

Incentive Mechanism Design for Joint Resource Allocation in Blockchain-based Federated Learning

Zhilin Wang, Qin Hu, Ruinian Li, Minghui Xu, and Zehui Xiong

Abstract—Blockchain-based federated learning (BCFL) has recently gained tremendous attention because of its advantages such as decentralization and privacy protection of raw data. However, there has been few research focusing on the allocation of resources for clients in BCFL. In the BCFL framework where the FL clients and the blockchain miners are the same devices, clients broadcast the trained model updates to the blockchain network and then perform mining to generate new blocks. Since each client has a limited amount of computing resources, the problem of allocating computing resources into training and mining needs to be carefully addressed. In this paper, we design an incentive mechanism to assign each client appropriate rewards for training and mining, and then the client will determine the amount of computing power to allocate for each subtask based on these rewards using the two-stage Stackelberg game. After analyzing the utilities of the model owner (MO) (i.e., the BCFL task publisher) and clients, we transform the game model into two optimization problems, which are sequentially solved to derive the optimal strategies for both the MO and clients. Further, considering the fact that local training related information of each client may not be known by others, we extend the game model with analytical solutions to the incomplete information scenario. Extensive experimental results demonstrate the validity of our proposed schemes.

Index Terms—Federated learning, blockchain, resource allocation, incentive mechanism, game theory

1 INTRODUCTION

SINCE its emergence in 2016, federated learning (FL) has been greatly developed and widely applied in many fields, such as Internet of Things [1]–[3], smart transportation [4], [5] and healthcare [6]–[8]. One of the most important advantages of FL is that there is no transmission of raw data from mobile devices to the centralized server for model training; instead, by training models on local devices (a.k.a., clients) and averaging all local models, FL significantly reduces the possibility of leaking data privacy to a large extent [9]. However, there are still some challenges that may restrain the implementation and wide application of FL, e.g., the risk of the single point of failure, malicious attacks from participated clients, and the lack of participation incentives [10]–[13].

In recent years, researchers resort to blockchain technology to tackle the challenges of FL, where the blockchain system usually works as a decentralized system to provide incentives and data verification [14]–[16]. The combination of blockchain and FL is termed as blockchain-based FL

This work is partly supported by the US NSF under grant CNS-2105004.

- Zhilin Wang and Qin Hu (corresponding author) are with the Department of Computer and Information Science, Indiana University-Purdue University Indianapolis, IN, 46202, USA. E-mail: {wangzhiqinhu}@iu.edu
- Ruinian Li is with the Department of Computer Science, Bowling Green State University, Bowling Green, Ohio, 43551, USA. E-mail: lir@bgsu.edu
- Minghui Xu is with the School of Computer Science and Technology, Shandong University, China. E-mail: mhxu@sdu.edu.cn
- Zehui Xiong is with Pillar of Information Systems Technology and Design, Singapore University of Technology Design, Singapore. E-mail: zehui_xiong@sutd.edu.sg

(BCFL). In the BCFL framework, model updates submitted by clients will be verified by miners before the global aggregation algorithm is conducted. Once the global model is obtained, it will be updated into the main chain that can be accessed by all qualified participants. Though BCFL can partially address the aforementioned challenges of traditional FL, there are still some remaining issues that need to be addressed.

One of the most critical problems in BCFL is the resource allocation on local devices. Firstly, local devices with heterogeneous computational power usually have their own tasks to finish, so a universal resource allocation scheme for all the mobile devices is not practical. In addition, the whole system may not work effectively and sustainably if there are no reasonable rewards allocated to clients. Furthermore, both training and mining in the framework of BCFL consume significant amount of resources and time, and thus it is difficult for clients to appropriately allocate their limited resources to ensure the performance of the global model during the required time period. Lastly, since the system may not be aware of the amount of training data that each client owns, it can be challenging for the model owner (MO), i.e., the BCFL task publisher, to make proper decisions regarding the reward distribution.

There exist very few studies that tackle the above challenges [17], [18]. The existing studies are based on two assumptions which are not practical: 1) all clients have identical computational power and data volume; and 2) the system knows all the information about clients. To fill the gap, we propose an incentive mechanism for joint resource allocation on mobile devices in BCFL that is applicable to incomplete information scenario.

For the first challenge regarding the unbalanced distribution of resources on mobile devices, we let the clients decide how much computational power they are willing to devote into the training and mining tasks by themselves. By this means, clients can flexibly allocate computation resources for their own tasks. In addition, in our model, training and mining are performed sequentially, and the amount of computational power devoted to these two tasks can be different.

To overcome the second challenge of motivating clients to join BCFL, we design an incentive mechanism to reward clients. Training and mining are two different tasks that require different amount of computational power, and thus the rewards should also be different. To ensure a fair distribution of rewards to all clients, we employ the approach of Shapely Value (SV) [19] to determine the contributions of clients in the training process, which will affect the constraints in their respective optimization problems.

To address the last two challenges, we build the Stackelberg game model under both the complete and incomplete information situations, which are solved separately but with different insights. Based on the derived optimal solutions, our system can make optimal decisions in different information conditions.

In summary, our contributions in this work can be summarized as below:

- We model the BCFL resource allocation problem as a two-stage Stackelberg game to help the MO make decisions on assigning how many rewards to each client for training and mining and to assist clients in determining the corresponding amount of computational power to be devoted in each subtask, via maximizing their respective utilities.
- In order to maintain the stability and sustainability of the whole BCFL system, we design a fair reward allocation scheme inspired by SV to calculate the rewards for clients based on their contributions in the training process.
- Considering that the training related information of devices may not be known to others in the practical application scenario, we further study the resource allocation mechanism under the incomplete information situation and derive the optimal solutions accordingly.
- We test our proposed resource allocation mechanisms through extensive experiments. The experimental results show that these mechanisms are effective.

The rest of this paper is organized as follows. We introduce the system model and problem formulation based on the two-stage Stackelberg game in Section 2. The detailed models and solutions under complete and incomplete information scenarios are reported in Section 3 and Section 4, respectively. Experimental evaluations are presented in Section 5. We present the related work in Section 6. Finally, we conclude this paper in Section 7.

2 SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we will illustrate the system model of our considered blockchain-based federated learning (BCFL) and

then formulate the problem from the perspective of resource allocation and incentive mechanism design based on Stackelberg game. For convenience, we list the key notations in Table 1.

TABLE 1: Key Notations.

Notation	Meaning
\mathcal{N}	The set of clients
N	The total number of clients
q_i	The maximum number of client i 's CPU cycle per second
q_{ti}	The number of CPU cycles per second used to train
q_{mi}	The number of CPU cycles per second used to mine
p_{ti}	The unit price for training to client i
p_{mi}	The unit price for mining to client i
π	The number of training iterations for clients during one round of BCFL to submit model update
D_i	The data size of client i
d_i	The number of CPU cycles used for training each data sample
μ_i	The total CPU cycles required to finish the local training for generating model updates
T_i	The time spent on training for client i
ψ	The total CPU cycles used to mine for each client
T_{mi}	The time spent on mining for client i
U_i	The utility of client i in one round of BCFL
U_{mo}	The utility of the MO in one round of BCFL
q_{ti}^*	The optimal CPU cycles per second for training
q_{mi}^*	The optimal CPU cycles per second for mining
p_{ti}^*	The optimal unit price for training to client i
p_{mi}^*	The optimal unit price for mining to client i

2.1 System Overview

Inspired by [20], we consider a fully coupled BCFL system, which runs FL on a consortium blockchain network. In such a decentralized BCFL system, the participants in FL work as the blockchain nodes as well. Specifically, there are multiple local devices, termed as clients, working collaboratively to train a machine learning model, i.e., the global model. The set of clients can be denoted as $\mathcal{N} = \{1, \dots, i, \dots, N\}$ with N representing the total number of clients in the BCFL system. For simplicity, we refer to the work done by the blockchain for FL as *mining* in a uniform way, which does not imply that clients perform mining jobs consuming excessive amount of computing power like Proof of Work (PoW) [21]. In our system, we consider that lightweight consensus algorithms, such as Practical Byzantine Fault Tolerance (PBFT) [22] and Delegated Proof of Stake (DPoS) [23], are utilized in the consortium blockchain system.

In our considered BCFL system, each client should be responsible for both training and mining. Since the workflow of the fully coupled BCFL is that mining starts only after the training is completed, we assume that training and mining are not parallel in this paper. Once the training is finished, all the clients will upload their local model updates to the blockchain network so as to be recorded on the blockchain. Here we define one round of BCFL as finishing both the training and mining processes. Since mobile devices usually have their own tasks to finish rather than only contributing to BCFL, we assume that they will not use all of their computation resources. In other words, CPU cycles per second for training and mining can be adjusted strategically in each round of BCFL.

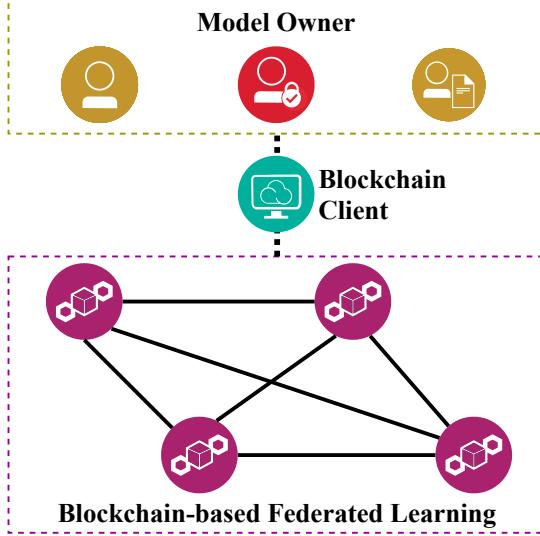


Fig. 1: An illustration of the BCFL system.

The MO is the requester of the FL task, aiming at to receive a well-trained final global model from the BCFL system. After the FL task is published on the blockchain, clients start to train their local models and then broadcast the obtained model updates to the blockchain network once the local training process is finished. By this means, the MO can only access the model updates from all clients rather than raw data of devices, thus preventing the leakage of private information for participants. An illustration of our system model is shown in Fig. 1.

The detailed workflow to finish one BCFL task is as below:

- The MO publishes an FL task, with the rewards for training and mining.
- Clients determine the computational resources, i.e., CPU cycles per second, used to train the model and mine for the blockchain based on the rewards provided by the MO.
- Each client trains the local model, and then broadcasts the model updates to the blockchain network. Then, clients start to mine the block.
- Once the block is generated, the model updates are stored on the blockchain, and the rewards will be delivered to each client.
- Clients calculate the global model with the verified model updates on chain. As long as the expected performance of the global modal is not reached, clients will start the next round of training based on the aggregated global model.

2.2 Utility Models

Since the computational resource of client i is limited and we assume that client i has other tasks to finish rather only working for the BCFL, it is essential to design a decision mechanism for client i to allocate CPU cycles for training and mining, respectively. What's more, an incentive mechanism is necessary because clients will be reluctant to contribute to the BCFL task without receiving enough compensation for their efforts. It is difficult to design such

a mechanism because the MO has limited rewards budget, and it is necessary to make sure that the time consumption of training and mining can be shortened and a well performed global ML model can be obtained at the same time. Thus, in the following part, we build the utility models of both the MO and client from the perspectives of resource allocation and incentive mechanism.

2.2.1 Client's Utility

We assume that the maximum number of client i 's CPU cycles per second is q_i , and the number of CPU cycles per second used to train and mine are q_{ti} and q_{mi} , respectively. Then we have $q_{ti}, q_{mi} \leq q_i$. Let π be the number of training iterations for clients during one round of BCFL to submit model update, which is usually fixed for all clients. Let D_i be the number of the data size of client i , and d_i be the number of CPU cycles used for training each data sample. Therefore, we can define the total CPU cycles required to finish the local training to generate model updates as $\mu_i = \pi d_i D_i$.

Since any client i can decide its CPU cycles used to train the local model, the time used to finish the local training varies for each client. We can calculate the time spent on training for client i via $T_{ti} = \frac{\mu_i}{q_{ti}}$. Besides, we denote the total CPU cycles used to mine for each client as ψ , which is the same for all clients since mining a new block in blockchain system usually consumes fixed computational resources. Thus, the time spent on mining can be calculated as $T_{mi} = \frac{\psi}{q_{mi}}$. So we can have the total time cost of client i to finish a round of BCFL task as $T_i = T_{ti} + T_{mi}$. Since it is not possible to let T_{ti} and T_{mi} be limitless according to the convergence time requirement, we denote the upper bound of time consumption in one round of BCFL by T . Thus, we have $T_i \leq T$.

In order to encourage clients to join BCFL, the MO provides some rewards to clients, where the prices per second for training and mining are denoted as p_{ti} and p_{mi} , respectively. Clients can allocate unit CPU cycles for training and mining based on the unit prices given by the MO. Then the rewards of client i for training and mining to generate one round of local model updates are $R_{ti} = T_{ti} p_{ti}$ and $R_{mi} = T_{mi} p_{mi}$, respectively. Thus, the total rewards for client i in one round of BCFL is $R_i = R_{ti} + R_{mi}$.

Next, we can calculate the energy costs for training and mining as $C_{ti} = \rho_i \mu_i q_{ti}^2$ and $C_{mi} = \rho_i \psi q_{mi}^2$ base on a widely used energy consumption model [24], where ρ_i is the parameter correlated to the chip architecture.

In this way, the total cost¹ can be calculated as $C_i = C_{ti} + C_{mi}$.

Finally, we can obtain the utility of client i in one round of BCFL as

$$U_i = R_i - C_i = \frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi q_{mi}^2. \quad (1)$$

2.2.2 MO's Utility

The main concerns related to the utility of the MO are the performance of the global model, the time consumption and

1. As for the communication cost, since the submissions of clients are the same, we can consider it as a constant value, which cannot be optimized anymore and thus is omitted here.

the rewards paid to all participants in each round of BCFL, where the first one is a sort of revenue and the last two are related to the cost for the MO.

Generally, the performance of an ML model will be affected by the number of CPU cycles spent for training. Thus, we define the performance of the global model after one round of local training and mining as G which can be calculated by $G = f(\sum_{i=1}^N \mu_i)$. Here $f(\cdot)$ is a monotonically increasing function, indicating that more CPU cycles used for the local training by all clients, the better performance of the global model after aggregation will be achieved. As for the MO, its utility depends on the performance of the BCFL system (G), total time cost ($\sum_{i=1}^N (T_i)$), and total rewards distributed to clients ($\sum_{i=1}^N (R_i)$). Thus, the utility of the MO in one round of BCFL can be expressed as

$$\begin{aligned} U_{mo} &= f\left(\sum_{i=1}^N \mu_i\right) - \xi \sum_{i=1}^N (T_i + R_i) \\ &= f\left(\sum_{i=1}^N \mu_i\right) - \xi \sum_{i=1}^N \left(\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} + \frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi}\right), \end{aligned} \quad (2)$$

where $\xi > 0$ is a scalar parameter to balance the revenue and cost.

2.3 Problem Formulation using Two-stage Stackelberg Game

According to the above analysis of our system model, client i provides its computational power to finish BCFL tasks based on the rewards given by the MO. In other words, the unit prices p_{ti} and p_{mi} determine the unit computational power q_{ti} and q_{mi} . We can formulate the interactions between clients and the MO as a two-stage Stackelberg game, which is widely used for the complete information dynamic game [25]. In this game, the MO determines the unit prices of the CPU-cycle frequency used for training and mining, and then client i decides its CPU cycles per second based on the received prices, which means that the decision of client i is impacted by the decision of the MO. In this case, we can define the process of the two-stage Stackelberg game as below:

- Stage I: The MO sets the unit prices per second for training and learning for each client, i.e., p_{ti} and p_{mi} , via maximizing its own utility, which is specifically based on its budget and the total number of CPU cycles consumed for training submitted by each client. Taking into account the fairness of the distribution for setting price, we need to design a fair reward allocation scheme here.
- Stage II: After receiving the unit prices from the MO, clients determine their corresponding computational power, i.e., q_{ti} and q_{mi} , through optimizing their respective utilities.

In practical situations, q_{ti} and q_{mi} are not independent of each other because of time and reward budget constraints; similarly, p_{ti} and p_{mi} influence each other as well. Therefore, we should take these constraints into account when modeling to make the decisions reasonable.

Recall μ_i in Section 2.2.1, we know that μ_i is a variable correlated to the data size of client i and the performance of the corresponding device, which may not always known to the MO or the system. As for ψ , it can be predefined by the system since generating a new block usually consumes a constant amount of resources. Therefore, we can classify the two-stage Stackelberg game into information complete and incomplete scenarios based on whether μ_i is known to the MO. The models derived for these two scenarios are different, and hence the strategies of the MO and clients are different accordingly, which will be explored in Sections 3 and 4.

3 RESOURCE ALLOCATION WITH COMPLETE INFORMATION

In this section, we will elaborate the expressions of the proposed Stackelberg game model and the corresponding solutions for the clients and the MO in the scenario of complete information, which means that the MO makes its decisions when μ_i of each client is known as a prior. First, we propose a fair reward allocation scheme for clients, and then we transfer the two-stage Stackelberg game into two separate optimization problems that are resolved sequentially. The methodology we adopt to solve the two problems is backward induction, which requires analyzing the optimal strategies of Stage II first and then the strategies of Stage I.

3.1 Fair Reward Allocation

Before we formulate the game model, we should clarify the fair reward allocation scheme first. In our system, we consider that each client has an equal chance to participate in both the training and mining processes with fair rewards. And since the allocation of rewards to each client in training and mining has a significant impact on the system fairness and further the participation willingness, we need to design a fair reward allocation scheme. Although we have already defined the payoff of each client during the training and mining processes in the above section, it is necessary to investigate their upper bounds based on the MO's rewards budget. And the rewards distribution should not only be associated with the computing power of the device, but also take into account the performance of its work. On the one hand, the reward budget of the MO and the rewards that each client can get are limited; on the other hand, if the resources are allocated only based on the computing power devoted, it could lead to the situation where devices with sufficient computing power take most of the rewards, while devices with less power cannot get enough rewards, making the system unstable and unsustainable.

3.1.1 Upper Bound of Rewards for Mining

For simplicity, we set a fixed total reward budget η in each round of BCFL. Since the computational power consumed by generating a new block is constant, with η_m denoting the upper bound of the reward for mining that all clients can receive, we have:

$$\frac{\psi}{q_{mi}} p_{mi} \leq \bar{R}_{mi} = \frac{\eta_m}{N}, \quad (3)$$

where \bar{R}_{mi} is the upper bound of the reward for mining that each client can get.

3.1.2 Upper Bound of Rewards for Training

Since the devices in our BCFL system are assumed to be homogeneous, and they may have different computing capabilities, we cannot simply distribute the rewards evenly to each client. To guarantee the fairness of reward distribution, we allocate rewards based on the contribution of each client in the training process. Considering that Shapely Value (SV) [19] is a methodology which can distribute the rewards to participants according to their respective contributions, here we apply it to facilitate reward distribution. The SV of client i is defined as

$$SV_i(\mathcal{N}, v) = \sum_{i \notin S, S \subseteq \mathcal{N}} \frac{s!(N-s-1)!}{N!} (v(S \cup i) - v(S)), \quad (4)$$

where $S \subseteq \mathcal{N}$ is a subset of clients and $s = |S|$ is the number of devices in the set S ; $v(S)$ is a function describing the performance of the training result with the client set S . Then, we give the expression of function $v(S)$. Recall G in Section 2.2.2, we can assume that $v(S)$ is a function correlated to G and it can be defined as

$$v(S) = W - \left\| \frac{\sum_{i=1}^s G}{s} - g \right\|_2, \quad (5)$$

where $W = \max_{S \subseteq \mathcal{N}} \left\| \frac{\sum_{i=1}^s G}{s} - g \right\|_2$ and $\|\cdot\|_2$ is the Euclidean norm; g is the targeted performance value. Then, we can calculate the upper bound of the reward distributed to each device for training as:

$$\bar{R}_{ti} = \frac{SV_i(\mathcal{N}, v)}{v(\mathcal{N})} (\eta - \eta_m). \quad (6)$$

For each client, its rewards should not exceed the upper bound, so we can have the following constraint:

$$\frac{\mu_i}{q_{ti}} p_{ti} \leq \bar{R}_{ti}. \quad (7)$$

3.2 Stage II: Clients Set CPU Cycles Per Second based on Unit Rewards

Since each client i has a limited amount of computational resource and should follow the working rules of BCFL, the goal of client i is to maximize its utility as follows:

Problem 1: $\max : U_i,$

$$\text{s.t. } \frac{\mu_i}{q_{ti}} p_{ti} - \rho_i \mu_i q_{ti}^2 \geq 0, \quad (8)$$

$$\frac{\psi}{q_{mi}} p_{mi} - \rho_i \psi q_{mi}^2 \geq 0, \quad (9)$$

$$\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} \leq T, \forall i \in \mathcal{N}, \quad (10)$$

where the first two constraints (8) and (9) mean that client i wishes to gain non-negative payoffs in both training and mining; and the last constraint (10) indicates that client i should finish the working process, including training and mining, within the time period T .

It is clear that Problem 1 is a nonlinear optimization problem with inequality constraints, so we adopt the

method of Karush-Kuhn-Tucker (KKT) conditions to solve it. First, we need to demonstrate that Problem 1 can be resolved. We can calculate $\frac{\partial U_i}{\partial q_{ti}} = -\frac{\mu_i p_{ti}}{q_{ti}} - 2\rho_i q_{ti} < 0$ and $\frac{\partial U_i}{\partial q_{mi}} = -\frac{\mu_i p_{mi}}{q_{mi}} - 2\rho_i q_{mi} < 0$, so it can be proved that U_i is concave and it has the maximum value. By solving Problem 1, we get the following theorem:

Theorem 3.1. *The optimal strategies of client i in the scenario of complete information are given by*

$$q_{ti}^* = \left(\frac{p_{ti}}{\rho_i} \right)^{\frac{1}{3}}, \quad (11)$$

$$q_{mi}^* = \frac{\psi}{T - \mu_i \left(\frac{\rho_i}{p_{ti}} \right)^{\frac{1}{3}}}. \quad (12)$$

The detailed proof of Theorem 3.1 is in Appendix A. From the above theorem, we can see that the number of optimal CPU cycles per second client i putting into training grows as the unit price for training given by the MO increases. The optimal CPU cycles per second devoted to mining is constraint by ψ , indicating that if the mining work requires more CPU cycles, client i should mine with a larger q_{mi}^* .

3.3 Stage I: MO Sets Unit Prices for Clients

The MO expects to get a global model with good performance consuming time and cost for rewards as less as possible, so its goal is to maximize the utility function U_{mo} , and the optimization problem can be formulated as follows:

$$\begin{aligned} \text{Problem 2: } \max : & U_{mo}, \\ \text{s.t. } & \frac{\mu_i}{q_{ti}} p_{ti} \leq \bar{R}_{ti}, \end{aligned} \quad (13)$$

$$\frac{\psi}{q_{mi}} p_{mi} \leq \bar{R}_{mi}, \forall i \in \mathcal{N}, \quad (14)$$

where (13) and (14) are the constraints of individual rewards from training and mining to meet the MO's budget.

It is clear that Problem 2 is also a nonlinear optimization problem, and U_{mo} is also concave, so we can list all the KKT conditions to find its maximum value. Via solving Problem 2, we can have:

Theorem 3.2. *The optimal strategies of the MO in the scenario of complete information are:*

$$p_{ti}^* = \left(\frac{1}{\rho_i} \right)^{\frac{1}{2}} \left(\frac{\bar{R}_{ti}}{\mu_i} \right)^{\frac{3}{2}}, \quad (15)$$

$$p_{mi}^* = \frac{\bar{R}_{mi}}{T - (\mu_i \rho_i)^{\frac{3}{2}} \left(\frac{1}{\bar{R}_{ti}} \right)^{\frac{1}{2}}}. \quad (16)$$

The proof of Theorem 3.2 is shown in Appendix B. In the optimal solutions above, p_{mi}^* and p_{ti}^* are highly correlated. This is because there are time and budget constraints so that p_{ti}^* and p_{mi}^* are not independent variables from each other. In other words, the MO needs to balance p_{ti}^* and p_{mi}^* to satisfy the constraints when making decisions. Furthermore, we can find that μ_i and ψ influence the optimal decisions as well.

We summarize the resource allocation mechanism with complete information in Algorithm 1. The MO calculates the

unit prices given to the client for training and mining first, and then calculates its utility based on the previous unit prices (Lines 1-2). If U_{mo} is the optimal utility for the MO, then the optimal decisions of MO can be obtained (Lines 3-5). Next, the MO sends the unit prices to clients, and each client calculates the the numbers of CPU cycles per second used for training and mining; if the utility for client i is optimal, client i can make its optimal decisions and start to train and mine (Lines 6-12).

Algorithm 1 Resource Allocation Mechanism with Complete Information

Require: $T, \mu_i, \psi, \rho_i, \eta, \bar{R}_{mi}$
Ensure: $q_{ti}^*, q_{mi}^*, p_{ti}^*, p_{mi}^*$

- 1: The MO calculates \hat{p}_{ti} and \hat{p}_{mi} via (15) and (16)
- 2: The MO calculates U_{mo} based on \hat{p}_{ti} and \hat{p}_{mi} via (2)
- 3: **if** $U_{mo}(\hat{p}_{ti}, \hat{p}_{mi}) \geq U_{mo}(p_{ti}, p_{mi})$ **then**
- 4: $p_{ti}^* \leftarrow \hat{p}_{ti}, p_{mi}^* \leftarrow \hat{p}_{mi}$
- 5: **end if**
- 6: The MO sends p_{ti}^* and p_{mi}^* to the client i
- 7: **for** $i \in \mathcal{N}$ **do**
- 8: Calculate \hat{q}_{ti} and \hat{q}_{mi} via (??) and (12)
- 9: **if** $U_i(\hat{q}_{ti}, \hat{q}_{mi}) \geq U_i(q_{ti}, q_{mi})$ **then**
- 10: $q_{ti}^* \leftarrow \hat{q}_{ti}, q_{mi}^* \leftarrow \hat{q}_{mi}$
- 11: Client i uses q_{ti}^* to train and q_{mi}^* to mine
- 12: **end if**
- 13: **end for**
- 14: **return** $q_{ti}^*, q_{mi}^*, p_{ti}^*, p_{mi}^*$

In general, the case of complete information is an ideal situation, and we find that it mainly influences the optimal decisions of the MO. Therefore, we can study the optimal decisions in the case of incomplete information by adjusting the decision mechanism of the MO.

4 RESOURCE ALLOCATION WITH INCOMPLETE INFORMATION

In this section, we will discuss the game model in the case of incomplete information where the MO has no knowledge of the true value of μ_i for each client. Thus, the MO needs to set the unit price in such a way that each client has a non-negative payoff, while ensures that the clients report the value of μ_i honestly. Before designing the resource allocation mechanism, we first give two definitions below.

Definition 4.1. (*Individual Rationality*). *The incentive mechanism for resource allocation is individually rational if the utility of client i given the rewards provided by the MO is non-negative, i.e.,*

$$U_i(q_{ti}, q_{mi}, p_{ti}, p_{mi}, \mu_i) \geq 0, \forall i. \quad (17)$$

Definition 4.2. (*Incentive Compatibility*). *The incentive mechanism for resource allocation is incentive compatible if each client can get the optimal utility by reporting its μ_i truthfully, i.e.,*

$$U_i(q_{ti}, q_{mi}, p_{ti}, p_{mi}, \mu_i) \geq U_i(q_{ti}, q_{mi}, p_{ti}, p_{mi}, \hat{\mu}_i), \forall i, \quad (18)$$

where $\hat{\mu}_i$ represents any value of μ_i .

Based on the previous analysis, we know that clients' decisions are made based on their non-negative utility. Since

clients should ensure that the rewards they receive are not less than the total costs they spend, in such a situation, they can participate in the BCFL task. So in the situation of incomplete information, the MO needs to guarantee that its decisions should satisfy (17) to encourage clients to join the work. Besides, μ_i of client i is not known by the MO, and the decisions of the MO are required to be based on the correct value of μ_i reported by clients, so the MO needs to satisfy (18) when making the decisions.

Since the client sets the CPU cycles per second after the unit prices are given by the MO, the decisions of the client in the case of incomplete information are the same as those made under the complete information case as discussed in Section 3.2. Therefore, we will only focus on the derivation of the optimal strategies of the MO in this section.

With incomplete information, the MO has to ensure that the allocation of rewards to all clients is fair, the clients' utilities are non-negative, and clients report μ_i truthfully. Thus, the decision-making problem of the MO with incomplete information can be transformed into the following optimization problem:

$$\begin{aligned} \text{Problem 3: } \max : \quad & U_{mo} \\ \text{s.t. } & (13), (14), (17), (18), \\ & \forall i \in \mathcal{N}, \end{aligned}$$

where (17) and (18) are the constraints of individual rationality and incentive compatibility for the mechanism; (13) and (14) are the constraints of individual rewards for meeting the MO's budget.

To solve Problem 3, we can first write it in Lagrangian form according to its optimization objective and constraints, and then analyze its KKT conditions. The optimal solutions can be solved as below:

Theorem 4.1. *The optimal strategies of the MO in the scenario of incomplete information are*

$$p_{ti}^* = \left(\frac{1}{\rho_i} \right)^{\frac{1}{2}} \left(\frac{\bar{R}_{ti}}{\mu_i} \right)^{\frac{3}{2}}, \quad (19)$$

$$p_{mi}^* = \frac{\rho_i \psi^3}{\left(T - \mu_i \left(\frac{\rho_i}{p_{ti}^*} \right)^{\frac{1}{3}} \right)^3}. \quad (20)$$

The proof of Theorem 4.1 is presented in Appendix C. The optimal solution for p_{ti}^* in the incomplete information case is the same as the optimal solution in the complete information case, while p_{mi}^* is different. Since the decision of the MO in the case of incomplete information is not only influenced by the budget of the reward, but also required to satisfy the two conditions (17) and (18) in the above definitions. In other words, the decisions in this case is more conservative so the MO would prefer to minimize its cost by reducing the payments to training and mining. We will illustrate the specific differences about the decisions in the two scenarios through experiments in Section 5.

The resource allocation mechanism in the incomplete-information case is presented in Algorithm 2, which is similar to Algorithm 1, except the decision process of the MO. In the scenario, the MO should make sure that its utility

is optimal and the utility for each client is non-negative (Lines 2-7).

Algorithm 2 Resource Allocation Mechanism with Incomplete Information

Require: $T, \mu_i, \psi, \rho_i, \eta, \bar{R}_{mi}$
Ensure: $q_{ti}^*, q_{mi}^*, p_{ti}^*, p_{mi}^*$

- 1: The MO calculates \hat{p}_{ti} and \hat{p}_{mi} via (19) and (20)
- 2: **if** $U_{mo}(\hat{p}_{ti}, \hat{p}_{mi}) \geq U_{mo}(p_{ti}, p_{mi})$ **then**
- 3: The MO calculates the expected utility \hat{U}_i of client i
- 4: **if** $\hat{U}_i \geq 0$ **then**
- 5: $p_{ti}^* \leftarrow \hat{p}_{ti}, p_{mi}^* \leftarrow \hat{p}_{mi}$
- 6: **end if**
- 7: **end if**
- 8: The MO sends p_{ti}^* and p_{mi}^* to client i
- 9: **for** $i \in \mathcal{N}$ **do**
- 10: Calculate \hat{q}_{ti} and \hat{q}_{mi}
- 11: **if** $U_i(\hat{q}_{ti}, \hat{q}_{mi}) \geq U_i(q_{ti}, q_{mi})$ **then**
- 12: $q_{ti}^* \leftarrow \hat{q}_{ti}, q_{mi}^* \leftarrow \hat{q}_{mi}$
- 13: Client i uses q_{ti}^* to train and q_{mi}^* to mine
- 14: **end if**
- 15: **end for**
- 16: **return** $q_{ti}^*, q_{mi}^*, p_{ti}^*, p_{mi}^*$

We can see that the time complexity of both Algorithm 1 and Algorithm 2 is $O(n)$, which means that the time consumption of solving these two optimization problems increases with the number of clients linearly. Therefore, our proposed algorithms can work efficiently in practice.

5 EXPERIMENTAL EVALUATION

In this section, we will conduct numerical experiments to verify and support our designed mechanism. We first clarify the experimental settings and then illustrate the results. We implement the simulations using Matlab 2019b in macOS 11.0.1 running on Intel i7 processor with 32 GB RAM and 1 TB SSD.

5.1 Experimental Setting

In our experiments, we mainly focus on the impacts of four variables (i.e., μ_i , ψ , p_{ti} and p_{mi}) on our designed models under complete and incomplete situations. The basic setting for these simulations are slight different, and we will clarify the different parts of the settings in each experiment. For simplicity of calculation and presentation, we use GHz as the unit of CPU cycles per second and minute as the unit of time. We first set $\eta = 1500$ and $\bar{R}_{mi} = 5$. Since we adopt SV to calculate the total rewards distributed to individual client and SV is correlated to the value of μ_i (see (4) and (5)), we let $G = \frac{\sum_{i=1}^N \mu_i}{N}$. By running the algorithm of SV we can get the value of \bar{R}_{ti} for each client. The settings for other parameters are $\rho_i = 0.01$, $\xi = 0.1$, $g = 10$ and $T = 15$. Note that we conducted extensive experiments with other experimental settings, while we found that different values of the parameters would not influence the trends of the results. So we only present the results of the experiments based on the above settings.

5.2 Experimental Results

First, we prove the correctness of the optimal strategies derived from our models. We assume there are 50 clients in total and each client has the same data size, so we set $\mu_i = 10$. In our experiments, for clients and the MO, there are four strategy combinations, i.e., both sides choose the optimal strategies, one chooses the random strategies while the other chooses optimal strategies, and both choose the random strategies. For example, we define the strategy combination *Random vs. Optimal* as the clients choose the random strategies and the MO chooses the optimal strategy. We compare the utilities of clients and the MO with random strategies and optimal strategies, respectively. The results in Fig. 2 illustrate that clients and the MO can obtain the higher utilities than all other strategies when they both choose the optimal strategies, proving the validity of our proposed optimal strategies.

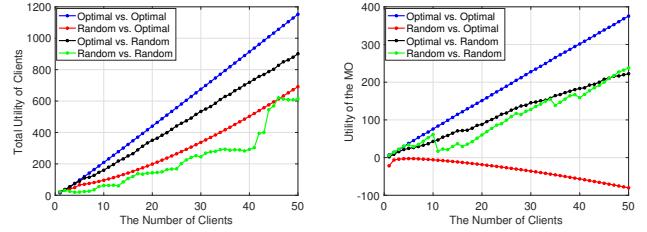


Fig. 2: Utilities changing with strategy pairs.

Then, the experiments will be designed to study the impacts of μ_i and ψ on the utility of clients and the MO under the situations of complete and incomplete information. We set $\mu_i \in [0, 5]$ and $\psi \in [0, 5]$. The simulation results are shown in Fig. 3. We can see that both μ_i and ψ have a significant impact on the utility of the MO. That is because the higher CPU power will shorten the time in each round and improve the performance of the global model. However, for clients, devoting more CPU cycles does not result in more utility due to higher energy consumption.

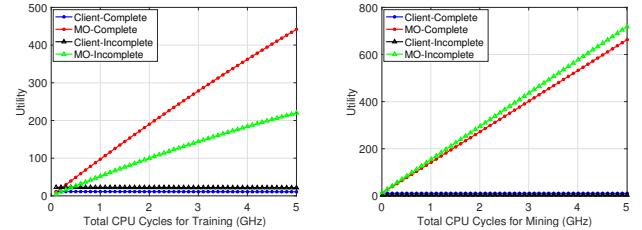


Fig. 3: Utilities of the client and the MO changing with the total CPU cycles for training and mining.

We then study the effect of p_{ti} and p_{mi} on the utility of the MO and clients. We set $p_{ti} \in [0, 10]$ and $p_{mi} \in [0, 10]$. The results are shown in Fig. 4. If the unit price of training increases, clients can be stimulated to provide more computing power, which reduces the time cost and improves the model performance, so the MO utility will be improved. However, the revenue of clients does not grow significantly with the increase of the unit price of training, because the

cost of energy consumption also rises. p_{mi} has the same effect on utility for both complete and incomplete information cases, and the results are shown on the right side of Fig. 4. When the unit price of mining increases, the utility of both clients and the MO can be improved. This is because with the increase of p_{mi} , clients can receive more mining revenue by devoting more q_{mi} . At the same time, the MO can reduce the time cost and improve its utility by encourage clients to devote more CPU cycles per second for mining.

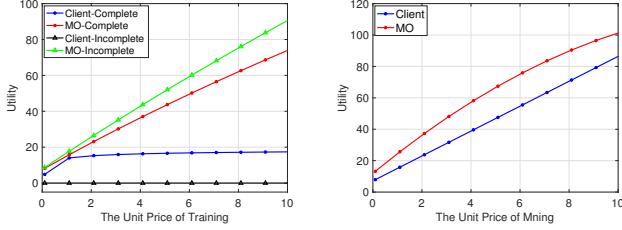


Fig. 4: Utilities of the client and the MO changing with the unit prices of training and mining.

Next, we conduct experiments to analyze the relationship between μ_i and the unit price for training and mining. We set $\mu_i \in [0, 5]$, and the results are illustrated in Fig. 5. We can see that both the unit price and the number of CPU cycles for training increase with μ_i . This is because if μ_i increases, more rewards are needed to motivate clients to put more computational resources in training. In general, μ_i does not affect p_{mi} and q_{mi} a lot, as the benefits of mining are relatively constant and are more influenced by the resource allocation scheme.

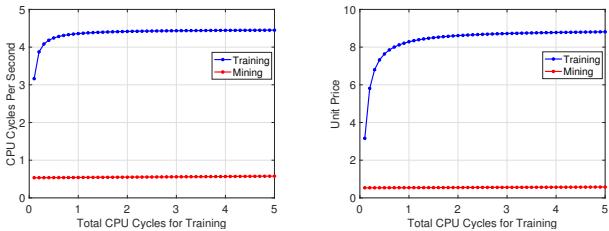


Fig. 5: Impacts of μ_i on CPU cycles per second and unit prices for training and mining.

In the end, we explore the influence of p_{ti} on both q_{ti} and q_{mi} to figure out how the decisions of MO influence the decisions of client i . We set $\mu_i = 10$ and $p_{ti} \in [0, 10]$. In this setting, the simulation results are shown in Fig. 6. We can see that the unit CPU cycles used in local model training has a positive relationship with the unit price of training offered by the MO, because more unit rewards for training will incentivize clients to put more computational power on model training. As for CPU cycles per second used in mining, it decreases with the increase of p_{ti} . This makes sense because if clients are motivated to put more computing power into training, the training time will be reduced and the mining time will be correspondingly increased. In this way, clients do not need to set a high q_{mi} for mining.



Fig. 6: CPU cycles per second for training and mining changing with p_{ti} .

6 RELATED WORK

Most of the existing studies related to BCFL focus on protecting privacy, achieving decentralization and improving the performance of model training [26]–[30]. In our paper, we mainly focus on the resource allocation and incentive mechanism design in BCFL. Thus, we provide the literature review about these two areas in this section.

6.1 Resource Allocation in BCFL

As for resource allocation, researchers mainly consider the homogeneous computational power of all clients, and make decisions through the reinforcement learning approach.

In [17], the resource allocation problem is resolved for the local devices with the same computational power in BCFL. An upper-bound of the global loss function was proposed to evaluate the performance of training; in the meantime, the relationship among update rounds, block generation rate, and learning rate was explored. Although the proposed method can easily control the training and mining time by adjusting the number of updates to allocate resources, it is based on the assumption that all clients have the same amount of computing resources and local data, which is not practical.

Hieu et al. [18] designed a deep reinforcement learning approach to help mobile devices determine the data volume and energy used for training and to assist the system to decide the block generation rate. Neither of the two works considers how to motivate clients to work honestly and efficiently.

According to the above discussion, it can be seen that the studies related to resource allocation in BCFL are insufficient. One of the reasons is that the research regarding BCFL is still in the early stage. Another reason is that there are many types of BCFL structures depending on the role of the blockchain playing in FL, making it difficult to have a common framework for resource allocation. In order to assist the MO and the clients of the BCFL system to make the proper decisions, we design the mechanisms based on the two-stage Stackelberg game in this paper. Besides, we consider allocating resources in the fully coupled BCFL with FL clients working as blockchain nodes.

6.2 Incentive Mechanism in BCFL

There are some studies about BCFL that focus on regulating the behaviors of clients through incentive mechanism

design, thus encouraging them to work honestly and efficiently according to the predefined rules.

Toyoda et al. [31] proposed an economic approach based on the assumption that clients would act rationally, where the repeated competition method was utilized to ensure that clients will follow the protocol. Bao et al. [32] designed an incentive mechanism to attract more data and computational power contributing to the framework of BCFL. In their proposed system, honest clients can gain fairly partitioned rewards while the malicious clients will be punished via timely behaviour detection scheme. In [33], an incentive mechanism that integrated reputation and contract theory was proposed to encourage clients to provide high-quality data to train the local models. As for the fairness of reward allocation, Liu et al. [34] used Shapley Value (SV) to calculate the contributions of clients of the FL system and then allocate the rewards accordingly. However, this approach is not able to make incentive decisions for training and mining, respectively.

The existing studies about incentive mechanism design in BCFL focus on how to provide incentives for FL through blockchain, without considering the incentives for blockchain and FL in a systematical manner. In other words, blockchain and FL are in different phases for BCFL, so they should both have reasonable incentives. In our paper, we design a pricing mechanism for the MO based on the computing power provided by clients, thus providing incentives to the whole BCFL system.

In general, the existing studies have paid little attention to resource allocation for mobile devices in BCFL and assume that clients join the task voluntarily. To address this challenge, we design a resource allocation mechanism for mobile devices, which also offers reward suggestions to the MO so as to motivate clients to participate in BCFL.

7 CONCLUSION

This paper studies the resource allocation of clients in BCFL by designing incentive mechanism. We describe the interactions between clients and the MO as a two-stage Stakelberg game. Within our model, clients with varying computing power can determine the resources to invest in training and mining based on the rewards provided by the MO through maximizing their utilities, while the MO can also obtain the optimal utility. Since the local training related information of clients may be not known to the MO, we further study the game model and optimal solutions in the incomplete information case. Numerous experimental results show that our proposed mechanisms are effective.

APPENDIX A

PROOF OF THEOREM 3.1

The Lagrangian correlated to **Problem 1** is expressed as:

$$\begin{aligned} \mathcal{L}_1 = & \frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi (q_{mi})^2 \\ & - \lambda_1 \left(\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti} \right) - \lambda_2 \left(\rho_i \psi q_{mi}^2 - \frac{\psi}{q_{mi}} p_{mi} \right) \\ & - \lambda_3 \left(\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T \right), \forall i, \end{aligned} \quad (21)$$

where λ_1 , λ_2 , and λ_3 are non-negative parameters correlated to the constraints of **Problem 1**.

The KKT conditions are as below:

$$\frac{\partial \mathcal{L}_1}{\partial q_{ti}} = \frac{\partial \mathcal{L}_1}{\partial q_{mi}} = 0, \forall i, \quad (22)$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0, \forall i, \quad (23)$$

$$\lambda_1 \left(\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti} \right) = 0, \forall i, \quad (24)$$

$$\lambda_2 \left(\rho_i \psi q_{mi}^2 - \frac{\psi}{q_{mi}} p_{mi} \right) = 0, \forall i, \quad (25)$$

$$\lambda_3 \left(\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T \right) = 0, \forall i, \quad (26)$$

$$\frac{\mu_i}{q_{ti}} p_{ti} - \rho_i \mu_i q_{ti}^2 \geq 0, \forall i, \quad (27)$$

$$\frac{\psi}{q_{mi}} p_{mi} - \rho_i \psi q_{mi}^2 \geq 0, \forall i, \quad (28)$$

$$\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} \leq T, \forall i. \quad (29)$$

According to (22), we can have $\frac{\partial \mathcal{L}_1}{\partial q_{ti}} = \frac{u_i \lambda_3}{q_{ti}^2} - 2(1 + \lambda_1) \rho_i u_i q_{ti}, \forall i$. Let the above equation equal to 0 and we have

$$\frac{\lambda_3}{q_{ti}^2} = 2(1 + \lambda_1) \rho_i q_{ti}, \forall i. \quad (30)$$

Similarly, we have

$$\frac{\lambda_3}{q_{mi}^2} = 2(1 + \lambda_2) \rho_i q_{mi}, \forall i. \quad (31)$$

Then, let's consider equation (29). Assume that $\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T \neq 0, \forall i$, according to (29), we have $\lambda_3 = 0$. From (30), we can see that if $\lambda_3 = 0$, this equation will be $2(1 + \lambda_1) \rho_i q_{ti} = 0, \forall i$, then we have $\lambda_1 = -1 < 0$. Since λ_1 is constrained by (23), it should always be non-negative. Therefore, this assumption is invalid. We can obtain the same conclusion from (31) as well. So we can conclude that for any i , equation $\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0$ is always satisfied. In this way, $\lambda_3 > 0$ can be deduced.

Based on the KKT conditions and $\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T \neq 0, \forall i$, we can analyze the optimal solutions of **Problem 1** as follows:

Case 1: $\lambda_1 = \lambda_2 = 0, \frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0, \forall i$.

In this case, since $\lambda_1 = \lambda_2 = 0$, we can derive $q_{ti} = q_{mi} = \sqrt[3]{\frac{\lambda_3}{2\rho_i}} \geq 0$ using (30) and (31), respectively. But λ_3 is a non-negative parameter, and it is not a constant value, so we still can not get the optimal solutions of **Problem 1**. Thus, this case is not suitable.

Case 2: $\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti} = \rho_i \psi q_{mi}^2 - \frac{\psi}{q_{mi}} p_{mi} = 0, \forall i, \frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0, \forall i$.

By solving $\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti} = 0, \forall i$ and $\rho_i \psi q_{mi}^2 - \frac{\psi}{q_{mi}} p_{mi} = 0, \forall i$, we have $q_{ti} = \sqrt[3]{\frac{p_{ti}}{\rho_i}}, \forall i$ and $q_{mi} = \sqrt[3]{\frac{p_{mi}}{\rho_i}}, \forall i$.

Since $\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0, \forall i$, even though the above two functions can give the expression of the solution of **Problem 1**, it is still constrained by this function. In other words, the one of the KKT conditions, i.e., (29), is not satisfied. Thus, this case is not suitable.

Case 3: $\left(\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti} \right) = 0, \lambda_2 = 0, \frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0, \forall i$.

From (29), we can get the relationship between q_{ti} and q_{mi} is $q_{mi} = \frac{\psi}{T - \frac{\mu_i}{q_{ti}}}$. Solving $(\rho_i \mu_i q_{ti}^2 - \frac{\mu_i}{q_{ti}} p_{ti}) = 0$ yields $q_{ti} = \sqrt[3]{\frac{p_{ti}}{\rho_i}}$, $\forall i$. Based on $q_{mi} = \frac{\psi}{T - \frac{\mu_i}{q_{ti}}}$, we let $q_{ti} = \sqrt[3]{\frac{p_{ti}}{\rho_i}}$, $\forall i$, then we can derive that $q_m(t) = \frac{\psi}{T - \frac{\mu_i}{\sqrt[3]{\frac{p_{ti}}{\rho_i}}}}$, $\forall i$. From (30) and (31), we have $\lambda_1 = \frac{\lambda_3}{2\rho_i q_{ti}^3} - 1$ and $\lambda_2 = \frac{\lambda_3}{2\rho_i q_{mi}^3} - 1$. Since λ_3 , q_{ti} , q_{mi} and ρ_i are positive, so $\lambda_1 = \frac{\lambda_3}{2\rho_i q_{ti}^3} > 0$, and λ_3 can be large enough to make sure $\lambda_1 = \frac{\lambda_3}{2\rho_i q_{ti}^3} \geq 1$, thus $\lambda_1 \geq 0$ can be guaranteed. Similarly, $\lambda_2 \geq 0$ can be derived. From the above analysis, **Case 3** satisfies all the KKT conditions, therefore the optimal solutions are obtained.

Case 4: $(\rho_i \psi q_{mi}^2 - \frac{\psi}{q_{mi}} p_{mi}) = 0$, $\forall i$, $\lambda_1 = 0$, $\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} - T = 0$, $\forall i$. This case is similar to **Case 3**.

Based on the above analysis, the optimal solutions of **Problem 1** are $q_{ti}^* = \left(\frac{p_{ti}}{\rho_i}\right)^{\frac{1}{3}}$, $\forall i$, and $q_{mi}^* = \frac{\psi}{T - \mu_i \left(\frac{p_{ti}}{\rho_i}\right)^{\frac{1}{3}}}$.

Thus **Theorem 3.1** is proved.

APPENDIX B PROOF OF THEOREM 3.2

The Lagrangian correlated to Problem 2 is

$$\mathcal{L}_2 = f\left(\sum_{i=1}^N \mu_i\right) - \xi \sum_{i=1}^N \left(\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} + \frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi}\right) - \theta_1 \left(\frac{\mu_i}{q_{ti}} p_{ti} - \bar{R}_{ti}\right) - \theta_2 \left(\frac{\psi}{q_{mi}} p_{mi} - \bar{R}_{mi}\right), \quad (32)$$

where θ_1 and θ_2 are the Lagrange multipliers correlated to the constraints of Problem2. The following constrains should be met:

$$\frac{\partial \mathcal{L}_2}{\partial p_{ti}} = \frac{\partial \mathcal{L}_2}{\partial p_{mi}} = 0, \forall i, \quad (33)$$

$$\theta_1 \geq 0, \theta_2 \geq 0, \forall i, \quad (34)$$

$$\theta_1 \left(\frac{\mu_i}{q_{ti}} p_{ti} - \bar{R}_{ti}\right) = 0, \forall i, \quad (35)$$

$$\theta \left(\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \omega\right) = 0, \forall i, \quad (36)$$

$$\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} \leq \omega, \forall i. \quad (37)$$

$$\frac{\mu_i}{q_{ti}} p_{ti} \leq \bar{R}_{ti}, \forall i, \quad (38)$$

$$\frac{\psi}{q_{mi}} p_{mi} \leq \bar{R}_{mi}, \forall i. \quad (39)$$

First, let $q_{ti} = q_{ti}^*$ and $q_{mi} = q_{mi}^*$.

Case 1: $\theta_1 = 0$, $\frac{\psi}{q_{mi}} p_{mi} - \bar{R}_{mi} = 0$, $\forall i$.

In this case, we can have $\frac{\partial \mathcal{L}_2}{\partial p_{ti}} = \frac{-(\mu_i(\xi p_{mi} + 2p_{ti}) + \theta_2 p_{mi})}{3p_{ti} \sqrt[3]{\frac{p_{ti}}{\rho_i}}}$, $\forall i$. Setting this equation equal to 0 yields $p_{ti} = \frac{-p_{mi}(\xi + \theta_2)}{2\xi}$. Obviously, we cannot find a positive θ_2 to satisfy this equation, making this solution invalid.

Case 2: $\theta_2 = 0$, $\frac{\psi}{q_{ti}} p_{ti} - \bar{R}_{ti} = 0$, $\forall i$.

This case is similar to Case 1.

Case 3: $\frac{\psi}{q_{ti}} p_{ti} - \bar{R}_{ti} = 0$, $\frac{\psi}{q_{mi}} p_{mi} - \bar{R}_{mi} = 0$, $\forall i$.

By solving $\frac{\psi}{q_{ti}} p_{ti} - \bar{R}_{ti} = 0$, $\frac{\psi}{q_{mi}} p_{mi} - \bar{R}_{mi} = 0$, $\forall i$, we can get (15) and (16). We can also prove that this case satisfy the rest of the KKT conditions.

Thus, **Theorem 3.2** is proved.

APPENDIX C PROOF OF THEOREM 4.1

Then, we will provide the solution of Problem 3. The Lagrangian of Problem 3 can be written as

$$\begin{aligned} \mathcal{L}_3 = & f\left(\sum_{i=1}^N \mu_i\right) - \xi \sum_{i=1}^N \left(\frac{\mu_i}{q_{ti}} + \frac{\psi}{q_{mi}} + \frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi}\right) \\ & - \alpha_1 \left(\frac{\mu_i}{q_{ti}} p_{ti} - \bar{R}_{ti}\right) - \alpha_2 \left(\frac{\psi}{q_{mi}} p_{mi} - \bar{R}_{mi}\right) \\ & - \alpha_3 \left(\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi q_{mi}^2\right). \end{aligned} \quad (40)$$

where α_1 , α_2 and α_3 are the Lagrange multipliers. The KKT conditions are similar with Problem 2 except the following three conditions:

$$\alpha_3 \geq 0, \forall i, \quad (41)$$

$$\alpha_3 \left(\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi q_{mi}^2\right) = 0, \forall i, \quad (42)$$

$$\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi q_{mi}^2 \geq 0, \forall i. \quad (43)$$

We then analyze the solutions under different cases. Actually, there should be nine cases in this problem, but we only give two of them to analyze since the other situation can be interpreted similarly.

Case1 : $\alpha_1 = \alpha_2 = \alpha_3 = 0$, $\forall i$.

In this case, we can have $\frac{\partial \mathcal{L}_3}{\partial p_{ti}} = \frac{-(\mu_i \xi (p_{mi} + 2p_{ti}))}{3\rho_i \sqrt[3]{\frac{p_{ti}}{\rho_i}}}$, and let

it equal to 0 we can get $p_{ti} = \frac{-p_{mi}}{2}$. Obviously, since p_{ti} and p_{mi} are non-negative values, we cannot find a p_{mi} to satisfy the above equation. So this case is invalid.

Case2 : $\frac{\mu_i}{q_{ti}} p_{ti} + \frac{\psi}{q_{mi}} p_{mi} - \rho_i \mu_i q_{ti}^2 - \rho_i \psi q_{mi}^2 = 0$, $\frac{\psi}{q_{ti}} p_{ti} - \bar{R}_{ti} = 0$, $\alpha_2 = 0$, $\forall i$. By solving the above equations, we get (19) and (20). We can verify that the solutions above is incentive compatible and satisfy all the KKT conditions.

Thus **Theorem 4.1** is proved.

REFERENCES

- [1] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system," 2020.
- [2] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2020.
- [3] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.
- [4] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.
- [5] G. Hua, L. Zhu, J. Wu, C. Shen, L. Zhou, and Q. Lin, "Blockchain-based federated learning for intelligent control in heavy haul railway," *IEEE Access*, vol. 8, pp. 176 830–176 839, 2020.

- [6] J. Passerat-Palmbach, T. Farnan, R. Miller, M. S. Gross, H. L. Flannery, and B. Gleim, "A blockchain-orchestrated federated learning architecture for healthcare consortia," *arXiv preprint arXiv:1910.12603*, 2019.
- [7] S. Aich, N. K. Sinai, S. Kumar, M. Ali, Y. R. Choi, M.-I. Joo, and H.-C. Kim, "Protecting personal healthcare record using blockchain & federated learning technologies," in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2021, pp. 109–112.
- [8] R. Kumar, A. A. Khan, J. Kumar, A. Zakria, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, and W. Wang, "Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging," *IEEE Sensors Journal*, 2021.
- [9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.
- [11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.
- [12] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.
- [13] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghanian, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.
- [14] P. Ramanan and K. Nakayama, "Baffle: Blockchain based aggregator free federated learning," in *2020 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2020, pp. 72–81.
- [15] Y. J. Kim and C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2019, pp. 1–4.
- [16] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, "Fedcoin: A peer-to-peer payment system for federated learning," in *Federated Learning*. Springer, 2020, pp. 125–138.
- [17] J. Li, Y. Shao, K. Wei, M. Ding, C. Ma, L. Shi, Z. Han, and H. V. Poor, "Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation," pp. 1–12, 2021. [Online]. Available: <http://arxiv.org/abs/2101.06905>
- [18] N. Q. Hieu, T. T. Anh, N. C. Luong, D. I. Niyato, D. I. Kim, and E. Elmroth, "Resource Management for Blockchain-enabled Federated Learning: A Deep Reinforcement Learning Approach," 2020. [Online]. Available: <http://arxiv.org/abs/2004.04104>
- [19] X. Qu, Q. Hu, and S. Wang, "Privacy-preserving model training architecture for intelligent edge computing," *Computer Communications*, vol. 162, pp. 94–101, 2020.
- [20] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," *arXiv preprint arXiv:2110.02182*, 2021.
- [21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.
- [22] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.
- [23] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.
- [24] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2, pp. 203–221, 1996.
- [25] J. Zhang and Q. Zhang, "Stackelberg game for utility-based cooperative cognitiveradio networks," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, 2009, pp. 23–32.
- [26] Z. Peng, J. Xu, X. Chu, S. Gao, Y. Yao, R. Gu, and Y. Tang, "Vfchain: Enabling verifiable and auditable federated learning via blockchain systems," *IEEE Transactions on Network Science and Engineering*, 2021.
- [27] H. B. Desai, M. S. Ozdayi, and M. Kantarciooglu, "Blockfla: Accountable federated learning via hybrid blockchain architecture," in *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, 2021, pp. 101–112.
- [28] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.
- [29] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-device federated learning via blockchain and its latency analysis," *arXiv preprint arXiv:1808.03949*, 2018.
- [30] Q. Hu, Z. Wang, M. Xu, and X. Cheng, "Blockchain and federated edge learning for privacy-preserving mobile crowdsensing," *IEEE Internet of Things Journal*, 2021.
- [31] K. Toyoda and A. N. Zhang, "Mechanism design for an incentive-aware blockchain-enabled federated learning platform," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 395–403.
- [32] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*. IEEE, 2019, pp. 151–159.
- [33] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10700–10714, 2019.
- [34] Y. Liu, S. Sun, Z. Ai, S. Zhang, Z. Liu, and H. Yu, "FedCoin: A Peer-to-Peer Payment System for Federated Learning," 2020. [Online]. Available: <http://arxiv.org/abs/2002.11711>



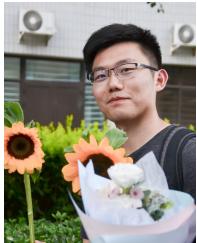
Zhilin Wang received his B.S. from Nanchang University in 2020. He is currently pursuing his Ph.D. degree of Computer and Information Science In Indiana University-Purdue University Indianapolis (IUPUI). He is a Research Assistant with IUPUI, and he is also a reviewer of 2022 IEEE International Conference on Communications (ICC). His research interests include blockchain, federated learning, edge computing, and Internet of Things (IoT).



Qin Hu received her Ph.D. degree in Computer Science from the George Washington University in 2019. She is currently an Assistant Professor with the Department of Computer and Information Science, Indiana University-Purdue University Indianapolis (IUPUI). She has served on the Editorial Board of two journals, the Guest Editor for two journals, the TPC/Publicity Co-chair for several workshops, and the TPC Member for several international conferences. Her research interests include wireless and mobile security, edge computing, blockchain, and crowdsensing.



Ruinian Li received the PhD degree in computer science from the George Washington University in 2018. He is currently an assistant professor at the Department of Computer Science, Bowling Green State University (BGSU), USA. His research interests include security and privacy-preserving computations, applied cryptography, and blockchain technology. He has been working in a wide area of social networks, auction systems, and IoT, and his work has been published in top-tier journals, such as IEEE Transactions on Services Computing, and IEEE Transactions on Network Science and Engineering.



Minghui Xu received the BS degree in Physics from the Beijing Normal University, Beijing, China, in 2018, and the PhD degree in Computer Science from The George Washington University, Washington DC, USA, in 2021. He is currently an Assistant Professor in the School of Computer Science and Technology, Shandong University, China. His current research focuses on blockchain, distributed computing, and quantum computing.



Zehui Xiong is currently an Assistant Professor in the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design. Prior to that, he was a researcher with Alibaba-NTU Joint Research Institute, Singapore. He received the PhD degree in Nanyang Technological University, Singapore. He was the visiting scholar at Princeton University and University of Waterloo. His research interests include wireless communications, network games and economics, blockchain, and

edge intelligence. He has published more than 140 research papers in leading journals and flagship conferences and many of them are ESI Highly Cited Papers. He has won over 10 Best Paper Awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He is now serving as the editor or guest editor for many leading journals including IEEE JSAC, TVT, IoTJ, TCCN, TNSE, ISJ, JAS. He is the recipient of IEEE TCSC Early Career Researcher Award for Excellence in Scalable Computing, IEEE CSIM Technical Committee Best Journal Paper Award, IEEE SPCC Technical Committee Best Paper Award, IEEE VTS Singapore Best Paper Award, Chinese Government Award for Outstanding Students Abroad, and NTU SCSE Best PhD Thesis Runner-Up Award. He is the Founding Vice Chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee.