Probabilistic Client Sampling and Power Allocation for Wireless Federated Learning

Wen Xu*, Ben Liang*, Gary Boudreau†, and Hamza Sokun†
*Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada
†Ericsson, Ontario, Canada

***Abstract*—Despite the many known benefits of Federated Learning (FL), in the wireless environment, its performance is significantly impacted by the statistical and system heterogeneities among the local data sets and local clients. Therefore, judicious sampling of clients and resource allocation among them are of vital importance in FL. In this work, we consider the online joint optimization of probabilistic client sampling and power allocation to improve the training performance of wireless FL. Our optimization is based on a new convergence bound for non-convex loss functions under probabilistic client sampling, which considers the different data ratios and gradient norms among clients. We propose a new algorithm based on the Lyapunov optimization framework, termed PCSPA, that accounts for how the statistical and system heterogeneities affect both the convergence rate and training time of FL, as well as the long-term power constraints and the expected number of sampled clients. Experiments on image classification with wireless FL show that the proposed algorithm can substantially outperform conventional separate optimization strategies and a state-of-the-art joint optimization method.**

I. Introduction

In Federated Learning (FL), multiple clients collaboratively train a machine learning (ML) model without transmitting their raw data [1]. In the typical *cross-device* setting, an edge server coordinates the model training, and the clients are usually mobile or IoT devices. Learning occurs over a sequence of training rounds. In each round (a) the server selects a subset of all clients and sends the current global model to them, (b) each selected client returns an updated model based on its local data, and (c) the server updates the global model by aggregating all client updates. Although FL enjoys many advantages such as data privacy and workload sharing, a main challenge is statistical and system heterogeneity [2]–[4]. For example, statistical heterogeneity includes different sizes of local data and non-identical data distributions among clients, and system heterogeneity includes time-varying communication conditions and uneven computing capabilities. If clients are blindly chosen, both the data quality and communication efficiency may suffer.

In the pioneering work by McMahan et al. [1], the server uniformly randomly chooses a fraction of all clients for participation in each round. This simple approach considers neither statistical heterogeneity nor system heterogeneity. An early work on client selection was presented in [5], aiming to maximize the number of sampled clients under round time constraints. A tier-based FL system was proposed in [6], which divides clients into tiers based on performance and only selects clients of the same tier in each round. A meta algorithm of geometrically increasing client participation to tackle stragglers was proposed in [7]. All of the above studies considered deterministic client sampling, leading to challenging combinatorial optimization problems.

Relaxing the binary constraint on client sampling to a probability can lead to substantial reduction in computational complexity. An optimal probabilistic client sampling solution was proposed in [8], which explicitly uses the norms of the gradients to measure the *importance* of the data on clients. However, this work did not take into account system aspects of FL such as channel conditions for communication. Probabilistic sampling with consideration for wireless communications was proposed in [9] to minimize the convergence time and maximize the performance of FL. However, this work required a fixed client to always be connected to the server in each training round, which may not be realistic. A recent work [10] proposed a method to minimize the expected wall-clock time with constraints on convergence and sampling probabilities. However, its computed sampling probabilities are fixed in all training rounds, which does not capture the dynamics of system conditions such as time-varying channels.

Furthermore, it has been recognized in the literature that client sampling in FL is tightly coupled with communication resource allocation among the clients. Joint FL algorithm design and wireless resource allocation was studied in [11]. However, in this work, the client sampling and resource allocation were fixed in all training rounds, which does not capture the dynamics of system conditions. A recent work [12] addressed this issue by combining probabilistic client sampling and Lyapunov optimization [13]. The authors formulated a novel stochastic network optimization problem with time average transmission power as constraints. However, their formulation does not take into account different data ratios and gradient norms of the clients, and it has no control over the number of sampled clients, which can be important in real-world systems that have limited communication and computation capacity.

To address these deficiencies, we employ a probabilistic client sampling paradigm in FL and jointly optimize client sampling and client power allocation for each training round, with consideration for the time-varying learning progress and communication environment. Our formulation also accounts for heterogeneous client data ratios and gradient norms, as well as a constraint on the number of sampled clients. Our contributions are as follows:

This work was funded in part by Ericsson and the Natural Sciences and Engineering Research Council of Canada.

- We first derive a new convergence bound for non-convex loss functions in FL with probabilistic client sampling, taking into account different data ratios and gradient norms of the clients. Based on this convergence analysis, we then formulate a general framework for online joint optimization of probabilistic client sampling and power allocation over the training rounds. Our objective considers both the convergence rate and the training time of FL, subject to constraints on the long-term power usage and the expected number of sampled clients.
- Under the framework of Lyapunov optimization to solve the aforementioned stochastic optimization problem, we derive the per-round problems and observe that they are bi-convex. Furthermore, we utilize a separation structure to find globally optimal solutions to the per-round problems. This leads to an iterative Probabilistic Client Sampling and Power Allocation (PCSPA) algorithm that dynamically adapts to the per-round norms of stochastic gradients, without needing to know the statistics of the training data, the gradients, or the wireless channels.
- We conduct numerical experiments on FL-based image classification over a random wireless environment. Our results demonstrate that PCSPA can substantially outperform conventional FL strategies with separate client sampling and power allocation, as well as the state-of-the-art joint optimization method in [12].

The rest of this paper is structured as follows. In Section II, we define a general form of FL with probabilistic client sampling and derive its convergence bound for non-convex loss functions. In Section III, we describe our formulation of online joint optimization of probabilistic client sampling and power allocation, leading to the proposed PCSPA algorithm and its performance bound. In Section IV, we present numerical experiment results on wireless FL for image classification. Finally, concluding remarks are given in Section V.

II. FL WITH PROBABILISTIC SAMPLING

We consider wireless FL with one parameter server and N clients, where each client n holds a set of training samples $\mathcal{D}_n$. We use $[N]$ as an abbreviation for the set $\{1,\ldots,N\}$. Let D_n be the size of $\mathcal{D}_n$, and $D=\sum_{n\in[N]} D_n$. Let $p_n=\frac{D_n}{D}$, and without loss of generality we assume $p_n>0, \forall n\in[N]$. Let $w\in\mathbb{R}^d$ be the global model to be learned, and let $f_n(w)$ be the local loss function given the training data at client n. Then the global loss $f(w)$ is a weighted sum of $f_n(w)$ and the objective of FL is:

$$\min_{w\in\mathbb{R}^d} f(w), \text{where } f(w)=\sum_{n=1}^{N} p_n f_n(w). \tag{1}$$

We first formalize a more general version of `FedAvg` [1] with probabilistic client sampling, by first replacing the original uniform client sampling and then adopting a new aggregation rule to compensate for the non-uniformity in sampling. We assume that each client runs L steps of local stochastic gradient descent (SGD) for each round of model aggregation at the server (i.e., a *training round*). We define w_t as the parameters of the global model after the t-th round and $w_{t,i}^n$ as the parameters of the local model after the t-th round and the i-th local SGD step on client n. Let T be the number of global rounds, η be the local learning rate, and g_n be a (stochastic) gradient of local loss function f_n. We further denote by q_t^n the probability that client n is sampled in round t. We assume that the server selects client n by running an independent Bernoulli trial with q_t^n as the selection probability in each training round t. Let a_t^n be an indicator function representing whether client n is sampled in round t.

Algorithm 1: FEDERATED LEARNING WITH PROBABILISTIC CLIENT SAMPLING

Input: local learning rate η, local steps L, global rounds T, sampling probability $\{q_t^n\}_{n=1}^N$ in each round t.
Output: $\{w_t\}_{t=0}^{T-1}$.

Server executes:
  initialize w_0;
  for each round $t=0,1,\ldots,T-1$ **do**
    select a subset of clients S_t based on $\{q_t^n\}_{n=1}^N$;
    for each client $n\in S_t$ **in parallel do**
      $w_{t,L}^n \leftarrow \text{ClientUpdate}(n, w_t)$;
    end for
    $w_{t+1} \leftarrow w_t + \sum_{n=1}^N \frac{p_n a_t^n}{q_t^n}(w_{t,L}^n - w_t)$;
  end for

ClientUpdate(n, w_t)**:** ▷ Run on client n
  $w_{t,0}^n \leftarrow w_t$;
  for each local step $j=0,1,\ldots,L-1$ **do**
    pick a mini-batch of samples $z_j\in\mathcal{D}_n$;
    $w_{t,j+1}^n \leftarrow w_{t,j}^n - \eta g_n(w_{t,j}^n; z_j)$;
  end for
  return $w_{t,L}^n$ to server.

Algorithm 1 details the general probabilistic client sampling version of FL. It is similar to the FL with arbitrary client sampling in [10], except that (a) their sampling probability q is fixed over training rounds, (b) they require $\sum_{n\in[N]} q_t^n = 1$ and obtain S_t by sampling N times with replacement, and (c) they have an extra $\frac{1}{N}$ multiplicative term in the aggregation rule.

Note that the aggregation rule of local models in `FedAvg` [1] has been modified in Algorithm 1 to make sure that each aggregation results in an unbiased estimator of the weighted sum of all local results:

$$w_{t+1} = w_t + \sum_{n=1}^{N} \frac{p_n a_t^n}{q_t^n}(w_{t,L}^n - w_t). \tag{2}$$

It is not hard to check that $\mathbb{E}[w_{t+1}] = \sum_{n=1}^N p_n w_{t,L}^n$ by the linearity of expectation and the fact that $\mathbb{E}[a_t^n] = q_t^n$ for any n and t as any a_t^n is one with probability q_t^n.

As far as we are aware, there is no existing convergence analysis on FL in the form of Algorithm 1. In [10], the

convergence bound considers the different data ratios and different gradient norms among clients, but it only applies to smooth and strongly-convex loss functions. In [12], the convergence bound applies to smooth and non-convex loss functions but without the aforementioned consideration on heterogeneity. Here we provide a new convergence bound that combines the advantages of the prior analyses. We require the following assumptions, which are common in the ML literature [14].

Assumption 1 (Smoothness)**.** *Each f_n is β-smooth:*

$$\|\nabla f_n(x) - \nabla f_n(y)\| \leq \beta \|x - y\|, \quad \forall x, y \in \mathrm{dom}(f_n). \tag{3}$$

Remark 1. *Assumption 1 implies the global loss $f(x)$ is also β-smooth, which can be proved by Jensen's inequality.*

Assumption 2 (Unbiased local stochastic gradient)**.** *The stochastic gradient g_n is an unbiased estimator of the true gradient for any local loss function f_n:*

$$\mathbb{E}[g_n(w)] = \nabla f_n(w), \forall w \in \mathrm{dom}(f_n). \tag{4}$$

Remark 2. *This assumption is standard in the analysis of SGD or mini-batch SGD in ML [14].*

Assumption 3 (Bounded stochastic gradients)**.** *There exists $G_n > 0$ such that*

$$\mathbb{E}[\|g_n(w)\|^2] \leq G_n^2 \tag{5}$$

holds for all $w \in \mathrm{dom}(f_n)$ and $n \in [N]$, where g_n is a stochastic gradient of f_n.

Based on these assumptions, we are able to derive a convergence bound for any non-convex loss functions, which is stated in Theorem 1.

Theorem 1. *Suppose Assumptions 1-3 hold, we have*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(w_t)\|^2] \leq{}& \frac{2(f(w_0)-f^*)}{\eta T L} \\ &+ \frac{\eta^2\beta^2(L-1)(2L-1)}{6T}\sum_{t=0}^{T-1}\sum_{n=1}^{N} p_n G_n^2 \\ &+ \frac{\beta\eta}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_t^n}\sum_{i=0}^{L-1}\mathbb{E}[\|g_n(w_{t,i}^n)\|^2],\end{aligned} \tag{6}$$

where w_0 is the initial model at the beginning of training and f^* is the optimal solution of (1)*.*

Proof. We only provide a proof sketch here due to the page limit. The first step is to derive an upper bound between the difference of the expectation of $f(w_{t+1})$ and the actual $f(w_t)$, by considering the local SGD updates, the aggregation rule, and β-smoothness of global loss f. Then, we further upper bound the two terms in the upper bound of the first step, leveraging the independence between client sampling and the randomness in data sampling of SGD, as well as tools including Jensen's inequality and Young's inequality. Finally, we sum the inequality from the previous step over t from 0 to $T-1$, take total expectation, and rearrange terms to obtain the stated result. □

III. Joint Probabilistic Client Sampling and Power allocation

Besides the convergence bound in Theorem 1, the performance of FL also depends on the communication overhead between the server and clients. Without loss of generality, let us model the uplink transmission rate r_t^n of client n in training round t by the Shannon bound:

$$r_t^n = B\log_2\left(1 + \frac{h_t^n P_t^n}{N_0}\right), \tag{7}$$

where B is the bandwidth between the clients and the server, N_0 is the noise power, h_t^n is the channel power gain of client n, and P_t^n is the allocated transmission power of client n. Then the communication time of client n in training round t is

$$T_{\mathrm{comm},t}^n = \frac{M}{r_t^n}, \tag{8}$$

where M is the size of the transmitted model. Therefore, the expected total communication time by all clients in round t is

$$\mathbb{E}[T_{\mathrm{total},t}] = \mathbb{E}_{\{q_t^n\}_{n=1}^N}\left[\sum_{n=1}^{N} a_t^n T_{\mathrm{comm},t}^n\right] \tag{9}$$

$$= \sum_{n=1}^{N} q_t^n \left(\frac{M}{B\log_2(1+\frac{h_t^n P_t^n}{N_0})}\right), \tag{10}$$

A. Optimization Formulation

Our optimization objective considers both the convergence rate as shown in Theorem 1 and the expected round time. We observe that only the third term of the upper bound in (6) depends on the sampling probabilities p_t^n. Let $G_t^n = \mathbb{E}[\sum_{i=0}^{L-1}\|g_n(w_{t,i}^n)\|^2]$. Furthermore, the downlink communication time is fixed since the server simply broadcasts the global model, and the computation time is independent of client sampling. Therefore, we only need to consider the uplink communication time in (10). Thus, our objective contains a trade-off between the convergence bound and the communication time similarly to [12]. Specifically, we define

$$y_0(t) = \sum_{n=1}^{N}\left(\frac{p_n}{q_t^n}(G_t^n)^2 + \lambda q_t^n \frac{M}{B\log_2(1+\frac{h_t^n P_t^n}{N_0})}\right), \tag{11}$$

where λ is a hyperparameter.

We have two transmission power constraints. All clients have maximum power $P_{\max}$, as well as a long-term average power constraint $\bar{P}_n$, which reflects the need for energy conservation in devices with limited battery capacity. We also require the expected number of sampled clients to be bounded by some m. Thus, we obtain the following stochastic optimization problem:

$$\textbf{P1:} \underset{\{q_t^n\},\{P_t^n\}}{\text{minimize}} \quad \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1} y_0(t) \tag{12}$$

$$\text{subject to} \quad \lim_{T\to\infty} 1/T \sum_{t\in\{0,\ldots,T-1\}} P_t^n q_t^n \le \bar{P}_n, \forall n \in [N] \tag{13}$$

$$0 \le P_t^n \le P_{\max}, \quad \forall n \in [N] \tag{14}$$

$$\sum_{n=1}^{N} q_t^n \le m, \tag{15}$$

$$0 \le q_t^n \le 1, \quad \forall n \in [N], \tag{16}$$

where (12) is a time average of $y_0(t)$, (13) and (14) are the constraints on client power, (15) bounds the expected number of sampled clients, and (16) ensures that q_t^n is a probability. Note that the time average in (13) is on $P_t^n q_t^n$, which is the expectation of actual power usage under probabilistic client sampling, i.e., $\mathbb{E}[P_t^n a_t^n]$.

B. Per-round Subproblems and Solutions

We begin to solve the optimization problem in **P1** under the general min drift-plus-penalty framework [13]. We first transform the long-term power constraints into queue stability. Let

$$y_n(t) = P_t^n q_t^n - \bar{P}_n, \quad \forall n \in [N]. \tag{17}$$

Define virtual queues

$$Z_n(t+1) = \max\{Z_n(t) + y_n(t), 0\}. \tag{18}$$

For convenience, we use $\mathbf{\Theta}(t)$ to represent all the queue backlogs at time t by stacking them into one vector. We use the following standard Lyapunov function:

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{n=1}^{N} Z_n(t)^2. \tag{19}$$

Then, the Lyapunov drift is:

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|L(\mathbf{\Theta}(t))\}. \tag{20}$$

The drift-plus-penalty expression is defined as

$$\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(t)\}, \tag{21}$$

where $V \in \mathbb{R}_+$ balances the trade-off between the Lyapunov drift of the Lyapunov function and minimizing the objective functions. By [13, Lemma 4.6], we have the following upper bound on the drift-plus-penalty expression:

$$\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(t)\} \le B_0 + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(t)\} + \sum_{n=1}^{N} Z_n(t)\mathbb{E}\{y_n(t)|\mathbf{\Theta}(t)\}, \tag{22}$$

where B_0 is a positive constant. This leads to the following per-round problem:

$$\textbf{P2:} \underset{\{q_t^n\}_{n=1}^N, \{P_t^n\}_{n=1}^N}{\text{minimize}} \quad Vy_0(t) + \sum_{n=1}^{N} Z_n(t)y_n(t) \tag{23}$$
$$\text{subject to} \quad (14), (15), (16).$$

We observe that the optimization problem **P2** is non-convex in general. Also, we cannot directly separate it into N subproblems, one for each client n, as we have a constraint on the sum of sampling probabilities in (15) which coalesces different clients. However, we observe the following special structure that enables an efficient solution to this problem:

- For any feasible $\{q_t^n\}_{n=1}^N$, the objective and constraints are convex in $\{P_t^n\}_{n=1}^N$. Also, $\{q_t^n\}_{n=1}^N$ can be exactly factored out of this subproblem, so the optimal values of $\{P_t^n\}_{n=1}^N$ in the subproblem are unrelated to $\{q_t^n\}_{n=1}^N$;
- If $\{P_t^n\}_{n=1}^N$ are fixed, the objective and constraints are convex in $\{q_t^n\}_{n=1}^N$.

Hence, the following two-step approach suffices to find a globally optimal solution: we first fix $\{q_t^n\}_{n=1}^N$ to solve for $\{P_t^n\}_{n=1}^N$; then based on the solutions of $\{P_t^n\}_{n=1}^N$, we solve for $\{q_t^n\}_{n=1}^N$.

For the first step, we derive a closed-form solution for the optimal $\{P_t^n\}_{n=1}^N$. For any feasible $\{q_t^n\}_{n=1}^N$, the subproblem of $\{P_t^n\}_{n=1}^N$ is

$$\underset{\{P_t^n\}_{n=1}^N}{\text{minimize}} \quad V\sum_{n=1}^{N} \lambda q_t^n \left(\frac{M}{B\log_2(1+\frac{h_t^n P_t^n}{N_0})}\right) + Z_n(t)P_t^n q_t^n \tag{24}$$

$$\text{subject to} \quad 0 \le P_t^n \le P_{\max}, \quad \forall n \in [N]. \tag{25}$$

This problem can be solved by separate optimization of each P_t^n, and we observe that the q_t^n terms in the objective can be eliminated:

$$\underset{P_t^n}{\text{minimize}} \quad V\lambda \left(\frac{M}{B\log_2(1+\frac{h_t^n P_t^n}{N_0})}\right) + Z_n(t)P_t^n \tag{26}$$

$$\text{subject to} \quad 0 \le P_t^n \le P_{\max}. \tag{27}$$

The optimization problem in (26)-(27) is a single-variable convex optimization problem with a convex objective and a box constraint.

In the following we derive a closed-form solution to the above problem. For simplicity, let $A_1 = V\lambda M\log(2)/B$, $A_2 = h_t^n/N_0$, and $A_3 = Z_n(t)$. The convex objective becomes $\frac{A_1}{\log(1+A_2P_t^n)} + A_3P_t^n$. Setting its derivative to zero, we have

$$(1+A_2P_t^n)(\log(1+A_2P_t^n))^2 = \frac{A_1A_2}{A_3}. \tag{28}$$

Let $x = \frac{\log(1+A_2P_t^n)}{2}$. Plugging x into (28), we obtain $x\exp(x) = \sqrt{\frac{A_1A_2}{4A_3}}$. Therefore $x = W_0(\sqrt{\frac{A_1A_2}{4A_3}})$, where W_0 is the principal branch of the Lambert W function. Since $\sqrt{\frac{A_1A_2}{4A_3}} \ge 0$, x is non-negative and unique. Then we can recover P_t^n from x. If this P_t^n falls within the range $[0, P_{\max}]$, it is the optimal power. Otherwise, the optimal power is $P_{\max}$. Summarizing the above, we have

$$P_t^n = \begin{cases} \frac{N_0}{h_t^n}\left(\exp(2W_0(\sqrt{A}/2)) - 1\right), & \text{if } P_t^n \le P_{\max}, \\ P_{\max}, & \text{otherwise,} \end{cases}$$

where $A = \frac{V\lambda M\log(2)h_t^n}{BZ_n(t)N_0}$.

For the second step, with fixed $\{P_t^n\}_{n=1}^N$, the optimization problem becomes

$$\underset{\{q_t^n\}_{n=1}^N}{\text{minimize}} \quad V\sum_{n=1}^{N}\left(\frac{p_n}{q_t^n}(G_t^n)^2+\lambda q_t^n\left(\frac{M}{B\log_2(1+\frac{h_t^nP_t^n}{N_0})}\right)\right)+\sum_{n=1}^{N}P_t^nq_t^nZ_n(t) \tag{29}$$

$$\text{subject to} \quad \sum_{n=1}^{N}q_t^n\le m, \tag{30}$$

$$0\le q_t^n\le 1, \quad \forall n\in[N]. \tag{31}$$

This is a convex optimization problem as the objective is convex and the constraints are affine in $\{q_t^n\}_{n=1}^N$. We can use a standard convex optimization solver to find an optimal solution in polynomial time [15].

C. PCSPA Algorithm

With the sampling probabilities and power allocation for each round t, obtained as the solution to the above per-round problems, we can extend Algorithm 1 to construct our PCSPA algorithm, which is summarized in Algorithm 2. We note that in each round t, the server sends not only the current global model w_t but also the calculated power allocation P_t^n to the sampled clients. Furthermore, before the actual sampling of clients in each round, all clients need to perform local computation to determine the actual norm of local stochastic gradients and transmit these norms to the server on a control channel. This transmission can be efficiently performed piggyback on the standard channel estimation procedure between the server and clients.

Finally, we remark that since the per-round optimization problems in PCSPA can be efficiently solved as detailed in Section III-B, PCSPA is computationally efficient. Furthermore, since the per-round optimization solution can achieve an arbitrary precision of optimality, from [13, Theorem 4.8], PCSPA provides the following performance guarantee:

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[y_0(t)]\le y_0^{\text{opt}}+\frac{B_0}{V}, \tag{32}$$

where y_0^{opt} is the optimal objective of problem (12)-(16), while it is guaranteed that PCSPA satisfies all time average constraints.

Algorithm 2: JOINT PROBABILISTIC CLIENT SAMPLING AND POWER ALLOCATION (PCSPA)

Input: learning rate η, local epochs L, global rounds T.
Output: $\{w_t\}_{t=1}^T$.

Server initializes w_0 and virtue queues $\{Z_n(0)\}$.
for each round $t=0,1,\ldots,T-1$ **do**
  Server broadcasts the current model w_t to all clients.
  for each client n **do**
    run L steps of training with learning rate η.
    send G_t^n back to the server.
  end for
  Server calculates $\{q_t^n\}_{n=1}^N$ and $\{P_t^n\}_{n=1}^N$.
  Server selects clients S_t based on $\{q_t^n\}_{n=1}^N$.
  Server broadcasts P_t^n to client $n\in S_t$.
  for each client $n\in S_t$ **do**
    send the local model $w_{t,L}^n$ back to the server.
  end for
  Server aggregates the local models via (2).
  Server updates the virtual queues via (18).
end for

IV. NUMERICAL EVALUATION

We conduct numerical experiments using JAX [16] and FedJAX [17] for the learning framework and CVXPY [18] for convex optimization. We perform FL for image classification on the Fashion-MNIST dataset [19] under two scenarios of data distribution: IID and non-IID. In the IID case, all clients have equal-size training datasets that are drawn from all 10 classes of Fashion-MNIST. In the non-IID case, each client holds only one class of training data, and the number of samples on client n is $100n$. The learning model is a fully-connected neural network with two hidden layers of 300 and 100 neurons each. There are $266,610$ trainable parameters, which are of size $8,531,520$ bits using the default single precision float number representation. The training lasts for $T=1000$ rounds. For each local step, a client samples a mini-batch of size 2 from its local dataset to update its local model by SGD with learning rate 0.01.

We consider an FL system of one server and $N=10$ clients. We assume bandwidth $B=22$ MHz, noise power $N_0=2\times10^{-8}$ W, and the Rayleigh fading channel from each client to the server that result in h_t^n following an exponential distribution with mean 2×10^{-5}. The long-term power constraints $\bar{P}_n=0.01$ W and maximum power $P_{\max}=1$ W. We set $\lambda=1$ and $V=1$.

A. Comparison Benchmarks

We consider the following three benchmarks.

- **Separate uniform**: For sampling, in each round t solve

$$\underset{\{q_t^n\}_{n=1}^N}{\text{minimize}} \quad \sum_{n=1}^{N}\frac{1}{q_t^n} \tag{33}$$
$$\text{subject to} \quad \sum_{n=1}^{N}q_t^n\le m \text{ and } 0\le q_t^n\le 1,\ \forall n\in[N].$$

  Note that the solution to (33) is $q_t^n=\frac{m}{N}$, i.e., each client receives the same sampling probability. It corresponds to the probabilistic client sampling version of the vanilla `FedAvg` [1], i.e., Algorithm 1 with $q_t^n=\frac{m}{N}$. To satisfy the long-term power constraints, each sampled client transmits with power $\frac{\bar{P}_n}{q_t^n}$ in round t.
- **Separate gradient-based**: For sampling, in each round t solve

$$\underset{\{q_t^n\}_{n=1}^N}{\text{minimize}} \quad \sum_{n=1}^{N}\frac{p_n(G_t^n)^2}{q_t^n} \tag{34}$$

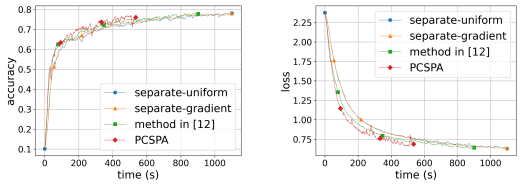

Fig. 1. Accuracy and loss in the IID scenario for $m = 8$.

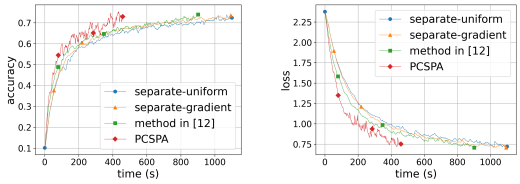

Fig. 2. Accuracy and loss in the non-IID scenario for $m = 8$.

$$\text{subject to} \quad \sum_{n=1}^{N} q_t^n \leq m \text{ and } 0 \leq q_t^n \leq 1, \forall n \in [N].$$

This benchmark uses the same power allocation method as **separate uniform**. It further considers the data ratio and the actual norms of the gradients, which is inspired by [8], [10].

- **Joint optimization method in [12]**. Since [12] does not support our constraint on the expected number of sampled clients, we modify it as follows. For any resulting sampling probability vector with $\sum_{n=1}^{N} q_t^n > m$, we sequentially set the lowest among q_t^n to 0 until the sum of q_t^n no longer exceeds m.

B. IID Scenario

In Fig. 1, for the scenario where all clients have the same data distribution, we plot the test accuracy and loss over time for $m = 8$. We observe PCSPA requires substantially less time to achieve the same learning performance as the other methods. Also, in this IID scenario, since each client has same the distribution, the model performance does not differ much for the two benchmarks that use separate optimization.

C. Non-IID Scenario

In Fig. 2, for the scenario where each client holds only one class of data samples, we plot the test accuracy and loss over time for $m = 8$. As each client holds a different data distribution, the curves are noisier than in the IID scenario. Therefore, we use an exponential moving average to smooth the curves. We observe that the performance gain of PCSPA is even more pronounced in this scenario. In particular, the method in [12] and both of the separate optimization methods experience substantial performance degradation due to data heterogeneity, as indicated by their slower decrease in loss and slower increase in accuracy. In contrast, PCSPA retains much of its rate of improvement over time as in the IID scenario.

V. Conclusion

We consider online joint optimization of probabilistic client sampling and power allocation over the training rounds of wireless FL. The optimization objective is based on a new convergence bound that takes into account the statistical heterogeneity and real-time learning dynamics. The proposed PCSPA algorithm accommodates heterogeneous and time-varying communication channels, constraints on the long-term power usage of clients, and a limit on the expected number of sampled clients in each round. It is computationally efficient and provides provable performance guarantee. Our numerical experiments with image classification examples demonstrate that PCSPA can substantially outperform the state-of-the-art method in [12] and other alternatives, especially when the clients hold training datesets with heterogeneous distributions.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, ser. Proc. Mach. Learn. Res., vol. 54, 20–22 Apr 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[3] T. Li, A. K. Sahu, A. S. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," *IEEE Signal Process. Mag*, vol. 37, pp. 50–60, 2020.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019.

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

[6] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo *et al.*, "Tifl: a tier-based federated learning system," in *Proc. ACM Int. Symp. High-Performa. Paral. and Distrib. Comput. (HPDC)*, 2020, p. 125–136.

[7] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[8] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *Trans. Mach. Learn. Res.*, Aug. 2022.

[9] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time minimization of federated learning over wireless networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020.

[10] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022.

[11] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2021.

[12] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022.

[13] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool, 2010.

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

[15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[16] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin *et al.*, "JAX: composable transformations of python+numpy programs," 2018. [Online]. Available: http://github.com/google/jax

[17] J. H. Ro, A. T. Suresh, and K. Wu, "FedJAX: federated learning simulation with JAX," *arXiv preprint arXiv:2108.02117*, 2021.

[18] S. Diamond and S. Boyd, "CVXPY: a python-embedded modeling language for convex optimization," *J. of Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[19] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. [Online]. Available: https://github.com/zalandoresearch/fashion-mnist