Efficient Federated Learning in Wireless Communication Systems: A Multi-Objective Optimization Perspective

Zhou, Yu; Zhao, Xiaohui; Liu, Xuemei; You, Lei; Lei, Lei

Published in:
Proceedings of 2023 IEEE/CIC International Conference on Communications in China (ICCC)

Link to article, DOI:
10.1109/ICCC57788.2023.10233536

Publication date:
2023

Document Version
Peer reviewed version

Citation (APA):
Zhou, Y., Zhao, X., Liu, X., You, L., & Lei, L. (2023). Efficient Federated Learning in Wireless Communication Systems: A Multi-Objective Optimization Perspective. In *Proceedings of 2023 IEEE/CIC International Conference on Communications in China (ICCC)* IEEE. https://doi.org/10.1109/ICCC57788.2023.10233536

Efficient Federated Learning in Wireless Communication Systems: A Multi-Objective Optimization Perspective

Yu Zhou[1], Xiaohui Zhao[1], Xuemei Liu[2], Lei You[3], and Lei Lei[1,4]

[1]School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, P. R. China
[2]Institute of Rail Transit, Tongji University, Shanghai, China
[3]Department of Engineering Technology, Technical University of Denmark, Lyngby, Denmark
[4]National Mobile Communications Research Laboratory, Southeast University, Nanjing, P. R. China

***Abstract*—This paper focuses on a federated learning (FL) system that employs a base station as a central server while clients with limited computation capabilities perform local training. The limited bandwidth leads to that only a portion of clients can participate in each FL training round, and picking different clients can impact the performance of FL systems, requiring effective allocation of their computing resources. In FL systems, both model convergence and energy consumption are important performances. To this end, we formulate a multi-objective optimization problem (MOP) to simultaneously speed up model convergence and reduce energy consumption. To address the MOP, we propose a multi-objective algorithm (MOA) for FL systems to obtain a Pareto optimal solution set, where Tchebycheff approach is adopted to divide MOP into multiple single-objective problems and optimize them by differential evolution. The extensive experiments on Fashion-MNIST dataset in both i.i.d and non-i.i.d data settings illustrates that MOA outperforms other algorithms.**

***Index Terms*—Federated learning, multi-objective algorithm, differential evolution, decomposition.**

I. Introduction

Centralized machine learning (ML) has been widely applied to a number of fields in 5G wireless communication [1] [2]. ML centrally trains and learns a global model through collecting data, leading to the leakage of privacy data. To deal with the problem, a novel distributed learning called federated learning (FL) is proposed, which aims to train a global model in a distributed manner. In FL systems, clients need to be picked in each round and different selection strategies could affect the performance of the system because clients have different computing resources, storage capacity and. To this end, many works concentrates on how to establish an efficient client selection strategy. Huang *et al.* [3] studied a random client selection method to address the tradeoff between model convergence and learning stability. Xu *et al.* [4] designed a FL system and jointly optimized client selection and bandwidth allocation. The allocation of computing resources, such as CPU frequency, has a significant impact on the performance of FL systems and has been extensively studied. Zhang *et al.* [5] proposed a novel scheme for the joint allocation of datasets and computing resources in FL systems and formulated an energy consumption minimization problem with the constraint of completing training within a certain time frame. Huang *et al.* [6] introduced a problem of minimizing model convergence time while adhering to an energy consumption constraint, which involved joint optimization of CPU frequency and phase shifts.

Most of the previous studies focus on single-objective optimization problems. The objective function separately takes energy consumption or model convergence into account. In fact, both energy consumption and model convergence are important performances of FL systems. To this end, we design a multi-objective optimization problem to simultaneously minimize energy consumption and model convergence. A multi-objective algorithm (MOA) is proposed to address this multi-objective optimization problem (MOP). We remark that this is an early application of multi-objective optimization in the field of federated learning. The main contributions of this work are summarized as follows:

- By considering client selection and CPU frequency, a MOP for FL systems is developed to reduce model convergence and energy consumption concurrently.
- A multi-objective optimization approach called MOA is suggested to solve this MOP. MOA divides the MOP into N single objective problems, which are then simultaneously optimized using differential evolution (DE) method.
- The experimental results show that, for both i.i.d and non-i.i.d data settings, MOA outperforms other algorithms on Fashion-MNIST dataset.

II. System Model and Problem Formulation

Figure 1 illustrates the studied FL system, which comprises a single-antenna base station (BS) and a group of K single-antenna clients with energy constraints. Each client k has its own local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ labeled data samples $(\boldsymbol{x}_n, y_n)_{n=1}^{D_k} \in \mathcal{D}_k$, where $(\boldsymbol{x}_n, y_n)$ denotes the input-output data sample including a feature $\boldsymbol{x}_n$ and its corresponding label y_n. Because of bandwidth limitations, only a subset of clients denoted by $\mathcal{K}'$ are chosen to participate in the training process. The overall workflow is given as follows, which contains T training rounds. The BS initializes the parameters of the global model denoted as $\boldsymbol{w}_1$ and picks up clients to participate in the t-th ($t \in \mathcal{T} = \{1, 2, ..., T\}$) round. Then, it sends $\boldsymbol{w}_t$ to the chosen clients. When receiving $\boldsymbol{w}_t$, the chosen client k ($k \in \mathcal{K}'$) trains the parameters of its local model $\boldsymbol{w}_{k,t} = \boldsymbol{w}_t$ via $\mathcal{D}_k$. The goal of training is to minimize the local loss function $F_k(\boldsymbol{w}_{k,t})$ by $\boldsymbol{w}^*_{k,t} = \boldsymbol{w}_{k,t} - \delta \bigtriangledown F_k(\boldsymbol{w}_{k,t})$, where $\delta = 0.1$. Here, $F_k(\boldsymbol{w}) := \frac{1}{D_k}\sum_{n \in \mathcal{D}_k} f_n(\boldsymbol{x}_n, y_n; \boldsymbol{w})$, where $f_n(\boldsymbol{x}_n, y_n; \boldsymbol{w})$ denote the loss function on $(\boldsymbol{x}_n, y_n)$. Then, the chosen client k sends $\boldsymbol{w}^*_{k,t}$ to the BS. The BS performs the aggregation and then updates $\boldsymbol{w}_t$ by $\boldsymbol{w}_{t+1} = \sum_{k \in \mathcal{K}'} \frac{D_K}{D} \boldsymbol{w}^*_{k,t}$, where D is the number of total data samples. Then, the computing model, communication model and model convergence are presented, respectively.

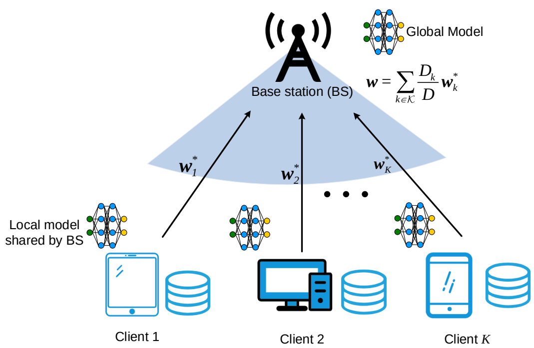


Figure 1. An illustration of a wireless FL system.

A. Energy Consumption Models

Computation Model: The CPU cycles required to execute one data simple $(\boldsymbol{x}_n, y_n)$ in the studied system are denoted by a constant c_k [7]. Therefore, the total number of CPU cycles required for client k to perform one local training iteration is $c_k D_k$. The computing time in the t-th round can then be expressed as:

$$TIME^{cmp}_{k,t} = \frac{c_k D_k}{f_{k,t}}, \tag{1}$$

where $f_{k,t}$ is the CPU frequency of client k in the t-th round. The computing energy consumption in the t-th round can be expressed as

$$E^{cmp}_{k,t} = \frac{\alpha_k}{2} c_k D_k f^2_{k,t}, \tag{2}$$

where $\frac{\alpha_k}{2}$ is the capacitance coefficient of client k.

Communication Model: The uplink communication is taken into account, in which the BS assigns the chosen clients O subchannels where each subchannel has the same bandwidth $b = \frac{B}{O}$. Here, B is the total bandwidth. Additionally, chosen client k is also given the channel gain $h_{k,t}$ in the t-th round. Consequently, the transmission data rate of client k can be expressed as

$$r_k = b \log_2 \left(1 + \frac{P_k h^2_{k,t}}{\sigma^2}\right), \tag{3}$$

where σ^2 denotes the Gaussian channel noise and P_k represents the transmit power of client k. The transmission data rate r_k is set to the threshold transmission data rate R_k in order to get the minimum transmit power P_k. Therefore, the communication time of client k in the t-th round can be represented as

$$TIME^{com}_{k,t} = \frac{H}{b \log_2 \left(1 + \frac{P_k h^2_{k,t}}{\sigma^2}\right)}. \tag{4}$$

where H is a constant to measure the size of $\boldsymbol{w}_t$.

The communication energy consumption for client k in the t-th round can be given by

$$E^{com}_{k,t} = \frac{J P_k}{b \log_2 \left(1 + \frac{P_k h^2_{k,t}}{\sigma^2}\right)}, \tag{5}$$

To simplify the calculation, we convert the loss function into the model convergence value via the staleness method [7]

$$S_t = -\frac{\left(\sum_{k=1}^{K} s_{k,t} D_k F_k(\boldsymbol{w}_{k,t}) C_{k,t}\right)^{\mu}}{\mu}, \tag{6}$$

where τ is a constant.$C_{k,t+1}$ is the staleness value as follow:

$$C_{k,t+1} = (C_{k,t} + 1)(1 - s_{k,t}). \tag{7}$$

B. Problem Formulation

The total energy consumption in the t-th round can be calculated by

$$E_t = \sum_{k=1}^{K} s_{k,t} \left(\phi E^{cmp}_{k,t} + E^{com}_{k,t}\right), \tag{8}$$

where ϕ represents the number of local training iterations and $s_{k,t}$ denotes the client selection variable of client k in the t-th round. If client k is chosen, $s_{k,t} = 1$; otherwise, $s_{k,t} = 0$.

In this paper, we optimize the FL model convergence and energy consumption simultaneously by considering client selection variables and CPU frequencies. MOP is defined as

$$\min_{f_{k,t}, s_{k,t}} \mathcal{F} = (S_t, E_t), \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{9a}$$

$$\text{s.t.} \quad \mathcal{C}1: f_{min} \leq f_{k,t} \leq f_{max}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \tag{9b}$$

$$\mathcal{C}2: s_{k,t} \in \{0, 1\}, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \tag{9c}$$

$$\mathcal{C}3: \sum_{k=1}^{K} s_{k,t} \leq O, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \tag{9d}$$

where $\mathcal{C}1$ defines the lower and upper bounds of the CPU frequency for each client k; $\mathcal{C}2$ indicates whether each client k ($k \in \mathcal{K}$) is chosen to take part in the training process; $\mathcal{C}3$ makes sure that the number of chosen clients does not go above the maximum number of subchannels in the t-th round because of the bandwidth limitation.

III. Proposed Method

A. MOA

To address (9), a multi-objective evolutionary algorithm for FL systems called MOA is proposed. MOA aims to utilize a decomposition approach to divide the MOP into several single objective problems and introduces DE algorithm, including the mutation, crossover and selection operators, to search for a Pareto optimal solution set.

The general framework of MOA is presented in **Algorithm 1**. In detail, $\boldsymbol{w}$ are first initialized (line 1). Then, MOA decomposes (9) into single objective problems based on a uniform spread of weight vectors denoted as $\{\lambda^1, \ldots, \lambda^N\}$ via the simplex method (line 2). Then, the weight vectors are input to perform the optimization process in each round. Specifically, we first utilize the candidate Pareto optimal solution set CP to collect the non-dominated solutions (line 4), where CP is set to $\varnothing$. After that, for each weigh vector, the neighbors $B(i)$ (denoted as $\{i_1, \ldots, i_J\}$) is initialized as well as its J closest weigh vectors (denoted as $\{\lambda^{i_1}, \ldots, \lambda^{i_J}\}$) for λ^i are obtained via the Euclidean distances between any two weight vectors.

Next, a population is initialized randomly $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ (line 5), where each individual corresponds to a solution for a weight vector. Each individual is represented as the following encoding scheme:

$$\boldsymbol{z}_i = \{f_{i,1}, ..., f_{i,k}, ..., f_{i,K}, s_{i,1}, ..., s_{i,k}, ..., s_{i,K}\}, \forall i \in \mathcal{N}, \tag{10}$$

where $f_{i,k} = f_{min} + rand * (f_{max} - f_{min})$ is the CPU frequency and $s_{i,k} = \llcorner rand \lrcorner$ represent the client selection variable of client k for λ^i, respectively; $rand$ denotes a random number within $[0, 1]$; and N is the population size. Then, we initialize the reference point $RF^* = (RF^*_m | m = 1, 2)$ and the F-value FV, respectively (line 6). Then, the population are evolved via DE algorithm.

When we update the population, it makes sense to choose individuals from outside the neighbors as the mating parent population to expand the variety of the population. The selection range of the mating parents needs to be defined in each generation. Specifically, we generate a random number β within [0,1], which is used to choose the mating parent population (line 9)

$$P = \begin{cases} B(i), & \text{if } \beta \leq \eta, \\ \{1, 2, ..., N\}, & \text{otherwise}, \end{cases} \tag{11}$$

where η represents a pre-set probability to choose $B(i)$. If $\beta \leq \eta$ for λ^i, we employ $B(i)$ to update population. Otherwise, the entire population is selected as the range of the mating parents. Then, we perform the mutation and crossover operations to obtain a new solution according to P.

The mutation operator is to gain a mutant vector $\boldsymbol{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,2*K})$ based on $\boldsymbol{z}_i$ for λ^i. In this paper, a famous mutation operator called DE/current-to-rand/1 is as follows:

$$\boldsymbol{u}_i = \boldsymbol{z}_i + F \cdot (\boldsymbol{z}_{r1} - \boldsymbol{z}_i) + F \cdot (\boldsymbol{z}_{r2} - \boldsymbol{z}_{r3}), \tag{12}$$

where $r1$, $r2$ and $r3$ are three random integers chosen from P and different from i, and F represents the mutation control parameter.

Then, the crossover operator is performed to increase the diversity of the population. In detail, the crossover vector $\boldsymbol{v}_i$ is obtained according to $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$. Here, the binomial crossover technique is utilized, which is a widely used method and can be expressed as:

$$v_{i,n} = \begin{cases} u_{i,n}, & \text{if } rand_n \leq CR \text{ or } n = n_{rand}, \\ z_{i,n}, & \text{otherwise}, \end{cases} \tag{13}$$

where n_{rand} denotes a integer randomly chosen within $[1, 2 * K]$; $rand_n$ represents a random number within $[0, 1]$ for each n; and CR is the crossover control parameter. Then, $\boldsymbol{v}_i$ is to generate a solution as $\boldsymbol{y}$. We fix the invalid elements of $\boldsymbol{y}$ and generate $\boldsymbol{y}'$.

Then, we evaluate the fitness value of two objective functions for $\boldsymbol{z}_i$ in the t-th round, which can be expressed as (line 11)

$$\mathcal{F}(\boldsymbol{z}_i) = (\mathcal{F}_1(\boldsymbol{z}_i), \mathcal{F}_2(\boldsymbol{z}_i)), \forall i \in \mathcal{N}. \tag{14}$$

Here,

$$\mathcal{F}_1(\boldsymbol{z}_i) = -\frac{\left(\sum_{k=1}^{K} s_{k,t} D_k F_k(\boldsymbol{w}_{k,t}) C_{k,t}\right)^{\mu}}{\mu}, \tag{15}$$

and

$$\mathcal{F}_2(\boldsymbol{z}_i) = \sum_{k=1}^{K} s^i_{k,t}(E^{cmp,i}_{k,t} + E^{com,i}_{k,t}), \forall i \in \mathcal{N}, \tag{16}$$

where $s^i_{k,t}$ represents client selection variable of client k for λ^i in the t-th round. Therefore, the fitness function for λ^i in the t-th round can be formulated as

After that, the MOP is divided into N single objective problems by the Tchebycheff method [8], where the i-th single objective problem can be given as

$$\min g^{te}(\boldsymbol{z}|\lambda^i, RF^*) = \max_{1 \leq m \leq 2} \lambda^i_m |\mathcal{F}_m(\boldsymbol{z}) - RF^*_m|, \forall i \in \mathcal{N}, \tag{17}$$

Here, RF^*_m for the m-th objective function is initialized by $RF^*_m = \min_{1 \leq m \leq 2} F_m(\boldsymbol{z})$.

Then, RF^* is updated as follows (line 12):

$$RF^*_m = \begin{cases} \mathcal{F}_m(\boldsymbol{y}'), & RF^*_m \leq \mathcal{F}_m(\boldsymbol{y}'); \\ RF^*_m, & \text{otherwise.} \end{cases} \tag{18}$$

Next, we update the solutions in $B(i)$ and FV are updated in each j ($j \in B(i) = \{i_1, i_2, \ldots i_J\}$) as follows (line 13):

$$\boldsymbol{z}^j = \begin{cases} \boldsymbol{y}', & g^{te}(\boldsymbol{y}'|\lambda^{i_j}, RF^*) \leq g^{te}(\boldsymbol{z}^j|\lambda^{i_j}, RF^*), \\ \boldsymbol{z}^j, & \text{otherwise}, \end{cases} \tag{19}$$

and

$$FV^j = \begin{cases} F(\boldsymbol{y}'), & g^{te}(\boldsymbol{y}'|\lambda^{i_j}, RF^*) \leq g^{te}(\boldsymbol{z}^j|\lambda^{i_j}, RF^*), \\ FV^j, & \text{otherwise.} \end{cases} \tag{20}$$

Finally, we update CP based on the domination relationship. Specifically, if $\boldsymbol{y}'$ is dominated by all the solutions in CP, $\boldsymbol{y}'$ is removed from CP. $\boldsymbol{y}'$ is added to CP (line 14).

The local model training and transmission are executed in line 5. Next, for λ^i, the optimal client selection variables and the CPU frequencies are obtained (line 18) and the corresponding S_t^i and E_t^i are calculated (line 19). To this end, the total model convergence and energy consumption are updated, respectively (line 20). Finally, we perform the global aggregation, and update $\boldsymbol{w}_t$ (line 22). When overall FL training process termination, a Pareto optimal solution set, the corresponding minimal model convergence, and energy consumption are output.

Algorithm 1 General framework of MOA

Input: Initialize $S^i = 0$ and $E^i = 0$ for $\lambda^i (i \in \mathcal{N})$;
1: Initialize the global model parameters $\boldsymbol{w}$;
2: Initialize a uniform spread of weight vectors to divide (9) into N single objective problems;
3: **for** training round $t = 1 : T$ **do**
4: Initialize CP, $B(i)$ and its H closest weight vectors;
5: Initialize $\mathcal{Z}^1 = \{\boldsymbol{z}_1^1, \boldsymbol{z}_2^1, ..., \boldsymbol{z}_N^1\}$;
6: Initialize RF^* and FV^i for the i-th objective;
7: **for** generation $g = 1 : g_{\max}$ **do**
8: **for** each weight vector λ^i $(i \in \mathcal{N})$ **do**
9: Generate β to set P based on (11);
10: Execute the mutation and crossover operation to get $\boldsymbol{y}$ and obtain $\boldsymbol{y}^{'}$;
11: Execute the fitness evolution to obtain $F(\boldsymbol{y}^{'})$ and $g^{te}(\boldsymbol{y}^{'}|\lambda^i, RF^*)$ for the i-th problem;
12: Update RF^* according to (18);
13: Update the solutions in $B(i)$ and FV according to (19) and (20);
14: Update CP;
15: **end for**
16: **end for**
17: **for** each weight vector λ^i $(i \in \mathcal{N})$ **do**
18: Obtain $o_{k,t}^{i*}$ and $f_{k,t}^{i*}$ for λ^i;
19: Get the corresponding S_t^i and E_t^i for $o_{k,t}^{i*}$ and $f_{k,t}^{i*}$;
20: Calculate S^i and E^i via $S^i = S^i + S_t^i$, $E^i = E^i + E_t^i$;
21: **end for**
22: Execute the global aggregation and update $\boldsymbol{w}_t$;
23: **end for**
Output: A Pareto optimal solution set, and the minimal S^i $(\forall i \in \mathcal{N})$ and E^i $(\forall i \in \mathcal{N})$.

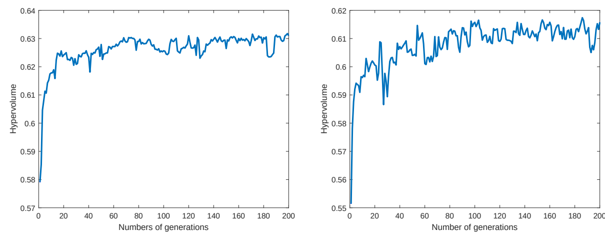
IV. Experimental Study

A. Experimental Settings

The system consists of multiple clients randomly distributed within a rectangular area with vertices located at $[70, 0, 0]$, $[90, 10, 0]$, $[90, 50, 0]$, and $[70, 50, 0]$ m. The BS is located at $[0, 30, 1]$ m. The path loss is determined using the formula $PL = PL_0 - 10\beta \log(\frac{d}{d_0})$ where $PL_0 = 30$ dB represents the path loss at a reference distance of $d_0 = 1$ m, and d is the distance between the transmitter to the receiver, with the value of β being determined by the system. The total size of the model parameters and dataset is set to 86.6 KB and 47.04 MB, respectively, while the size of each client's local dataset D_k is determined based on their CPU frequencies. For more details on the parameters of the studied FL system, we provide them in Table I.

Table I
Parameter Settings of the wireless edge FL system

Parameter	Value	Parameter	Value
K	15	O	10
B	1.25 Mbps	R	500 Kbps
η	0.9	g_{max}	100
α_k	2e-28	P_k^0	1e-2 W
c_k	20 cycle/bit	P_k	1e-1 W
CR	0.9	F	0.9
T_0	1	N	30



(a) i.i.d data setting (b) non-i.i.d data setting

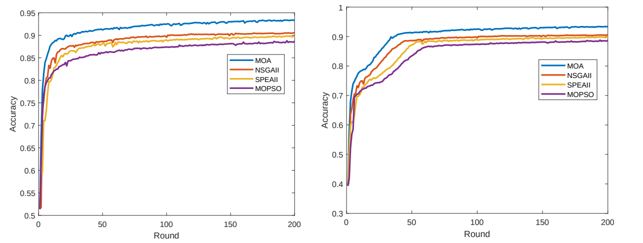
Figure 2. Hypervolume with Fashion-MNIST dataset.

The FL training process involves the classification of handwritten digits via Fashion-MNIST dataset [9]. The training set contains 60,000 samples, while the test set includes 10,000 samples. Fashion-MNIST dataset can be used in both i.i.d and non-i.i.d data settings. In the i.i.d data setting, the data distribution among clients is uniform and balanced, whereas in the non-i.i.d data setting, the distribution of the dataset among clients is unbalanced, with varying dataset sizes among different clients. To train Fashion-MNIST dataset, a multi-layer perceptron network is used, which includes a hidden layer with 50 neurons and employs the Tanh activation function. To evaluate the effectiveness of MOA, three multi-objective approaches, including NSGAII [10], SPEAII [11] and MOPSO [12], are used as baselines.

B. Performance Evaluation

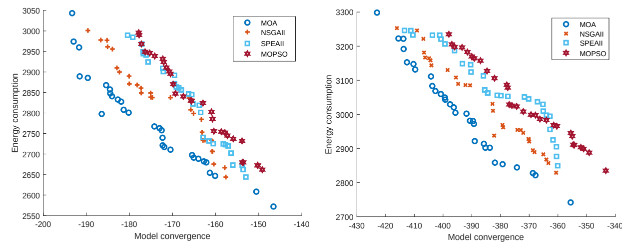
To begin with, we will discuss the convergence of MOA and introduce the hypervolume (HV) metric commonly used to evaluate the convergence. To evaluate the convergence of MOA, we calculate the HVs using the obtained Pareto optimal solution sets with the Fashion-MNIST dataset for two data settings. The results are presented in Fig. 2. Fig. 2(a) shows that in the i.i.d data setting, the HV with Fashion-MNIST dataset initially rises and then fluctuates within a certain range after the 50-th generation. In Fig. 2(b), the HV with Fashion-MNIST dataset in the non-i.i.d data setting rises initially and then starts to fluctuate within a certain range after the 50-th generation. These results demonstrate that MOA can identify the Pareto optimal solution after 50 generations for each problem with Fashion-MNIST dataset. Consequently, MOP is capable of reaching the Pareto optimal solution set.

For the four multi-objective optimization algorithms, they were run 30 times to obtain the average and standard deviation of their supercolumes, and rank sum tests were performed. The



(a) I.i.d data setting (b) Non-i.i.d data setting

Figure 3. Test accuracy among the four multi-objective approaches with Fashion-MNIST dataset.



(a) I.i.d data setting (b) Non-i.i.d data setting

Figure 4. Solutions in the last generation among the seven approaches with Fashion-MNIST dataset.

experimental results are shown in Table II, where +, ≈, and - denote that MOA is better than, similar to, and worse than other baselines, and std is a function to calculate the standard deviation of the members in a vector. This table indicates that MOEA-FL can achieve the best HV under different data settings with Fashion-MNIST dataset, which means it can obtain a better approximate Pareto optimal solution set than the other three multi-objective methods.

Table II
EXPERIMENTAL RESULTS OF MOA AND OTHER THREE BASELINE WITH FASHION-MNIST DATASET

Data setting	MOA Mean HV ± variance	NSGAII Mean HV ± variance	SPEA Mean HV ± variance	MOPSO Mean HV ± variance
i.i.d	**6.13E-01 ± 1.87E-03**	5.87E-01 ± 4.35E-02 +	5.74E-01 ± 7.88E-02 +	5.72E-01 ± 3.30E-02 +
Non i.i.d	**6.07E-01 ± 6.14E-02**	5.75E-01 ± 1.41E-02 +	5.56E-01 ± 6.57E-02 +	5.59E-01 ± 2.00E-02 +
+/≈/-	-	2/0	2/0	2/0

Next, we discuss the accuracy of the FL system used by MOA. We compare the four multi-objective approaches. In order to compare them, we first take the mean value for the Pareto optimal solution set obtained by MOA, NSGAII, SPEAII, and MOPSO, respectively. Then, the solution closest to the mean value and corresponding accuracy curve is selected. The results are shown in Fig. 3. In Fig. 3(a), we can observe that the accuracy for MOA reaches 91.57% and the accuracy for NSGAII reaches 90.24% while the accuracy for SPEAII is 88.93% and the accuracy for MOPSO is 88.06%. Meanwhile, MOA is faster to converge than the other three multi-objective approaches. In Fig. 3(b), among the four multi-objective approaches, MOA obtains the highest accuracy and faster convergence in the non-i.i.d data setting.

Fig. 4 presents the solutions obtained by the four approaches on Fashion-MNIST dataset. It can be observed that MOA obtains the approximate Pareto optimal solution set in both the i.i.d and the non-i.i.d data settings. In contrast, NSGAII, SPEAII, and MOPSO do not perform better than MOA. The reason may be that these methods keep the solutions that need to be discarded in the search process.

V. CONCLUSIONS

This paper investigated an FL system with bandwidth constraints and formulated a MOP that jointly reduces model convergence and energy consumption. To solve the MOP, we proposed MOA, which utilized the Tchebycheff approach to decompose the MOP into single objective problems and employed DE algorithm to optimize them. The results demonstrated that MOA outperforms other algorithms on Fashion-MNIST dataset.

REFERENCES

[1] Y. Yuan, L. Lei, T. X. Vu, Z. Chang, S. Chatzinotas, and S. Sun, "Adapting to dynamic LEO-B5G systems: Meta-critic learning based efficient resource scheduling," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9582–9595, 2022.

[2] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, S. Sun, and B. Ottersten, "Energy minimization in UAV-aided networks: Actor-critic learning for constrained scheduling optimization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5028–5042, 2021.

[3] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 055–20 070, 2022.

[4] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2020.

[5] J. Kim, D. Kim, J. Lee, and J. Hwang, "A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 898–902, 2022.

[6] N. Huang, T. Wang, Y. Wu, S. Bi, L. Qian, and B. Lin, "Delay minimization for intelligent reflecting surface assisted federated learning," *China Communications*, vol. 19, no. 4, pp. 216–229, 2022.

[7] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[8] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[9] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[11] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1787–1796, 2020.

[12] H.-G. Han, Z. Liu, W. Lu, Y. Hou, and J.-F. Qiao, "Dynamic mopso-based optimal control for wastewater treatment process," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2518–2528, 2019.