Scheduling Policy and Power Allocation for Federated Learning in NOMA Based MEC

Xiang Ma*, Haijian Sun†, and Rose Qingyang Hu*
*Department of Electrical and Computer Engineering, Utah State University, Logan, UT
†Department of Computer Science, University of Wisconsin-Whitewater, Whitewater, WI

Abstract—**Federated learning (FL) is a highly pursued machine learning technique that can train a model centrally while keeping data distributed. Distributed computation makes FL attractive for bandwidth limited applications especially in wireless communications. There can be a large number of distributed edge devices connected to a central parameter server (PS) and iteratively download/upload data from/to the PS. Due to limited bandwidth, only a subset of connected devices can be scheduled in each round. There are usually millions of parameters in the state-of-art machine learning models such as deep learning, resulting in a high computation complexity as well as a high communication burden on collecting/distributing data for training. To improve communication efficiency and make the training model converge faster, we propose a new scheduling policy and power allocation scheme using non-orthogonal multiple access (NOMA) settings to maximize the weighted sum data rate under practical constraints during the entire learning process. NOMA allows multiple users to transmit on the same channel simultaneously. The user scheduling problem is transformed into a maximum-weight independent set problem that can be solved using graph theory. Simulation results show that the proposed scheduling and power allocation scheme can help achieve a higher FL testing accuracy in NOMA based wireless networks than other existing schemes within the same learning time.**

Index Terms—**Federated Learning, scheduling policy, power allocation, maximum-weight independent set, NOMA.**

I. INTRODUCTION

The rapidly growing data availability has gradually enabled training based artificial intelligence applications such as image recognition, autonomous driving, and natural language processing to become reality [1]. Unlike the traditional model-based problem solving approaches, machine learning (ML) is more data-driven and less depends on the knowledge of the models. State-of-the-art ML techniques especially deep learning [2] has demonstrated remarkable performance, such as AlphaGo and Tesla Autopilot, which can outperform human beings in certain areas. Since processing big data may exceed the computation capability of a single server, processing through multiple distributed [3] yet collaborative devices becomes a highly promising and feasible direction to pursue. Further motivated by the increasing computational/storage capacities of wireless local devices as well as the ever increasing concerns on sharing data due to privacy and security, next-generation communications/computation networks will encounter a paradigm shift from conventional cloud/central computing to mobile edge computing (MEC) [4], which largely deploys computational power to the wireless network edge devices to meet the needs of applications that demand very high computations, low

latency, as well as high privacy requirement. In this paradigm, a large ML task is partitioned into multiple pieces that can be performed in parallel by multiple distributed mobile edge devices based on locally collected data.

Although data can be processed locally and do not need to be sent in the primitive format to the central parameter server (PS), data with reduced size may still need to be exchanged for joint processing in order to reach a global consensus on the model learning. Recently, a novel ML technique called federated learning (FL) [5] is proposed to address this issue. It allows devices to collect data from their local environment and then train models locally. No raw data transmission to the PS is needed. Instead the trained model with a much reduced data size is uploaded to the PS. There are usually a large number of edge devices connected to one PS. To achieve efficient learning with limited wireless bandwidth, FL only selects a subset of edge devices for model update in each round. Devices collect data from their respective wireless local environment so the data collected across different devices can be heterogeneous or non-i.i.d. The significance of user scheduling is to make a decision on selecting a subset of devices (most important devices based on certain scheduling criteria) to upload model update in each round. The study in [6] gave three different scheduling policies, i.e., random scheduling, round robin, proportional fair to schedule devices randomly, in group and according to channel conditions, respectively. They considered the number of devices and the channel conditions in scheduling but did not consider the data distribution. [7] proposed a coordinated scheduling and power control scheme in cloud radio access networks. To maximize the weight sum data rate, the maximum weight sum data rate problem was transformed to a maximum-weight clique problem. Then the power allocation problem was solved using [8] to achieve weighted throughput maximization through power control. It considered user scheduling by using the orthogonal time divsion multiplexing access (TDMA) and frequency division multiplexing access (FDMA). [9] investigated the spectrum efficient resource management problem (SERMP) under non-orthogonal multiple access (NOMA) by transforming the SERMP problem into a maximum weighted independent set problem and solved it using graph theory. However, none of the above can be directly applied to our scheme.

There are usually millions of model parameters in the modern deep learning models such as ResNet-50 has 25 million parameters, AlexNet has 62 million parameters. Most of the

existing works consider a computer-science based methodology to reduce the model size by compression. [10] utilized quantization and sparsification to perform model compression. Furthermore, advanced communication mechanisms have been developed to improve the spectral efficiency and to enhance the data rate, which is very instrumental to facilitate the ML methods from communications perspective. When the transmission takes place in TDMA or FDMA, different devices should work in different time slot or frequency channel. NOMA allows multiple devices to transmit simultaneously on the same channel so that data rate is increased and communication latency is reduced when implementing FL [11].

In this work, we focus on NOMA-based FL uplink communication by considering wireless fading channel. The user scheduling and power allocation are formulated as a maximum weighted sum rate problem, which is further transformed to a maximum weight independent set problem and solved with graph theory. The contributions are summarized as follows.

- We apply NOMA as the transmission scheme in FL model update to reduce aggregation latency. Our simulations reveal more than 2x time reduction compared with TDMA-based scheme.
- To find the optimal user scheduling and power allocation scheme, we utilize the efficient maximum weight independent set based on graph theory.

The rest of the paper is organized as follows. Section II introduces the system model, NOMA transmission scheme and problem formulation. Section III presents the solution for user scheduling and power allocation. Simulation results are shown in Section IV, where experiments are conducted to verify the proposed schemes. Lastly, Section V concludes the paper.

II. SYSTEM MODEL

For the distributed learning task on device k, there exist a dataset $\boldsymbol{x}_k$ and a corresponding label $\boldsymbol{y}_k$. At round t, ML learns the mapping from $\boldsymbol{x}_k^t$ to $\boldsymbol{y}_k^t$. Model parameters θ_k^t are used to describe the mappings. $f(\boldsymbol{x}_k, \boldsymbol{y}_k; \theta_k^t)$ is the loss function used to capture the error of the mappings. Each user performs the machine learning task locally aiming to solve the following problem [12]:

$$\min_{\boldsymbol{\theta}_k^t} F_k(\boldsymbol{\theta}_k^t) = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} f(\mathbf{x}_k^t(i), \mathbf{y}_k^t(i); \theta_k^t), \quad (1)$$

In what follows, we remove the index of t as the equation is true for each round. $|\mathcal{D}_k|$ is the cardinality of the dataset on user k.

FL training relies on the distributed stochastic gradient descent (DSGD) [13] using dataset $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$ across K different devices. The loss function in (1) can be generalized as:

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} F_k(\boldsymbol{\theta}_k), \quad (2)$$

where $\boldsymbol{\theta}$ is the global model that generated from sub-model $\boldsymbol{\theta}_k$, $|\mathcal{D}| = \sum_{k=1}^{K} |\mathcal{D}_k|$.

As shown in Fig. 1, each round of the FL process starts with the downlink communication for sharing central model $\boldsymbol{\theta}$, followed by the learning process at local devices to obtain $\boldsymbol{\theta}_k$, and ends with the uplink communication from device k to the PS for $\boldsymbol{\theta}_k$ transmission. For uplink, we apply NOMA scheme that allows multiple distributed devices to update simultaneously.
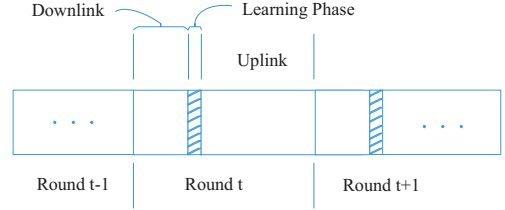

Fig. 1. One round of the FL process

In our system, there are a total of M edge devices connected to the PS. The maximum number of devices that can be scheduled to participate model update in NOMA is K. The total number of iterations or rounds for the training model to converge is T. Let $\mathcal{M}$ be the set of all the devices, $\mathcal{K}$ be the set of devices for model update and $\mathcal{T}$ be the set of all the rounds. Usually the number of devices participating the model updating is much smaller than the total number of devices connected to PS, due to the bandwidth limitation and signaling overhead, i.e., $M \gg K$. With the existence of massive devices, for the sake of fairness, each device is scheduled to participate the model update at most once. We also assume $M \geq K \times T$.

Fig. 2 gives the system model of the FL update. At each round, only the right side K devices are scheduled to upload their model update while all the M devices receive the aggregated model from the PS.
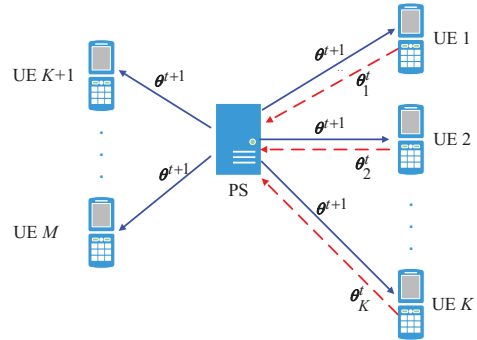

Fig. 2. FL Update Model

At the beginning, PS initializes the model as $\boldsymbol{\theta}^0$ and broadcasts it to all the users. Each user performs the local training task and calculates the gradient $\mathbf{g}_k = \nabla F_k(\boldsymbol{\theta})$ by using its local data. In the round t, user k calculates $\boldsymbol{\theta}_k^t = \boldsymbol{\theta}_k^t - \eta \nabla F_k(\boldsymbol{\theta})$ to get gradients $\mathbf{g}_k$, where η is the learning rate. All the scheduled users then send their gradients to the PS for aggregation. The PS further calculates $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \nabla F_k(\boldsymbol{\theta})$ and sends $\boldsymbol{\theta}^{t+1}$ to all the users for the next round update. This

so-called *FedAvg* learning process continues until the training on the model converges [5].

A. Uplink NOMA Transmission

NOMA allows multiple devices to transmit on the same channel simultaneously. We consider a practical fading channel in typical wireless settings. The channel gain of device k at round t is $h_k^t = L_k^t h_0^t$, which is considered constant during each t but varies across different rounds. L_k^t is the large-scale fading and h_0^t is the small-scale fading. L_k^t follows the free-space path loss model $L_k^t = \frac{\sqrt{\delta_k^t}\lambda}{4\pi d_k^{\alpha/2}}$, δ_k^t is the transmitter and receiver antenna gain at t, λ is the signal wavelength, d_k is the distance between user k and the PS, and α is the path-loss exponent. Small-scale factor h_0^t is a normal Gaussian variable, i.e., $h_0^t \sim \mathcal{CN}(0,1)$. The transmit power of device k at round t is denoted as p_k^t, $p_k^t \leq p_k^{t\,\max}$, where $p_k^{t\,\max}$ is the maximum transmission power. Let $\mathbf{s_k^t}$ be the encapsulated gradient update from user k at round t. For simplicity, we normalize the transmitted symbols $||\mathbf{s_k^t}||_2^2 = 1$. Due to the superposition nature of the transmitted signal in NOMA, the received signal at the PS at round t thus can be expressed as:

$$y^t = \sum_{k=1}^{K} \sqrt{p_k^t} h_k^t s_k^t + n^t, \tag{3}$$

where $n^t \sim \mathcal{CN}(0,\sigma^2)$ is the additive noise.

NOMA applies SIC at PS side to decode the signals from different devices sequentially. The decoding process starts with the strongest signal first by regarding other signals as interference. After successful decoding, PS subtracts the decoded signal from the superposed signal and proceeds to decode the next strongest signal. This process continues until the PS decodes all the signals. Without loss of generality, we assume $p_1^t(h_1^t)^2 > p_2^t(h_2^t)^2 > \cdots > p_K^t(h_K^t)^2$. Therefore, the signal-to-noise-plus-interference ratio (SINR) of user k at round t, γ_k^t is :

$$\gamma_k^t = \frac{p_k^t(h_k^t)^2}{\sum_{j=k+1}^{K} p_j^t(h_j^t)^2 + \sigma^2}, \forall k = \{1,\ldots,K-1\}. \tag{4}$$

The achievable data rate for user k in round t becomes [14]:

$$R_k^t = \log_2\{1+\gamma_k^t\}, \forall k = \{1,\ldots,K-1\}. \tag{5}$$

Data rate of the last decoded user K is $R_K^t = \log_2(1 + \frac{p_K^t(h_K^t)^2}{\sigma^2})$.

B. Adaptive Model Compression

Interference exists within each uplink NOMA group, which inevitably impacts the signal quality of different devices. Data rate of each user in a NOMA-based dense wireless network can thus be limited, which may hamper the model update accuracy at each round. A common approach allows each device to further compress their model to alleviate this limitation. Since the channel coefficients may change across different communication rounds, adaptive model compression is applied here. Standard machine learning techniques typically use a 32-bit floating point number to represent each model parameter.

However, the gradients in machine learning tasks are usually in the range $[-1, 1]$ or in a even smaller range. So less bits can be used to represent the gradients and help reduce the model size. Here limited-bit quantization is applied. DoReFa scheme [15] is suitable for quantizing gradients within $[-1, 1]$. The mapping between full-bit number and less-bit number is established as $q_k(\pi) = \frac{1}{a}\lfloor a\pi \rceil$, $\lfloor \cdot \rceil$ maps to the nearest integer, π is the full-bit gradient value, and $a = 2^b - 1$, where b is the quantization bit length.

We employ adaptive compression to meet different rate limitations. The compression rate r_k for user k can be calculated as $r_k = \max\{\frac{I}{c_k}, 1\}$, I is the total bit length of gradients, $c_k^t = R_k^t t$ is the allowable transmission bit length for user k. The quantization bit length b_k is calculated by $b_k = \lfloor \frac{1}{r_k} 32 \rfloor$, $\lfloor \cdot \rfloor$ is the floor operation. Further, the compression rate r_k may vary in different rounds, so we can use the average compression rate to represent the compression performance. **Algorithm 1** summarizes the proposed adaptive aggregation scheme with compression applied.

Algorithm 1 FL Adaptive Model Aggregation under NOMA

1: **Initialization:** $\boldsymbol{\theta}^0$, T.
2: **for** each FL update round t **do**
3:     PS sends $\boldsymbol{\theta}^t$ to all users then selects K users.
4:     **for** each selected user k in parallel **do**
5:         Calculate local gradients: $\boldsymbol{\theta}_k^t = \boldsymbol{\theta}_k^t - \eta\nabla F_k(\boldsymbol{\theta})$.
6:         Apply quantization on gradients.
7:         Send gradients $\nabla F_k(\boldsymbol{\theta})$ to the PS.
8:     **end for**
9:     PS applies SIC to decode gradient from K users.
10:     PS performs weighted average to get updated model: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta\sum_{k=1}^{K}\frac{|\mathcal{D}_k|}{|\mathcal{D}|}\nabla F_k(\boldsymbol{\theta})$.
11: **end for**

C. Problem Formulation

Here we provide the formulated optimization problem with the following three constraints considered in our system model.

- $C1$: Each device can be scheduled at most once across different rounds.
- $C2$: At most K devices are allowed to participate the FL update in each round under NOMA.
- $C3$: Transmission power of each device in each round is bounded by a maximum value.

We aim to maximize a weighted sum rate of all participated devices, the optimization problem is formulated as

$$\max \sum_{m,t} w_m^t \Lambda_m^t R_m^t \tag{6a}$$

$$s.t. \quad \sum_t \Lambda_m^t \leq 1, \forall m, \tag{6b}$$

$$\sum_m \Lambda_m^t \leq K, \forall t, \tag{6c}$$

$$0 \leq p_m^t \leq p_m^{t\,\max}, \forall(m,t) \in \mathcal{M} \times \mathcal{T}, \tag{6d}$$

$$\Lambda_m^t \in \{0,1\}, \forall(m,t) \in \mathcal{M} \times \mathcal{T}, \tag{6e}$$

where w_m^t is the data rate weight of device m scheduled at round t. In FL, PS performs weighted average to generate the current global model, hence a natural selection for the data rate weight can be $w_m^t = \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$, which also clearly outlines the significance of each device's update. $\Lambda_m^t = \{0, 1\}$ is a binary variable that equals 1 if device m is scheduled at t and is 0 otherwise. Here, the constraint in (6b) corresponds to constraint $C1$, constraint in (6c) corresponds to constraint $C2$ and constraint in (6d) corresponds to constraint $C3$. Finding the maximum weight sum data rate under these constraints involves traversing all possible scheduling patterns, which possess very high complexity when the number of total devices is large and selected devices for scheduling is small, i.e., $M \gg K$. Towards that, we propose the following scheduling algorithm to address this complexity issue and power allocation to solve the optimization problem (6a).

III. SCHEDULING ALGORITHM AND POWER ALLOCATION

Fig. 3 shows the diagram of the user scheduling. Each column represents a FL round for model update, and there are a total of T columns. Each block in a specific column represents a scheduled user and at most K users are scheduled to participate FL update in each round. The power of the scheduled user k in round t is p_k^t. $(i_1, i_2, \ldots, i_K)$, $(j_1, j_2, \ldots, j_K)$ and $(l_1, l_2, \ldots, l_K)$ are different user combinations.

Fig. 3. Scheduling Diagram

For the proposed joint scheduling and power allocation scheme, first, all possible user schedules are found. Then optimal power allocation is applied for each schedule to find the optimal one. The scheduling problem which aims to maximize weighted sum rate is transformed under graph theory. Specifically, we introduce the maximum weight independent set problem first. An independent set is a sub-graph of an undirected graph where there exists no edge between any two vertices. When the weight of each vertex is set to be equal to the sum data rate of users scheduled in the specific round, the sum of the weight of all vertices in an independent set equals to the sum data rate of a possible user schedule. The maximum weight independent set then corresponds to the schedule pattern that maximizes the sum data rate. The maximum weight independent set problem involves searching for all possible independent sets and then finding the maximum weight one. Thus a critical step is to construct the scheduling graph in order to find all the scheduling patterns.

A. Scheduling Graph Construction

Let $\mathcal{S}$ be the set that includes all the possible scheduling patterns for all the devices and rounds. $s \in \mathcal{S}$ is a possible schedule. The scheduling graph can be constructed as follows. First, we need to generate vertices. In this graph, a vertex $v_j = (j_1, j_2, \ldots, j_K)t$ indicates that devices $j_1, j_2, \ldots, j_K$ are scheduled at time t. There are a total of $\binom{M}{K} \times T$ vertices. When creating the edges, the following constraints need to be satisfied.

- $C1$: Each device can be scheduled at most once.
- $C2$: At most K devices can be scheduled in one round.

For two vertices $v_i = (i_1, i_2, \ldots, i_K)t_i$ and $v_j = (j_1, j_2, \ldots, j_K)t_j$, if $i_k \in \{j_1, j_2, \ldots, j_K\}, \forall k = \{1, \ldots K\}$ (violates $C1$) or $t_i = t_j$ (violates $C2$), v_i and v_j are connected and an edge exists between these two vertices. Then when we select vertices from independent set, both $C1$ and $C2$ will be satisfied. Let us construct a scheduling graph example with $M = 4$, $K = 1$, and $T = 2$, as shown in Fig. 4. In this case there are $\binom{4}{1} \times 2 = 8$ vertices. From this figure, we can find out that the possible independent sets for vertex $(1)1$ (green node) is $\{\{(1)1, (2)2\}, \{(1)1, (3)2\}, \{(1)1, (4)2\}\}$. Similarly, we can find all the independent sets for each vertex in the graph. Because of the edge connection constraints, each independent set has at most T vertices. Since the FL rounds are continuous and the number of FL rounds is T, the independent sets with T vertices are only considered.
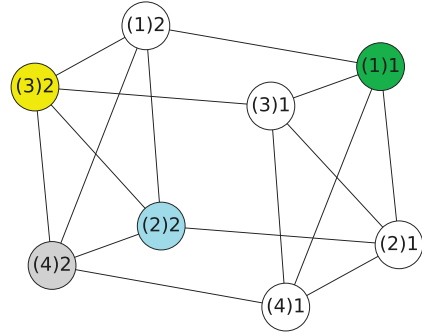

Fig. 4. A Scheduling Graph Example

B. Optimal scheduling Pattern

When scheduling graph is constructed, we calculate the weight of each vertex as sum data rate of users scheduled in a specified round, that is

$$w(v_j) = \sum_{k \in v_j} w_k^t R_k^t, \forall t \in s. \tag{7}$$

Then the sum of the weight of all vertices in an independent set equals the sum data rate of a possible schedule, that is

$$\sum_j w(v_j) = \sum_{k,t} w_k^t R_k^t, \forall (k, t) \in S. \tag{8}$$

where v_j represents vertex in an independent set.

The objective function in (6a) is actually equal to the problem maximizing the (8), which is the maximum weight

independent set problem. The maximum weight sum rate problem then can be transformed as a maximum weight independent set problem. And the optimal schedule can be selected in the **Algorithm 2**: here, $\mathbf{O}$ is the maximum weight independent

---

Algorithm 2 Optimal Scheduling Selection

1: **Require:** $\mathcal{M}, \mathcal{K}, \mathcal{T}, p_m^t$, and h_m^t.
2: Initialize $\mathbf{O} = \emptyset$
3: Construct scheduling graph G
4: Compute $w(v), \forall v \in G$
5: **while** $G \neq \emptyset$ **do**
6:     $Q = \left\{ v | w(v) \geq \sum_{u \in J(v)} \frac{w(u)}{\beta(u)+1} \right\}$
7:     Select $v^* = \arg\max_{v \in Q} \frac{w(v)}{\beta(v)+1}$
8:     Set $\mathbf{O} = \mathbf{O} \cup \{v^*\}$
9:     Set $G = G - J(v^*)$
10: **end while**
11: Output $\mathbf{O}$

---

set in the graph, which is the schedule pattern corresponding maximum weight sum data rate. $J(v)$ is the sub-graph of G containing vertex v and the vertices adjacent to v, $\beta(v)$ is the degree of v, which is the number of vertices adjacent to v. Q is the set of vertices where the weight of vertex v is larger than the average weight of $J(v)$. v^* is selected by making the average weight of $J(v)$ maximization.

C. Power Allocation

Once the user scheduling is determined, device power can be allocated according to the channel condition to achieve the maximum sum data rate. Power allocation in NOMA has been extensively investigated in the existing works. To achieve the maximum sum data rate under fairness constraints, a similar algorithm to [8] is used here. We notice that the objective function (6a) as a logarithmic function of SINR is monotonically increasing. It can be transformed into a product of exponential linear fraction functions. Due to the properties of logarithm function, the optimal power allocation problem for a specified user combination is

$$\max \prod_{k=1}^{K} \left(\frac{\mu_k(\mathbf{p})}{\phi_k(\mathbf{p})} \right)^{w_k}, \tag{9a}$$

$$s.t. \quad 0 \leq p_k \leq p_k^{\max}, \forall k \in \mathcal{K}. \tag{9b}$$

where $\mathbf{p} = (p_k, \forall k \in \mathcal{K})$ is the power vector, $\mu_k(\mathbf{p}) = \sum_{j=k}^{K} p_j h_j^2 + \sigma^2$ and $\phi_k(\mathbf{p}) = \sum_{j=k+1}^{K} p_j h_j^2 + \sigma^2$. Let $\mathbf{z}_k = \frac{\mu_k(\mathbf{p})}{\phi_k(\mathbf{p})}$ for all k, the problem then can be re-formulated as

$$\max \prod_{k=1}^{K} (\mathbf{z}_k)^{w_k} \tag{10a}$$

$$s.t. \quad 0 \leq \mathbf{z}_k \leq \frac{\mu_k(\mathbf{p})}{\phi_k(\mathbf{p})}, \forall k \in \mathcal{K}, \tag{10b}$$

$$0 \leq p_k \leq p_k^{\max}, \forall k \in \mathcal{K}. \tag{10c}$$

Notice that $\tau(\mathbf{e}) = \prod_{k=1}^{K} (e_k)^{w_k}$ is an increasing function for all positive e_k, where $\mathbf{e}$ is the collection of all e_k. Besides, for two vectors $\mathbf{e}_l$ and $\mathbf{e}_m$, if $\mathbf{e}_l \succeq \mathbf{e}_m$, where $\succeq$ means element-wise greater than, we have $\tau(\mathbf{e}_l) > \tau(\mathbf{e}_m)$. Clearly, the optimal solution occurs where $\mathbf{z}_k^* = \frac{\mu_k(\mathbf{p}^*)}{\phi_k(\mathbf{p}^*)}$, and p_k in the feasible set. This can be regarded as a multiplicative linear fractional programming (MLFP) problem, where K linear equations are formulated as below:

$$z_k^* \phi_k(\mathbf{p}^*) - \mu_k(\mathbf{p}^*) = 0, \forall k \in \mathcal{K}. \tag{11}$$

Notice that (11) contains random channel gain components hence those K linear equations are independent with probability 1, which suggests a unique optimal power allocation $\mathbf{p}^*$. To solve (11) efficiently, however, requires constructing of feasible polyblock and sequentially reduce its size, see [8] for the detailed algorithm.

IV. SIMULATION RESULTS

This section first gives simulation results to compare two schemes, namely the TDMA based FedAvg scheme [5] and NOMA compression based FedAvg scheme. Both schemes use the maximum power transmission for all the devices thus no power control is applied on the uplink. After that we compare the performance for the following four schemes, 1) the scheme using optimal joint scheduling with power allocation (our proposed scheme), 2) the scheme using the optimal scheduling but with no power control (all the devices transmit at the maximum power), 3) the scheme using a random scheduling with optimal power allocation, and 4) the scheme using random scheduling with no power control (maximum power transmission). All the simulations run using image recognition as the learning task trained on the MNIST (Modified National Institute of Standards and Technology) dataset [16]. Testing accuracy, which is defined as number of correct predictions divided by total number of predictions, is used to measure the performance of all the schemes. A fully connected neural network called LeNet-300-100 with two hidden layers is used, which has 300 neurons in the first layer and 100 neurons in the second layer. Thus the total number of model parameters including bias is $266,610$. The system parameter settings are given as follows. The uplink bandwidth is $B = 4$ MHz, path loss exponent is $\alpha = 3$, additive noise power density is $\sigma^2 = -174$ dBm/Hz. The total number of user is $M = 300$ and the number of model update user in each round is $K = 3$. The maximum transmission power of each user is $p^{\max} = 0.01$ watts. Cell size of PS is 500 m. Users are uniformly distributed in the cell and the positions of users are fixed during the learning process. Uplink transmission time slot is $t = 0.2$ s. For downlink transmission from PS, FL uses broadcast with no compression. Transmission time is $T_d = \max_k \frac{I}{B_d \log_2(1+p_d\gamma_k)}$, where I is the total bit length of model, B_d is the downlink bandwidth and is 10 MHz. $p_d = 0.2$ watts is the PS transmission power, γ_k is the SINR from the PS to k-th user.

The hyperparameters are given in Table I. The learning phase is partitioned into training and testing stages at each

device. Also the dataset are split into training and testing sets correspondingly, which are shown in Table I, where 90% samples belong to the training set and the the rest belong to the testing set. To make the model more general and robust, data are made non-i.i.d across different devices, i.e., the sizes and distributions of data at each device are both different. To evaluate the model validation, in every communication round, each device first does the training based the received model from the PS and local data, followed by the testing process. With iterative learning, more and more data are fed into the model so that the testing accuracy keeps increasing.

TABLE I
HYPERPARAMETERS

Learning rate size (η)	Batch size ($\mathcal{B}$)	FL Round (T)	Training set size	Testing set size
0.01	10	35	90%	10%

We first demonstrate that NOMA compression based FedAvg achieves better performance than the traditional TDMA based FedAvg. As said, both schemes use the maximum power transmission for all the devices thus no power control is applied on the uplink. In the NOMA based scheme, quantization is used for compression while there is no compression for the TDMA based scheme. Fig. 5 shows that FL using the NOMA based scheme achieves a better testing accuracy compared with the TDMA based scheme during the same communication time. Each round takes $t_k + T_d$ time in the NOMA based scheme while it takes $Kt_k + T_d$ time for the TDMA based scheme. So for a given time, NOMA based FedAvg performs more rounds of FL training than the TDMA based FedAvg. In Fig. 5, the NOMA based FedAvg update starts to converge and achieves 70% of accuracy after 10s while the TDMA based FedAvg takes about 22s to achieve the similar accuracy.
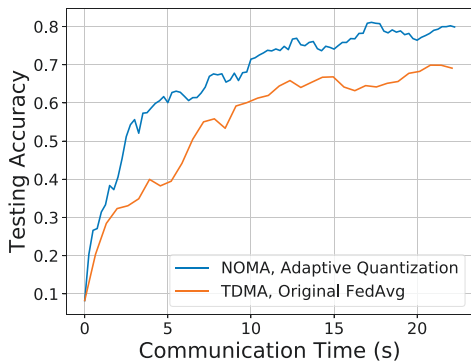

Fig. 5. Testing Accuracy vs Communication time

Fig. 6 shows the comparison among 4 different scheduling and power control schemes as defined above. It is observed that all schemes except the 4th one (random scheduling with maximum power transmission) can get above 60% testing accuracy after 35 rounds of communication/training. The optimal joint scheduling and power allocation scheme consistently achieves the best performance among all the schemes during

the entire training process. Both scheduling and power control play an important role in achieving better FL training through improving the communication quality, which leads to more accurate model update during the training process.
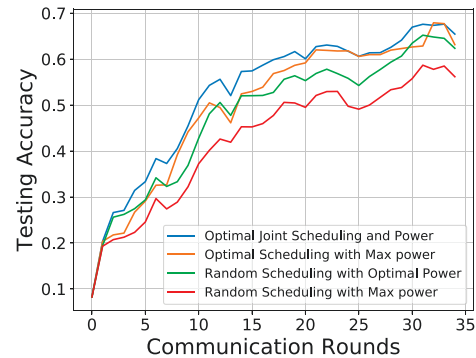

Fig. 6. Testing Accuracy vs Communication Rounds

V. CONCLUSIONS

In this work, we proposed to apply NOMA in the FL based model update. To maximize the system sum data rate, the maximum weight sum data rate problem was transformed to a maximum weight independent set problem that can be solved using graph theory based approach. The user scheduling and power allocation were employed to obtain the maximum sum data rate. NOMA based scheme can achieve similar accuracy as TDMA one while reducing the communication latency significantly. Besides, our results show that proper user scheduling and power allocation during wireless communication stage can help to get a higher testing accuracy.

REFERENCES

[1] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A Survey of Machine Learning for Big Data Processing," *EURASIP J. Adv. Signal Process.*, vol. 67, pp. 1-16, Dec. 2016.

[2] L. T. Tan and R. Q. Hu, "Mobility-Aware Edge Caching and Computing in Vehicle Networks: A Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190-10203, Nov. 2018.

[3] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling Distributed Machine Learning with the Parameter Server," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI).*, pp. 583-598, 2014.

[4] L. T. Tan, R. Q. Hu, and L. Hanzo, "Twin-Timescale Artificial Intelligence Aided Mobility-Aware Edge Caching and Computing in Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3086-3099, April 2019.

[5] J. Konečny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," [Online]: https://arxiv.org/abs/1610.05492, 2017.

[6] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling Policies for Federated Learning in Wireless Networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317-333, Jan. 2020.

[7] A. Douik, H. Dahrouj, T. Y. Al-Naffouri, and M. Alouini, "Coordinated Scheduling and Power Control in Cloud-Radio Access Networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2523-2536, April 2016.

[8] L. P. Qian, Y. J. Zhang, and J. Huang, "MAPEL: Achieving Global Optimality for a Non-convex Wireless Power Control Problem," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553-1563, March 2009.

[9] D. Zhai and J. Du, "Spectrum Efficient Resource Management for Multi-Carrier-Based NOMA Networks: A Graph-Based Method," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 388-391, June 2018.

[10] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[11] H. Sun, X. Ma, and R. Q. Hu, "Adaptive Federated Learning With Gradient Compression in Uplink NOMA," *IEEE Trans. Veh. Technol. to appear*, 2020.

[12] F. Li and Y. Yang, "A Loss Function Analysis for Classification Methods in Text Categorization," *Proc. 20th Int'l Conf. Machine Learning (ICML '03).*, pp. 472-479, 2003.

[13] L. Bottou, "Large-scale Machine Learning with Stochastic Gradient Descent," *Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186

[14] Z. Yang, Z. Ding, P. Fan, and G. K. Karagiannidis, "On the Performance of Non-orthogonal Multiple Access Systems With Partial Channel Information," *IEEE Trans. Wireless Commun.*, vol. 64, no. 2, pp. 654-667, Feb. 2016.

[15] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," [Online]: https://arxiv.org/abs/1606.06160, 2018.

[16] Y. Lecun and C. Cortes, "The MNIST Database of Handwritten Digits," [Online]: http://yann.lecun.com/exdb/mnist/, 1998.