Edge-to-cloud Latency Aware User Association in Wireless Hierarchical Federated Learning

Rung-Hung Gau[1], Di-Chun Liang[1], Ting-Yu Wang[1] and Chun-Hung Liu[2]
Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan[1]
Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, USA[2]
e-mail: gaurunghung@nycu.edu.tw, ldc.ee02@nycu.edu.tw, wty880113.ee06@nycu.edu.tw, chliu@ece.msstate.edu

2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring) | 979-8-3503-8741-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-SPRING62846.2024.10683582

***Abstract*—In this paper, we propose a backbone-aware user association algorithm for heterogeneous hierarchical federated learning. We consider the scenario in which mobile devices have different computation and communication capabilities, while edge servers have different model uploading delays to the cloud server. To find an optimal user association, we formulate a combinatorial optimization problem that takes into consideration mobile-to-edge delays and edge-to-cloud delays. To reduce the computational complexity, we put forward the backbone-aware greedy algorithm. In addition, we prove that it is not always optimal for a mobile device to connect to the edge server with the minimum mobile-to-edge delay. Furthermore, we propose using dynamic bandwidth allocation after assigning users to edge servers to further reduce the latency. We also use simulation results to show the advantages of the proposed approach.**

***Index Terms*—Hierarchical federated learning, user association, heterogeneous system, end-to-end latency, combinatorial optimization.**

I. Introduction

Federated learning is a framework of distributed machine learning and is designed to protect data privacy of users [1]. In a basic federated learning system, users directly upload their local models to the parameter server for updating the global model. Abad *et al.* [2] studied hierarchical federated learning in a cellular network that contains a macro base station and small base stations. Specifically, each small base station aggregates local models of associated users, the macro base station is responsible for updating the global model and there is no cloud server. To combine the advantages of cloud and edge servers, Liu *et al.* [3] proposed using a client-edge-cloud hierarchical federated learning (HFL) system. We focus on client-edge-cloud HFL systems in this paper. Federated learning consists of multiple rounds. In a round of a client-edge-cloud HFL system, mobile devices use the received global model and their own data to perform local model updates and upload their local models to the associated edge servers. After aggregating received local models, each edge server uploads its model to the cloud server for updating the global model. Then, the cloud server broadcasts the latest global model to edge servers which forward the global model to associated mobile devices for local model updates.

User association and wireless resource allocation are important for HFL systems. In a heterogeneous HFL system, mobile devices have different computation capabilities, wireless links have distinct average channel gains and edge-to-cloud links have unequal delays. Thus, it is important to take into consideration heterogeneous devices and communications links in order to make intelligent decisions on user association and wireless resource allocation. Luo *et al.* [4] formulated an optimization problem to minimize the weighted sum of energy consumption and delay of hierarchical federated learning. They derived analytical results on the optimal bandwidth and computation capacity allocations. For deciding user association, they proposed a heuristic algorithm based on device transferring adjustments and device exchanging adjustments. Liu *et al.* [5] studied user association and wireless resource allocation for wireless hierarchical federated learning systems. For mobile devices with IID data, they claimed that the optimal user association is for each mobile device to choose the edge server with the largest signal-to-noise ratio (SNR). The SNR-based algorithm is called Max-SNR in this paper. Liu *et al.* [6] sought to minimize the total model parameter communication and computation delay by optimal user-edge association and wireless resource allocation. For deciding user-edge association, they proposed a greedy algorithm based on signal-to-noise ratios. In this paper, we show that the Max-SNR algorithm does not necessarily lead to optimal user-edge association when the edge-to-cloud delays are not negligible in comparison with mobile-to-edge delays.

Wen *et al.* [7] investigated sub-channel allocation and helper scheduling in a hierarchical federated learning system in which the base station plays the role of parameter server and the helpers connect to the base station through wireless links. Liu *et al.* [8] derived a tighter convergence bound for HFL with neural network quantization. Based on the derived analytical results, they optimized the two aggregation intervals in HFL. However, they [7] [8] did not explicitly deal with the optimal user-helper association problem. Wu *et al.* [9] proposed using deep reinforcement learning based staleness control and heterogeneity-aware client-edge association to improve the system efficiency of HFL. Wang *et al.* [10] put forward FedCH that adopts bipartite matching to partition clients into clusters based on their training capacities for accelerating HFL. The heterogeneity-aware client-edge association algorithm [9] and the cluster construction algorithm [10] assumed that the communication latency between mobile device k and edge server m is independent of the number of mobile devices associated with edge server k. In contrast, we take into consideration the impacts of the number of mobile devices associated with an edge server on the communication latency.

Chen *et al.* [11] proposed a deep reinforcement learning approach for client selection and resource allocation in HFL systems. Feng *et al.* [12] developed a theoretical model for studying the impact of user mobility on the performance of HFL systems. Machine learning and mobility management for HFL is beyond the scope of this paper.

Our major technical contributions are summarized as follows.

- To minimize the length of a round in heterogeneous hierarchical federated learning, we formulate a combinatorial optimization problem for optimal user-edge association. We consider the general case in which edge-to-cloud links have different delays and mobile devices have different computation and communication capabilities.
- To reduce the computational complexity, we propose the backbone-aware greedy (BAG) algorithm that takes into consideration heterogeneous edge-to-cloud delays for efficiently assigning mobile devices to edge servers.
- We derive novel analytical results on user association in HFL. We show that it is not necessarily optimal for a mobile device to connect to the edge server with the minimum latency of wireless communications when edge-to-cloud delays are different and not negligible.
- We use large-scale simulations to show that the proposed approach could significantly reduce the length of an HFL round when mobile devices have different computing capabilities and backbone communication links have different delays.

The rest of the paper is organized as follows. In Section II, we include the system models and formulate a combinatorial optimization problem for backbone-aware user association in heterogeneous hierarchical federated learning. In Section III, we elaborate on the proposed backbone-aware greedy algorithm for efficient user association. In Section IV, we derive novel analytical results on optimal user association in HFL. In Section V, we propose using dynamic bandwidth allocation to further reduce the HFL latency. In Section VI, we include simulation results that show the advantages of the proposed approach. In Section VII, we draw conclusions.

II. System Models

We consider a heterogeneous hierarchical federated learning system that consists of one cloud server, $N \geq 2$ edge servers and $M \geq 2$ mobile devices. Each edge server is colocated with a base station (BS). Specifically, edge server n is associated with BS n, $\forall n$. Let $\mathbb{N}$ be the set of positive integers. For each $n \in \mathbb{N}$, let $[n] = \{1, 2, .., n\}$. For each set S, denote its cardinality by $|S|$.

The federated learning process is composed of rounds. For each $t \in \mathbb{N}$, in the tth round, a mobile device is associated with an edge server. Let $x_{m,n}(t) \in \{0, 1\}$ be a binary variable, $\forall m \in [M], n \in [N], t \in \mathbb{N}$. Specifically, if mobile device m is associated with edge server n in round t, $x_{m,n}(t) = 1$. Otherwise, $x_{m,n}(t) = 0$. If $x_{m,n}(t) = 1$, mobile device m uploads its local model to edge server n through BS n in round t. Since a mobile device is associated with a single edge server, we have

$$\sum_{n=1}^{N} x_{m,n}(t) = 1, \forall m \in [M], t \in \mathbb{N}. \quad (1)$$

Let $A_n(t)$ be the set that is composed of the indexes of mobile devices that are associated with edge server n in the tth round of federated learning, $\forall n \in [N], t \in \mathbb{N}$. Then,

$$A_n(t) = \{m \in [M] | x_{m,n}(t) = 1\}, \forall n \in [N], t \in \mathbb{N}. \quad (2)$$

Note that $(A_1(t), A_2(t), .., A_N(t))$ is a partition of $[M]$. Namely, $\cup_{n=1}^{N} A_n(t) = [M]$ and $A_i(t) \cap A_j(t) = \emptyset$, $\forall i \neq j$.

Let $\mathbf{w}(t)$ be the global model vector at the beginning of round t. At the beginning of round t, the cloud server broadcasts the value of $\mathbf{w}(t)$ to the M mobile devices through the N base stations. Upon receiving the value of $\mathbf{w}(t)$, each mobile device performs local model updates based on its local training data and $\mathbf{w}(t)$. Let $\alpha_m(t)$ be the amount of time required for mobile device m to perform a local model update in round t, $\forall m \in [M], t \in \mathbb{N}$.

Let B_n be the total bandwidth that BS/edge server n owns. Let $\theta_{m,n}(t)$ be the fraction of bandwidth that edge server n allocates to mobile device m for uploading the local model in the tth HFL round, $\forall m \in [M], n \in [N], t \in \mathbb{N}$. If $x_{m,n}(t) = 1$, $\theta_{m,n}(t) > 0$. Otherwise, $\theta_{m,n}(t) = 0$. Let $p_m(t)$ be the transmit power of mobile device m in the tth HFL round. Let $g_{m,n}(t)$ be the channel gain of the link from mobile device m to BS/edge server n in the tth HFL round. Let N_0 be the power spectral density of the additive white Gaussian noise at each edge server. Let $r_{m,n}(t)$ be the data transmission rate from mobile device m to BS/edge server n in the tth HFL round. Then, for each (m, n, t), where $m \in [M], n \in [N], t \in \mathbb{N}$,

$$r_{m,n}(t) = \theta_{m,n}(t) B_n \times \log_2(1 + \frac{p_m(t) g_{m,n}(t)}{\theta_{m,n}(t) B_n N_0}). \quad (3)$$

Let L be the number of bits in a local model of machine learning. Let $t_{m,n}(t)$ be the amount of time required to upload

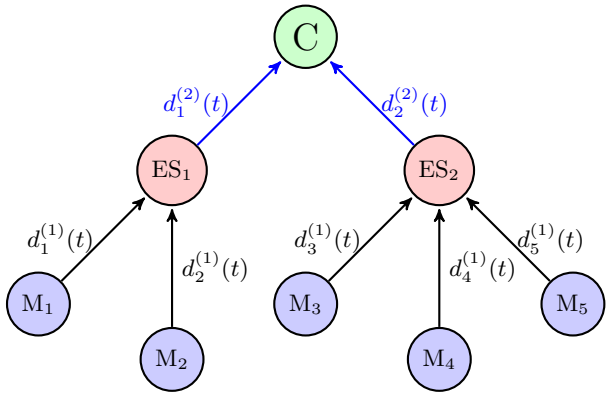

Fig. 1: An illustration for a hierarchical federated learning system.

the local model of mobile device m to edge server n in round t. Then,

$$t_{m,n}(t) = \frac{x_{m,n}(t)L}{r_{m,n}(t)}, \forall m \in [M], n \in [N], t \in \mathbb{N}. \quad (4)$$

Let $\beta_{m,n}(t)$ be the amount of time required for uploading the local model of mobile device m to edge server n in round t when $\theta_{m,n}(t) = 1$, $\forall m \in [M], n \in [N], t \in \mathbb{N}$. Namely,

$$\beta_{m,n}(t) = \frac{L}{B_n \times \log_2(1 + \frac{p_m(t)g_{m,n}(t)}{B_n N_0})}. \quad (5)$$

It is assumed that all mobile devices that are associated with edge server n equally share the bandwidth of BS n. Namely, equal bandwidth allocation (EBA) is adopted and

$$\theta_{m,n}(t) = \frac{x_{m,n}(t)}{|A_n(t)|}, \forall t \in \mathbb{N}, n \in [N], m \in A_n(t). \quad (6)$$

Then, if $x_{m,n}(t) = 1$,

$$\begin{aligned} t_{m,n}(t) &= \frac{L}{\theta_{m,n}(t)B_n \times \log_2(1 + \frac{p_m(t)g_{m,n}(t)}{\theta_{m,n}(t)B_n N_0})} \\ &= \frac{|A_n(t)| \cdot L}{B_n \times \log_2(1 + \frac{p_m(t)g_{m,n}(t)}{B_n N_0/|A_n(t)|})} \\ &\leq |A_n(t)| \cdot \beta_{m,n}(t). \end{aligned} \quad (7)$$

Let $d_m^{(1)}(t)$ be the amount of time required for mobile device m to perform local model updates and upload its local model to the associated edge server in round t, $\forall m \in [M], t \in \mathbb{N}$. Specifically, for each (m, t), where $m \in [M], t \in \mathbb{N}$,

$$d_m^{(1)}(t) = \alpha_m(t) + \sum_{k=1}^{N} x_{m,k}(t) \times \beta_{m,k}(t) \times |A_k(t)|. \quad (8)$$

Note that $x_{m,n}(t) = 1$ if and only if $m \in A_n(t)$.

Consider edge server n in round t. After receiving local models from all associated mobile devices, edge server n aggregates the local models to obtain the edge model. Next, the edge server uploads the edge model to the cloud server. After receiving all edge models, the cloud server updates the global model. Then, the cloud server broadcasts the latest global model to the mobile devices through the edge servers.

Let $d_n^{(2)}(t)$ be the amount of time required for edge server n to upload its model to the cloud server, $\forall n \in [N], t \in \mathbb{N}$. In general, the geographical distances from different edge servers to the cloud server are different. In addition, two edge servers might connect to the cloud server through different routing paths with different congestion levels. Thus, the value of $d_n^{(2)}(t)$ depends on n and is not always constant.

In Fig. 1, we illustrate a hierarchical federated learning system that consists of one cloud server, two edge servers and five mobile devices. In the figure, $N = 2$, $M = 5$, the cloud server is marked by C, the nth edge server is marked by ES_n, $\forall n \in [N]$ and the mth mobile device is marked by M_m, $\forall m \in [M]$. In addition, the first two mobile devices are associated with the first edge server, while the last three mobile devices are associated with the second edge server. In this case, $x_{1,1}(t) = x_{2,1}(t) = 1$ and $x_{3,2}(t) = x_{4,2}(t) = x_{5,2}(t) = 1$.

Let $y(t)$ be the length of the tth HFL round. In particular,

$$y(t) = \max_{n:n\in[N]} \big[\max_{m:m\in A_n(t)} d_m^{(1)}(t) + d_n^{(2)}(t) \big], \forall t \in \mathbb{N}. \quad (9)$$

For each $n \in [N]$, $\max_{m:m\in A_n(t)} d_m^{(1)}(t) + d_n^{(2)}(t)$ is the amount of time required for mobile devices associated with edge server n to update and upload their local models. In addition, the cloud server has to receive models from all edge servers in order to update the global model. Thus, we have the above equality.

For each $t \in \mathbb{N}$, let $\mathbf{x}(t) \in \{0, 1\}^{M \times N}$ be a matrix such that $[\mathbf{x}(t)]_{m,n} = x_{m,n}(t)$, $\forall m \in [M], n \in [N]$. For each $t \in \mathbb{N}$, to find an optimal user association for round t, we formulate the following combinatorial optimization problem.

$$\begin{aligned} &\min_{\mathbf{x}(t)} \max_{n:n\in[N]} \big[\max_{m:m\in A_n(t)} d_m^{(1)}(t) + d_n^{(2)}(t) \big] \\ &\text{subject to} \\ &x_{m,n}(t) \in \{0, 1\}, \forall m \in [M], n \in [N] \\ &\sum_{n=1}^{N} x_{m,n}(t) = 1, \forall m \in [M] \\ &A_n(t) = \{m \in [M] | x_{m,n}(t) = 1\}, \forall n \in [N]. \end{aligned} \quad (10)$$

Since each mobile device could be assigned to one of the N edge servers and there are M mobile devices, there are N^M feasible solutions for (10).

III. The backbone-aware greedy algorithm

In this section, we introduce the proposed backbone-aware greedy (BAG) algorithm for user association. The BAG algorithm is backbone-aware rather than backbone-oblivious, since it takes into consideration edge-to-cloud delays as well as mobile-to-edge delays when making decisions on user-edge association. In addition, the BAG algorithm consists of M iterations and tries to find an optimal edge server for a mobile

Algorithm 1. The backbone-aware greedy (BAG) algorithm.

Require: M, N, t, $\alpha_m(t)$'s, $\beta_{m,n}(t)$'s, $d_n^{(2)}(t)$'s.
Ensure: $(A_1(t), A_2(t), .., A_N(t))$.
1: Obtain $(\phi_1(t), \phi_2(t), .., \phi_M(t))$ by sorting $\alpha_m(t)$'s.
2: $A_n(t) \leftarrow \emptyset$, $\tilde{A}_{0,n}(t) \leftarrow \emptyset$, $\forall n \in [N]$.
3: **for** $r = 1$ to M **do**
4: **for** $n = 1$ to N **do**
5: $\tilde{A}_{r,k}(t) \leftarrow \tilde{A}_{r-1,k}(t)$, $\forall k \in [N], k \neq n$.
6: $\tilde{A}_{r,n}(t) \leftarrow \tilde{A}_{r-1,n}(t) \cup \{\phi_r(t)\}$.
7: Obtain $d_m^{(1)}(t)$'s based on (8).
8: $\lambda_{r,n}(t) \leftarrow f_t(\tilde{A}_{r,1}(t), \tilde{A}_{r,2}(t), .., \tilde{A}_{r,N}(t))$.
9: **end for**
10: $s_r(t) \leftarrow \arg\min_{n:n\in[N]} \lambda_{r,n}(t)$.
11: $A_{s_r(t)}(t) \leftarrow A_{s_r(t)}(t) \cup \{\phi_r(t)\}$.
12: $\tilde{A}_{r,k}(t) \leftarrow A_k(t)$, $\forall k \in [N]$.
13: **end for**

device in each iteration. Pseudo codes for the BAG algorithm are included in Algorithm 1.

Consider HFL round t. The BAG algorithm works as follows. First, the BAG algorithm sorts $\alpha_m(t)$'s in decreasing order. Let $\phi_r(t)$ be the index of the mobile device with rank r, $\forall r \in [M]$. Then,

$$\alpha_{\phi_r(t)}(t) \geq \alpha_{\phi_{r+1}(t)}(t), \forall r \in [M-1]. \tag{11}$$

Note that mobile device $\phi_1(t)$ is the device with the worst computation capability and the largest computation latency in round t.

Since $x_{m,n}(t) = 1$ if and only if $m \in A_n(t)$, (10) is equivalent to the following combinatorial optimization problem.

$$\begin{aligned}
&\min_{A_1(t),A_2(t),..,A_N(t)} \max_{n:n\in[N]} \big[\max_{m:m\in A_n(t)} d_m^{(1)}(t) + d_n^{(2)}(t) \big] \\
&\text{subject to} \\
&\cup_{n=1}^{N} A_n(t) = [M] \\
&A_i(t) \cap A_j(t) = \emptyset, \forall i \neq j \\
&x_{m,n}(t) = 1, \forall n \in [N], m \in A_n(t).
\end{aligned} \tag{12}$$

Based on (12), for each $(A_1(t), A_2(t), .., A_N(t))$, which is a partition of $[M]$, we define $f_t(A_1(t), A_2(t), .., A_N(t))$ as follows.

$$\begin{aligned}
&f_t(A_1(t), A_2(t), .., A_N(t)) \\
= &\max_{n:n\in[N]} \big[\max_{m:m\in A_n(t)} d_m^{(1)}(t) + d_n^{(2)}(t) \big].
\end{aligned} \tag{13}$$

The BAG algorithm is composed of M iterations. Specifically, for each $r \in [M]$, in the rth iteration of the tth HFL round, the BAG algorithm finds an edge server for mobile device $\phi_r(t)$. Let $\tilde{A}_{r,k}(t)$ be the set composed of the indexes of mobile devices that have been assigned to edge server k by the BAG algorithm in the first r iterations of the tth HFL round, $\forall r \in [M], k \in [N], t \in \mathbb{N}$. For each (r, n), where $r \in [M], n \in [N]$, we define $\lambda_{r,n}(t)$ as follows.

$$\begin{aligned}
\lambda_{r,n}(t) = \; & f_t(\tilde{A}_{r-1,1}(t), \tilde{A}_{r-1,2}(t), .., \tilde{A}_{r-1,n-1}(t), \\
& \tilde{A}_{r-1,n}(t) \cup \{\phi_r(t)\}, \tilde{A}_{r-1,n+1}(t), .., \\
& \tilde{A}_{r-1,N}(t)).
\end{aligned} \tag{14}$$

Note that $\lambda_{r,n}(t)$ is equal to the length of HFL round t when $A_n(t) = \tilde{A}_{r-1,n}(t) \cup \{\phi_r(t)\}$ and $A_k(t) = \tilde{A}_{r-1,k}(t)$, $\forall k \in [N], k \neq n$.

Let $s_r(t)$ be the index of the edge server to which mobile device $\phi_r(t)$ is assigned by the BAG algorithm in iteration r of HFL round t, $\forall r \in [M], t \in \mathbb{N}$. To minimize the length of HFL round t, the BAG algorithm sets the value of $s_r(t)$ as follows.

$$s_r(t) = \arg \min_{n:n\in[N]} \lambda_{r,n}(t). \tag{15}$$

Namely, in iteration r of HFL round t, the BAG algorithm assigns mobile device $\phi_r(t)$ to edge server n^*, where $\lambda_{r,n^*}(t) = \min_{n:n\in[N]} \lambda_{r,n}(t)$. If there are two or more minimum elements in the set $\{\lambda_{r,n}(t)|n \in [N]\}$, the proposed BAG algorithm breaks the tie by selecting the element with the minimum index for $s_r(t)$. According to the BAG algorithm,

$$A_n(t) = \{r \in [M] | s_r(t) = n\}, \forall n \in [N], t \in \mathbb{N}. \tag{16}$$

To sum up, the proposed BAG algorithm sequentially assigns mobile devices to edge servers. Specifically, in the rth iteration of the tth HFL round, mobile device $\phi_r(t)$ is assigned to edge server $s_r(t)$.

We now analyze the computational complexity of the proposed BAG algorithm. In line 1 of Algorithm 1, it takes $O(M \log_2(M))$ time to sort the M real numbers in the set $\{\alpha_m(t)|m \in [M]\}$. Since $\sum_{k=1}^{N} |\tilde{A}_{r-1,k}(t)| \leq M$, given $(\tilde{A}_{r-1,1}(t), \tilde{A}_{r-1,2}(t), .., \tilde{A}_{r-1,N}(t))$, it takes $O(M)$ time to obtain $(\tilde{A}_{r,1}(t), \tilde{A}_{r,2}(t), .., \tilde{A}_{r,N}(t))$ in lines 5-6. Given $(A_1(t), A_2(t), .., A_N(t))$, it takes $O(MN)$ time to obtain $\mathbf{x}(t)$. For each $m \in [M]$, it takes $O(N)$ time to obtain $d_m^{(1)}(t)$. Thus, in line 7, it takes $O(MN)$ time to obtain $(d_1^{(1)}(t), d_2^{(1)}(t), .., d_M^{(1)}(t))$. For each (r, n, t), given $(\tilde{A}_{r,1}(t), \tilde{A}_{r,2}(t), .., \tilde{A}_{r,N}(t))$, based on (13), it takes $O(MN)$ time to acquire the value of $\lambda_{r,n}(t)$. Thus, the computational complexity of an iteration of the inner for loop in lines 5-8 is equal to $O(M) + O(MN) + O(MN) = O(MN)$. Hence, for each (r, t), it takes $N \times O(MN) = O(MN^2)$ time to obtain $(\lambda_{r,1}(t), \lambda_{r,2}(t), .., \lambda_{r,N}(t))$. Given $(\lambda_{r,1}(t), \lambda_{r,2}(t), .., \lambda_{r,N}(t))$, in line 10, it takes $O(N)$ time to obtain $s_r(t)$. The computational complexity of line 11 is at most $O(M)$. Since $\sum_{k=1}^{N} |A_{r,k}(t)| \leq M$, $\forall r, t$, the computational complexity of line 12 is $O(M)$. Thus, the computational complexity for an iteration of the outer for loop between line 4 and line 12 is $O(MN^2) + O(N) + O(M) + O(M) = O(MN^2)$. Therefore, the overall computational complexity for Algorithm 1 is $M \times O(MN^2) = O(M^2N^2)$.

IV. Analytical Results

In this section, we derive novel analytical results on user association in heterogeneous hierarchical federated learning. We find a scenario in which the well-known Max-SNR algorithm [5] does not produce an optimal solution for (10).

According to the Max-SNR algorithm, for each (m, t), $x_{m,n}(t) = 1$ only if $\beta_{m,n}(t) \leq \beta_{m,k}(t)$, $\forall k \in [N]$. The following theorem shows that the Max-SNR algorithm does not always produce an optimal solution for (10).

Theorem 1: Consider a fixed $t \in \mathbb{N}$. If $N = 2$, $M = 2K$, where $K \in \mathbb{N}$, $\alpha_m(t) = \alpha \in \mathbb{R}$, $\forall m \in [M]$, $\beta_{m,1}(t) < \beta_{m,2}(t)$, $\forall m \in [K]$, $\beta_{m,1}(t) > \beta_{m,2}(t)$, $\forall m \in \{K+1, K+2, .., M\}$ and $d_1^{(2)}(t) - d_2^{(2)}(t) > \sum_{m=1}^{M} \sum_{n=1}^{N} \beta_{m,n}(t)$, the Max-SNR algorithm does not produce an optimal solution for (10).

Proof:

1. Let $\mathbf{x}'(t)$ be the solution produced by the Max-SNR algorithm. Since $N = 2$, $M = 2K$, $\beta_{m,1}(t) < \beta_{m,2}(t)$, $\forall m \in [K]$, $\beta_{m,1}(t) > \beta_{m,2}(t)$, $\forall m \in \{K+1, K+2, .., M\}$, $x'_{m,n}(t) = 1$, $\forall m \in [K]$ and $x'_{m,n}(t) = 2$, $\forall m \in \{K+1, K+2, .., M\}$. Define $A'_n(t) = \{m \in [M] | x'_{m,n}(t) = 1\}$, $\forall n \in [N]$. Let $f'_t = f_t(A'_1(t), A'_2(t))$.

2. Then, $d_m^{(1)}(t) = \alpha + \beta_{m,1}(t)$, $\forall m \in [K]$ and $d_m^{(1)}(t) = \alpha + \beta_{m,2}(t)$, $\forall m \in \{K+1, K+2, .., M\}$. Thus,

$$\begin{aligned} & \max_{m:m\in A_1'(t)} d_m^{(1)}(t) + d_1^{(2)}(t) \\ = \; & \alpha + d_1^{(2)}(t) + \max_{m:m\in[K]} \beta_{m,1}(t). \end{aligned}$$

In addition,

$$\begin{aligned} & \max_{m:m\in A_2'(t)} d_m^{(1)}(t) + d_2^{(2)}(t) \\ = \; & \alpha + d_2^{(2)}(t) + \max_{m:m\in\{K+1,K+2,..,M\}} \beta_{m,2}(t). \end{aligned}$$

Hence,

$$\begin{aligned} f_t' &= \max_{n:n\in[N]} \Big[\max_{m:m\in A_n'(t)} d_m^{(1)}(t) + d_n^{(2)}(t) \Big] \\ &= \max\{\alpha + d_1^{(2)}(t) + \max_{m:m\in[K]} \beta_{m,1}(t), \\ &\quad \alpha + d_2^{(2)}(t) + \max_{m:m\in\{K+1,K+2,..,M\}} \beta_{m,2}(t)\} \\ &= \alpha + d_1^{(2)}(t) + \max_{m:m\in[K]} \beta_{m,1}(t) \\ &> \alpha + d_2^{(2)}(t) + \sum_{m=1}^{M}\sum_{n=1}^{N} \beta_{m,n}(t). \end{aligned}$$

The third equality and the inequality are due to that $d_1^{(2)}(t) - d_2^{(2)}(t) > \sum_{m=1}^{M}\sum_{n=1}^{N} \beta_{m,n}(t)$ and $\beta_{m,n}(t) > 0$, $\forall m, n$.

3. Define $x_{m,1}^{\dagger}(t) = 0$ and $x_{m,2}^{\dagger}(t) = 1$, $\forall m \in [M]$. Define $A_n^{\dagger}(t) = \{m \in [M] | x_{m,n}^{\dagger}(t) = 1\}$, $\forall n \in [N]$. Let $f_t^{\dagger} = f_t(A_1^{\dagger}(t), A_2^{\dagger}(t))$. Then,

$$\begin{aligned} f_t^{\dagger} &= \max\{0, \alpha + d_2^{(2)}(t) + \max_{m:m\in[M]} \beta_{m,2}(t)\} \\ &= \alpha + d_2^{(2)}(t) + \max_{m:m\in[M]} \beta_{m,2}(t) \\ &< \alpha + d_2^{(2)}(t) + \sum_{m=1}^{M}\sum_{n=1}^{N} \beta_{m,n}(t). \end{aligned}$$

4. Based on 2 and 3, $f_t' > f_t^{\dagger}$ and therefore $\mathbf{x}'(t)$ is not an optimal solution of (10). In this case, the Max-SNR algorithm does not produce an optimal solution for (10). ■

The above theorem implies that one has to take into consideration the edge-to-cloud delays as well as the mobile-to-edge delays for optimal user association in HFL.

V. Dynamic Bandwidth Allocation

To further reduce the latency of HFL, we adopt dynamic bandwidth allocation (DBA) after assigning mobile users to edge servers. Recall that $A_n(t)$ is the set composed of indexes of mobile users that are associated with edge server n in round t and $\theta_{m,n}(t)$ is the fraction of bandwidth that edge server n allocates to mobile user m in round t. Define $\theta_n(t) = (\theta_{1,n}(t), \theta_{2,n}(t), .., \theta_{M,n}(t))$. To find an optimal bandwidth allocation at edge server n in round t, we formulate the following optimization problem.

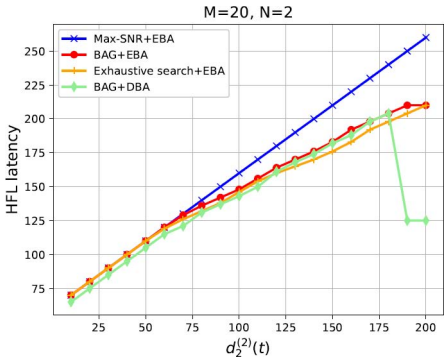

Fig. 2: The impact of $d_2^{(2)}(t)$ on the HFL latency.

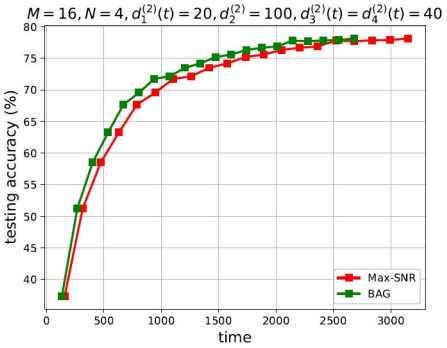

Fig. 3: The testing accuracy of HFL.

$$\begin{aligned} & \min_{\theta_n(t)} \max_{m:m\in A_n(t)} \alpha_m(t) + \frac{\beta_{m,n}(t)}{\theta_{m,n}(t)} \\ & \text{subject to} \\ & \theta_{m,n}(t) \in [0,1], \forall m \in [M] \\ & \sum_{m=1}^{M} \theta_{m,n}(t) \leq 1 \\ & \theta_{m,n}(t) = 0, \forall m \notin A_n(t). \end{aligned} \tag{17}$$

Note that $\max_{m:m\in A_n(t)} \alpha_m(t) + \frac{\beta_{m,n}(t)}{\theta_{m,n}(t)}$ is the latency for all mobile users associated with edge server n in round t to update and upload their local models to edge server n in round t. After defining $\mu_n(t) = \max_{m:m\in A_n(t)} \alpha_m(t) + \frac{\beta_{m,n}(t)}{\theta_{m,n}(t)}$, one can reformulate (17) as a convex optimization problem. Let $\theta_n^*(t) = (\theta_{1,n}^*(t), \theta_{2,n}^*(t), .., \theta_{M,n}^*(t))$ be an optimal solution of (17). It can be proved that there exists a positive real number $\gamma_n(t)$ such that

$$\begin{aligned} \alpha_m(t) + \frac{\beta_{m,n}(t)}{\theta_{m,n}^*(t)} &= \gamma_n(t), \forall m \in A_n(t) \\ \sum_{m:m\in A_n(t)} \theta_{m,n}^*(t) &= 1. \end{aligned} \tag{18}$$

We use binary search to numerically obtain the value of $\gamma_n(t)$. In addition, $\theta_{m,n}^*(t) = \frac{\beta_{m,n}(t)}{\gamma_n(t) - \alpha_m(t)}$, $\forall m \in A_n(t)$.

VI. Simulation Setup and Results

In this section, we include simulation setup and results. We wrote Python programs to obtain simulation results. We evaluate four algorithms for HFL. The first algorithm is the well-known Max-SNR algorithm with equal bandwidth allocation (EBA), the second algorithm adopts the proposed BAG algorithm and EBA, the third algorithm uses EBA and exhaustive search to obtain an optimal user association and the fourth algorithm adopts the BAG algorithm and dynamic bandwidth allocation (DBA). For proof of concept, we study the case in which $M = 20$ and $N = 2$. The coordinates of the first edge server are $(0, 0)$ and the coordinates of the second edge server are $(5, 0)$. The first 10 mobile devices are located around $(1, 0)$, while the remaining 10 mobile devices are located around $(3, 0)$. In this section, for each $t \in \mathbb{N}$, $\beta_{m,1}(t) = 1$, $\forall 1 \leq m \leq 10$ and $\beta_{m,1}(t) = 9$, $\forall 11 \leq m \leq 20$. In addition, for each $t \in \mathbb{N}$, $\beta_{m,2}(t) = 16$, $\forall 1 \leq m \leq 10$ and $\beta_{m,2}(t) = 4$, $\forall 11 \leq m \leq 20$. There are two types of mobile devices in terms of computation capability. Specifically, for each $t \in \mathbb{N}$, $\alpha_m(t) = 10$, $\forall m \in \{1, 2, .., 5\} \cup \{11, 12, .., 15\}$ and $\alpha_m(t) = 20$, $\forall m \in \{6, 7, .., 10\} \cup \{16, 17, .., 20\}$. Moreover, for each $t \in \mathbb{N}$, $d_1^{(2)}(t) = 10$ and $d_2^{(2)}(t) \in [10, 200]$.

In Fig. 2, we show the impacts of $d_2^{(2)}(t)$ on the HFL latency. For the first three algorithms that adopt EBA, the HFL latency increases as the value of $d_2^{(2)}(t)$ increases. It is due to that the HFL latency depends on the mobile-to-edge delays and the edge-to-cloud delays. When $d_2^{(2)}(t) \in [10, 70]$, the proposed BAG algorithm is as good as the Max-SNR algorithm in terms of the HFL latency. On the other hand, when $d_2^{(2)}(t) \in [80, 200]$, the proposed BAG algorithm is superior to the Max-SNR algorithm in terms of the HFL latency. In comparison with the Max-SNR algorithm, the proposed BAG algorithm could reduce the HFL latency by up to 19.2%. When $|d_2^{(2)}(t) - d_1^{(2)}(t)|$ is small, one can safely ignore the edge-to-cloud delays when making decisions on user association. However, when $|d_2^{(2)}(t) - d_1^{(2)}(t)|$ is large, one has to take into consideration the edge-to-cloud delays for optimally assigning mobile devices to edge servers. The simulation results also show that DBA could further reduce the HFL latency especially when the number of mobile users associated with an edge server is large.

We adopt PyTorch to evaluate two user association algorithms in a federated learning system for image recognition. We use the CIFAR-10 [13] dataset for training and testing machine learning models. There are 10 classes of images in the CIFAR-10 dataset. Each class contains 5000 training images and 1000 testing images. We adopt a convolutional neural network (CNN) that is composed of 10 layers including 1 input layer, 3 convolutional layers, 2 pooling layers, 1 flatten layer and 3 fully connected layers. The CNN has 591066 trainable parameters. In Fig. 3, we show the testing accuracy of HFL for two algorithms when $M = 16$ and $N = 4$. The BAG algorithm outperforms the Max-SNR algorithm in terms of the convergence speed.

VII. Conclusion

We have proposed a novel algorithm for user association in heterogeneous hierarchical federated learning systems. We have studied the scenario in which mobile devices have different computation and communication capabilities, while edge servers have different model uploading delays to the cloud server. To find an optimal user-edge association, we have formulated a combinatorial optimization problem based on mobile-to-edge delays and edge-to-cloud delays. To reduce the computational complexity, we have put forward the backbone-aware greedy algorithm. Furthermore, we have derived novel analytical results on user association. Moreover, we have used computer simulation to reveal the advantages of the proposed approach. Future work includes jointly optimizing wireless resource allocation, user association and backbone routing for hierarchical federated learning.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. 2017 International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, FL, USA, Apr. 20-22, 2017.

[2] M. S. H. Abad, E. Ozfatura, D. GUndUz and O. Ercetin, "Hierarchical Federated Learning ACROSS Heterogeneous Cellular Networks," in *Proc. 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8866-8870, Barcelona, Spain, May 4-8, 2020.

[3] L. Liu, J. Zhang, S. H. Song and K. B. Letaief, "Client-Edge-Cloud Hierarchical Federated Learning," in *Proc. 2020 IEEE International Conference on Communications (ICC)*, pp. 1-6, Dublin, Ireland, June 7-11, 2020.

[4] 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.

[5] S. Liu, G. Yu, X. Chen and M. Bennis, "Joint User Association and Resource Allocation for Wireless Hierarchical Federated Learning With IID and Non-IID Data," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7852-7866, Oct. 2022.

[6] C. Liu, T. J. Chua and J. Zhao, "Time Minimization in Hierarchical Federated Learning," in *Proc. 2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, Seattle, WA, USA, Dec. 2022, pp. 96-106.

[7] W. Wen, Z. Chen, H. H. Yang, W. Xia and T. Q. S. Quek, "Joint Scheduling and Resource Allocation for Hierarchical Federated Edge Learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5857-5872, Aug. 2022.

[8] L. Liu, J. Zhang, S. Song and K. B. Letaief, "Hierarchical Federated Learning With Quantization: Convergence Analysis and System Design," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 2-18, Jan. 2023.

[9] Q. Wu et al., "HiFlash: Communication-Efficient Hierarchical Federated Learning With Adaptive Staleness Control and Heterogeneity-Aware Client-Edge Association," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560-1579, May 2023.

[10] Z. Wang, H. Xu, J. Liu, Y. Xu, H. Huang and Y. Zhao, "Accelerating Federated Learning With Cluster Construction and Hierarchical Aggregation," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3805-3822, 1 July 2023.

[11] Q. Chen, Z. You, D. Wen and Z. Zhang, "Enhanced Hybrid Hierarchical Federated Edge Learning Over Heterogeneous Networks," in *IEEE Transactions on Vehicular Technology*, doi: 10.1109/TVT.2023.3287355.

[12] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. S. Quek and G. Min, "Mobility-Aware Cluster Federated Learning in Hierarchical Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8441-8458, Oct. 2022.

[13] Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009, https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.