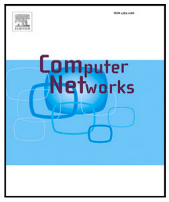
Computer Networks 250 (2024) 110512

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Client selection for federated learning using combinatorial multi-armed bandit under long-term energy constraint☆

Konglin Zhu [a,c], Fuchun Zhang [a], Lei Jiao [b,*], Bowei Xue [a], Lin Zhang [a]

[a] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
[b] Department of Computer Science, University of Oregon, Eugene, OR, USA
[c] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, China

ARTICLE INFO

Keywords:
Client selection
Combinatorial multi-armed bandit
Federated learning
Lyapunov optimization

ABSTRACT

In a federated learning system, it is often the case that the more clients it involves, the less increment of the outcome it achieves. It is thus essential to design a client selection strategy to choose an appropriate subset of the clients to participate in federated learning. However, client selection is not easy due to the heterogeneity of clients and the long-term energy budget of each client. Moreover, long-term energy budgets intertwined with the short-term client selection often make the problem NP-hard. In this paper, we propose an online strategy *Energy-Aware Client Selection for Federated Learning* (EACS-FL) to address this problem. The problem is formulated with a joint energy and delay optimization objective, and the Combinatorial Multi-Armed Bandit (CMAB) is introduced to solve the problem in an online manner. We take advantage of Lyapunov optimization to manage energy consumption of clients, which enables us to deal with independent energy budgets through minimizing virtual energy deficit queues. Theoretical analysis shows that EACS-FL achieves sublinear regret and keeps all queues stable. Experiment results exhibit that the proposed approach outperforms the existing works and achieves close-to-optimal delay and energy consumption performance.

1. Introduction

Federated learning (FL) has been extensively studied recently in machine learning as an approach to training models while keeping the raw data decentralized. On one hand, FL gathers models from clients and bridges the data islands among clients; on the other hand, as clients only upload models instead of data, privacy can be well preserved. An FL system consists of a central server and a crowd of clients (*e.g.*, mobile phones, wearable devices, or autonomous vehicles) [1], aiming at learning a global model on the central server by iteratively aggregating local models trained on clients using the clients' own data. As shown in Fig. 1, the server X is used for the global model aggregation, and the clients A, C, and E are selected for the local model training.

The FL system often has "submodularity". That is, when the number of the clients reaches a certain scale, the increment of the outcome by involving more clients will become less, but the cost of using more clients usually continues to increase. In other words, the increment of the outcome by using more clients is not worth the cost [2]. Therefore, it is of great importance to select a group of clients with effective contributions to the FL system. Indeed, there are many factors making the client selection problem non-trivial. First of all, as clients in the FL system are often heterogeneous, varying in data quality, hardware performance, etc., an arbitrary selection decision could have a negative impact on the training effectiveness [3]. Furthermore, mobile clients typically have limited energy. The selection strategy has to abide by the energy constraints through the whole training process. Thus, it is never easy to design a client selection strategy that considers these factors. The fundamental challenges behind these factors are as follows.

First, clients in FL systems often have different hardware settings and software systems, making their computational performance and energy consumption different. In general, the overall training effectiveness of the selected set of clients in each round depends on the slowest one, namely the straggler [4,5]. As depicted in Fig. 1, the client D is a mobile device with very limited computational capability. If it participates in the FL local model training, the FL aggregation

☆ This work was partially supported by the National Key Research and Development Program of China (2023YFB2704500), the Beijing Natural Science Foundation (4222033), the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, and the U.S. National Science Foundation (CNS-2047719 and CNS-2225949).

* Corresponding author.
E-mail addresses: klzhu@bupt.edu.cn (K. Zhu), zhangfc@bupt.edu.cn (F. Zhang), jiao@cs.uoregon.edu (L. Jiao), xue0316@bupt.edu.cn (B. Xue), zhanglin@bupt.edu.cn (L. Zhang).

https://doi.org/10.1016/j.comnet.2024.110512
Received 8 January 2023; Received in revised form 26 March 2024; Accepted 14 May 2024
Available online 22 May 2024

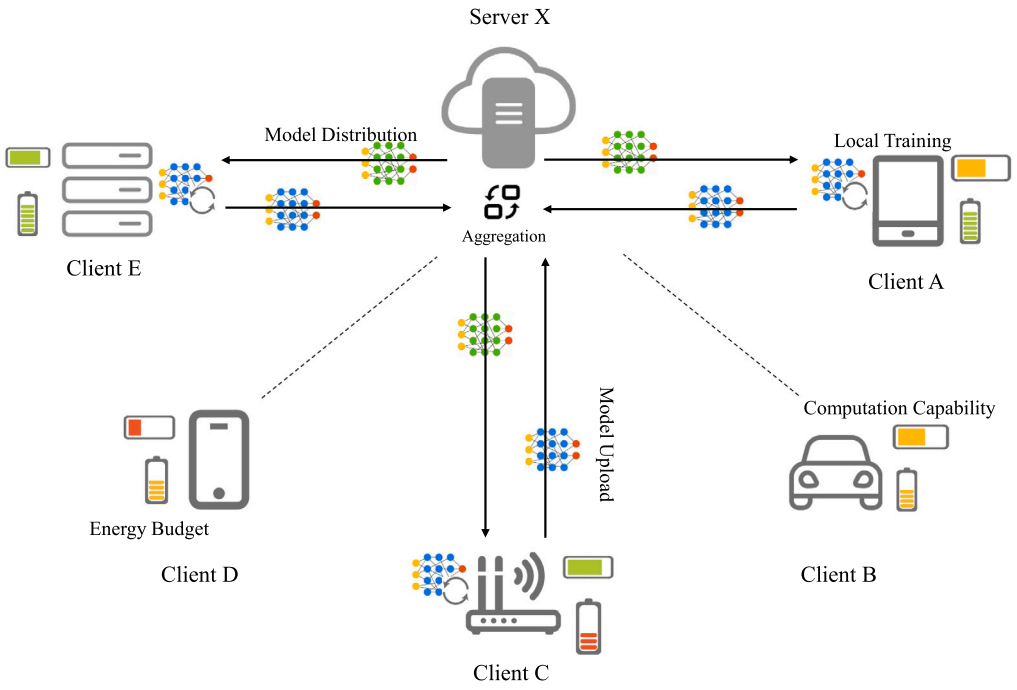


Fig. 1. A federated learning system.

may be delayed due to the extended local training process of D. Moreover, the central server X lacks priori knowledge about such client information. Despite the server's inability to foresee future information such as training time and energy consumption, it must make selection decisions. Given that these inputs are time-varying and unpredictable, the problem is inherently online.

Second, the heterogeneity of clients may mislead the selection strategy. As some clients outperform others, the strategy may frequently select those *good* clients after some iterations, while overlooking others. This is not acceptable for the preferred clients, as they are not specialized equipments for model training and have other tasks to process [6]. Even worse, the above-mentioned phenomenon could lead to unbalanced data being engaged in the training process, resulting in a poor global model. For instance, as shown in Fig. 1 again, the client A might be chosen too frequently for local model training due to its high computation capability. However, the data collection of A may be so slow that there is no new data collected between different FL training rounds, resulting in no update to local model parameters. Hence, careful design is needed for the strategy to avoid this phenomenon.

Third, each client has its own energy budget. The energy budget is a long-term constraint that the client selection strategy needs to respect. With such long-term constraints, the problem can often be NP-hard, making it difficult to design online algorithms and strike the balance between training effectiveness and energy consumption.

Existing studies cannot fully address the aforementioned challenges. Those works on client selection fall insufficient, as the central server requests resource information of clients and estimates the their performance in prior [7,8]. Other works [9,10] adopt the Multi-Armed Bandit (MAB) theory and focus on selection fairness while overlooking the energy budget of clients. The rest [11–13] focus on client scheduling and radio/network resource allocation, out of the scope of our study.

In this paper, we design an online algorithm named *Energy-Aware Client Selection for Federated Learning* (EACS-FL), based on the Combinatorial Multi-Armed Bandit (CMAB) [14–20] with heterogeneous energy budget constraints, for FL systems to select clients dynamically without foreseeing future information of clients. We formulate the client selection problem to optimize the energy consumption and the training delay, which is an NP-hard problem even in the offline setting. We further reformulate the problem as an online problem for each single round. Then we propose the EACS-FL algorithm by leveraging CMAB to estimate the quality function based on energy consumption and training delay using the UCB1 algorithm [21]. We define a virtual energy deficit queue and apply Lyapunov optimization to manage the energy budget of clients and balance the client selection frequency. The theoretical analysis suggests that our proposed algorithm can reach a close-to-optimal solution. The simulation results show that our proposed algorithm saves 54.5% energy compared with the state-of-the-art while maintaining the same accuracy level of FL models.

The main contributions of this paper are summarized as follows:

- We study the client selection in FL systems and develop a novel CMAB-based algorithm to alleviate the straggler effect by jointly considering training time, model download and upload time, and energy consumption. We rigorously prove that the time-averaged regret bound of our algorithm is $\mathcal{O}(\ln T/T)$.
- We combine Lyapunov optimization with CMAB to deal with the heterogeneous energy budget constraints. We creatively use a virtual energy deficit queue into the objective function of CMAB as a penalty to control the selection, which enables our algorithm to be aware of the selection frequency of clients. An upper bound of the time-averaged total queue is found.
- Extensive simulations are carried out to evaluate the performance of the proposed algorithm and validate our theoretical findings. The results show that our algorithm outperforms existing works and achieves close-to-optimal performance compared with the *Oracle* [22] that knows information such as training time and energy consumption of the clients in advance.

2. Related works

Nishio et al. [7] estimate the time required for model downloading, updating, and uploading using information sent from clients such as wireless channel states, computational capacities, and the size of data resources relevant to the current training task. They develop an algorithm based on the estimated time to select as many clients as possible before a defined deadline. They formulate the problem into 0–1

knapsack problem and solve it greedily. AbdulRahman et al. [8] define resource utilization as constraint considering availability of resources such as CPU, memory and energy and predict it based on linear regression. Their algorithm selects as many clients that satisfy the resource utilization constraint as possible in each round. They prove that their problem can reduce to classical knapsack problem and solve it greedily too. Xu et al. [12] take account of energy budgets and wireless channel conditions of clients, and define virtual energy queue to manage energy budgets of clients under the framework of Lyapunov optimization. The utility function is based on dataset size of clients and the defined virtual energy queue. They maximize utility under constraint of bandwidth limit using a selection priority metric based on queue length and channel gain. Wadu et al. [11] focus extensively on radio resource allocation, aiming to minimize the empirical loss function of clients within the constraints of bandwidth limits. They formulate the problem as a stochastic optimization problem and subsequently transform it into a series of optimization problems, which are solved at each round t using the Lyapunov framework. On the other hand, Xia et al. [10] take into account factors such as training time, client availability, and selection fairness, and formulate a reward function based on training time with the objective of minimizing regret. They employ the UCB algorithm to solve the problem. Huang et al. [9] investigate how the fairness of selection affects the training performance and define a selection metric named model exchange time which equals to total of model distribution time, training time and model upload time. Then they designed a policy based on Contextual Combinatorial Multi-Armed Bandit (CCMAB) to minimize model exchange time under the constraint of fairness while overlooking energy budget of clients. Li et al. [23] characterize the relationship between FL system performance and optimal aggregation round K, parameters, and the proportion of inactive clients. Deng et al. [24] concentrated on the reduction of training delays in FL while maintaining acceptable learning performance. Jin et al. [13] incorporated the cumulative aggregation constraint in FL in a non-stochastic setting. [25,26] placed excessive emphasis on various different client selection strategies.

Our research in this paper differs from aforementioned works. Those works such as [11,12] aim to optimize the utility of radio resource which is impractical as the information of radio resource is not available for central server in the real FL system. Related works such as [7,8] design some criteria for selection based on technique such as linear regression which is insufficient to describe relevant metrics. Other works such as [9,10] concentrate on training time and fairness of selection while neglecting energy consumption of clients. The recent work such as [13,24–26] primarily address specific facets of FL optimization and do not adequately consider the overarching constraint optimization problem. To sum up, none of the existing research, to the best of our knowledge, have studied client selection problem of FL from an online perspective with guaranteed training effectiveness under long-term heterogeneous energy budget constraint.

3. System model and problem formulation

In this section, we introduce the system model firstly and then describe the problem in detail. We further give the formulation of the problem.

3.1. System overview

We study client selection problem in FL system, which includes a central server and a crowd of heterogeneous clients. Each client has an energy budget for training model. The whole FL process is carried out under constraint that no client exceeds its energy budget. By defining virtual energy deficit queue for each client and ensuring that all virtual energy deficit queues remain stable, we make sure that there is no energy budget is exceeded at the end of FL process. Moreover, the computational resource of each client is heterogeneous and limited. We define the quality function based on training delay for each client and minimize it to avoid impact of *straggler*. There are several global models on the central server. The initial global models are some random generated models.

An initial phase in the whole process as shown in Fig. 2, in which the server selects all clients to obtain their delay and energy consumption information for initializing quality functions. At the beginning of each training round, we assume that the server selects the same number of clients, denoted by k, for each global model based on quality functions of previous round. The total number of selected clients is denoted by K. Then each global model is distributed to the corresponding selected clients. After trained on the clients, these models are uploaded back to the server for aggregation by Federated Average (FedAvg). At the meantime, Parameters such as training delay and energy consumption are uploaded to server for updating quality functions.

3.2. System modeling

Basic Model: Consider a FL system, composed of a central server and a set of clients (or *arms* in the language of MAB theory), represented as $\mathcal{N} = \{1, 2, \dots, N\}$, indexed by i, which can communicate with the server via wireless networks. Considering that the system is dynamic, we study it over a series of rounds $\mathcal{T} = \{0, 1, 2, \dots, T\}$, indexed by t. Each client i is equipped with built-in storage, CPUs and GPUs for participating in FL, powered by the local battery of which available capacity is B_i. There are a set of global models to be trained in the system, denoted by $\mathcal{J} = \{1, 2, \dots, J\}$, indexed by j. Specifically, these models are trained on clients and aggregated on the server.

System Workflow: In each round t, the server selects K clients altogether and divides them into groups of k, which gradually approaches the theoretical optimal value with training instead of a fixed value. In each round of training, the number of clients selected for each global model j to be trained is k, which ensures the fairness of each model training. Then the server distributes global models based on data collected by clients, namely clients holding data required by some global models are assigned to those models as much as possible. We use notation $S_j(t)$ to capture the set of the k clients involved for model j. Here we assume that there are enough clients to satisfy all global models. Then every selected client downloads its corresponding global model, performs gradient-descent steps on the local training data to update the model and then uploads the new model back to the server.

Energy Consumption of Clients: The energy consumption of clients in each round is formulated as $c_i(t) = \kappa \cdot C_i |D_i| f_i^2$ [27] where κ is the effective switched capacitance that depends on CPU architecture, C_i is the number of CPU cycles required for computing one sample data, $|D_i|$ is the size of data of client i, f_i is the CPU cycles per second.

According to [28,29], we can estimate that the energy consumption of transmission is about 10^0 J, and the computational energy consumption is about 10^2 J. The data upload for model updating may incur a minor portion of the total energy budget, especially when network bandwidth is ample. Despite potential network latency issues due to geographical dispersion of clients, our client selection algorithm mitigates the impact on system performance. Therefore, we neglect the energy consumption of transmission and formulate only the computational energy consumption in our objective function for simplicity.

Reward of Clients: Aiming at improving training effectiveness, we concern the model training time, model download and upload time. We use $D_i(t)$ to describe the total of model training time, model download and upload time of client i (if chosen) in round t. $D_i(t)$ is calculated as $D_i(t) = d_i(t) + d_i^{dl}(t) + d_i^{up}(t)$ where $d_i(t)$, $d_i^{dl}(t)$, $d_i^{up}(t)$ are model training time, model download and upload time of client i (if chosen) respectively. Considering the transmission errors [30], we set an upper bound D_{max} for $D_i(t)$. In a certain round, if a client's $D_i(t)$ exceeds the upper bound due to unstable wireless network or other reasons, our algorithm will set its $D_i(t)$ to D_{max} and ignore the client during global aggregation. We summarize the commonly used notations throughout the paper in Table 1.

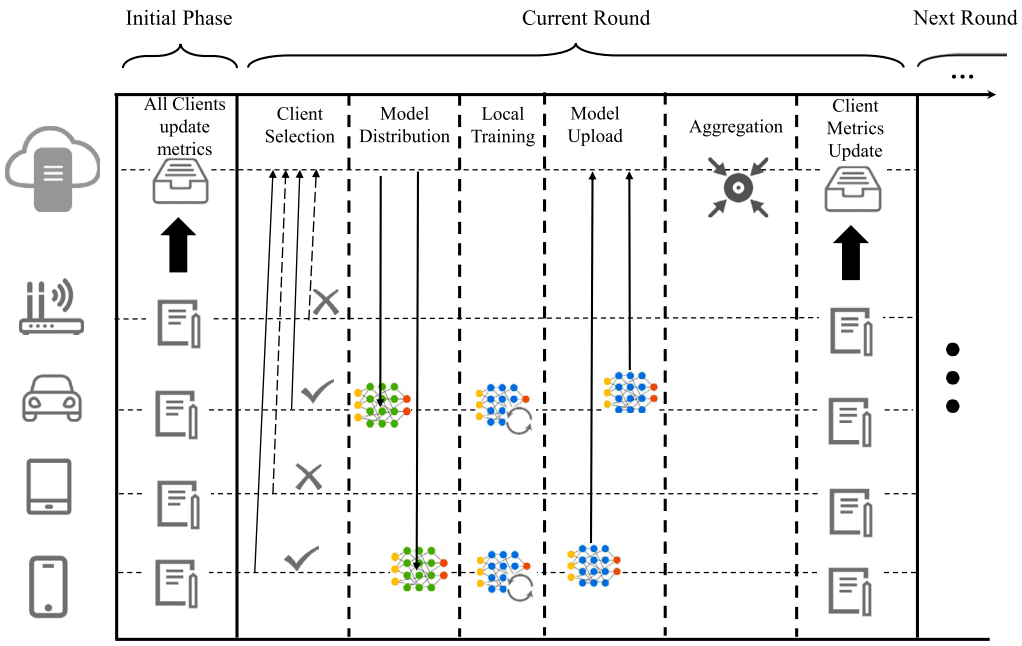


Fig. 2. System workflow.

Table 1
Description of notations.

Variable	Description
$\mathcal{N}$, $\mathcal{J}$	the sets of clients and global models, respectively.
B_i	the energy budget of client i.
i, j, t	the indexes for clients, global models and rounds.
k	the number of selected clients for each global model.
K	the total number of selected clients in each round.
$S_j(t)$	the set of selected clients for global model j in round t.
$S(t)$	the set of all selected clients in round t.
$c_i(t)$	the energy consumption of client i (if selected) in round t.
$D_i(t)$	the training delay of client i (if selected) in round t.
$n_i(t)$	the number of client i being selected until round t.
$q_i(t)$	the quality function of client i until round t.
$\bar{q}_i(t)$	the estimated quality function of client i in round t.
$\hat{q}_i(t)$	the UCB-based quality function of client i in round t.
V	the parameter for Lyapunov optimization.
$\Theta_i(t)$	the virtual energy deficit queue of client i in round t.
$\mathbb{I}\{\cdot\}$	the indicator function.

3.3. Problem formulation

In each round t, the FL server decides whether client i is chosen or not, which is captured by $I_i(t) = \{0, 1\}$. $I_i(t) = 1$ means client i is selected in round t while $I_i(t) = 0$ means otherwise. We use $S(t)$ to describe the set of all selected clients in round t, *i.e.* $S(t) = \cup_{j\in\mathcal{J}} S_j(t)$, and let $K = |S(t)|$. Let $\mathcal{P} = \{S | S \in 2^{\mathcal{N}}, |S| = K\}$. Firstly, a quality function is defined to describe performance of each client which equals the reward of clients, namely $D_i(t)$. Because the training speed of all users depends on the slower computational training among users, we define a utility function for $S_j(t)$ to capture the utility of the set of clients who train the same global model j, which is $u_j(t) = \max_{i\in S_j(t)} D_i(t)$. The objective is to minimize the utility function. Based on the above definitions, we formulate the client selection problem as follows:

$$\textbf{(P1)} \quad \min_{\{S_j(1), S_j(2), \ldots, S_j(T)\}_{j\in\mathcal{J}}} \sum_{t=1}^{T} \sum_{j\in\mathcal{J}} u_j(t),$$

$$\textbf{s.t.} \quad \sum_{t=1}^{T} \sum_{j\in\mathcal{J}} \mathbb{I}\{i \in S_j(t)\} c_i(t) \le B_i, \forall i \in \mathcal{N}, \tag{1}$$

$$|S_j(t)| = k, \forall j \in \mathcal{J}, t = 1, 2, \ldots, T. \tag{2}$$

where $S_j(t)$ is our optimized target which captures the selected clients for global model j in each round. Constraint (1) is a long-term constraint which means that the total energy consumption of each client i should not exceed its budget B_i. Constraint (2) means that the server selects k clients for global model j in each round.

Note that **(P1)** is a time-coupling optimization problem, with respect to the long-term objective and the energy budget constraint (1), meaning that the required training parameters can only be uploaded to the server after users participate in training. Such a problem is NP-hard as the problem can be reduced to a standard knapsack problem where the energy budget is corresponding to capacity of knapsack; the energy consumption is corresponding to weight of item and $u_j(t)$ is as value of item. Furthermore, the information on quality function can only be observed after involving the clients in training. Nevertheless, the server is supposed to make a selection decision before the real training process when the actual value of quality function is unachievable. The nature behind this fact is that the server have to make decision without foreseeing future information. Therefore, for an alternative sub-optimal solution, we transform the offline problem to a round-by-round online problem under the framework of MAB theory.

(P1) is to minimize the total maximum training delay, the training delay $u_i(t) = max_{i\in S_j(t)} D_i(t) = max_{i\in S_j(t)}(d_i(t) + d_i^{dl}(t) + d_i^{up}(t))$. The $d_i(t)$ may increase if the loss function involves complex calculations or a large amount of data manipulation. In addition, the effectiveness of the loss function directly affects the speed of convergence of the system model; a more effective loss function will allow the model to converge faster, thus reducing training latency.

3.4. Problem transformation

P1 is a time coupled optimization problem, taking into account long-term goals and energy budget constraints. In the overall training process, if we are only aiming to find the optimal decision that means

single optimal user in each round of training, the MAB theory will be a good solution. However, in this paper, the problem is to find the optimal set of clients $S(t)$ for each global model in each round of training, which correspond to the CMAB theory. Therefore, we adopt CMAB theory to transform **(P1)** to online fashion. The estimated *Upper Confidence Bound* of quality function is defined as follows,

$$\hat{D}_i(t) = \bar{D}_i(t) - Q_i(t), \tag{3}$$

where

$$\bar{D}_i(t) = \frac{\sum_{\tau=0}^{t-1} \mathbb{I}\{i \in S(\tau)\} D_i(\tau)}{n_i(t-1)}, \tag{4}$$

is the empirical mean of $D_i(t)$, $Q_i(t) = \sqrt{\frac{(K+1)\ln t}{n_i(t-1)}}$ [31], and $n_i(t)$ denotes the number of times that client i has been chosen till round t.

Remark 1. The notation $t = 0$ in Eq. (4) means the initialization phase of our algorithm rather than the literal meaning. Actually the *Upper Confidence Bound* aforementioned is the Lower Confidence Bound of quality function. This is the language of MAB theory and does not change the nature of our algorithm as we minimize it. Therefore, we still adopt the expression of *Upper Confidence Bound*.

Replacing $D_i(t)$ with $\hat{D}_i(t)$ in **P1**, we now introduce the transformed form of our problem as follows,

$$\textbf{(P2)} \quad \min_{\{S_j(t)\}_{j\in\mathcal{J}}} \sum_{j\in\mathcal{J}} \max_{i\in S_j(t)} \hat{D}_i(t),$$

s.t. (1), (2).

According to [32], CMAB will stick to some certain group of clients after several iterations, which is not acceptable for those clients as they have other tasks to process. To balance selection among clients, we define a virtual energy deficit queue for each client i shown as follows:

$$\Theta_i(t+1) = \left[\Theta_i(t) + \sum_{j\in\mathcal{J}} \mathbb{I}_{i,j}(t) c_i(t) - \frac{B_i}{T_0}\right]^+, \tag{5}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$, $\mathbb{I}_{i,j}(t) = \mathbb{I}\{i \in S_j(t)\}$. T_0 means upper bound of total round of our algorithm. In real implementations, it is difficult to obtain the exact value of T_0. However, a reasonably good estimate of T_0 can be obtained based on the history data, *e.g.*, setting T_0 as the maximum T that has been observed. One can notice that the virtual energy deficit queue $\Theta_i(t)$ will remain small if a client is less selected otherwise it will increase rapidly. Through minimize $\Theta_i(t)$, we can guide our algorithm to choose clients who are less selected. Another effect of virtual energy deficit queue $\Theta_i(t)$ is to manage energy consumption of clients and constraint (1) shall be satisfied as long as the queues remain stable, formally (see Theorem 3):

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i\in\mathcal{N}} \mathbb{E}\{\Theta_i(t)\} < \infty. \tag{6}$$

Now we present Theorem 1 to justify the rationale for this statement.

Theorem 1. *Long-term energy budget constraint (1) holds if all virtual energy deficit queues remain stable across the FL process.*

Proof. According to the queue theory (Theorem 2.5, [33]), if all the virtual queues $\Theta_i(t)$ remain stable across the FL process, we have:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\left[\sum_{j\in\mathcal{J}} \mathbb{I}_{i,j}(t) c_i(t) - \frac{B_{\min}}{T_0}\right] \le 0. \tag{7}$$

We transform the **(P1)** offline problem into a round by round online problem **(P2)** through the MAB theoretical framework. Under this framework, we can divide the total energy budget in constraint (1) into each $t (t = 1, 2, \ldots, T_0)$ round on average, T_0 is an upper bound of the number of training rounds. In actual implementation, it is difficult to obtain the exact value of T_0 directly. Thus we can apply expectations and limits to solve it, which provides convenience for subsequent calculations. Rearranging $\frac{B_{\min}}{T_0}$ in inequality (7) to its right-hand side and multiplying both side by T yields constraint (1). This completes the proof. □

Combining CMAB and Lyapunov [34,35], we rewrite the quality function as follows:

$$q_i(t) = V \cdot D_i(t) + \Theta_i(t) c_i(t), \tag{8}$$

and the estimate of $q_i(t)$ is

$$\bar{q}_i(t) = V \cdot \bar{D}_i(t) + \Theta_i(t) \bar{c}_i(t), \tag{9}$$

where $\bar{c}_i(t) = \frac{\sum_{\tau=0}^{t-1} \mathbb{I}\{i\in S(\tau)\} c_i(\tau)}{n_i(t-1)}$, then the *Upper Confidence Bound* of quality function is

$$\hat{q}_i(t) = \bar{q}_i(t) - Q_i(t). \tag{10}$$

We give the final form of the problem, as shown in the following:

$$\textbf{(P3)} \quad \min_{\{S_j(t)\}_{j\in\mathcal{J}}} \sum_{j\in\mathcal{J}} \max_{i\in S_j(t)} \hat{q}_i(t),$$

s.t. (6),(2)

The final problem **(P3)** is indeed theoretically solvable as the server only needs to sort the estimated quality function values of all clients, and then select the K smallest ones.

4. Energy-aware client selection algorithm

In this section, we present basic solution of problem **(P3)**, describe the detail of our proposed algorithm, and give the related analysis.

4.1. Basic solution

To address **(P3)**, we model it as a heterogeneous budget-limited CMAB problem, where each client is seen as an arm, training delay is seen as corresponding reward, and selecting clients is treated as pulling arms. Pulling an arm will generate corresponding energy consumption and the actions must be taken under constraint of no client exceeds its energy budget. To deal with the heterogeneous budget, we combine Lyapunov optimization with CMAB algorithm. By minimizing the defined virtual energy deficit queue and ensuring its stability, the constraint on energy budgets will not be violated. Before the beginning of FL process, the proposed algorithm will select all clients to estimate their delay and energy consumption to initialize virtual energy deficit queue and quality function. Our algorithm selects the K arms with smallest quality function and updates estimate of training delay and energy consumption using feedback from clients iteratively. Each group of k clients is assigned to its corresponding global model as shown in Fig. 3 where the clients are sorted by the UCB-based quality function in ascending order. The correspondence between global models and groups of clients is based on the information about datasets of clients in those groups.

4.2. Detailed algorithm

We propose Energy Aware Client Selection FL (EACS-FL) algorithm (Algorithm 1) to solve the online problem **(P3)**. The proposed algorithm is comprised of two main phases. The first is initial phase shown in Line 1 where the server selects all clients and observes the feedback. In the initial phase, all of the clients are selected and D_i^t and c_i^t of all clients are gathered by the server to initialize our algorithm. The second is main loop shown in Line 3~8, where the server selects clients according to historical performance of clients. One could notice that in Line 3, $\hat{q}_i^{t-1}$, rather than $\hat{q}_i^t$, because the server cannot access $\hat{q}_i^t$ at the beginning of round t since it is observed at the end of round t. The reason is the server cannot access $\hat{q}_i^t$ at the beginning of round t since it is

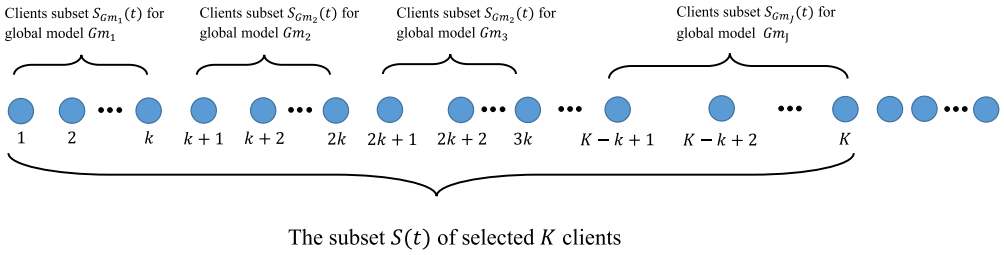


Fig. 3. Assignment of K selected clients.

observed at the end of round t. The CMAB theory is utilized to estimate performance of clients. The optimization problem, which is a sorting problem with computational complexity $O(N \log N)$, in Line 3 aims to minimize a weighted sum of the delay and energy consumption where the weight depends on the current virtual energy deficit queue length and is varying over time. Here the Lyapunov optimization enables us to manage energy consumption of clients, making sure that no client exceeds its energy budget. After selection, invokes algorithm 2 to assign a group of clients to each global model in Line 4. Next clients involved conduct FL process and upload local model to server in Line 5, then server observes and updates relevant parameters as shown in Line 6∼9.

The data held by each client is different, and the data required by each global model on the server may also be different. Therefore, an algorithm is needed to match clients possessing data required by some global models. Algorithm 2 determines the correspondence between global models and client groups. Specifically, a bipartite graph $\mathcal{G} = \{\mathcal{J} \cup \boldsymbol{\Phi}(t), \mathcal{E}, \mathcal{X}\}$ is built to describe the correlation between datasets of client groups and each global model, where $\boldsymbol{\Phi}(t) = \{S_l(t), l = 1, 2, \dots, J\}$. $\mathcal{E}$ indicates the set of edges (*i.e.*, client group-global model pairs) in the bipartite graph. $S_l(t)$ represents the selected client set of global model j in round t. We use $S_l(t)$ represent the best matching relationship between the clients and the global models to be trained based on the maximum weight matching algorithm. $\mathcal{X} = \{x_{l,j}, \forall j \in \mathcal{J}, S_l(t) \in \boldsymbol{\Phi}(t)\}$ represents the weights of corresponding edges. The weight $x_{l,j}$ is defined as total size of data relevant to global model j on clients in $S_l(t)$. We use $m_{i,j}$ to capture size of data relevant to global model j on client i, which represents the weight value of the edge between the user set $\boldsymbol{\Phi}(t)$ and the global model J. Then $x_{l,j} = \sum_{i \in S_l(t)} m_{i,j}$. Algorithm 2 takes global models $\mathcal{J}$, $\boldsymbol{\Phi}(t)$, $m_{i,j}$ as input and outputs the correspondence. Each weight $x_{l,j}$ in the bipartite graph $\mathcal{G}$ is initialized as 0 as shown in Line 1. In Line 2∼6, Algorithm 2 calculates weight $x_{l,j}$ for all $S_l(t) \in \boldsymbol{\Phi}(t)$ and $j \in \mathcal{J}$. Finally in Line 7, the maximum weight matching algorithm [36] is invoked to find a subset of edges $\mathcal{X}$ that has the maximum sum of weights. This yields the desired correspondence. The computational complexity of Algorithm 2 is $O(N^{\frac{5}{2}})$ according to [36]. Although the optimization problem mentioned in Line 3 of Algorithm 1 involves a computational complexity associated with sorting, its execution frequency throughout the entire algorithm is relatively limited. Moreover, the time complexity of $O(N \log N)$ for this sorting optimization problem is considerably smaller compared to the $O(N^{\frac{5}{2}})$ complexity of Algorithm 2. The remaining operations within the loop are of constant time complexity. Therefore, the overall time complexity of Algorithm 1 is dominated by the time complexity of Algorithm 2, specifically $O(N^{\frac{5}{2}})$, where N is the number of edges in the bipartite graph $\mathcal{G}$. Since Algorithm 1 and Algorithm 2 are both applied to the FL system, which is generally composed of multiple clients, meaning that the algorithms can be well extended in clients.

4.3. Theoretical analysis

The client assignment shown in Fig. 3 is a solution of **(P3)** which presented as follows:

Algorithm 1 The EACS-FL algorithm

Input: $\mathcal{J}, \mathcal{N}, B_i$ for $i \in \mathcal{N}$, k, V, T_0
Output: $S_j(t)$ for $j \in \mathcal{J}, t = 0, 1, 2, 3, \dots, T$
1: **Initialization**:
$t \leftarrow 0, \Theta_i(0) \leftarrow 0, n_i(0) \leftarrow 1, c_i(0) \leftarrow 0, B_i(0) \leftarrow B_i$; select all clients, *i.e.* $S(0) \leftarrow \mathcal{N}$, observe $D_i(0)$ and $c_i(0)$ for all $i \in \mathcal{N}$.
2: **for** t *in* $1, 2, \cdots, T$ **do**
3: Get $S_j(t)$ for $\forall j \in \mathcal{J}$ such that

$$\{S_j(t)\}_{j \in \mathcal{J}} \in \underset{S_j(t) \in \mathcal{P}, j \in \mathcal{J}}{\arg\min} \sum_{j \in \mathcal{J}} \max_{i \in S_j(t)} \hat{q}_i(t-1)$$

4: Invoke algorithm 2 to assign selected clients for each global model
5: Conduct FL process at selected clients
6: Observe $D_i(t), c_i(t)$, for $i \in \cup_{j \in \mathcal{J}} S_j(t)$
7: Update $n_i(t), \bar{D}_i(t), \Theta_i(t), \bar{q}_i(t), Q_i(t), \hat{q}_i(t)$
8: $B_i(t) \leftarrow B_i(t-1) - c_i(t)$ for $i \in \cup_{j \in \mathcal{J}} S_j(t)$
9: $t \leftarrow t + 1$
10: **end for**

Theorem 2. *Given the sequence of quality functions of clients in $\mathcal{N}$, the sequence of subsets $S_{Gm_j}(t), \forall Gm_j \in \mathcal{J}$ shown in Fig. 3 is a valid solution of* ***(P3)****.*

Proof. Suppose that a sequence of subsets $S'_j(t)$ denotes any assignment that split any K clients into several groups of k clients. For convenience of description, we assume that clients in all $S'_j(t)$ are sorted by the UCB-based quality function in ascending order. Then we have

$$\sum_{j \in \mathcal{J}} \max_{i \in S'_j(t)} \hat{q}_i(t) = \sum_{j \in \mathcal{J}} \hat{q}_{k_j}(t),$$

where $\hat{q}_{k_j}(t)$ is the UCB-based quality function of kth client in subset $S'_j(t)$. For all $S_{Gm_j}(t)$, we have

$$\sum_{Gm_j \in \mathcal{J}} \max_{i \in S'_{Gm_j}(t)} \hat{q}_i(t) = \sum_{i=1}^{J} \hat{q}_{ik}(t).$$

Since the sequence of $S_{Gm_j}(t)$ is generated in ascending order of quality function, it is obviously that

$$\sum_{i=1}^{J} \hat{q}_{ik}(t) \le \sum_{j \in \mathcal{J}} \hat{q}_{k_j}(t).$$

As the sequence of $S'_j(t)$ denotes any assignment, the sequence of $S_{Gm_j}(t)$ is a valid solution of **(P3)**. This complete the proof. □

In MAB theory, *regret* is used to measures the performance gap between a given policy and the optimal policy. For ease of analysis, we define time average regret of proposed algorithm.

Definition 1.

$$R(T) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left\{ \sum_{j \in \mathcal{J}} \max_{i \in S_j(t)} q_i(t) - \sum_{j \in \mathcal{J}} \max_{i \in S_j^*} q_i(t) \right\}, \tag{11}$$

Algorithm 2 The algorithm for assigning selected clients

Input: $\mathcal{J}, \Phi(t), m_{i,j}$
Output: the correspondence between $\mathcal{J}$ and $\Phi(t)$
1: **Initialization**: $x_{l,j} = 0$
2: **for** $j = 1, 2, \cdots, J$ **do**
3: **for** $l = 1, 2, \cdots, J$ **do**
4: $x_{l,j} = \sum_{i \in S_l(t)} m_{i,j}$
5: **end for**
6: **end for**
7: Conduct the maximum weight matching algorithm [36] in terms of the weight $x_{l,j}$, and output the result.

where we use S_j^* to represent the selection made by the optimal policy.

Before our proof, there is an assumption we need to introduce.

Assumption 1. Let T_0 denotes the upper bound of total round of our algorithm, then

$$\frac{B_{\min}}{T_0} > c_{\max},$$

where $B_{\min} = \min_{i \in \mathcal{N}}\{B_i\}$, $c_{\max}$ denotes the maximum of energy consumption in a single round.

Assumption 1 means that clients have enough energy budget to participate in each round of FL training. With definitions and assumptions above, we now present strict upper bound of time-averaged total queue length and time average regret as follows.

Theorem 3. *Suppose system satisfies Assumption 1, then our algorithm can achieve queue stability in the systems, i.e. there exist constants $\mathcal{B}$ and ϵ, such that*

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in\mathcal{N}} \mathbb{E}\{\Theta_i(t)\} \le \frac{1}{\epsilon}(\mathcal{B} + VNc_{\max}D_{\max}), \tag{12}$$

where $\mathcal{B} = \frac{1}{2}N|\mathcal{J}|^2 c_{\max}^2 + \frac{1}{2}N\frac{B_{\max}^2}{T_0^2}$, $\epsilon = \left(\frac{B_{\min}}{T_0} - c_{\max}\right)$

Proof. See Appendix. □

Theorem 4. *The worst time-averaged regret of our algorithm is bounded as*

$$R(T) \le \frac{\mathcal{B}}{V} + Nc_{\max}D_{\max} + \frac{1}{T}N(\lambda_1 \ln T + \lambda_2)\Delta_{\max}, \tag{13}$$

where $\lambda_1 = \frac{4K^2(K+1)}{\Delta_{\min}^2}$, $\lambda_2 = 1 + \frac{K\pi^2}{3}$

Proof. See Appendix. □

4.4. Convergence analysis of FL

In this work, we consider the following distributed optimization model:

$$\min_{\mathbf{w}}\left\{F(\mathbf{w}) \triangleq \sum_{k=1}^{N} p_k F_k(\mathbf{w})\right\},$$

where N is the number of devices, and p_k is the weight of the kth device such that $p_k \ge 0$ and $\sum_{k=1}^{n}$. Suppose the kth device holds the n_k training data: $x_{k,1}, x_{k,2}, \ldots, x_{k,n_k}$. The local objective $F_k(\cdot)$ is defined by

$$F_k(\mathbf{w}) \triangleq \frac{1}{n_k}\sum_{j=1}^{n_k} \ell(\mathbf{w}; x_{k,j}),$$

where $\ell(\cdot;\cdot)$ is a user-specified loss function.

Here, we describe one around (say the tth) of the standard FedAvg algorithm. First, the central server broadcasts the latest model, w_t, to all the devices. Second, every device (say the kth) lets $w_t^k = w_t$ and then performs local updates:

$$\mathbf{w}_{t+1}^k \iff \mathbf{w}_t^k - \eta_t \nabla F_k(\mathbf{w}_t^k, \xi_t^k),$$

where η_t is the learning rate and ξ_t^k is a sample uniformly chosen from the local data belonging to kth client in tth round.

In each round t, the server chooses a subset$S(t) \subseteq \mathcal{N}$ of the clients and then distributes the weight vector $x(t)$ of the global model to the selected clients. After receiving the global model weights, each of the selected clients individually updates the global model by computing the gradients of their local loss functions based on their own private data and then uploads the updated gradients to the server for model aggregation, *i.e.*

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \gamma v(t), \tag{14}$$

where $v(t) = \sum_{k=1}^{n} p_k \nabla F_k(x(t))$.

Then, we highlight the necessary assumptions and then provide the convergence guarantee.

Assumption 2. $F_1, \ldots, F_N$ are all L-smooth: for all $\mathbf{v}$ and $\mathbf{w}$, we have

$$F_k(\mathbf{v}) \le F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{L}{2}\|\mathbf{v}-\mathbf{w}\|_2^2.$$

Assumption 3. $F_1, \ldots, F_N$ are all μ-strongly convex: for all $\mathbf{v}$ and $\mathbf{w}$,we have

$$F_k(\mathbf{v}) \ge F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2.$$

Assumption 4 (*Bounded Variance*)**.:** It is assumed that the difference between any correct gradient estimator $\nabla F(x(t))$ in any round t and the gradient estimator $\nabla F_k(x(t))$ of k users has upper bounded variance:

$$\mathbb{E}\left\|\nabla F(x(t)) - \sum_{k=1}^{n} p_k \nabla F_k(x(t))\right\|_2^2 \le \sigma_0^2.$$

Theorem 5. *Under Assumptions 2–4 and taking $\gamma \le \frac{1}{L} \le \frac{1}{\mu}$, we have*

$$\mathbb{E}[F(\mathbf{x}(T)) - F(\mathbf{x}^*)] \le \frac{\delta_0^2}{2\mu} + (1-\gamma\mu)^{T-1}\mathbb{E}\left[F(\mathbf{x}(1)) - F(\mathbf{x}^*) - \frac{\delta_0^2}{2\mu}\right], \tag{15}$$

where x^* denotes the optimal weights.

Proof. See Appendix. □

5. Experiments

In this section, we exhibit our experimental setup and experiment results.

5.1. Experimental setup

We implement the client selection algorithm, and construct the FL framework using Flower [37] as the training pipeline. We conduct our experiments on a desktop server with an Intel Xeon E5 CPU, 32 GB RAM, 2TB HDD, 512 GB SSD, and the Linux Ubuntu 16.04.1 LTS operating system. There are 20 clients emulated by Docker containers. The energy budgets of clients vary from 1300 J to 1600 J randomly. To maximize the global accuracy, the value of k is set to 2 according to a series of experiments.

As the t increases, the system model will gradually converge; however, at this point, the amount of model updates for each client $|D_i|$ will be very limited, and the selected client $S_j(t)$ may be basically the same in each round; In addition, the total energy consumption budget B_i of each client will increase, but the clinet's consumption budget

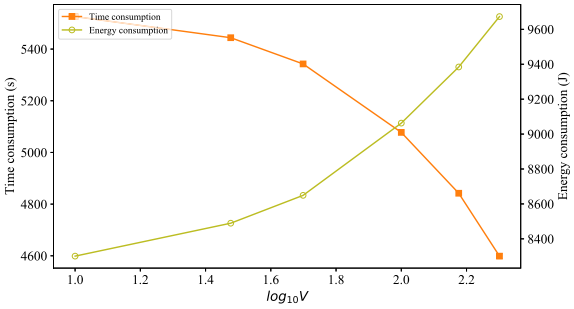


Fig. 4. Energy and time consumption over parameter V (300 rounds).

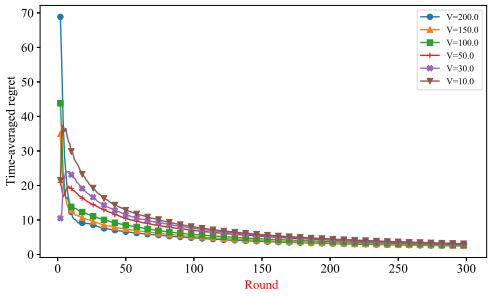


Fig. 5. Time-averaged regret under different V over round.

$c_i(t)$ for each round will become smaller because $|D_i|$ will decrease as t increases; the training delay $D_i(t)$ will also become smaller due to the decrease in the amount of data. t can indeed be used as a decision variable, but since t is less relevant to our optimization objective, we set t to a fixed value.

Dataset and Global models: We utilize extended handwritten digit classification dataset EMNIST [38] for FL. The digits part of EMNIST contain 28 × 28 gray-scale images in 10 classes, a training set of 240,000 examples and a test set of 40,000 examples. We combine all 240,000 examples in training set and 32,000 examples in test set together and split these examples into 20 groups for 20 clients as their training set. The remaining 8,000 examples are utilized as the test set for all clients. The are 3 global models of fully connected network composed as follows: a fully connected layer as the input layer followed by ReLU activation, another fully connected layer followed by ReLU activation as hidden layer, and a final fully connected layer with Softmax as output layer.

To evaluate the performance of the proposed EACS-FL algorithm, we vary the value of V and compare our algorithm with two baseline algorithms.

- Random: in each round, the clients participating in FL process are selected randomly.
- CS-UCB-Q [10]: the server selects clients with smallest delay under the constraint of fairness.

Metrics including training delay (or total time consumption), total energy consumption, time-averaged regret of selection, time-averaged total queue, total quality functions, test accuracy of global models are evaluated in our experiments.

5.2. Experimental results

Fig. 4 depicts the impact of the control parameter V on the total time consumption and total energy consumption. As seen in Fig. 4, parameter V influences the focus of our algorithm. By increasing V from 10 to $10^{2.3}$, EACS-FL cares more about the delay performance, and thus the total time consumption decreases from 5,500 s to 4,600 s. However, with less concern on the energy consumption when V increases from 10 to $10^{2.3}$, the total energy consumption increases from 8,300 J to 9,700 J. In addition, it obviously shows that the time and energy

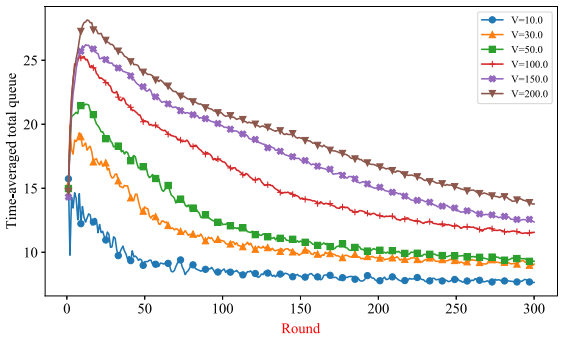


Fig. 6. Time-averaged total queue under different V over round.

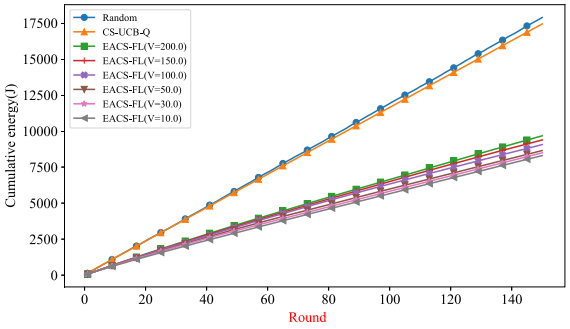


Fig. 7. Cumulative energy consumption of different strategies over round.

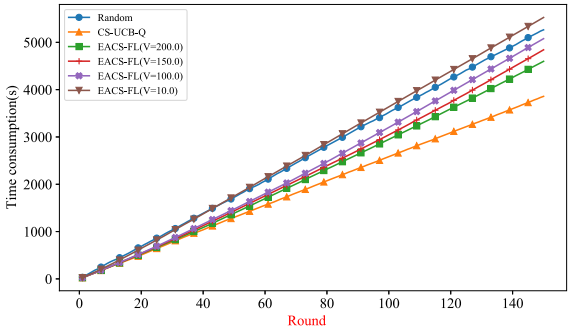


Fig. 8. Training delay of different strategies over round.

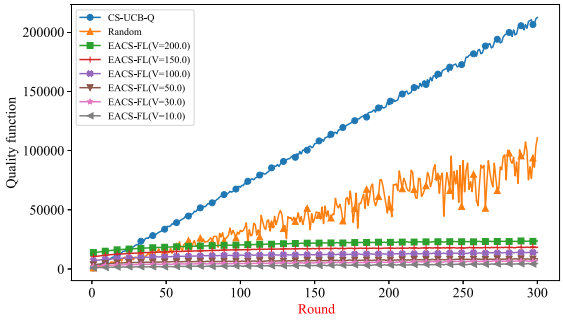


Fig. 9. Quality function of different strategies over round.

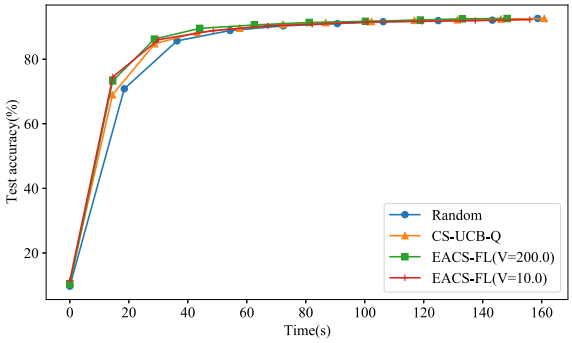


Fig. 10. Global accuracy over time.

consumption follows the $[\mathcal{O}(1/V), \mathcal{O}(V)]$ tradeoff. With the increase of V, the total energy consumption increases linearly and the time consumption decreases inversely.

We explore time-averaged regret of our algorithm under different settings of parameter V, as shown in Fig. 5. After several iterations, the bigger parameter V is, the smaller time-averaged regret returns. One can easily find this trend from $t = 6$ to $t = 100$. When $t = 250$, the time-averaged regret gradually becomes stable and finally stabilizes at about 4. Furthermore, the time-averaged regret increases first and then decreases over round which has the form of $\mathcal{O}(\ln T/T)$, by which Theorem 4 is verified.

Fig. 6 demonstrates the trend of time-averaged total queue over round. According to Fig. 6, the time-averaged total queue increases first, then decreases and becomes stable finally. Moreover, as parameter V increases, the upper bound of time-averaged total queue increases too, which verifies Theorem 3.

We compare our algorithm with two baseline strategies that are commonly used in the field, *i.e.* random and CS-UCB-Q. Note that we have made an adaptation to CS-UCB-Q in order to tailor it to our context, but the basic idea is the same as the vanilla one.

We evaluate the energy performance of our algorithm which is presented in Fig. 7. Since CS-UCB-Q does not consider the influence of energy [10], its energy consumption is close to random selection strategy and is significantly higher than our algorithm under different settings of control parameter V. We can find that when $V = 200$, the energy consumption of our algorithm is 54.6% of CS-UCB-Q and it is only 45.5% of CS-UCB-Q when $V = 10$. Compared with random strategy, the energy consumption of our algorithm is 54.0% of it with $V = 200$ and 45.0% with $V = 10$.

Fig. 8 exhibits delay performance of different strategies. As CS-UCB-Q is more concerned with the effect of delay, the total time consumption of it is smaller than that of ours, since we focus on both delay and energy consumption. The total time consumption of our algorithm is 44% more than that of CS-UCB-Q with $V = 10$ and it is only 19% more than that of CS-UCB-Q with $V = 200$. Compared with random policy, our algorithm is more concerned about the influence of energy consumption when the parameter V is small, so that the delay performance of EACS-FL is worse than it, yet when the parameter V becomes big, our algorithm outperforms the random strategy. It should be noted that the design motivation of the CS-UCB-Q is to enable server to select clients with minimal latency under the constraint of fairness. In our approach, we consider a holistic set of criteria when choosing clients, including training delay, total energy consumption, time-averaged regret of selection. In comparison to our algorithm, CS-UCB-Q tends to excessively prioritize performance in terms of training latency while potentially undervaluing the impact of other constraints on the federated system. While this may lead to outstanding performance in terms of training latency, it could result in sub-optimal overall system performance.

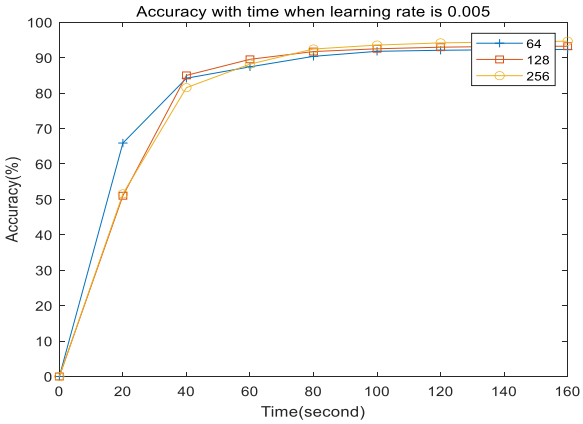


Fig. 11. Global accuracy over time when $\eta = 0.005$.

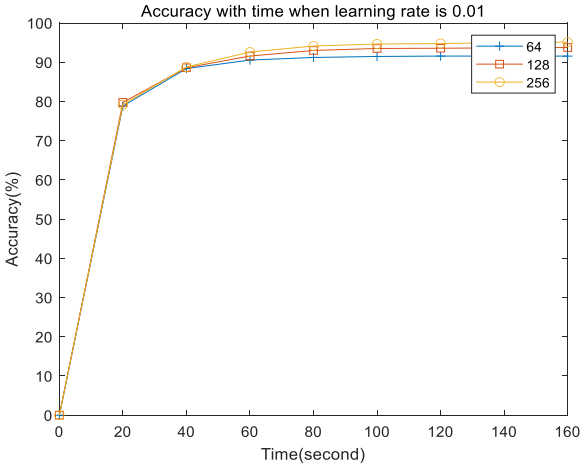


Fig. 12. Global accuracy over time when $\eta = 0.01$.

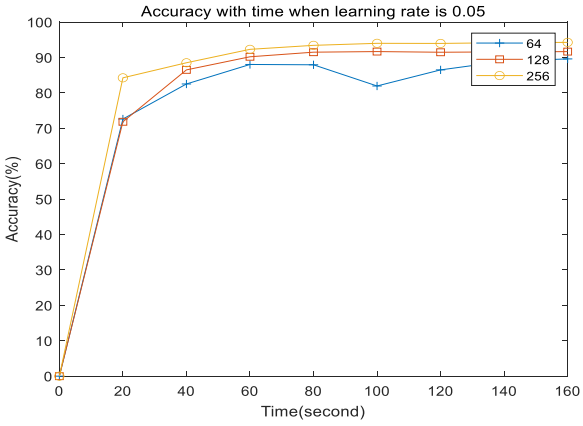


Fig. 13. Global accuracy over time when $\eta = 0.05$.

The performance of quality function is shown in Fig. 9. The quality function is a composite indicator combining delay and energy, which comprehensively reflects the performance of a selection strategy. With reference to Fig. 4, we can see that increasing the value of V will increase the energy budget and delay. Therefore, when we increase the value of V, the starting point of the quality function does not start from 0. The reader can see that quality function of our algorithm is obviously lower than that of CS-UCB-Q and random. It is obviously

shown that quality function of CS-UCB-Q increases linearly and that of random fluctuates a lot but also grows almost in linear pattern.

As shown in Fig. 10, we simulate the average accuracy of global model. The simulation results validate that our algorithm takes less time than random under the condition of achieving the same accuracy, and achieves the same performance as CS-UCB-Q in general. Take global accuracy reaching 80% for example, the training delay of EACS-FL is 89.53% of CS-UCB-Q and 74.35% of random strategy.

We validate the FL settings also by adapting to different hyperparameters, with batch sizes of 64, 128, and 256, and learning rates of 0.005, 0.01, and 0.05. V is set to 200. Figs. 11–13 shows the accuracy of FL in learning rate of 0.005, 0.01 and 0.05. Three figures show the same trend that the accuracy is higher when the batch size is set bigger. Overall, Fig. 12 illustrates that a batch size of 256 and a learning rate of 0.01 achieve the best accuracy in the shortest time. This combination is chosen for our experiments due to its high accuracy, albeit with a higher energy cost during the process. This heightened energy cost accentuates the changes in results, making this combination more pronounced.

6. Conclusion

In this paper, we propose a novel algorithm for client selection in FL system, considering the long term energy constraints of the client, and design an online selection strategy based on CMAB theory and Lyapunov optimization to minimize the energy consumption and delay at the same time. By theoretical proof and experimental verification, our algorithm can achieve close to optimal performance of delay and energy. In addition, by adjusting the control parameter V, our algorithm can make a trade-off between delay and energy consumption to meet the needs of the FL system. Experiment results show that the proposed algorithm saves 54.5% energy compared with the state-of-the-art while maintaining the same level of test accuracy of FL system.

CRediT authorship contribution statement

Konglin Zhu: Formal analysis, Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing. **Fuchun Zhang:** Methodology, Investigation, Formal analysis, Conceptualization, Software, Validation, Visualization, Writing – original draft. **Lei Jiao:** Conceptualization, Investigation, Methodology, Supervision, Funding acquisition, Writing – review & editing. **Bowei Xue:** Writing – review & editing, Visualization, Validation, Software, Investigation, Formal analysis, Data curation. **Lin Zhang:** Resources, Project administration, Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix

Before proof, we introduce a counter $C_i(t)$ for each client which is defined as follows. Each round a non-optimal set of arms is selected, that is $S(t) \neq S^*$, we increase the smallest counter in $S(t)$:

$$C_j(t) \leftarrow C_j(t-1) + 1, j = \arg\min_{j \in S(t)} C_j(t-1)$$

Now we introduce a lemma shown as follows,

Lemma 1. *Upon termination of proposed algorithm, the upper bound of* $\mathbb{E}\left[C_i(T)\right]$ *is:*

$$\mathbb{E}\left[C_i(T)\right] \leq \frac{4K^2(K+1)\ln T}{\Delta_{\min}^2} + 1 + \frac{K\pi^2}{3}$$

Proof.

$$\begin{aligned}
C_i(\tau) &= \sum_{t=1}^{\tau} \mathbb{I}\left\{I_i(t) = 1\right\} \\
&= l + \sum_{t=1}^{\tau} \mathbb{I}\left\{I_i(t) = 1, C_i(t) \geq l\right\} \\
&= l + \sum_{t=1}^{\tau} \mathbb{I}\left\{\sum_{j \in \mathcal{J}} \max_{i' \in S_j(t)} \hat{q}_{i'}(t-1) \leq \sum_{j \in \mathcal{J}} \max_{i' \in S_j^*} \hat{q}_{i'}(t-1), C_i(t) \geq l\right\} \\
&= l + \sum_{t=1}^{\tau} \mathbb{I}\left\{\sum_{j \in \mathcal{J}} \left[\sum_{i' \in S(t)} \psi_j^{i'}(t) \hat{q}_{i'}(t-1)\right]\right. \\
&\leq \left.\sum_{j \in \mathcal{J}} \left[\sum_{i' \in S}^{*} \psi^{*i'}{}_j \hat{q}_{i'}(t-1)\right], C_i(t) \geq l\right\} \\
&= l + \sum_{t=2}^{\tau} \mathbb{I}\left\{\sum_{i' \in S(t)} \lambda_{i'}(t-1)\hat{q}_{i'}(t-1) \leq \right. \\
&\left. \sum_{i^* \in S^*} \lambda_{i^*}(t-1)\hat{q}_{i^*}(t-1),\ C_i(t) \geq l\right\}
\end{aligned} \tag{16}$$

where

$$\psi_j^{i'}(t) = \mathbb{I}\{i' = \arg\max_{i'' \in S_j(t)} \hat{q}_{i''}(t-1)\}$$

$$\psi^{*i'}{}_j = \mathbb{I}\{i' = \arg\max_{i'' \in S_j^*} \hat{q}_{i''}(t-1)\}$$

$$\lambda_{i'}(t-1) = \sum_{j \in \mathcal{J}} \mathbb{I}\left\{i' = \arg\max_{i'' \in S_j(t)} \hat{q}_{i''}(t-1)\right\}$$

$$\lambda_{i^*}(t-1) = \sum_{j \in \mathcal{J}} \mathbb{I}\left\{i^* = \arg\max_{i'' \in S_j}^{*} \hat{q}_{i''}(t-1)\right\}$$

Then, we continue Eq. (16)

$$\begin{aligned}
C_i(\tau) \leq l + \sum_{t=1}^{\tau} \mathbb{I}\left\{\min_{l \leq n_{s(1)} \leq \cdots \leq n_{s(K)} \leq t} \sum_{x=1}^{K} \lambda_{s(x)}(t-1)\hat{q}_{s(x)}(t-1) \leq \right. \\
\left. \max_{1 \leq n_{s^*(1)} \leq \cdots \leq n_{s^*(K)} \leq t} \sum_{x=1}^{K} \lambda_{s^*(x)}(t-1)\hat{q}_{s^*(x)}(t-1)\right\}
\end{aligned}$$

where $s(x)$ denotes the xth element of $S(t)$, and $s^*(x)$ denotes the xth element of S^*.

Therefore

$$\begin{aligned}
C_i(\tau) \leq l + \sum_{t=1}^{\tau} \sum_{n_{s(1)}=l}^{t} \cdots \sum_{n_{s(K)}=l}^{t} \sum_{n_{s^*(1)}=1}^{t} \cdots \sum_{n_{s^*(K)}=1}^{t} \\
\mathbb{I}\left\{\sum_{x=1}^{K} \lambda_{s(x)}(t-1)\hat{q}_{s(x)}(t-1) \leq \right. \\
\left. \sum_{x=1}^{K} \lambda_{s^*(x)}(t-1)\hat{q}_{s^*(x)}(t-1)\right\}
\end{aligned} \tag{17}$$

Following event holds:

$$\sum_{x=1}^{K}\lambda_{s(x)}(t-1)\left[\bar{q}_{s(x)}(t-1)-Q_{s(x)}(t-1)\right]\le \sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)\left[\bar{q}_{s^*(x)}(t-1)-Q_{s^*(x)}(t-1)\right]$$

means that at least one of the following three events must holds:

$$\sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)\left[q_{s^*(x)}+Q_{s^*(x)}(t-1)\right] \le \sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)\bar{q}_{s^*(x)}(t-1) \tag{18}$$

$$\sum_{x=1}^{K}\lambda_{s(x)}(t-1)\bar{q}_{s(x)}(t-1)\le \sum_{x=1}^{K}\lambda_{s(x)}(t-1)\left[q_{s(x)}-Q_{s(x)}(t-1)\right] \tag{19}$$

$$\sum_{x=1}^{K}\lambda_{s(x)}(t-1)\left[q_{s(x)}-2Q_{s(x)}(t-1)\right]< \sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)q_{s^*(x)} \tag{20}$$

For event (18),

$$\mathbb{P}\left\{\sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)[q_{s^*(x)}+Q_{s^*(x)}(t-1)] \le \sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)\bar{q}_{s^*(x)}(t-1)\right\}$$

$$\le \sum_{x=1}^{K}\mathbb{P}\left\{\bar{q}_{s^*(x)}(t-1)\ge q_{s^*(x)}+Q_{s^*(x)}(t-1)\right\} \tag{21}$$

Apply Chernoff–Hoeffding bound to Eq. (21), we have

$$\mathbb{P}\left\{\bar{q}_{s^*(x)}(t-1)\ge q_{s^*(x)}+Q_{s^*(x)}(t-1)\right\}$$

$$\le e^{-2n_{s^*(x)}(t-1)\left(\frac{(K+1)\ln(t)}{n_{s^*(x)}(t-1)}\right)}=t^{-2(K+1)}$$

Then the upper bound of event (18) is

$$\mathbb{P}\left\{\sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)\bar{q}_{s^*(x)}(t-1) \ge \sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)[q_{s^*(x)}+Q_{s^*(x)}(t-1)]\right\} \le K\cdot t^{-2(K+1)} \tag{22}$$

For event (19), we have similar upper bound.

When $l\ge\frac{4K^2(K+1)\ln T}{\Delta_{\min}^2}$, event (20) is always false because

$$\sum_{x=1}^{K}\lambda_{s(x)}(t-1)q_{s(x)}-\sum_{x=1}^{K}\lambda_{s^*(x)}(t-1)q_{s^*(x)} -2\sum_{x=1}^{K}\lambda_{s(x)}(t-1)Q_{s(x)}(t-1)$$

$$\ge \Delta_t-2K\sqrt{\frac{(K+1)\ln t}{l}}$$

$$\ge \Delta_{\min}-2K\sqrt{\frac{(K+1)\ln T}{\frac{4K^2(K+1)\ln T}{\Delta_{\min}^2}}}=0$$

Now we continue inequality (17)

$$C_i(\tau)\le\left\lceil\frac{4K^2(K+1)\ln T}{\Delta_{\min}^2}\right\rceil+1+ \sum_{t=1}^{\tau}\sum_{n_{s(1)}=l}^{t}\cdots\sum_{n_{s(K)}=l}^{t}\sum_{n_{s^*(1)}=1}^{t}\cdots\sum_{n_{s^*(K)}=1}^{t}2K\cdot t^{-2(K+1)} \tag{23}$$

$$\le\frac{4K^2(K+1)\ln T}{\Delta_{\min}^2}+1+\sum_{t=1}^{\tau}2K\cdot t^{-2} \le \underbrace{\frac{4K^2(K+1)}{\Delta_{\min}^2}}_{=:\lambda_1}\ln T+\underbrace{1+\frac{K\pi^2}{3}}_{=:\lambda_2} \quad\square \tag{24}$$

Recall Theorem 3

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{i\in\mathcal{N}}\mathbb{E}\{\Theta_i(t)\}\le\frac{1}{\epsilon}(\mathcal{B}+VNc_{\max}D_{\max})$$

and Theorem 4

$$R(T)\le\frac{\mathcal{B}}{V}+Nc_{\max}D_{\max}+\frac{1}{T}N(\lambda_1\ln T+\lambda_2)\Delta_{\max}$$

Proof. To obtain the results of Theorem 3 and Theorem 4, we firstly define some useful notation:

Lyapunov function: $L[\vec{\Theta}(t)]\triangleq\frac{1}{2}\sum_{i\in\mathcal{N}}\Theta_i^2(t)$

Lyapunov drift: $\Delta[\vec{\Theta}(t)]=L[\vec{\Theta}(t+1)]-L[\vec{\Theta}(t)]$

Next we bound Lyapunov drift $\mathbb{E}\{\Delta[\vec{\Theta}(t)]\}$ as follows,

$$\mathbb{E}\{\Delta[\vec{\Theta}(t)]\}=\mathbb{E}\{\frac{1}{2}\sum_{i\in\mathcal{N}}\Theta_i^2(t+1)-\frac{1}{2}\sum_{i\in\mathcal{N}}\Theta_i^2(t)\}$$

$$\le\mathbb{E}\left\{\frac{1}{2}\sum_{i\in\mathcal{N}}[\Theta_i(t)+\sum_{j\in\mathcal{J}}\mathbb{I}_{i,j}(t)c_i(t)-\frac{B_i}{T_0}]^2-\frac{1}{2}\sum_{i\in\mathcal{N}}\Theta_i^2(t)\right\}$$

$$=\mathbb{E}\left\{\frac{1}{2}\sum_{i\in\mathcal{N}}\left[\sum_{j\in\mathcal{J}}\mathbb{I}_{i,j}(t)c_i(t)\right]^2+\frac{1}{2}\sum_{i\in\mathcal{N}}\left(\frac{B_i}{T_0}\right)^2 +\sum_{i\in\mathcal{N}}\Theta_i(t)\left[\sum_{j\in\mathcal{J}}\mathbb{I}_{i,j}(t)c_i(t)-\frac{B_i}{T_0}\right]\right\}$$

$$\le\underbrace{\frac{1}{2}N|\mathcal{J}|^2c_{\max}^2+\frac{1}{2}N\frac{B_{\max}^2}{T_0^2}}_{=:\mathcal{B}}+\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\left[\sum_{j\in\mathcal{J}}\mathbb{I}_{i,j}(t)c_i(t)-\frac{B_i}{T_0}\right]\right\}$$

$$=\mathcal{B}-\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\frac{B_i}{T_0}\right\}+\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\sum_{j\in\mathcal{J}}\mathbb{I}_{i,j}(t)c_i(t)\right\}$$

$$\le\mathcal{B}-\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\}\frac{B_{\min}}{T_0}+\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\}c_{\max}$$

$$\le\mathcal{B}-\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\}\frac{B_{\min}}{T_0}+\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\left[V\bar{D}_i(t)+\Theta_i(t)\right]\right\}c_{\max}$$

$$\le\mathcal{B}-\underbrace{\left(\frac{B_{\min}}{T_0}-c_{\max}\right)}_{=:\epsilon}\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\}+VNc_{\max}D_{\max} \tag{25}$$

We bring in Eq. (5) and obtained the first inequality. Aiming to the second and third inequality,we according to the implication of $\mathbb{I}_{i,j}(t)$ and $c_i(t)$ to scaling down to obtain threshold range for energy budgeting. In the fourth inequality, we introduce $\bar{D}_i(t)$ for the subsequent optimization of the objective function connection. Then we focus on

drift-plus-penalty function, which is shown as follows,

$$\mathbb{E}\{\Delta[\overrightarrow{\Theta}(t)]\} + V\mathbb{E}\left\{\sum_{j\in\mathcal{J}}\max_{i\in S_j(t)} q_i(t) - \sum_{j\in\mathcal{J}}\max_{i\in S_j^*} q_i(t)\right\} \tag{26}$$

Substituting inequality (25) into inequality (26), then according to use the sum transformation, the operation of summing the order of model $j \in \mathcal{J}$ is transformed into the operation of summing user $i \in S(t)$ and $i \in S^*$.

$$\begin{aligned}
&\mathbb{E}\{\Delta[\overrightarrow{\Theta}(t)]\} + V\mathbb{E}\left\{\sum_{j\in\mathcal{J}}\max_{i\in S_j(t)} q_i(t) - \sum_{j\in\mathcal{J}}\max_{i\in S_j^*} q_i(t)\right\} \\
&\le \mathcal{B} + VNc_{\max}D_{\max} - \epsilon\cdot\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\} \\
&+ V\mathbb{E}\left\{\sum_{i\in S(t)} r_i(t) - \sum_{i\in S^*} r_i^*\right\}
\end{aligned} \tag{27}$$

where

$$r_i(t) \triangleq \sum_{j\in\mathcal{J}}\mathbb{I}\left\{i = \arg\max_{i'\in S_j(t)} q_{i'}(t)\right\} q_i(t)$$

$$r_i^* \triangleq \sum_{j\in\mathcal{J}}\mathbb{I}\left\{i = \arg\max^{*}_{i'\in S_j} q_{i'}(t)\right\} q_i(t)$$

Summing both sides of inequality (27) over $t = 1, 2, 3, \ldots, T$ yields

$$\begin{aligned}
&L[\overrightarrow{\Theta}(T+1)] - L[\overrightarrow{\Theta}(1)] + V\sum_{t=1}^{T}\mathbb{E}\left\{\sum_{j\in\mathcal{J}}\max_{i\in S_j(t)} q_i(t) - \sum_{j\in\mathcal{J}}\max_{i\in S_j^*} q_i(t)\right\} \\
&\le T(\mathcal{B} + VNc_{\max}D_{\max}) - \epsilon\sum_{t=1}^{T}\mathbb{E}\left\{\sum_{i\in\mathcal{N}}\Theta_i(t)\right\} \\
&\quad + V\sum_{t=1}^{T}\mathbb{E}\left\{\sum_{i\in S(t)} r_i(t) - \sum_{i\in S^*} r_i^*\right\}
\end{aligned}$$

Rearranging inequality 1 yields

$$\epsilon\cdot\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{\sum_{i\in\mathcal{N}}\Theta_i(t)\} \le \mathcal{B} + VNc_{\max}D_{\max} \tag{28}$$

taking lim-sup over $T \to \infty$ yields Theorem 3.

Rearranging inequality 1 also yields

$$\begin{aligned}
R(T) &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{\sum_{j\in\mathcal{J}}\max_{i\in S_j(t)} q_i(t) - \sum_{j\in\mathcal{J}}\max_{i\in S_j^*} q_i(t)\right\} \\
&\le \frac{\mathcal{B}}{V} + Nc_{\max}D_{\max} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{\sum_{i\in S(t)} r_i(t) - \sum_{i\in S}^{*} r_i^*\right\} \\
&= \frac{\mathcal{B}}{V} + Nc_{\max}D_{\max} + \frac{1}{T}\sum_{S(t)\neq S^*}\mathbb{E}\left\{\sum_{i\in S(t)} r_i(t) - \sum_{i\in S^*} r_i^*\right\} \\
&\le \frac{\mathcal{B}}{V} + Nc_{\max}D_{\max} + \frac{1}{T}\sum_{i\in\mathcal{N}} C_i(T)\Delta_{\max} \\
&\le \frac{\mathcal{B}}{V} + Nc_{\max}D_{\max} + \frac{1}{T}N(\lambda_1\ln T + \lambda_2)\Delta_{\max}
\end{aligned} \tag{29}$$

This completes the proof of Theorem 4. □

Proof of Theorem 5. According to Assumption 3 and taking $\gamma \le \frac{1}{L}$, we have

$$\begin{aligned}
F(\mathbf{x}(t+1)) - F(\mathbf{x}(t)) &\le -\gamma\nabla F(\mathbf{x}(t))\mathbf{v}(t) + \frac{\gamma}{2}\|\mathbf{v}(t)\|_2^2 \\
&= -\frac{\gamma}{2}\|\nabla F(\mathbf{x}(t))\|_2^2 + \frac{\gamma}{2}\|\nabla F(\mathbf{x}(t)) - \mathbf{v}(t)\|_2^2.
\end{aligned} \tag{30}$$

The initial inequality involves substituting $\mathbf{x}(t+1) = v$ and $\mathbf{x}(t) = w$ into Assumption 3, followed by incorporating Eq. (17). In the first equation, we utilize the inverse operation of the square difference formula to expand the aforementioned inequality. Subsequently, we take the expectation over $v(t)$, leading to the following expression:

$$\begin{aligned}
\mathbb{E}[F(\mathbf{x}(t+1)) - F(\mathbf{x}(t))] &\le -\frac{\gamma}{2}\|\nabla F(\mathbf{x}(t))\|_2^2 \\
&+ \frac{\gamma}{2}\mathbb{E}\left\{\|\nabla F(\mathbf{x}(t)) - \sum_{k\subseteq S(t)}|\nabla F_k(\mathbf{x}(t))\|_2^2\right\} \\
&\le -\frac{\gamma}{2}\|\nabla F[\mathbf{x}(t)]\|_2^2 + \frac{\gamma\delta_0^2}{2},
\end{aligned} \tag{31}$$

where the last inequality holds because of Assumption. 4. Now we define a new function

$$\mathcal{F}(\hat{\mathbf{x}}) = F(\mathbf{x}(t)) + \nabla F(\mathbf{x}(t))(\hat{\mathbf{x}} - \mathbf{x}(t)) + \frac{\mu}{2}\|\hat{\mathbf{x}} - \mathbf{x}(t)\|_2^2 \tag{32}$$

whose minimal value is achieved when all the partial derivatives are 0's.

$$\frac{\partial\mathcal{F}(\hat{\mathbf{x}})}{\partial\hat{\mathbf{x}}} = \nabla F(\mathbf{x}(t)) + \mu[\hat{\mathbf{x}}(t) - \mathbf{x}(t)] = 0 \tag{33}$$

The optimal point is $\hat{\mathbf{x}}^* = \mathbf{x}(t) - \frac{\nabla G(\mathbf{x}(t))}{\mu}$ and the minimal value is

$$\mathcal{F}_{\min} = F(\mathbf{x}(t)) - \frac{\|\nabla F(\mathbf{x}(t))\|^2}{2\mu} \tag{34}$$

We can derive the following inequality based on Eqs. (32), (34) and Assumption 3:

$$F(\mathbf{x}^*) \ge \mathcal{F}(\mathbf{x}^*) \ge \mathcal{F}_{\min}, \tag{35}$$

which implies that:

$$\begin{aligned}
F(\mathbf{x}^*) &\ge \mathcal{F}_{\min} \\
2\mu[F(\mathbf{x}(t)) - F(\mathbf{x}^*)] &\le \|\nabla F(\mathbf{x}(t))\|_2^2
\end{aligned} \tag{36}$$

Then we combine Eqs. (31) and (36), eliminating $\|\nabla F(\mathbf{x}(t))\|_2^2$:

$$\mathbb{E}[F(\mathbf{x}(t+1)) - F(\mathbf{x}(t))] \le -\gamma\mu[F(\mathbf{x}(t)) - F(\mathbf{x}^*)] + \frac{\gamma\delta_0^2}{2}. \tag{37}$$

We define $\Delta_{t+1} = F(\mathbf{x}(t+1)) - F(\mathbf{x}(t))$ rearrange Eq. (37) as:

$$\mathbb{E}[\Delta_{t+1}] \le (1-\gamma\mu)\Delta_t + \frac{\gamma\delta_0^2}{2}. \tag{38}$$

We just need to prove that Δ_{t+1} decreases as t increases, then we take $\mathbb{E}(\cdot)$ on both sides of the equation, and iterate t times on Δ_t to obtain:

$$\begin{aligned}
\mathbb{E}[\Delta_{t+1} - \frac{\delta_0^2}{2\mu}] &\le \mathbb{E}[(1-\gamma\mu)(\Delta_t - \frac{\delta_0^2}{2\mu})] \\
&\le \mathbb{E}[(1-\gamma\mu)^t(\Delta_1 - \frac{\delta_0^2}{2\mu})]
\end{aligned} \tag{39}$$

Because of $0 < \gamma \le \frac{1}{\mu}$, the $1-\gamma\mu < 1$, following that $(1-\gamma\mu)^t$ decreases as t increases, we take T = t+1 and we can obtain

$$\mathbb{E}[F(\mathbf{x}(T)) - F(\mathbf{x}^*)] \le \frac{\delta_0^2}{2\mu} + (1-\gamma\mu)^{T-1}\mathbb{E}\left[F(\mathbf{x}(1)) - F(\mathbf{x}^*) - \frac{\delta_0^2}{2\mu}\right] \tag{40}$$

It is observed from Eq. (40) that as t increases, the gap between $F(x(T))$ and $F(x^*)$ becomes smaller, suggesting a better convergence performance. □

References

[1] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, 2019, arXiv preprint arXiv:1912.04977.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[3] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, J. Liu, Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.

[4] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[5] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: Motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (6) (2020) 46–51.

[6] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.
[7] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–7.
[8] S. AbdulRahman, H. Tout, A. Mourad, C. Talhi, FedMCCS: Multicriteria client selection model for optimal IoT federated learning, IEEE Internet Things J. 8 (6) (2020) 4723–4735.
[9] T. Huang, W. Lin, W. Wu, L. He, K. Li, A. Zomaya, An efficiency-boosting client selection scheme for federated learning with fairness guarantee, IEEE Trans. Parallel Distrib. Syst. (2020).
[10] W. Xia, T.Q. Quek, K. Guo, W. Wen, H.H. Yang, H. Zhu, Multi-armed bandit-based client scheduling for federated learning, IEEE Trans. Wireless Commun. 19 (11) (2020) 7108–7123.
[11] M.M. Wadu, S. Samarakoon, M. Bennis, Federated learning under channel uncertainty: Joint client scheduling and resource allocation, in: 2020 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2020, pp. 1–6.
[12] J. Xu, H. Wang, Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective, IEEE Trans. Wireless Commun. (2020).
[13] Y. Jin, L. Jiao, M. Ji, Z. Qian, S. Zhang, N. Chen, S. Lu, Scheduling in-band network telemetry with convergence-preserving federated learning, IEEE/ACM Trans. Net. 31 (5) (2023) 2313–2328.
[14] V. Anantharam, P. Varaiya, J. Walrand, Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: IID rewards, IEEE Trans. Autom. Control 32 (11) (1987) 968–976.
[15] L. Tran-Thanh, A. Chapman, A. Rogers, N. Jennings, Knapsack based optimal policies for budget–limited multi–armed bandits, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 26, No. 1, 2012.
[16] B. Yu, M. Fang, D. Tao, Linear submodular bandits with a knapsack constraint, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[17] D. Zhou, C. Tomlin, Budget-constrained multi-armed bandits with multiple plays, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1, 2018.
[18] G. Gao, J. Wu, M. Xiao, G. Chen, Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 179–188.
[19] A. Badanidiyuru, R. Kleinberg, A. Slivkins, Bandits with knapsacks, in: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE, 2013, pp. 207–216.
[20] G. Gao, J. Wu, Z. Yan, M. Xiao, G. Chen, Unknown worker recruitment with budget and covering constraints for mobile crowdsensing, in: 2019 IEEE 25th International Conference on Parallel and Distributed Systems, ICPADS, IEEE, 2019, pp. 539–547.
[21] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2) (2002) 235–256.
[22] G. Caldarelli, Understanding the blockchain oracle problem: A call for action, Information 11 (11) (2020).
[23] J. Li, Y. Shao, K. Wei, M. Ding, C. Ma, L. Shi, Z. Han, H.V. Poor, Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation, IEEE Trans. Parallel Distrib. Syst. 33 (10) (2021) 2401–2415.
[24] X. Deng, J. Li, C. Ma, K. Wei, L. Shi, M. Ding, W. Chen, Low-latency federated learning with DNN partition in distributed industrial IoT networks, IEEE J. Sel. Areas Commun. 41 (3) (2022) 755–775.
[25] Y.J. Cho, J. Wang, G. Joshi, Towards understanding biased client selection in federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 10351–10375.
[26] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, Y. Liu, Context-aware online client selection for hierarchical federated learning, IEEE Trans. Parallel Distrib. Syst. 33 (12) (2022) 4353–4367.
[27] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, IEEE Trans. Wireless Commun. (2020).
[28] S. Yu, X. Chen, Z. Zhou, X. Gong, D. Wu, When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network, IEEE Internet Things J. 8 (4) (2020) 2238–2251.
[29] Y. Jiao, P. Wang, D. Niyato, B. Lin, D.I. Kim, Toward an automated auction framework for wireless federated learning services market, IEEE Trans. Mob. Comput. (2020).
[30] M. Chen, Z. Yang, W. Saad, C. Yin, H.V. Poor, S. Cui, A joint learning and communications framework for federated learning over wireless networks, IEEE Trans. Wireless Commun. 20 (1) (2020) 269–283.
[31] R. Agrawal, Sample mean based index policies with O (log n) regret for the multi-armed bandit problem, Adv. in Appl. Probab. (1995) 1054–1078.
[32] F. Li, J. Liu, B. Ji, Combinatorial sleeping bandits with fairness constraints, IEEE Trans. Netw. Sci. Eng. 7 (3) (2019) 1799–1813.
[33] M.J. Neely, Stochastic network optimization with application to communication and queueing systems, Synth. Lect. Commun. Netw. 3 (1) (2010) 1–211.
[34] Y. Sun, S. Zhou, J. Xu, EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2637–2646.
[35] X. Huang, Y. Tang, Z. Shao, Y. Yang, H. Xu, Joint switch-controller association and control devolution for sdn systems: An integration of online control and online learning, in: 2020 IEEE/ACM 28th International Symposium on Quality of Service, IWQoS, IEEE, 2020, pp. 1–10.
[36] J.E. Hopcroft, R.M. Karp, An n^5/2 algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2 (4) (1973) 225–231.
[37] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, N.D. Lane, Flower: A friendly federated learning research framework, 2020, arXiv preprint arXiv:2007.14390.
[38] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, EMNIST: Extending MNIST to handwritten letters, in: 2017 International Joint Conference on Neural Networks, IJCNN, IEEE, 2017, pp. 2921–2926.

Konglin Zhu received the master's degree in computer Science from the University of California, Los Angeles, CA, USA, and the Ph.D. degree from the University of Göttingen, Germany, in 2009 and 2014, respectively. He is now an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include Internet of Vehicles, Edge Computing and Distributed Learning.

Fuchun Zhang received the BEng degree in Communication Engineering from Beijing University of Posts and Telecommunications, China, in 2019. He is now a master student in School of Artificial Intelligence in Beijing University of Posts and Telecommunications. His research interests are in the areas of Online Optimization and Federated Learning.

Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently with the Department of Computer Science, University of Oregon, USA. Previously he worked at Nokia Bell Labs in Ireland. He is interested in optimization, control, learning, and game theory applied to computer and telecommunication systems, networks, and services. He has published about 70 peer-reviewed research papers, including many in leading journals and conferences. He is a recipient of the NSF CAREER Award. He also received several Best Paper Awards. He served as a guest editor for JSAC. He was on the program committees of INFOCOM, MOBIHOC, ICDCS, and IWQoS, and was also the program chair of multiple workshops with INFOCOM and ICDCS.

Bowei Xue received his bachelor's degree in information engineering from Beijing University of Posts and Telecommunications in 2021. He is now a doctoral student in the School of Artificial Intelligence of Beijing University of Posts and Telecommunications. His research interests are federated learning optimization algorithm and network structure optimization.

Lin Zhang received the B.S. and the Ph.D. degrees in 1996 and 2001, both from the Beijing University of Posts and Telecommunications, Beijing, China. From 2000 to 2004, he was a Postdoctoral Researcher with Information and Communications University, Daejeon, South Korea, and Nanyang Technological University, Singapore, respectively. He joined Beijing University of Posts and Telecommunications in 2004, where he has been a Professor since 2011. He is also the director of Beijing Big Data Center. His current research interests include mobile cloud computing and Internet of Things.