

Fig. 9. The completion time and loss function vs selection in IID case

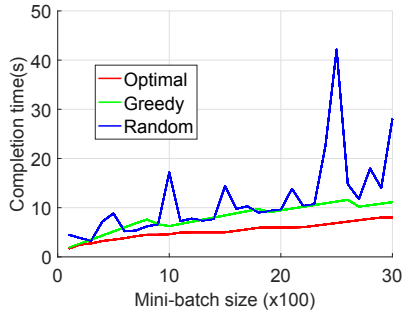

Fig. 10. Completion time vs selection in IID case

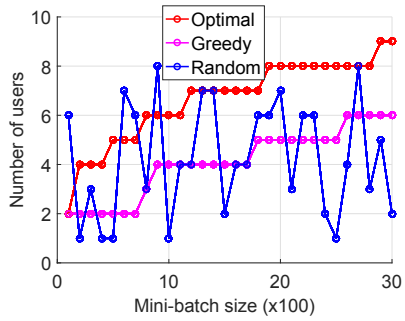

Fig. 11. The number of selected users vs selection in IID case

communication scheduling algorithms. We first describe the simulation setup, and then present the results and analysis.

A. Simulation Setup

We perform the simulations in terms of 4 design variables, which are individual mini-batch size, communication order, total mini-batch size, and user selection. For individual mini-batch size, we study the relation between the completion time and the total mini-batch size under the IID and non-IID data cases, respectively. These 4 algorithms are random allocation, average allocation, increasing with the communication order allocation, and the optimal allocation algorithm, where the random allocation allocates randomly to the users, the average allocation allocates equally to the uses, the increasing allocation allocates increasingly with the communication order to the users and the optimal algorithm is the algorithm we present above. We evaluate the optimal communication order which can depend on the optimal total mini-batch size. In this case, we consider 4 algorithms, which are the optimal algorithm, random algorithm, decreasing with p_t^i algorithm, and inverse to the optimal algorithm. The decreasing with p_t^i algorithm allocates a larger mini-batch size to the users with a smaller p_t^i, and the inverse to the optimal algorithm allocates mini-batch sizes in the non-increasing order of ${p_t^i}^2/C_t^i$. Finally, we evaluate several user selection algorithms, where we consider random selection, greedy selection and approximate optimal selection in IID case. The random selection selects the number of users under different mini-batch size randomly. The greedy algorithm will search for the minimum completion time with the different number of users under different mini-batch size.

ISBN 978-3-903176-37-9 © 2021 IFIP

B. Simulation Results

1) Impact of Individual Mini-Batch Size: We have considered 4 different design algorithms, which are random allocation, equal allocation, increasing allocation, and the optimal allocation algorithm. Figs. 3 and 4 show that if the mini-batch size is randomly allocated to each user, the completion time is the largest among the 4 algorithms. Figs. 3 and 4 show the increasing algorithm sometimes has the same completion time with our optimal algorithm, but for most cases it is much worse than our optimal algorithm. Fig. 5 shows that the system loss in problem (5) is increasingly affected by the completion time as the total mini-batch size increases.

2) Impact of Total Mini-Batch Size: We study the changes in our problems (3) and (5) with the total mini-batch size under the optimal algorithm. Fig.7 shows that the objective value is increasing with the number of selected users, because the completion time increasing by the number of selected users and this increase outweighs the decrease of the training loss. Fig. 8 shows the (3) is increasingly affected by the completion time as the mini-batch size increases. A too large mini-batch size leads to inefficient training and the optimal algorithm can find the optimal total mini-batch size.

3) Impact of Communication Order: Fig. 6 shows that the optimal communication order can decrease (5). If we change the communication order to be inverse to our optimal communication order, it will lead to a much larger increase in the objective function.

4) Impact of Users' Selection: The (2) is similar to the completion time function when the total mini-batch size is larger enough. The number of selected users increases with the total mini-batch size. It is non-optimal to select as many users as possible to minimize the objective function. This is because using more users incurs a larger communication time which can increase the completion time, and this increase can outweigh the reduction of the computation time.

VII. Conclusion and Future Work

In this paper, we have studied quality-aware distributed computation and communication scheduling for wireless federated learning (WFL), with the goal of minimizing the training loss and training time of WFL. For the case of IID data, we have characterized the optimal communication scheduling and the optimal mini-batch sizes. We have also develop a greedy algorithm that finds the optimal set of participating users with an approximation ratio. For the case of non-IID data, we have characterized the optimal communication structure and the optimal mini-batch sizes. Then we have developed algorithms that find the optimal communication order for some special cases. Our findings provide useful insights for the computation-communication co-design for WFL. We have evaluated the proposed algorithms using simulations.

For future work, one interesting direction is to consider asynchronous algorithms and/or non-convex problems for WFL. In this case, the optimal mini-batch size design and the optimal communication scheduling problem will be very different from in the synchronous and convex settings.

References

[1] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *Google Research Blog*, vol. 3, 2017.

[2] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication phy using neural network," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1364–1373, 2019.

[3] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in nb-iot networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[4] 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 *International Conference on Computer Communications (INFOCOM)*. IEEE, 2018.

[5] T. Chen, G. Giannakis, T. Sun, and W. Yin, "Lag: Lazily aggregated gradient for communication-efficient distributed learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5050–5060.

[6] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing systems*, 2017, pp. 1509–1519.

[7] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2019.

[8] 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, 2019.

[9] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, 2020.

[10] 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.

[11] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking (ToN)*, vol. 13, no. 2, pp. 411–424, 2005.

[12] I.-H. Hou, V. Borkar, and P. Kumar, "A theory of QoS for wireless," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2009.

[13] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 18, no. 3, pp. 960–972, 2010.

[14] X. Liu, E. K. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer networks*, vol. 41, no. 4, pp. 451–474, 2003.

[15] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 21, no. 5, pp. 1539–1552, 2013.

[16] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[17] Y. Kai, J. Tao, S. Yuanming, and D. Zhi, "Federated learning based on over-the-air computation," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019.

[18] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *IEEE Transactions on Signal Processing*, 2020.

[19] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *International Conference on Computer Communications (INFOCOM)*, 2021.

[20] X. Gong, "Delay-optimal distributed edge computing in wireless edge networks," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2020.

[21] Technical report. [Online]. Available: https://www.dropbox.com/s/up9mu3nv80lp35c/quality-WFL-fast-TR.pdf?dl=0

[22] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *SIAM Journal on Computing*, vol. 44, no. 5, pp. 1384–1402, 2015.

ISBN 978-3-903176-37-9 © 2021 IFIP