BARA: Efficient Incentive Mechanism with Online Reward Budget Allocation in Cross-Silo Federated Learning *

Yunchao Yang[1,2], **Yipeng Zhou**[3], **Miao Hu**[1,2 †], **Di Wu**[1,2], **Quan Z. Sheng**[3]
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[3]School of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia
yangych65@mail2.sysu.edu.cn
{humiao5, wudi27}@mail.sysu.edu.cn, {yipeng.zhou, michael.sheng}@mq.edu.au

Abstract

Federated learning (FL) is a prospective distributed machine learning framework that can preserve data privacy. In particular, cross-silo FL can complete model training by making isolated data islands of different organizations collaborate with a parameter server (PS) via exchanging model parameters for multiple communication rounds. In cross-silo FL, an incentive mechanism is indispensable for motivating data owners to contribute their models to FL training. However, how to allocate the reward budget among different rounds is an essential but complicated problem largely overlooked by existing works. The challenge of this problem lies in the opaque feedback between reward budget allocation and model utility improvement of FL, making the optimal reward budget allocation complicated. To address this problem, we design an online reward budget allocation algorithm using Bayesian optimization named BARA (Budget Allocation for Reverse Auction). Specifically, BARA can model the complicated relationship between reward budget allocation and final model accuracy in FL based on historical training records so that the reward budget allocated to each communication round is dynamically optimized so as to maximize the final model utility. We further incorporate the BARA algorithm into reverse auction-based incentive mechanisms to illustrate its effectiveness. Extensive experiments are conducted on real datasets to demonstrate that BARA significantly outperforms competitive baselines by improving model utility with the same amount of reward budget.

1 Introduction

Due to the rising concern on data privacy leakage in recent years, laws such as General Data Protection Regulation (GDPR) [Voigt and Von dem Bussche, 2017] have been made to regulate the collection and use of user data to protect data privacy. Federated learning (FL) [McMahan *et al.*, 2016], as an emerging distributed machine learning paradigm, enables decentralized clients to collaboratively train a shared model without disclosing their private data. The workflow of FL mainly includes: 1) The parameter server (PS) distributes the latest global model to participating clients. 2) Each client updates the model with its local dataset and returns the updated model to the PS. 3) Model aggregation is performed on the PS to obtain a new global model for the next communication round. The above process is repeated until the maximum number of training rounds is reached. Due to its capability in preserving data privacy, the FL market is proliferating. According to [Newswire, 2022], the global FL market size is projected to increase from 127 million dollars in 2023 to 210 million dollars by 2028, at a compound annual growth rate of 10.6% during the forecast period.

However, FL is unsealed without a mechanism to force clients to altruistically contribute their models, and thus an effective incentive mechanism motivating clients is very essential for the success of FL. It was reported in [Ng *et al.*, 2021] that the final model accuracy of FL can be substantially improved by an incentive mechanism, which can inspire more high-quality clients to participate in FL. Reverse auction [Myerson, 1981] has been widely studied in the incentive mechanism design for FL. For example, Deng *et al.* [Deng *et al.*, 2021] designed a quality-aware incentive mechanism based on reverse auction to encourage the participation of high-quality learning users. RRAFL [Zhang *et al.*, 2021] is an incentive mechanism for FL based on reputation and reverse auction theory. A few works even attempted to design incentive mechanisms for FL enhanced by differential privacy. For example, FL-Market [Zheng *et al.*, 2021] was proposed as a novel truthful auction mechanism enabling data owners to obtain rewards according to their privacy losses quantified by local differential privacy (LDP).

Yet, existing works focused on how to optimally allocate rewards between heterogeneous participating clients, ignoring the essential problem of how to allocate rewards between communication rounds. It usually consumes a large number of communication rounds to train advanced machine learning models. On the one hand, if excessive rewards are allocated per communication round, the reward budget will be used up

*An extended version of this paper (with the Appendix included) can be found in http://arxiv.org/abs/2305.05221.

†Corresponding author.

instantly without fully utilizing clients' data for model training. On the other hand, if the amount of allocated rewards is insufficient to solicit high-quality clients, FL fails as well because of the low training efficiency per round.

How to optimally allocate the reward budget between communication rounds is extremely difficult because of the complicated relationship between the final model accuracy and reward budget allocation giving rise to the following three challenges. *First*, once we change the amount of reward budget per communication round, it yields two opposite influences. If we increase the amount of reward budget per communication round, it diminishes the total number of conducted rounds but increases the model utility improvement per communication round, and vice verse. Thereby, it becomes vague how the change of the reward budget allocation will eventually affect the change of final model utility. *Second*, the prediction is problem-related susceptible to random factors, which will be different when training different models using different datasets or hyperparameters (*e.g.*, learning rate and batch size). *Third*, due to the constrained total reward budget, it is unaffordable to exhaustively make trials with different reward budget allocation strategies to search for the best one.

To address the reward budget allocation problem, we design a novel online reward budget allocation algorithm named BARA (Budget Allocation for Reverse Auction). Specifically, a Gaussian process (GP) model [Williams and Rasmussen, 2006] is established to analyze relationship between reward budget allocation and final model utility. Newton's polynomial interpolation is used to expand the training records for establishing the GP model based on historical information. Bayesian optimization is employed to dynamically search the reward budget allocation strategy that can maximize the predicted final model utility. Based on the above analysis, we design the BARA algorithm to determine reward budget allocation per round in an online fashion. It is worth noting that BARA is orthogonal to existing incentive mechanisms optimizing reward allocation between clients. Thus, it can be widely incorporated into existing incentive mechanisms to boost FL utility, which is exemplified by incorporating BARA into reverse auction-based incentive mechanisms in our work.

In summary, our main contributions are presented as follows:

- We establish a GP model to analyze the relation between reward budget allocation and model accuracy. Newton's polynomial interpolation is applied to enrich the training records for the GP model while the Bayesian optimization is employed to search the optimal reward budget allocation strategy.
- Based on our analysis, we propose an online reward budget allocation algorithm (BARA). To our best knowledge, we are the first to address the reward budget allocation across multiple communication rounds to incentivize FL clients.
- We conduct extensive experiments on four public datasets to evaluate our algorithm in comparison with other baselines. The results demonstrate the extraordinary performance and application value of BARA.

2 Related Work

In this section, we briefly discuss related works on incentive mechanism design for FL and Bayesian optimization.

2.1 Incentive Mechanisms in Federated Learning

Incentive mechanism design encouraging clients to contribute their resources for conducting FL has attracted intensive research in recent years. FAIR [Deng *et al.*, 2021] was proposed as an incentive mechanism framework in which reverse auction was employed to incentivize clients by rewarding them based on their model quality and bids. Clients contributing non-ideal model updates will be filtered out before the model aggregation stage. Zeng *et al.* [Zeng *et al.*, 2020] proposed an incentive mechanism with multi-dimensional procurement auction of winner clients. Theoretical results of this strategy was provided as well. RRAFL [Zhang *et al.*, 2021] was designed by combining reputation and reverse auction theory together to reward participants in FL. A reputation calculation method was proposed to measure the clients' quality. Zhou *et al.* [Zhou *et al.*, 2021] considered how to guarantee the completion of FL jobs with minimized social cost. They decomposed the problem into a series of winner determination problems, which were further solved by reverse auction.

It is a more challenging problem to incentivize clients in differentially private federated learning (DPFL) due to the disturbance of noises. FL-Market [Zheng *et al.*, 2021] is a personalized LDP-based FL framework with auction to incentivize the trade of private models with less significant noises. It can meet diversified privacy preferences of different data owners before deciding how to allocate rewards. Liu *et al.* [Liu *et al.*, 2021] introduced the cloud-edge architecture into FL incentive mechanism design to enhance privacy protection. Sun *et al.* [Sun *et al.*, 2021] proposed a contract-based personalized privacy-preserving incentive mechanism for FL by customizing a client's reward as the compensation for privacy leakage cost.

Nevertheless, existing works failed to optimize the reward budget allocation across multiple communication rounds due to the difficulty to explicitly analyze the relationship between a reward budget allocation strategy and final model utility, and this problem will be initially investigated by our work.

2.2 Bayesian Optimization

Bayesian optimization based on Gaussian process (GP) is particularly effective in analyzing a complicated process susceptible to various random factors without the need to derive a closed-form solution. Based on kernel methods and GP models, significant contribution has been made in machine learning [Shahriari *et al.*, 2015]. In [Williams and Rasmussen, 2006], smoothness assumptions of the objective function to be modelled are encoded through flexible kernels in a non-parametric fashion. Srinivas *et al.* [Srinivas *et al.*, 2009] proposed GP-UCB, an intuitive upper-confidence based algorithm. Its cumulative regret in terms of maximal information gain was bounded, and a novel connection between GP optimization and experimental design was established. Bogunovic *et al.* [Bogunovic *et al.*, 2016] considered a sequential Bayesian optimization problem with bandit feedback, which set the reward function to vary with time. GP-UCB was extended to provide an explicit characterization of the trade-off between the time horizon and the rate at which the function varies.

Given the excellent performance of Bayesian optimization based on GP in modeling complicated processes influenced by multiple unknown random factors, our work is novel in applying this approach in FL incentive mechanism design.

3 Preliminaries

We investigate a generic FL system with a single parameter server (PS) owning the test dataset $\mathcal{D}_{test}$ and N clients with local datasets $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$. FL training is completed by multiple communication rounds denoted by rounds $\{1, 2, \dots, t, \dots, T\}$. In communication round t, a typical FL system with a reverse auction-based incentive mechanism [Deng *et al.*, 2021] works as follows:

- **Step 1 (on clients):** Client i reports its bid $b_{t,i}$ to the PS, representing the reward client i desires for participating in FL.
- **Step 2 (on the PS):** The PS measures the quality of each client (*e.g.*, the size of local dataset) denoted by $q_{t,i}$. Based on $q_{t,i}$ and $b_{t,i}$, the PS selects n_t $(1 \leq n_t < N)$ as participating clients and $r_{t,i}$ represents the reward allocated to client i. Then, the PS sends out the latest global model ω_{t-1} to participating clients.
- **Step 3 (on participating clients):** Each participating client updates ω_{t-1} with its local dataset by using local update algorithm (*e.g.*, the gradient descent algorithm [Ruder, 2016]). Then, model updates are returned to the PS.
- **Step 4 (on the PS):** The PS performs model aggregation based on returned model updates to obtain the global model ω_t for the next communication round.

To effectively incentivize clients, reverse auction is widely adopted to determine the reward allocated between clients. More specifically, the PS ranks all clients in terms of the ratio of quality over bid, i.e., $\frac{q_{t,i}}{b_{t,i}}$, in a descending order. Then, the PS can get a list of ranked clients with $\frac{q_{t,1}}{b_{t,1}} > \frac{q_{t,2}}{b_{t,2}} > \cdots > \frac{q_{t,N}}{b_{t,N}}$. Due to the limited total reward budget, the PS sets a reward budget limit B_t to reward participating clients in communication round t. Constrained by B_t, the PS selects top clients from the rank list until B_t is used up. Suppose top n_t clients are selected, the reward for client i $(1 \leq i \leq n_t)$ is $r_{t,i} = \frac{b_{t,n_t+1}}{q_{t,n_t+1}} q_{t,i}$. n_t is determined by the constraint of B_t:

$$n_t = \arg\max_n \sum_{i=1}^{n} \frac{b_{t,n+1}}{q_{t,n+1}} q_{t,i} \leq B_t. \tag{1}$$

A straightforward strategy to determine B_t adopted by existing works is to set the target number of communication rounds T_{max}. If the total reward budget is B_{total}, the reward budget for each communication round is $B_t = \frac{B_{total}}{T_{max}}$.

4 Problem Formulation

In this work, we design a novel algorithm to adjust the number of participating clients based on the typical workflow of FL system with a reverse auction-based incentive mechanism in Section 3.

Let B_{total} denote the total reward budget provided by the PS to recruit participating clients. Let a_t denote the model accuracy after communication round t and Δa_t denote the incremental improvement of model accuracy, i.e., $\Delta a_t = a_t - a_{t-1}$. n $(1 \leq n < N)$ is the number of participating clients. Our objective is to tune the number of recruited clients to adjust consumed reward budget per communication round so as to maximize the final model accuracy. If T communication rounds are conducted in total, our problem can be formulated as:

$$\mathbb{P}1 \quad \max_{n:1 \leq n < N} a_0 + \sum_{t=1}^{T} \Delta a_t, \tag{2}$$

$$s.t. \sum_{t=1}^{T} B_t \leq B_{total}, \tag{3}$$

where a_0 is the test accuracy of the initial global model ω_0. However, it is a very challenging problem because: 1) Δa_t is a function of n as a bigger n brings a larger Δa_t. 2) B_t is a function of n since more participating clients consume more reward budget. 3) T is a function of n as well given a fixed B_{total}. Thus, if these variables in $\mathbb{P}1$ are expressed as functions of n, we can get:

$$\mathbb{P}2 \quad \max_{n:1 \leq n < N} a_0 + \sum_{t=1}^{T(n)} \Delta a_{t,n}, \tag{4}$$

$$s.t. \sum_{t=1}^{T(n)} B_t(n) \leq B_{total}. \tag{5}$$

With the knowledge of $T(n)$, $\Delta a_{t,n}$ and $B_t(n)$, we can solve $\mathbb{P}2$. The term $B_t(n)$ is determined by the reverse auction-based incentive mechanism once n is fixed. As discussed in Section 3, we can calculate the minimum reward budget $B_t(n)$ consumed by selecting n clients to participate in FL in the t-th communication round as:

$$B_t(n) = \sum_{i=1}^{n} \frac{b_{t,n+1}}{q_{t,n+1}} q_{t,i}. \tag{6}$$

Based on $B_t(n)$, $T(n)$ can be computed correspondingly. Let $x_{\tau,n} = 1$ denote if there are n participating clients in communication round τ. $T(n)$ can be estimated if n participating clients are selected in communication round t as:

$$T_t(n) \approx \frac{B_{total}}{\bar{B}_t(n)}, \tag{7}$$

where $\bar{B}_t(n) = \frac{\sum_{\tau=1}^{t} x_{\tau,n} B_t(n)}{\sum_{\tau=1}^{t} x_{\tau,n}}$ is the average reward budget consumption per communication round estimated in round t. Unfortunately, there is no prior work that explicitly defines $\Delta a_{t,n}$. Thus, the main challenge for solving $a_{t,n}$ is how to accurately estimate $\Delta a_{t,n}$.

5 Methodology

In this section, we propose an online reward budget allocation algorithm using Bayesian optimization to solve $\mathbb{P}2$. We first utilize Newton's polynomial interpolation to synthesize training records based on historical records observing reward budget allocation and model accuracy improvement. Based on training records, a Gaussian process (GP) is established to model the relationship between final model accuracy and reward budget allocation strategies. Next, Bayesian optimization is employed to search for the optimal reward budget allocation strategy in an online fashion to maximize the final model accuracy.

5.1 Training Records Synthesis

How to exactly estimate $\Delta a_{t,n}$ (representing model accuracy improvement with n participating clients in the t-th communication round) is a challenging open problem. When training different models, we can get different $\Delta a_{t,n}$. Until communication round t, we denote the number of participating clients in the τ-th communication round as n_τ $(1 \leq \tau \leq t)$. However, we cannot compute $\Delta a_{\tau,n}$ prior to model training and it is also impossible to obtain $\Delta a_{\tau,n}$ if $n \neq n_\tau$. Thereby, we approximate unknown training records during the model training process with historical records of $\Delta a_{\tau,n}$. We can use a matrix $\mathbf{M}_t \in \mathbb{R}^{t \times N-1}$ to denote $\Delta a_{\tau,n}$ until communication round t (i.e., $\mathbf{M}_t = [\Delta n a_{\tau,n}]$ $(1 \leq \tau \leq t$ and $1 \leq n < N)$. $\Delta a_{\tau,n}$ will be empty if $n \neq n_\tau$. Each row represents a communication round and note that only a single element in each row is from FL training records since we can only select a single n_τ for communication round τ.

However, to determine which n yields the highest final model accuracy, we need the knowledge of all elements in $\Delta a_{1,n}, \Delta a_{2,n}, \ldots, \Delta a_{T(n),n}$. With all known elements in the n-th column of $\mathbf{M}_t$, we employ the Newton's polynomial interpolation [Hildebrand, 1987] to approximate the values of those unknown elements.

Without loss of generality, we briefly explain how to apply the Newton's polynomial interpolation for a particular n-th column. Let $t_1, t_2, \ldots, t_J$ denote indices of elements in the n-th column with known value from past training records. Then, we can define a number of divided differences as follows:

$$y[t_1, t_2, \cdots, t_J] = \frac{y[t_1, \cdots, t_{J-1}] - y[t_2, \cdots, t_J]}{t_1 - t_J}. \quad (8)$$

Here $y[t_j] = \Delta a_{t_j,n}$ for all $1 \leq j \leq J$. We can easily compute $y[t_1, t_2, \cdots, t_J]$ with $y[t_j]$ and Eq. (8). Unknown values in $\mathbf{M}_t$ until $t = T(n)$ can be estimated by:

$$\begin{aligned} \Delta \hat{a}_{\tau,n} =& y[t_1] + y[t_1, t_2](\tau - t_1) + \cdots \\ &+ y[t_1, t_2, \cdots, t_J] \prod_{j=1}^{J-1} (\tau - t_j), \end{aligned} \quad (9)$$

for $\tau = 1, 2, \ldots, t, \ldots, T(n)$. Let $T = \max_{1 \leq n < N} T(n)$ denote the maximum number of communication rounds we can conduct by selecting n clients per communication round. Based on interpolation results, we can create the estimation matrix $\widehat{\mathbf{M}}_T = [\Delta \hat{a}_{t,n}]$ $(1 \leq t \leq T$ and $1 \leq n < N)$. Note that $\Delta \hat{a}_{t,n}$ is valid only if $t \leq T(n)$. Although $\Delta \hat{a}_{1,n}, \ldots, \Delta \hat{a}_{T(n),n}$ can be estimated through Eq. (9), the predicted model accuracy improvement is vulnerable to overfitting. In particular, in the first few communication rounds, the number of available historical records is insufficient for accurately finding the best n. Thus, we establish a learning process to dynamically predict the accuracy with different n before we can decide the optimal number of participating clients.

5.2 Searching for Optimal Reward Budget Allocation Strategy

It is known that the model accuracy performance is susceptible to various factors such as the data distribution among clients. It is difficult to accurately predict final model accuracy only based on the number of participating clients. In light of this complication, we use a Gaussian process (GP) to model the random evolution of final model accuracy when taking different reward budget allocation strategies.

As we collect more records of n_t and $\Delta \hat{a}_{t,n_t}$ along the training process, the matrix $\widehat{\mathbf{M}}_t$ expands gradually and the predicted value of each unknown element will be updated according to Eq. (9). Note that our goal is to predict the final model accuracy when using different n. For convenience, we define $\hat{a}_t(n)$ to represent the estimated final model accuracy predicted at communication round t. In other words, $\hat{a}_t(n) = a_0 + \sum_{t=1}^{T(n)} \Delta \hat{a}_{t,n}$ where $\Delta \hat{a}_{t,n}$ are elements in the n-th column of $\widehat{\mathbf{M}}_T$.

Note that $\hat{a}_t(n)$ is derived based on a few observations in matrix $\widehat{\mathbf{M}}_T$. It only utilizes elements in the n-th column for prediction failing to fully utilize all observations to predict $\hat{a}_t(n)$. To overcome this drawback, we model $\hat{a}_t(n)$ with a GP to capture the relationship between $\hat{a}_t(n)$ and $\hat{a}_{t'}(n')$ so that we can fully utilize all observations to more accurately predict $\hat{a}_t(n)$.

To distinguish with final model accuracy obtained by interpolation via Eq. (9), we define $z_t(n)$ as the final model accuracy sampled from the GP model in the t-th communication round assuming that n clients are recruited to conduct FL in each communication round. Specifically, $z_t(n_t)$ is modeled as a random variable sampling values from the distribution of $\mathrm{GP}(\mu(n_t), k(n_t, n_t))$ where $k(n_t, n_t)$ is the covariance (or kernel) function. The mean value of $z_t(n_t)$ is denoted by $\mu(n_t) = \mathbb{E}[z_t(n_t)]$. With multiple variables $z_t(n_t)$, we needs to consider the covariance $\mathbb{E}[(z_t(n_t) - \mu(n_t))(z_t(n_{t'}) - \mu(n_{t'}))]$ when modeling the relationship between two choices of n_t and $n_{t'}$ at two different communication rounds t and t', respectively.

According to [Srinivas *et al.*, 2009], the squared exponential kernel function is widely adopted to model covariance for a GP. Note that the elements in $\widehat{\mathbf{M}}_t$ (i.e., $\Delta \hat{a}_{t,n}$) are approximated by Newton's polynomial interpolation. Therefore, the approximation gets better over time as we collect more observation records of $\Delta a_{t,n}$, which means fresh observations are more valuable than stale ones. We construct the composite kernel to weigh stale and fresh observations differently based on Ornstein-Uhlenbeck temporal covariance function. Together with the squared exponential kernel function, the

Algorithm 1: The details of BARA algorithm

Input: clients' local datasets $\{\mathcal{D}_{i=1}\}_i^N$; initial global model parameters ω^0; test dataset D_{test}; GP prior (μ_0, σ_0, k); total reward budget B_{total}
Output: final global model parameters
PS:

```
Test ω_0 on D_test to get a_0
B ← 0
for t = 1 to T_max do
    Client i report b_{t,i} to PS, i ≤ N
    if t < T_0 then
        Randomly select n_t from {1, 2, ···, N − 1}
    else
        n_t ← arg max_{1≤n<N} μ_{t−1}(n) + sqrt(β_t) σ_{t−1}(n)
    end if
    Sort all clients in descending order of q_{t,i} / b_{t,i}
    B_t(n_t) ← Σ_{i=1}^{n_t} (b_{t,n_t+1} / q_{t,n_t+1}) q_{t,i}
    B ← B + B_t(n_t)
    if B > B_total then
        Return ω_{t−1}
    end if
    for i = 1 to n_t in parallel do
        r_{t,i} ← (b_{t,n_t+1} / q_{t,n_t+1}) q_{t,i}
        Send global model parameters ω_{t−1} to client i
        ∇l(ω_{t−1}; D_i) ← ClientUpdate(ω_{t−1}, D_i)
    end for
    Perform model aggregation based on ∇l(ω_{t−1}; D_i)
    to obtain ω_t and test it on D_test to obtain a_t
    Update M_t, M̂_t and perform Bayesian posterior
    update as in Eqs. (12) - (13) to obtain μ_t and σ_t
end for
```

ClientUpdate$(\omega_{t-1}, \mathcal{D}_i)$:

```
Receive ω_{t−1} from the PS
Calculate ∇l(ω_{t−1}; D_i) based on ω_{t−1} and D_i
Upload ∇l(ω_{t−1}; D_i) to the PS
```

covariance between $z_t(n_t)$ and $z_{t'}(n_{t'})$ is modeled by

$$k(n_t, n_{t'}) = (1-\lambda)^{\frac{|t-t'|}{2}} \exp\left(-\frac{||n_t - n_{t'}||^2}{2l^2}\right), \quad (10)$$

where $l > 0$ is a length scale hyperparameter to determine how much the two points n_t and $n_{t'}$ influence each other. Intuitively, if n_t is closer to $n_{t'}$, the value of $k(n_t, n_{t'})$ is bigger implying that $z_t(n_t)$ and $z_{t'}(n_{t'})$ are more correlated. Thus, the information of $z_t(n_t)$ is more useful for us to predict $z_{t'}(n_{t'})$. Moreover, stale observations should be weighted lighter and lighter over time. Here λ controls how fast the weights of stale observations decrease.

Until communication round t, we have made t different choices of n_t. Thus, we can establish a GP with t variables to predict the distribution of $z_{t+1}(n)$. To simplify our presentation, let $\mathbf{z}_t$ denote the vector of the first t choices of n, i.e., $\mathbf{z}_t = \{z_1(n_1), z_2(n_2), \ldots, z_t(n_t)\}$. Let $\mu(\mathbf{n}_t) = \{\mu(n_1), \mu(n_2), \ldots, \mu(n_t)\}$. The deviation between the observed value and the real value can be gauged by a zero-mean random noise $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$, which is independent with time. According to [Williams and Rasmussen, 2006], $(\mathbf{z}_t, z_{t+1}(n_{t+1}))$ is a sample drawn from the following distribution:

$$\mathcal{N}\left(\begin{bmatrix}\mu(\mathbf{n}_t)\\ \mu(n_{t+1})\end{bmatrix}, \begin{bmatrix}\mathbf{K}_t + \sigma^2\mathbf{I} & \mathbf{k}_t(n_{t+1})\\ \mathbf{k}_t^\top(n_{t+1}) & k(n_{t+1}, n_{t+1})\end{bmatrix}\right), \quad (11)$$

for $1 \leq n_{t+1} < N$. Note that n_{t+1} is the choice of round $t+1$, which has not occurred yet. Here $\mathbf{I}$ is a $t \times t$ identity matrix. $\mathbf{K}_t$ is the positive definite kernel matrix $[k(n, n')]_{\forall n, n' \in \mathbf{n}_t}$ and $\mathbf{k}_t(n_{t+1}) = [k(n_1, n_{t+1}), k(n_2, n_{t+1}), \ldots, k(n_t, n_{t+1})]^\top$. It is easy to see that the joint distribution given in Eq. (11) describes the relationship between $t+1$ variables. It can be regarded as the prior knowledge of Bayesian optimization to learn the posterior knowledge. At the end of communication round t, we can update the posterior knowledge, i.e., estimated mean and variance for $z_t(n)$ when choosing different n, as follows:

$$\mu_t(n) = \mathbf{k}_t^\top(n)[\mathbf{K}_t + \sigma^2\mathbf{I}]^{-1}\hat{\mathbf{a}}_t, \quad (12)$$

$$\sigma_t^2(n) = k(n, n) - \mathbf{k}_t^\top(n)[\mathbf{K}_t + \sigma^2\mathbf{I}]^{-1}\mathbf{k}_t(n), \quad (13)$$

for $1 \leq n < N$. Here, $\hat{\mathbf{a}}_t = \{\hat{a}_0(n), \hat{a}_1(n), \ldots, \hat{a}_t(n)\}$, which is computed based on the n-th column in matrix $\widehat{\mathbf{M}}_T$. In Eq. (12), we jointly utilize the results of Newton's polynomial interpolation, i.e., $\hat{\mathbf{a}}_t$, and GP which captures the correlation when choosing different n via the term $\mathbf{k}_t^\top(n)[\mathbf{K}_t + \sigma^2\mathbf{I}]^{-1}$ to predict $z_t(n)$. In this approach, we can fully utilize all historical records to make prediction. In Eq. (13), the variance of $z_t(n)$ is updated accordingly to gauge the uncertainty of the estimation in Eq. (12).

Note that $\mu_t(n)$ represents the expected final model accuracy by choosing n clients per communication round predicted at communication round t. As t increases, we will collect more and more information to continuously improve our prediction. For the $(t+1)$-th round, the decision aiming to maximize the final model accuracy should select the one that can maximize $\mu_t(n)$. However, considering the uncertainty of our prediction seized by $\sigma_t^2(n)$, it is better to add an exploration term based on $\sigma_t^2(n)$. Specifically, the decision of n for communication round $t+1$ is:

$$n_{t+1} = \arg\max_{n: 1 \leq n < N} \mu_t(n) + \sqrt{\beta_{t+1}}\sigma_t(n), \quad (14)$$

where $\sqrt{\beta_{t+1}}$ is a tuneable constant. $\sqrt{\beta_{t+1}}\sigma_t(n)$ is the exploration term, created based on previous empirical experience [Srinivas *et al.*, 2009]. The convergence property of Eq. (14) been proved in [Bogunovic *et al.*, 2016], which can guarantee that searched n_t will gradually approach to the optimal n^*. It will be further verified by our experiments in the next section.

It is worth mentioning that a random strategy to select n should be adopted at the early stage of FL training because observation records are insufficient to establish the GP for accurately learning posterior knowledge. Specifically, the online reward budget allocation algorithm has two stages: 1) a pure exploration stage and 2) an exploration-exploitation stage. In Stage 1, the PS randomly selects n_t

from $\{1, 2, \cdots, N-1\}$ in each communication round and observations will be recorded. In Stage 2, the Bayesian optimization is performed with enriched prior knowledge. The first pure exploration stage can be executed for a fixed T_0 communication rounds. Then, it will proceed to the exploration-exploitation stage in the remaining communication rounds. Note that Stage 1 should not exhaust the reward budget such that Stage 2 can be conducted.

We describe the detailed procedure of BARA in Algorithm 1. For each communication round $t < T_0$, the PS randomly selects n_t from $\{1, 2, \cdots, N-1\}$ in Stage 1 (line 6). After client selection and model update (lines 10-20), the PS updates the matrices $\mathbf{M}_t$ and $\widehat{\mathbf{M}}_t$, which employed to Bayesian posterior update to obtain μ_t and σ_t (line 22). In Stage 2, the PS balances exploration and exploitation based on the GP posterior (line 8). The time complexity of sorting all clients in descending order is $\mathcal{O}(N \log N)$ (line 10). Here $T = \max_{1 \le n < N} T(n)$. The time complexity of updating $\mathbf{M}_t$, $\widehat{\mathbf{M}}_t$ and Bayesian posterior update (line 22) are both $\mathcal{O}(T^2)$. The overall time complexity of BARA is $\mathcal{O}(\max\{N \log N, T^2\})$, which is lightweight in comparison with training advanced FL models.

6 Experiments

6.1 Experimental Setups

Datasets and Models

We use four public datasets for experiments: MNIST [LeCun *et al.*, 1998], Fashion-MNIST (also abbreviated as FMNIST) [Xiao *et al.*, 2017], CIFAR-10 and CIFAR-100 [Krizhevsky *et al.*, 2009] datasets. Similar to [McMahan *et al.*, 2016], we train a multilayer perceptron (MLP) model that consists of 2-hidden layers for classifying the MNIST dataset. A CNN (convolutional neural network) model that consists of two convolution layers (each followed by a max pooling layer and ReLU activation), then followed by a fully connected layer is trained for classifying the FMNIST dataset. For CIFAR-10 and CIFAR-100 datasets, we train the CNN model with the same structure as that in [Mills *et al.*, 2021], which consists of two convolutional pooling layers, two batch normalization layers and two fully connected layers.

In real scenarios, the typical data distribution on FL suffers from statistical heterogeneity due to the fact that the training data owned by a particular client is usually related with user-specific features. Therefore, we allocate training datasets to clients in a non-IID manner. According to [Chen *et al.*, 2022], for each dataset, we first sort samples by their labels and then split them into $2N$ shards equally. Each of N clients randomly selects 2 shards.

Parameter Settings

We set the total number of clients N as 20. Based on the empirical results in [Deng *et al.*, 2021], we set the maximum number of communication rounds, i.e., T_{max}, and the total reward budget of the PS, i.e., B_{total}, as 200 and $1,500$, respectively. We implement a typical reverse auction-based incentive mechanism for FL: bid price first (i.e., all clients are of equal quality) [Deng *et al.*, 2021]. Each client's bid for participating in each FL training round is independently sampled from a uniform distribution $U(0.5, 1.5)$. Referring to [Bogunovic *et al.*, 2016], we set hyperparameters in the GP as $\sqrt{\beta_t} = 0.8 \log(0.4t)$, the length scale parameter $l = 0.2$ for squared exponential kernel and λ as 0.001. We set the noise variance σ^2 to 0.01. The pure exploration stage runs for initial $T_0 = 40$ communication rounds.

Compared Baselines

We compare BARA with the following reward budget allocation baseline methods:

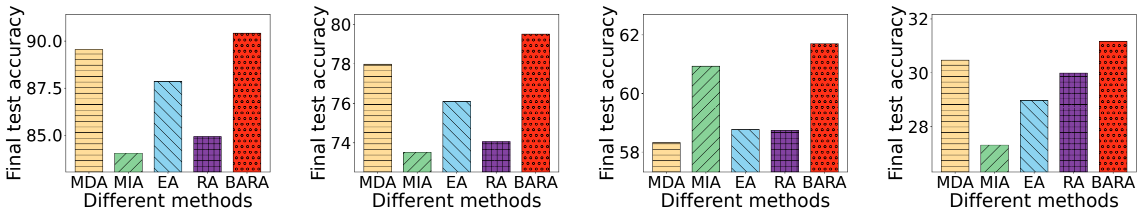
- **Even allocation (EA):** The PS allocates the total reward budget B_{total} evenly to each communication round (i.e., $B_t = \frac{B_{total}}{T_{max}}$), which is commonly adopted in existing works [Deng *et al.*, 2021; Zhang *et al.*, 2021; Zheng *et al.*, 2021].
- **Monotonically increasing allocation (MIA):** The allocated reward budget B_t for round t is a monotonically increasing function with t (i.e., $B_t = 2\frac{B_{total}}{T_{max}^2}t$).
- **Monotonically decreasing allocation (MDA):** The allocated reward budget B_t of round t is a monotonically decreasing function with t (i.e., $B_t = -2\frac{B_{total}}{T_{max}^2}t + 2\frac{B_{total}}{T_{max}}$).
- **Random allocation (RA):** In each communication round, the PS randomly selects the number of participating clients from $\{1, 2, \cdots, N-1\}$. The FL training process halts once the total reward budget is used up.

Evaluation Metrics

We adopt two metrics, *test accuracy* and *regret*, to evaluate our algorithm. Test accuracy evaluates the accuracy of ω^t on $\mathcal{D}_{test}$ in each communication round t. By comparing test accuracy, we can evaluate how much performance gain can be achieved by optimizing the reward budget allocation in FL. Regret evaluates the gap between the solution of our algorithm and the theoretically optimal solution. To obtain the theoretically optimal solution, we enumerate n in FL to find which $n*$ can achieve the highest final model accuracy on the test set. Due to the limited reward budget in practice, it is impossible to enumerate all possible n. Thus, the constraint of the reward budget is not considered for searching n^*. Once n^* is determined, we define the regret at round t as $Reg_t = \sum_{\tau=1}^{t}(a_{T(n^*)}(n^*) - \hat{a}_\tau(n_\tau))$ where $\hat{a}_\tau(n_\tau)$ is the estimation of model accuracy predicted by our model. Intuitively, if Reg_t approaches 0 with t, i.e., $\lim_{t\to\infty} Reg_t/t = 0$, it implies that BARA can approximately find the optimal solution after a certain number of rounds.

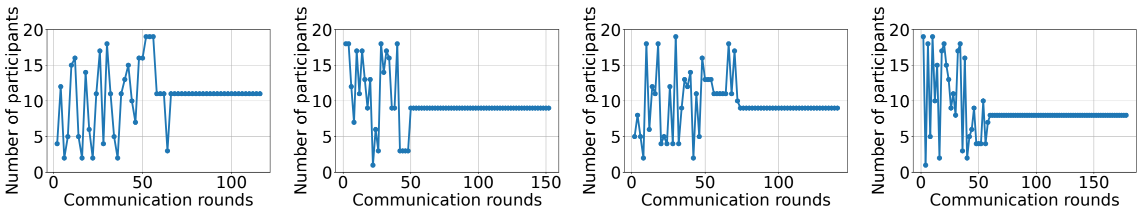
6.2 Experimental Results

We first conduct experiments to compare BARA with other reward budget allocation methods. The experimental results are plotted in Figure 1 with the x-axis representing different reward budget allocation methods and the y-axis representing the final model accuracy on $\mathcal{D}_{test}$. The results in Figure 1 manifest that BARA can significantly outperform other baselines in term of final model accuracy. The analysis of experimental results are presented in Appendix A.



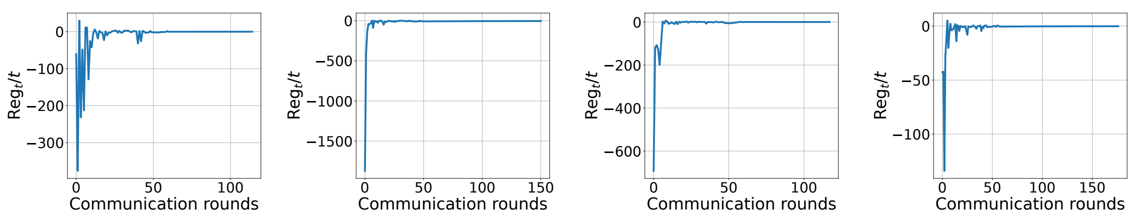
(a) MNIST (b) FMNIST (c) CIFAR-10 (d) CIFAR-100

Figure 1: Comparing final model accuracy of different reward budget allocation methods on four datasets.



(a) MNIST (b) FMNIST (c) CIFAR-10 (d) CIFAR-100

Figure 2: The number of participants per communication round selected by BARA on four datasets.



(a) MNIST (b) FMNIST (c) CIFAR-10 (d) CIFAR-100

Figure 3: The regret (showing the gap to the optimal value) per communication round of BARA.

Next, we investigate the learning process of BARA for searching the optimal number of participating clients. Based on experimental results in Figure 1, we plot the number of participating clients selected by BARA for each dataset in Figure 2. Here, the x-axis represents the communication round and the y-axis represents the number of participating clients per communication round. From Figure 2, we can observe that 1) In the initial $T_0 = 40$ communication rounds, the number of participating clients fluctuate over time as the PS randomly selects n for participating in FL. 2) The number of participating clients quickly converges to a stable value beyond the critical point $T_0 = 40$ indicating that BARA can efficiently explore n^* with sufficient historical records.

To further verify the effectiveness of BARA, we evaluate the regret of BARA for each dataset. In each communication round, we plot Reg_t/t in Figure 3 with the x-axis representing the communication round and y-axis representing Reg_t/t. From Figure 3, we can observe the fast convergence of the regret curve when $t > T_0$. As the regret approaches to 0, it implies that BARA finds n^* for determining the number of participating clients.

BARA is applicable for various different incentive mechanisms. To demonestrate this generic value of BARA, we implement two more typical reverse auction-based incentive mechanisms for FL. Their performance can be further enhanced by incorporating BARA into their mechanisms. The detailed experimental results are presented at Appendix B.

7 Conclusion

To our best knowledge, our work is the first one to investigate the reward budget allocation problem between training rounds in federated learning given a limited total budget. Due to the complicated relationship between reward budget allocation and final model utility, we established a Gaussian process (GP) model to predict model utility with respect to reward budget allocation. To expand the historical knowledge for building the GP model, Newton's polynomial interpolation was applied to generate artificial records. We further employed the Bayesian optimization to determine the reward budget allocation to maximize the predicted final model utility. Based on our analysis, an online reward budget allocation algorithm called BARA was proposed, which is lightweight for implementation. Finally, extensive experiments were conducted to demonstrate the effectiveness of BARA by extensively improving model accuracy compared with baselines.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U1911201, U2001209, 62072486, and the Natural Science Foundation of Guangdong Province under Grant 2021A1515011369.

References

[Bogunovic *et al.*, 2016] Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying Gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 314–323. PMLR, 2016.

[Chen *et al.*, 2022] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with a graph. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

[Deng *et al.*, 2021] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. FAIR: Quality-aware federated learning with precise user incentive and model aggregation. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10. IEEE, 2021.

[Hildebrand, 1987] Francis Begnaud Hildebrand. *Introduction to numerical analysis*. Courier Corporation, 1987.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Liu *et al.*, 2021] Tianyu Liu, Boya Di, Peng An, and Lingyang Song. Privacy-preserving incentive mechanism design for federated cloud-edge learning. *IEEE Transactions on Network Science and Engineering*, 8(3):2588–2600, 2021.

[McMahan *et al.*, 2016] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[Mills *et al.*, 2021] Jed Mills, Jia Hu, and Geyong Min. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):630–641, 2021.

[Myerson, 1981] Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[Newswire, 2022] PR Newswire. Global federated learning solutions market by application (drug discovery, industrial IoT), vertical (healthcare & life sciences, BFSI, manufacturing, retail & e-commerce, energy & utilities), and region - forecast to 2028. Technical report, Guinness Centre, Taylors Lane, Dublin 8, D08 AH31, Ireland., 2022.

[Ng *et al.*, 2021] Kang Loon Ng, Zichen Chen, Zelei Liu, Han Yu, Yang Liu, and Qiang Yang. A multi-player game for studying federated learning incentive schemes. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 5279–5281, 2021.

[Ruder, 2016] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[Shahriari *et al.*, 2015] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[Srinivas *et al.*, 2009] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2009.

[Sun *et al.*, 2021] Peng Sun, Haoxuan Che, Zhibo Wang, Yuwei Wang, Tao Wang, Liantao Wu, and Huajie Shao. Pain-fl: Personalized privacy-preserving incentive for federated learning. *IEEE Journal on Selected Areas in Communications*, 39(12):3805–3820, 2021.

[Voigt and Von dem Bussche, 2017] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[Williams and Rasmussen, 2006] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Zeng *et al.*, 2020] Rongfei Zeng, Shixun Zhang, Jiaqi Wang, and Xiaowen Chu. FMore: An incentive scheme of multi-dimensional auction for federated learning in mec. In *IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 278–288. IEEE, 2020.

[Zhang *et al.*, 2021] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the ACM Web Conference (WWW)*, pages 947–956, 2021.

[Zheng *et al.*, 2021] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. FL-Market: Trading private models in federated learning. *arXiv preprint arXiv:2106.04384*, 2021.

[Zhou *et al.*, 2021] Ruiting Zhou, Jinlong Pang, Zhibo Wang, John CS Lui, and Zongpeng Li. A truthful procurement auction for incentivizing heterogeneous clients in federated learning. In *IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 183–193. IEEE, 2021.