Dynamic Client Association for Energy-Aware Hierarchical Federated Learning

Bo Xu[†], Wenchao Xia[†], Jun Zhang[†], Xinghua Sun[§] and Hongbo Zhu[†]
[†]Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications
Nanjing 210003, China, e-mail: {2019010209, xiawenchao, zhangjun, zhuhb}@njupt.edu.cn
[§]Sun Yat-sen University, Guangzhou 510006, China, e-mail: sunxinghua@mail.sysu.edu.cn

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417267

***Abstract*—Federated learning (FL) has become a promising solution to train a shared model without exchanging local training samples. However, in the traditional cloud-based FL framework, clients suffer from limited energy budget and generate excessive communication overhead on the backbone network. These drawbacks motivate us to propose an energy-aware hierarchical federated learning framework in which the edge servers assist the cloud server to migrate the local models from the clients. Then a joint local computing power control and client association problem is formulated in order to minimize the training loss and the training latency simultaneously under the long-term energy constraints. To solve the problem, we recast it based on the general Lyapunov optimization framework with the instantaneous energy budget. We then propose a heuristic algorithm, which takes the importance of local updates into account, to achieve a suboptimal solution in polynomial time. Numerical results demonstrate that the proposed algorithm can reduce the training latency compared to the scheme with greedy client association and myopic energy control, and improve the learning performance compared to the scheme in which the associated clients transmit their local models with the maximal power.**

I. Introduction

Current approaches to training large-scale models are usually directly implemented at the cloud server, which has a powerful computation capacity and collects training samples from widely distributed mobile devices [1, 2]. However, since the training samples of mobile devices is private in nature, the data transmission between the mobile devices and the cloud server will cause the increasing privacy concern. Besides, the direct transmission of training samples between the clients and the cloud server can generate a lot of communication overhead For this reason, a recent approach called federated learning (FL) is proposed [3]. In FL, the mobile devices, referred to as clients, periodically download a global model from the cloud server, and then update and upload the local models to the cloud server to perform the global model aggregation and build an updated global model. This process is repeated until the global model is convergent.

However, an FL algorithm is trained in a distributed manner, and the clients must perform local model updates and transmit the local models over wireless links which can degrade the learning performance and introduce the training latency, due to limited computation and communication resources. Recently, the challenge has attracted attention form both academia and industry, and various client scheduling and resource allocation schemes have been proposed. For instance, in [5], a fair client scheduling policy was studied to make each client participate in a certain number of the training rounds. To minimize the training loss and meet the computation and communication resource constraints, a joint data sampling and client scheduling optimization was proposed in [6]. In terms of resource allocation, the authors in [7] achieved the tradeoff between the training latency and the energy consumption with a fixed set of the participating clients. Note that these works mainly focused on the cloud-based FL framework, in which the clients directly communicated with the cloud server and could cause excessive communication overhead on the backbone network.

Motivated by these facts, hierarchical FL framework turns out to be a promising solution by leveraging the power of Mobile Edge Computing (MEC) [8–10], in which the edge servers were fixedly deployed as intermediaries between the clients and the cloud server in order to assist the cloud server to collect the local models from the clients to perform the edge aggregation. The authors in [8] summarizes and looks forward to the hierarchical FL framework. In [9], the authors designed an MEC based FL framework, and formulated a joint resource allocation and client association problem to minimize the global cost. Besides, the convergence analysis of the hierarchical FL framework was studied in [10]. Generally, the benefits of a hierarchical FL framework include reducing the data traffic on the backbone network and increasing the reliability of communication due to a proximate access. However, there still have two practical challenges in the hierarchical FL framework: 1) *how to design the client association strategy for higher learning performance and lower training latency?* With limited spectrum resources, associating all the clients to the edge servers can significantly increase the transmission latency between the clients and the edge servers. Besides, in the scenario where the distribution of the local datasets is non-independent and identically distributed (non-i.i.d.) [11], the clients with different local updates are not equally important to the training. Recently, some works have proposed some scheduling criterions such as gradient variance, model variance, and local loss [12–14] to quantify the importance. As a result, from the perspective of the learning performance, it is better to associate more important clients, that let latency

This work of B. Xu, W. Xia, J. Zhang, and H. Zhu was supported in part by the National Natural Science Foundation of China (Grant No. 61871446, U1805262, 61771252, 61701201, and 62071247), and in part by the National Key Research and Development Program (No.2020YFB1806608 and 2019YFB2103004). This work of X. Sun was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515011906, and in part by the Open Project of the Key Laboratory of Wireless Sensor Network and Communication, Chinese Academy of Sciences under Grant 20190901.

and energy performance might be degraded. Therefore, it is of significant importance to strike a balance. 2) *how to meet the energy budget for the energy-aware clients after multiple rounds of training?* The clients often have finite energy budget such as a finite battery to participate in training. However, in the hierarchical FL framework, the clients have to perform repeated local computation and model transmission per round. Therefore, to meet the energy budget, the number of training rounds each client can participate is limited, and it is difficult to determine how much energy is allocated for training in each round To deal with the mentioned challenges, in this paper, we proposed EHFL, a novel energy-aware hierarchical framework for high-performance and low-latency FL. Instead of associating all the clients to participate in training [10], we prefer to associate the clients with more important local model updates so as to mitigate the learning performance degradation in the case of non-i.i.d. local datasets. Then a joint problem of resource allocation and client association is formulated based on the EHFL framework in order to minimize the training latency while achieving a target training loss and satisfying the long-term energy budget constraints of the individual clients. Since this optimization problem involving the long-term energy budget of the clients, we have to consider the future conditions effected by the strategy in the current round. Thanks to the development of the wildly used stability theory of Lyapunov optimization framework [15–17], we recast the problem into a queue-aware radio resource allocation problem with the instantaneous energy budget. We then propose a heuristic joint local computing power control and client association algorithm to achieve a good balance between the optimal value of the objective function and the implementation complexity.

II. System model

A. Learning model

As illustrated in Fig. 1, the proposed EHFL framework consists of a set $\mathcal{K}$ of K clients, a set $\mathcal{S}$ of S edge servers, and one cloud server. Each client k stores a local dataset $\mathcal{D}_k = \{x_{k,l}, y_{k,l}\}_{l=1}^{D_k}$, where $x_{k,l}$ is the l-th training data sample, $y_{k,l}$ is the corresponding ground-truth label, and D_k is the number of samples that client k owns. The model parameter is denoted as $\boldsymbol{\theta}$, which is trained collaboratively across the clients by leveraging their distributed local datasets. We introduce a loss function $\ell(\boldsymbol{\theta}, x_{k,l}, y_{k,l})$ to quantify the error between the data sample $x_{k,l}$ and its label $y_{k,l}$, and the loss function of the client k is denoted as $F_k(\boldsymbol{\theta}) = \frac{1}{D_k}\sum_{l=1}^{D_k} \ell(\boldsymbol{\theta}, x_{k,l}, y_{k,l})$. Accordingly, the global loss function minimization problem is formulated as

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \frac{1}{D}\sum_{k=1}^{K} D_k F_k(\boldsymbol{\theta}), \tag{1}$$

where $D = \sum_{k=1}^{K} D_k$ is the total number of training samples of all the clients. To solve the problem in (1), we adopt an iterative approach in each round, in which the number of training rounds is defined as the times that the global

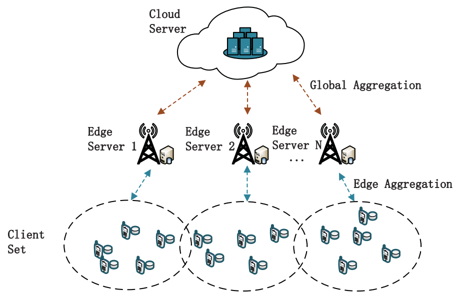

Fig. 1. The energy-aware hierarchical federated learning (EHFL) framework,

model aggregation has been performed. As shown in Fig. 1, after every κ_1 local updates, the clients upload their local models to the corresponding edge servers to perform the edge aggregations. Besides, after every κ_2 edge model aggregations, all edge servers upload their updated edge models to the cloud server via wireless backhaul links to perform the global model aggregation. As a result, the clients need to perform $\kappa_1\kappa_2$ local model updates in each round. We further define the local model of client k in the j-th iteration in round t as $\boldsymbol{\theta}_{k,j}(t)$, where $1 \le j \le \kappa_1\kappa_2$. In addition, to associate a proper set of client to the edge servers, we define the client association strategy as $\mathcal{A}(t) = \{a_{k,s}(t) | k \in \mathcal{K}, s \in \mathcal{S}\}$, where $a_{k,s}(t)$ is the client association strategy of client k. Specifically, $a_{k,s}(t) = 1$ indicates that client k uploads its local model updates to edge server s, and $a_{k,s}(t) = 0$ otherwise. Then the set of the associated clients that upload their local updates to server s is denoted as $\mathcal{K}_s(t) = \{k \in \mathcal{K} | a_{k,s}(t) = 1\}$, and the set of all the associated clients is denoted as $\mathcal{Q}(t) = \bigcup_{s=1}^{S} \mathcal{K}_s(t)$. Finally, according to the number of performing local updates, the evolution of $\boldsymbol{\theta}_{k,j}(t)$ is as the following three cases.

- Case 1: $j|\kappa_1 \neq 0$. The $\boldsymbol{\theta}_{k,j}(t)$ is updated by local computation: $\boldsymbol{\theta}_{k,j}(t) = \boldsymbol{\theta}_{k,j-1}(t) - \lambda\nabla F_k(\boldsymbol{\theta}_{k,j-1}(t))$, where $\lambda > 0$ is learning rate .
- Case 2: $j|\kappa_1 = 0$ and $j|\kappa_1\kappa_2 \neq 0$. The $\boldsymbol{\theta}_{k,j}(t)$ is updated by performing the edge model aggregation on server s: $\boldsymbol{\theta}_{k,j}(t) = \frac{\sum_{i\in\mathcal{K}_s(t)} D_i[\boldsymbol{\theta}_{i,j-1}(t) - \nabla F_i(\boldsymbol{\theta}_{i,j-1}(t))]}{\sum_{i\in\mathcal{K}_s(t)} D_i}$.
- Case 3: $j|\kappa_1\kappa_2 = 0$. The $\theta_{k,j}(t)$ is updated by performing the global model aggregation on the cloud server: $\boldsymbol{\theta}_{k,j}(t) = \frac{\sum_{i\in\mathcal{Q}(t)} D_i[\boldsymbol{\theta}_{i,j-1}(t) - \lambda\nabla F_i(\boldsymbol{\theta}_{i,j-1}(t))]}{\sum_{i\in\mathcal{Q}(t)} D_i}$.

To further improve the computation and communication efficiency of the proposed EHFL framework, we need to consider the training latency and the energy consumption caused by the local computation and the model transmission.

B. Computation and Transmission Model

1) Computation model: In round t, let $\mathcal{F} = \{f_k(t) | k \in \mathcal{K}\}$ denote the strategy of the local computing power control, where $f_k(t)$ is the local computing power of client k. Accordingly, the computation latency for client k to perform κ_1 local model updates is given by

$$T_k^{\mathrm{e}}(t) = \frac{\kappa_1 D_k C_k}{f_k(t)}, \tag{2}$$

where C_k is the number of CPU cycles required to compute one training sample of client k. We then define the energy consumption for per cycle as $\varrho f_k^2(t)$ [9], where ϱ is a coefficient based on chip architecture of CPU frequency. Hence, the energy consumption of client k to perform κ_1 local model updates is denoted as

$$E_k^{\mathrm{e}}(t) = \kappa_1 D_k C_k \varrho f_k^2(t). \tag{3}$$

2) Transmission Model: In round t, after every κ_1 local model updates, the associated clients upload their local models to the corresponding edge servers via the orthogonal frequency division multiple access (OFDMA). The system bandwidth is denoted by B, and the bandwidth allocated to the client $k \in \mathcal{Q}(t)$ is computed as $b_k(t) = \frac{B}{|\mathcal{Q}(t)|}$. Then the uplink rate between client k and edge server s is denoted as $r_{k,s}(t) = b_k(t)\log_2(1 + \frac{p_k(t)h_{k,s}(t)}{b_k(t)N_0})$, where $p_k(t)$ is the transmit power of client k, N_0 is the power spectral density of the Gaussian noise, and $h_{k,s}(t)$ is the average channel gain between client k and edge server s. Hence, we can define the transmission latency of client k for a local model update as

$$T_k^{\mathrm{u}}(t) = \sum_{s=1}^{S} \frac{a_{k,s} M}{r_{k,s}(t)}, \tag{4}$$

where M is the size of a local model update. The corresponding energy consumption incurred by model transmission is given by

$$E_k^{\mathrm{u}}(t) = \frac{b_k(t) T_k^{\mathrm{u}}(t) N_0}{\sum_{s=1}^{S} a_{k,s} h_{k,s}(t)} \left(2^{\frac{M}{b_k(t) T_k^{\mathrm{u}}(t)}} - 1\right). \tag{5}$$

After receiving the local models from the clients, according to the number of times of local model updates, each edge server s can broadcast the updated edge model to the clients in $\mathcal{K}_s(t)$ or upload the updated edge model to the cloud server via wireless backhaul links. In the former case, we assume that the broadcast channels of edge servers are orthogonal to each other. The broadcast rate between server s and client k is denote as $\widetilde{r}_{s,k}(t)$. Hence, the broadcast latency of the server s in round t is denoted as

$$\widetilde{T}_s^{\mathrm{b}}(t) = \frac{M}{\min_{k \in \mathcal{K}_s(t)} \widetilde{r}_{s,k}(t)}. \tag{6}$$

In the latter case, the transmission latency of edge server s can be derived as

$$\widetilde{T}_s^{\mathrm{u}}(t) = \frac{M}{\widetilde{r}_s^{\mathrm{u}}(t)}, \tag{7}$$

where $\widetilde{r}_s^{\mathrm{u}}(t)$ denotes the uplink rate of edge server s. After receiving the edge models, the cloud server updates and broadcasts the global model $\boldsymbol{\theta}(t+1)$ to all the clients for the next round of training. We can simply denote the broadcast latency of the cloud server as $T^{\mathrm{c}}(t)$.

C. Problem Formulation

We utilize the synchronous model aggregation in the proposed framework, and we require that all the associated clients to start model training simultaneously at the beginning of each round. Hence, the training latency in round t is denoted as

$$\begin{aligned} T^{\mathrm{r}}(t) &= T^{\mathrm{c}}(t) \\ &+ \max_{s \in \mathcal{S}} \left(\widetilde{T}_s^{\mathrm{u}}(t) + \kappa_2 \max_{k \in \mathcal{K}_s(t)} \left(T_k^{\mathrm{e}}(t) + T_k^{\mathrm{u}}(t) + \widetilde{T}_s^{\mathrm{b}}(t) \right) \right). \end{aligned} \tag{8}$$

Accordingly, the total energy consumption of client k is expressed as $E_k(t) = \kappa_2 \left(E_k^{\mathrm{e}}(t) + E_k^{\mathrm{u}}(t)\right)$. Since the proposed hierarchical FL algorithm requires multiple rounds of training, we emphasize the long-term energy consumption of the clients, and the average energy consumption of client k cannot exceed the energy budget $\bar{E}_k$, i.e., $\frac{1}{L}\sum_{t=1}^{L} E_k(t) \le \bar{E}_k$, where L is the number of training rounds. As mentioned before, our goal is to minimize the training loss and the training latency under the long-term energy constraints. However, since the training loss cannot be written in a closed expression and can only be observed when the global model is updated, it is hard to minimize the training loss by the traditional optimization method. Moreover, in a normal scenario where the distribution of the local datasets is non-i.i.d., the clients with different training samples are not equally important to the decrease of the training loss. Thus, it is a challenge for the edge servers to associate a proper set of the clients for the training. To solve these issues, recently works in [12–14] have shown that associating clients who have more important local model updates can significantly increase the learning performance. Inspired by these works, in this paper, we utilize the importance of local model updates to quantify the training loss.

Definition 1: In round t, given $\boldsymbol{\theta}(t)$, the importance of local model updates of client k is denoted as

$$\alpha_k(\boldsymbol{\theta}(t)) = e^{\frac{1}{\sqrt{D_k}} \sum_{l=1}^{D_k} \ell(\boldsymbol{\theta}(t), x_{k,l}, y_{k,l})}. \tag{9}$$

At the beginning of each round, based on the updated global model, all the clients first evaluate and upload their importance of local model updates to the cloud server for the resource allocation and the client association. Since $\alpha_k(\boldsymbol{\theta}(t))$ is only a scalar and can be easily evaluated without excessive computation, its computation latency and transmission latency can be ignored. Finally, we formulate the optimization problem as

$$\max_{\{\mathcal{A}(t), \mathcal{F}(t)\}} \frac{1}{L} \sum_{t=1}^{L} \left(\rho \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} a_{k,s}(t) \alpha_k(\boldsymbol{\theta}(t)) - (1-\rho) T^{\mathrm{r}}(t) \right) \tag{10a}$$

$$\frac{1}{L} \sum_{t=1}^{L} E_k(t) \le \bar{E}_k, \forall k \in \mathcal{K}, \tag{10b}$$

$$0 \le f_k(t) \le f_k^{\max}, \forall k \in \mathcal{K}, \tag{10c}$$

$$a_{k,s}(t) \in \{0,1\}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}, \tag{10d}$$

$$\sum_{s=1}^{S} a_{k,s}(t) \le 1, \forall k \in \mathcal{K}, \tag{10e}$$

where the constraint (10c) specifies the local computing power budget $f_k^{\max}$ of client k and (10e) indicates that each client k can be associated to at most one edge server in each round. Problem (10) is still difficult to be solved, because the constraint (10b) can be satisfied only if we know the CSI in all rounds, which is unpractical and infeasible. Fortunately, based on the Lyapunov optimization tools [17], this problem can be solved with instantaneous energy budget and channel information. We first construct a virtual energy queue $H_k(t)$ to capture the queue dynamics of the energy consumption, which is denoted as $H_k(t+1) = \left[H_k(t) - \bar{E}_k + E_k(t)\right]^+$. We then define the Lyapunov function as a scalar metric of queue congestion, i.e., $Q(t) = \frac{1}{2}\sum_{k=1}^{K} H_k^2(t)$, where $\mathcal{H}(t) = \{H_k(t) | k = 1, ..., K\}$ is the combined matrix of all virtual queues. Then the one-round Lyapunov drift of $Q(t)$ is defined as $\Delta_1(t) = Q(t+1) - Q(t)$. In round t, the underlying objective of the proposed resource allocation and client association strategy based on the general Lyapunov optimization framework is to maximum an infimum bound on the drift-plus-penalty expression, which is expressed as

$$V\eta(t) - \Delta_1(t), \tag{11}$$

where $\eta(t) = \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}} a_{k,s}(t)\alpha_k(\boldsymbol{\theta}(t)) - T^{\mathrm{r}}(t)$, and $V > 0$ is a parameter to control the tradeoff between the energy consumption and the optimal value of the objective function. Based on the combined matrix of all virtual queues and the Lyapunov drift defined in (11), we have the following lemma:

Lemma 1. *Assume that $H_k(1) = 0$, $\forall k \in \mathcal{K}$. We define $E^{max} = \max_{k\in\mathcal{K}} E_k(t)$, $\forall t$ and $\bar{E}^{min} = \min_{k\in\mathcal{K}} \bar{E}_k$. Then the drift-plus-penalty satisfies:*

$$V\eta(t) - \Delta_1(t) \geq V\eta(t) - \beta - \sum_{k=1}^{K} H_k(t)(E_k(t) - \bar{E}_k), \tag{12}$$

where β is a finite constant and satisfies $\beta \geq \frac{1}{2}\sum_{k=1}^{K}(E^{max} - \bar{E}^{min})^2$.

Proof: The proof is based on the Lyapunov optimization theorem and omitted here for the lack of space.

In order to maximize (11) while considering the long-term energy budget, we aim to maximize the right-hand-side (R.H.S) of (12) based on the instantaneous energy budget and CSI in each round. Then the problem to be solved in each round t is rewritten as

$$\max_{\{\mathcal{A}(t),\mathcal{F}(t)\}} V\eta(t) - \sum_{k=1}^{K} H_k(t)E_k(t) \tag{13a}$$

$$\text{s.t. } (10c) - (10e). \tag{13b}$$

The complexity to evaluate the optimal solution of (13) requires exponential time. Hence, our goal is to design a lower complexity resource allocation and client association algorithm while achieving competitive performance.

III. Problem Solution And Algorithm Design

By exploiting the structure of the objective function and the constrains in (13), we can decompose this problem into two subproblems with separated objectives: a local computing power control problem with fixed client association strategy, and an client association problem corresponding to the results of the local computing power control.

1) Local Computing Power Control: Given $\mathcal{A}(t)$, we can rewrite (13) as

$$\min_{\{\mathcal{F}(t)\}} VT^{\mathrm{r}}(t) + \sum_{k=1}^{K}\sum_{s=1}^{S} a_{k,s}(t)H_k(t)\kappa_1\kappa_2 D_k C_k \varrho f_k^2(t) \tag{14a}$$

$$\text{s.t. } 0 \leq f_k(t) \leq f_k^{\max}, \forall k \in \mathcal{K}. \tag{14b}$$

To minimize (14), we give the following theorem.

Theorem 1. *Given $\mathcal{A}(t)$, we set*

$$\hat{T}_k(t) = T^c(t) + \sum_{s=1}^{S} a_{k,s}(t)\tilde{T}_s^u(t) + \kappa_2\left(T_k^u(t) + \sum_{s=1}^{S} a_{k,s}(t)\tilde{T}_s^b(t)\right). \tag{15}$$

We then define

$$T^{min} = \max_{k\in\mathcal{K}} \sum_{s=1}^{S} a_{k,s}(t)\left(\hat{T}_k(t) + \frac{\kappa_1\kappa_2 D_k C_k}{f_k^{max}}\right), \tag{16}$$

and

$$\Upsilon(T^r(t)) = \sum_{k=1}^{K}\sum_{s=1}^{S} \frac{a_{k,s}(t)2H_k(t)\varrho\left(\kappa_1\kappa_2 D_k C_k\right)^3}{\left(T^r(t) - \hat{T}_k(t)\right)^3}, \tag{17}$$

then the optimal solution $f_k^*(t)$ in (14) satisfies

$$f_k^*(t) = \frac{\sum_{s=1}^{S} a_{k,s}(t)\kappa_1\kappa_2 D_k C_k}{T^{r,*}(t) - \hat{T}_k(t)}, \tag{18}$$

where

$$T^{r,*}(t) = \begin{cases} T^{min}, \text{ if } V > \frac{\Upsilon(T^r(t))}{1-\rho}, \\ \tilde{T}^r(t), \text{ otherwise}, \end{cases} \tag{19}$$

in which $\tilde{T}^r(t)$ is the solution of $V - \frac{\Upsilon(T^r(t))}{1-\rho} = 0$.

Proof: The proof is based on the convex optimization theory and omitted here for the lack of space.

2) Client Assocaition: Given $\mathcal{A}(t)$ and the optimal solutions of the local computing power control, the optimal value of the objective function in (13) is denoted as $\Gamma(\mathcal{A}(t))$. We can rewrite the client association problem as

$$\min_{\{\mathcal{A}(t)\}} \Gamma(\mathcal{A}(t)) \tag{20a}$$

$$\text{s.t. } a_{k,s}(t) \in \{0,1\}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S} \tag{20b}$$

$$\sum_{s=1}^{S} a_{k,s}(t) \leq 1, \forall k \in \mathcal{K}. \tag{20c}$$

Due to the nature of the client association problem, solving the optimal value of (20) in polynomial time is extremely difficult, and some common approaches such as exhaustive and bipartite matching will cause extremely complex. In order to ensure the system performance can also be applied in practice, we propose a low-complexity heuristic client association algorithm which can give a suboptimal solution in polynomial time. In round t, our algorithm starts with $a_{k,s}(t)=1, \forall k \in \mathcal{K}, s=\arg\max_{s\in\mathcal{S}} h_{k,s}(t)$, and repeatedly performs one of three operations, namely the *remove* operation, the *exchange* operation and the *add* operation. We first perform the *remove* operation to remove the clients from the current client set $\mathcal{Q}(t)$ in order improve the optimal value. We then perform the *exchange* operation which involves exchanging the client association strategy between two different clients. When there has no clients can be removed or exchanged to improve the optimal value, we then perform the *add* operation and check whether there exist other clients can be associated to the edge servers. The proposed heuristic client association algorithm is shown in **Algorithm 1**.

Algorithm 1 Heuristic client association Algorithm.

```
1: Initialize: Set $a_{k,s}(t)=1, \forall k \in \mathcal{K}, s=\arg\max_{s\in\mathcal{S}} h_{k,s}(t)$.
2: repeat: if there exits $a_{k',s'}(t)$, such that $\Gamma\left(\textbf{REMOVE}(\mathcal{A}(t), a_{k',s'}(t))\right) \geq \psi\Gamma(\mathcal{A}(t))$,
3:   Set $\mathcal{A}(t) \leftarrow \textbf{REMOVE}(\mathcal{A}(t), a_{k',s'}(t))$.
4: else if there exits $a_{k',s'}(t)$ and $a_{k'',s}(t)$, $k' \neq k''$, such that $\Gamma(\textbf{EXCHANGE}(\mathcal{A}(t), a_{k',s'}(t), a_{k'',s''}(t)) \geq \psi\Gamma(\mathcal{A}(t))$,
5:   Set $\mathcal{A}(t) \leftarrow \textbf{EXCHANGE}(\mathcal{A}(t), a_{k',s'}(t), a_{k'',s''}(t))$.
6: else if there exits $a_{k',s'}(t)$, such that $\Gamma(\textbf{ADD}(\mathcal{A}(t), a_{k',s'}(t))) \geq \psi\Gamma(\mathcal{A}(t))$,
7:   Set $\mathcal{A}(t) \leftarrow \textbf{ADD}(\mathcal{A}(t), a_{k',s'}(t))$.
8: end if
9: Until No client association adjustment is permitted, then stop and return $\mathcal{A}(t)$.
10: Function REMOVE ($\mathcal{A}(t)$, $a_{k',s'}(t)$)
11:     for $s \in \mathcal{S}$ do $a_{k',s}(t)=0$.
12:     end for
13: Return $\mathcal{A}(t)$.
14: Function EXCHANGE ($\mathcal{A}(t)$, $a_{k',s'}(t)$, $a_{k'',s''}(t)$)
15: for $s \in \mathcal{S}$ do $a_{k',s}(t)=0$, and $a_{k'',s}(t)=0$.
16:     end for $a_{k',s''}(t)=1$, and $a_{k'',s'}(t)=1$.
17: Return $\mathcal{A}(t)$.
18: Function ADD ($\mathcal{A}(t)$, $a_{k',s'}(t)$)
19:     for $s \in \mathcal{S}$ do $a_{k',s}(t)=0$.
20:     end for $a_{k',s'}(t)=1$.
21: Return $\mathcal{A}(t)$.
```

IV. Numerical Results

A. Experiment Settings

Unless otherwise specified, we consider a system where $K=50$ clients and $S=5$ edge servers are uniformly deployed in a area of size 500 m $\times$ 500 m with a cloud server at its center. The path loss model is given as $L[\text{dB}]=128.1+37.6\log_{10} d_{[\text{km}]}$, and the standard deviation of the log-normal shadowing fading is 8 dB [18]. The total bandwidth is $B=40$ MHz, and the spectral density of the noise is $N_0=-174$ dBm/Hz. Parameter C_k is uniformly distributed in $[1,3]\times 10^4$ cycles/sample. Model size $M=1$ Mbit. To simplify, we choose an equal transmit power $p_1(t)=...=p_K(t)=20$ dBm, an equal local computing power budget $f_1^{\max}(t)=...=f_K^{\max}(t)=2.5$ GHz, and an equal energy budget $\bar{E}_1=...=\bar{E}_K=0.05$ J. Besides, we set the coefficients $\varrho=10^{-28}$, $\rho=0.5$, and $V=0.01$. For the learning model, each client trains a multilayer perceptron model in the well-known dataset MNIST for classification, which consists of 10 categories ranging from digit "0" to "9". We sort all training samples according to their digit labels, and assign clients with 500 training samples with only two types of digits [12]. The number of training rounds $T=100$. In each training round, $\kappa_1=5$ and $\kappa_2=10$ [10]. We apply a standard multilayer perceptron model for local computation which has one hidden layer of 64 hidden nodes, We apply a standard multilayer perceptron (MLP) model for local model updates which has one hidden layer of 64 hidden nodes, and use ReLU activation. The MLP model has 50,890 weights. In addition, in this paper, we compare the performance of the proposed algorithm against the following baselines 1) Greedy client association with monotonous energy control (GAME): All the clients are selected and greedily associated to the edge servers that have the highest channel gains. Then all the clients make the local computing power control while satisfying the instantaneous energy budget, i.e., $E_k(t)=\bar{E}_k$, $\forall t \in \mathcal{T}$, $k \in \mathcal{K}$. 2) Latency-aware client association with maximal local computing power (DACM): Based on the proposed client association algorithm, all associated clients perform the local training with their maximal local computing power, i.e, $f_k(t)=f_k^{\max}(t), \forall t, k \in \mathcal{Q}(t)$.

B. Numerical Results

Figs. 2 and 3 demonstrate the training loss and the test accuracy versus the training latency. It is illustrated that the proposed scheme can achieve lower training loss and higher test accuracy with lower training latency compared with the baselines. More precisely, if the target training loss is 0.3, the proposed scheme can save half time compared with GAME. This is because the proposed scheme joint consider the learning performance and the training latency in each round, and GAME requires more time to finish the training. Besides, it is interesting to note that although DACM can finish the training with lower training latency, the learning performance

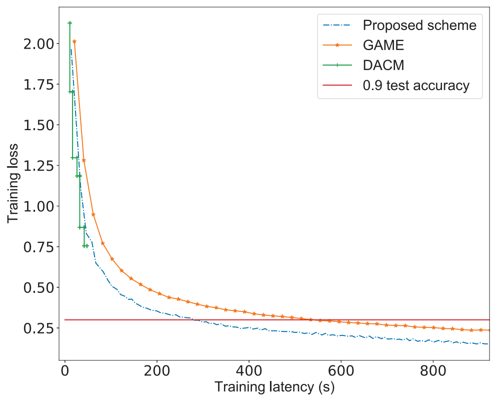

Fig. 2. The training loss versus the training latency.

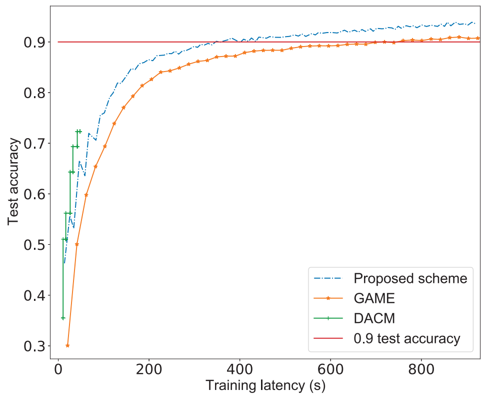

Fig. 3. The test accuracy versus the training latency.

is relatively poor. It attributes to the fact that the clients training with the maximum computing power can generate great energy consumption, and due to the energy budget constraints, the number of times that these clients participate in training is significantly decreased, which is not conducive to improving the learning performance.

V. Conclusion

In this paper, we have proposed an energy-aware hierarchical federated learning framework named EHFL. We aimed to minimize the training latency while achieving a target training loss and satisfying the long-term energy budget constraints of the individual clients. To mitigate the learning performance degradation in the case of non-i.i.d. local datasets and meet the energy budget, we considered the importance of local model updates, and formulated an alternative problem based on the general Lyapunov optimization framework. Then we proposed a heuristic algorithm to achieve a suboptimal solution. The results have shown that for non-i.i.d. local datasets, the proposed algorithm can meet the energy budget while achieving lower training loss with lower training latency compared with the benchmarks.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. Cambridge, MA: MIT Press, 2016.

[2] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, “A deep learning framework for optimization of miso downlink beamforming,” *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020.

[3] 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)*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1–10.

[4] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Federated learning for data privacy preservation in vehicular cyber-physical systems,” *IEEE Netw.*, vol. 34, no. 3, pp. 50–56, Jun. 2020.

[5] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, “Multi-armed bandit based client scheduling for federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Jul. 2020.

[6] C. Feng, Y. Wang, Z. Zhao, T. Q. Quek, and M. Peng, “Joint optimization of data sampling and user selection for federated learning in the mobile edge computing systems,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Virtual Conference, Jun. 2020, pp. 1–6.

[7] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy efficient federated learning over wireless communication networks,” *IEEE Trans. Wirel. Commun.*, early access, 10.1109/TWC.2020.3037554.

[8] Z. Zhao, C. Feng, H. H. Yang, and X. Luo, “Federated-learning-enabled intelligent fog radio access networks: Fundamental theory, key techniques, and future trends,” *IEEE Wirel. Commun.*, vol. 27, no. 2, pp. 22–28, Apr. 2020.

[9] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, “Hfel: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[10] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Virtual Conference, Jun. 2020, pp. 1–6.

[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of fedavg on non-iid data,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Millennium Hall, Addis Ababa, Ethiopia, Apr. 2020, pp. 5986–5994.

[12] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, “Scheduling in cellular federated edge learning with importance and channel awareness,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[13] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, “Update aware device scheduling for federated learning at the wireless edge,” in *Proc. IEEE Int. Symp. Inf. Theor. Proc. (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 2598–2603.

[14] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, “Active federated learning,” 2019, *arXiv:1909.12641*. [Online]. Available: https://arxiv.org/abs/1909.12641.

[15] R. Urgaonkar and M. J. Neely, “Opportunistic cooperation in cognitive femtocell networks,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 607–616, Apr. 2012.

[16] H. Ju, B. Liang, J. Li, and X. Yang, “Dynamic joint resource optimization for lte-advanced relay networks,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5668–5678, Nov. 2013.

[17] M. Peng, Y. Yu, H. Xiang, and H. V. Poor, “Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks,” *IEEE Trans. Multimedia.*, vol. 18, no. 5, pp. 879–892, May 2016.

[18] T. X. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Nov. 2018.