Incentive Mechanism for Rational Miners in Bitcoin Mining Pool

Gang Xue, Jia Xu*, Hanwen Wu, Weifeng Lu, and Lijie Xu

Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China
* Corresponding author
xujia@njupt.edu.cn

Abstract
Bitcoin is the most popular cryptocurrency in the world. Miners in the Bitcoin network reduce their risks through participating in mining pool. Existing mining pool systems do not consider the cost and strategy of miners. In this paper, we study two mining models: public cost model and private cost model. For the public cost model, we design an incentive mechanism, called *Mining* game, using a *Stackelberg* game. We show that *Mining* game is individually rational, profitable, and has the unique *Stackelberg Equilibrium*. For the private cost model, we formulate the *Budget Feasible Reward Optimization* (*BFRO*) problem to maximize the reward function under the budget constraint, and design a budget feasible reverse auction to solve the *BFRO* problem, which is computationally efficient, individually rational, truthful, budget feasible, and constant approximate. Through extensive simulations, we evaluate the performance and validate the theoretical properties of our incentive mechanisms.

Keywords Bitcoin, Mining pool, Incentive mechanism, Nash Equilibrium, Auction

1 Introduction

Bitcoin is the first decentralized digital currency over the world. It relies on a network of computers that synchronize transactions with a process called mining to find valid blocks. In this way, miners repeatedly compute hashes until one finds a numerical value, which is low enough, and thus get the reward from the block.

Finding a Bitcoin block is very profitable (at least B12.5, more than $110,000 today), but it is also very difficult for small miners who might find a block in expectation every a few months or even a few years. As a result, such small miners will participate in the mining pool to achieve large computing power in total and share the reward from blocks within the pool in order to receive a low but steady stream of income.

This is an extended and enhanced version of the paper [15] that appeared in the 2nd International Conference on Science of Cyber Security, SciSec2019

Incentive mechanisms are important for many human-involved cooperative systems, such as computation offloading [1], spectrum access [2], and crowdsourcing [3, 4]. Some research efforts have been focused on designing incentive mechanisms to entice miners to participate in mining pools. Rosenfeld *et al.* described the various scoring systems to calculate rewards of participants in Bitcoin pooled mining [5]. Schrijvers *et al.* introduced a game-theoretic model for reward function in Bitcoin mining pools [6]. Lewenberg *et al.* examined dynamics of pooled mining and the rewards that pools manage to collect, and use cooperative game theory to analyze how the pool members share these rewards [7]. However, none of them considers the cost of each miner.

We focus on designing incentive mechanisms for the rational miners with different cost. For example, people living in areas with higher electricity bill will have higher mining cost than others. Their mining strategies must be influenced by their cost. To address this issue, we design incentive mechanisms to motivate the rational miners to participate in the mining pool.

Our first incentive mechanism follows *Stackelberg* game, where the pool platform has the absolute control over the total payment to the miners affiliated, and miners can only determine the mining strategies based on the total payment decided by mining pool platform. The first incentive mechanism needs to know the cost of miners in advance in order to decide the total payment.

Our second incentive mechanism considers that the cost of miners is private information, and use budget feasible reverse auction to model the mining process. In the auction, each miner submits a bid including hash quantity and reserve price to the pool platform. Then the platform selects winners from all bidders, and decides the payment to them.

The main contributions of this paper are as follows:

- We present two models for pool mining system with rational miners: public cost model and private cost model.
- We model the mining process in public cost model as *Stackelberg* game, called *Mining* game in this paper. We show that Mining game is individually rational, profitable, and has unique *Stackelberg Equilibrium*.
- For the private cost model, we formulate the *Budget Feasible Reward Optimization* (*BFRO*) problem to maximize the reward function under the budget constraint. We design a budget feasible reverse auction to solve the *BFRO* problem based on *Proportional Share Allocation Rule* [14], which is computationally efficient, individually rational, truthful, budget feasible, and constant approximate.

The rest of the paper is organized as follows. Section 2 formulates the system models, and lists some desirable properties. Section 3 presents the detailed design of our incentive mechanism for the Bitcoin mining pool in the public cost model. Section 4 presents the detailed design of budget feasible reverse auction for the Bitcoin mining pool in the private cost model. Performance evaluation is presented in Section 5. We review the state-of-art research in Section 6, and conclude this paper in Section 7.

2 System Model and Problem Formulation

In this section, we provide two different models for mining pool: public cost model and private cost model. In public cost model, the platform knows the cost of miners and can decide the total payment to the miners by estimating the strategies of miners. In private cost model, we consider that the cost is the private information and known only to miner itself. The miners report their declared cost through reverse auction. This means the platform In order to make the miners remunerative, the platform loses the absolute control over the total payment to miners in fact. Since the reward of one valid block is fixed [20], we set a fixed budget for the total payment in this model.

Table 1 lists frequently used notations.

Table 1. Frequently used notations

Notation	Description
$\boldsymbol{M}$	miners set
$\boldsymbol{n}$	number of miners
$\boldsymbol{u_0}, \boldsymbol{u_i}$	utility of platform, utility of miner i
$\boldsymbol{P}, \boldsymbol{p_i}$	total payment, payment to miner i
$\boldsymbol{R}$	reward of a valid block
$\boldsymbol{h_i}, \boldsymbol{h}, \boldsymbol{h_{-i}}$	strategy (hash quantity) of miner i, strategy profile of all miners, strategy profile excluding i's strategy
$\boldsymbol{k_i}, \boldsymbol{b_i}$	unit cost of miner i, reserve price of miner i
$\boldsymbol{\beta_i}(\boldsymbol{h_{-i}})$	best response of miner i given h_{-i}
$\boldsymbol{u_i}, \boldsymbol{u_0}$	utility of miner i, utility of the platform
$\boldsymbol{A}$	total hash power in bitcoin network
$\boldsymbol{D}$	difficulty to find a valid block
$\boldsymbol{V}(\boldsymbol{S})$	reward function when the miners in set S is chosen
$\boldsymbol{V_i}(\boldsymbol{S})$	marginal value of miner i when the miners in set S is chosen
$\boldsymbol{B}$	budget of the platform

2.1 Public Cost Model

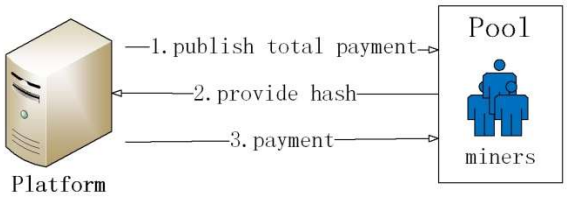

Fig. 1. Public cost mining pool system

We use Fig. 1 to illustrate public cost mining pool system. The system consists of a mining pool platform and a mining pool which is made up of many rational miners. Assume that there is a set $M = \{1, 2, \ldots, n\}$ of miners who participate in the mining

pool, where $n \geq 2$. The Miners provide hash quantity by consuming their computing power with different unit cost. Therefore, these rational miners expect the payment to compensate for their cost.

First, the platform publicizes a total payment to the miners. Taking the mining cost into consideration, each miner makes its own mining strategy (hash quantity) according to the total payment, and then submits the hash to the platform. After collecting the hashes from miners, the platform sends the payment to the miners. If the mining pool finds a valid block through the integrated efforts of the miners in the pool, the platform will receive the reward for the block. On the other side, if any other pool finds a valid block, the platform won't get the reward. Overall, the platform absorbs all the risks for the miners in the pool. This is the whole mining process.

The platform is only interested in maximizing its own utility. Since computing power is owned by different individuals, it is reasonable to assume that miners are selfish and rational. Hence each miner only wants to maximize its own utility and will not participate in mining pool unless there is sufficient incentive. The objective is designing an incentive mechanism, which is simple, scalable, and has provably properties. In this model, the mining strategy of a miner is in the form of its hash quantity. A miner participating in mining pool will earn a payment that is no lower than its cost. However, it needs to compete with other miners under a fixed total payment.

For mining a block, the platform announces a total payment $P > 0$, motivating miners to participate in the mining pool. Each miner decides its mining strategy based on the payment. The mining strategy of any miner $i \in M$ is represented by h_i, $h_i \geq 0$, the hash quantity he is willing to provide. Specifically, if $h_i = 0$, miner i indicates that he will not participate in the mining pool. The mining cost of miner i is $k_i h_i$, where $k_i > 0$ is its unit cost. Assume that the payment received by miner i is proportional to h_i. Then the utility of miner i can be defined as the difference between payment and cost:

$$u_i = \frac{h_i}{\sum_{j \in M} h_j} P - h_i k_i. \tag{2.1}$$

For the reason that the mining process is subject to Poisson process [1], we can get the utility of the platform in expectation:

$$u_0 = \frac{\sum_{j \in M} h_j}{A + \sum_{j \in M} h_j} R - P, \tag{2.2}$$

where $A = \frac{D \times 2^{32}}{T}$ is the total hash power in Bitcoin network. We can estimate A based on the difficulty D of finding a valid block. The value of D is adjusted periodically by the Bitcoin network to make sure the blocks are generated every $T = 600$ seconds averagely. We suppose that A is a constant because it is almost stable for two weeks (the approximate period when Bitcoin network adjusts the value of D). The probability of finding a valid block is proportional to its total computing power of the pool in the whole network. R is the reward the platform may obtain if it finds a valid block.

Under this model, the objective of the platform is to decide the optimal value of P so as to maximize (2.2), while each miner $i \in M$ decides its hash quantity h_i to max-

imize (2.1) for the given value of P. Since no rational miner is willing to mine with negative utility, miner i will set $h_i = 0$ when $P \leq k_i \sum_{j \neq i, j \in M} h_j$.

2.2 Private Cost Model

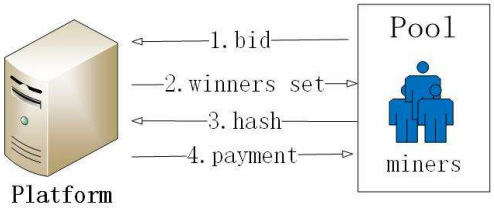

Fig. 2. Private cost mining pool system

We use Fig. 2 to illustrate the private cost mining pool system, which follows the reverse auction framework. First, each miner submits a bid $E_i = (h_i, b_i)$ to the pool platform according to its own hash power and unit cost k_i. b_i is the reserve price miner i wants to sell its hash of h_i. We consider the unit cost k_i is the private information and known only to miner i. After receiving all the bids from miners, the platform selects a subset S of miners as winners. Then each winner $i \in S$ provides hash of h_i to the platform. Finally, the platform determines and sends the payment p_i to each winning miner i. Due to the limited reward of finding a valid block, the pool platform is with a budget B to restrict the total payment. As the same in the public cost model, if the mining pool finds a valid block through the integrated efforts of the miners in the pool, the platform will receive the reward for the block.

We define the utility of any miner i as the difference between the payment and its real cost:

$$u_i = \begin{cases} p_i - h_i k_i, & i \in S \\ 0, & otherwise \end{cases}. \quad (2.3)$$

The reward function of finding a valid block in exception is defined as:

$$V(S) = R \frac{\sum_{i \in S} h_i}{\sum_{i \in S} h_i + A}. \quad (2.4)$$

Then the utility of platform can be defined as:

$$u_0 = v(S) - \sum_{i \in S} p_i. \quad (2.5)$$

Since we consider the miners are selfish and rational individuals, each miner can behave strategically by submitting a dishonest bid price to maximize its utility. One of our targets is making miner i get its maximal utility when he submits $b_i = k_i$ under situation that all miners are doing the same strategy as miner i.

The objective of the reverse auction is maximizing the reward function such that the total payment is not more than the budget. We refer to this problem as the *Budget Feasible Reward Optimization* (*BFRO*) problem, which can be formulated as follows:

$$\begin{aligned} \textbf{Objective}: &\quad \text{Maximize } V(S) \\ \textbf{Subject to}: &\quad \textstyle\sum_{i \in S} p_i \leq B \end{aligned}. \quad (2.6)$$

2.3 Desirable Properties

Our objective is to design incentive mechanisms for mining pool satisfying the following desirable properties:

- **Computational Efficiency:** A mechanism is computationally efficient if the outcome can be computed in polynomial time.
- **Individual Rationality:** Each winning miner will have a nonnegative utility while bidding its true cost, i.e., $u_i \geq 0, \forall i \in S$.
- **Profitability:** The platform should not incur a deficit. In other words, the value brought by the miners should be at least as large as the total payment to the miners, i.e., $u_0 \geq 0$. Note that profitability here is profitability in expectation because of the randomness of Bitcoin mining.
- **Truthfulness:** A mechanism is truthful if no miner can improve its utility by submitting a reserve price different from its true cost, no matter what other submit.
- **Budget Feasibility:** The total payments to the winners are no more than the budget, i.e., $\sum_{i \in S} p_i \leq B$.
- **Approximation:** We attempt to find solution with the maximum of reward function given in (2.4) only using polynomial-time algorithm. For $\chi \geq 1$, we say the incentive mechanism is α-approximate if the mechanism selects a winner set Π, such that $OPT \leq \chi V(\Pi)$, where OPT is the optimal solution of *BFRO* problem.

3 Incentive Mechanism for Public Cost Model

We model this mining process of public cost model as *Stackelberg* game [8], which can be called *Mining* game. There are two phases in *Mining Game*: In the first phase (called payment determination), the platform announces its payment *P*; in the second phase (called hash determination), each miner strategizes its mining plan to maximize its own utility. Therefore, the platform is the leader and the miners in the mining pool are the followers in our *Mining* game. The strategy of the platform is its payment *P*. The strategy of any miner i is its hash quantity h_i. Let $\boldsymbol{h} = (h_1, h_2, \ldots, h_n)$ denote the strategy profile of all miners. Let h_{-i} denote the strategy profile excluding h_i. For convenience, we write $\boldsymbol{h} = (h_i, h_{-i})$.

Note that the second process of *Mining Game* itself can be considered as a non-cooperative game, called the *Hash Determination* (*HD*) game. We introduce the following definitions:

Definition 1 (Nash Equilibrium, NE). *A set of strategies* $(h_1^{ne}, h_2^{ne}, \ldots, h_n^{ne})$ *is a Nash Equilibrium of the HD game if for any miner i,*

$$u_i(h_i^{ne}, h_{-i}^{ne}) \geq u_i(h_i, h_{-i}^{ne})$$

for any $h_i \geq 0$*, where* u_i *is defined in (2.1).*

Definition 2 (Subgame Perfect Nash equilibrium). *The Stackelberg game can be solved by finding the Subgame Perfect Nash Equilibrium (SPNE), i.e., given the strategies of the other players, the strategy profile serves best for each player, and entails every player playing in a Nash Equilibrium in every subgame.*

The existence of *NE* is important since the *NE* strategy profile is stable (no player has the incentive to make a unilateral change). The uniqueness of *NE* enables the platform to predict the behaviors of the miners, and thus enables the platform to select the optimal value of payment.

In Section 3.1, we will prove that for any given $P > 0$, the *HD* game has a unique *NE*, and present an algorithm for computing the *NE*. In Section 3.2, we will prove that the *Mining* game has a unique *SPNE*.

3.1 Hash Determination

We first introduce the concept of best response strategy.

Definition 3 (Best Response Strategy). *Given h_{-i}, the strategy is miner i's best response strategy, denoted by $\beta_i(h_{-i})$, if it maximizes $u_i(h_i, h_{-i})$ over all $h_i \geq 0$.*

Based on the definition of *NE*, every player is playing its best response strategy in a *NE*. From (2.1), we know that $h_i \leq \frac{P}{k_i}$ because u_i will be negative otherwise. To study the best response strategy of miner i, we compute the derivatives of u_i with respect to h_i:

$$\frac{\partial u_i}{\partial h_i} = \frac{1}{\sum_{j\in M} h_j} P - \frac{h_i}{(\sum_{j\in M} h_j)^2} P - k_i, \tag{3.1}$$

$$\frac{\partial^2 u_i}{\partial h_i^2} = -\frac{2P \sum_{j\in M\setminus\{i\}} h_j}{(\sum_{j\in M} h_j)^3} < 0. \tag{3.2}$$

Since the second-order derivative of u_i is negative, the utility u_i is a strictly concave function with h_i. Therefore, given any $P > 0$ and any strategy profile h_{-i} of the other miners, the best response strategy $\beta_i(h_{-i})$ of miner *i* is unique, if it exists. If the strategy of every other miner $j \neq i$ is $h_j = 0$, then miner *i* does not have a best response strategy, as it can have a utility arbitrarily close to *P*, by setting h_i to a sufficiently small positive number. Therefore, we are only interested in the best response for miner *i* when $\sum_{j\in M\setminus\{i\}} h_j > 0$. Setting the first derivative of u_i to 0, we have

$$\frac{1}{\sum_{j\in M} h_j} P - \frac{h_i}{(\sum_{j\in M} h_j)^2} P - k_i = 0. \tag{3.3}$$

Solving for h_i in (3.3), we obtain

$$h_i = \sqrt{\frac{P \sum_{j\in M\setminus\{i\}} h_j}{k_i}} - \sum_{j\in M\setminus\{i\}} h_j. \tag{3.4}$$

Remark: h_i is the total hash that can make *i* achieve maximum utility in the current mining pool. Of course, *i* can put the remaining hash power to any other pools.

If the right-hand side of (3.4) is positive, it is also the best response strategy of miner *i*, due to the concavity of u_i. If the right-hand side of (3.4) is less than or equal to 0, then miner *i* does not participate in the mining task by setting $h_i = 0$ (to avoid a deficit). Hence we have

$$\beta(h_i) = \begin{cases} 0, & if P \leq k_i \sum_{j\neq i \cap j\epsilon} h_j \\ \sqrt{\frac{P \sum_{j\in M\setminus\{i\}} h_j}{k_i}} - \sum_{j\in M\setminus\{i\}} h_j, & otherwise \end{cases} \tag{3.5}$$

These analyses lead to the following algorithm for computing an *NE* of the *HD* game.

Algorithm 1: Computation of the NE

Sort miners according to their unit costs, $k_1 \le k_2 \le \ldots \le k_n$;
$S \leftarrow \{1,2\}$, $i \leftarrow 3$;
while $i \le n$ **and** $k_i < \frac{k_i + \sum_{j\in S} k_j}{|S|}$ **do**
  $S \leftarrow S \cup \{i\}, i \leftarrow i+1$;
end
for each $i \in M$ **do**
  if $i \in S$ **then** $h_i^{ne} = \frac{(|S|-1)P}{\sum_{j\in S} k_j}\left(1 - \frac{(|S|-1)k_i}{\sum_{j\in S} k_j}\right)$;
  else $h_i^{ne} = 0$;
end
return $h^{\mathrm{ne}} = (h_1^{\mathrm{ne}}, h_2^{\mathrm{ne}}, \ldots, h_n^{\mathrm{ne}})$;

Theorem 1. *The strategy profile* $h^{ne} = (h_1^{ne}, h_2^{ne}, \ldots, h_n^{ne})$ *computed by Algorithm 1 is a NE of the HD game.*

PROOF: We first prove that the strategy profile h^{ne} is a *NE*. From Algorithm 1, we the following observations:

1) for $i \notin S$, $k_i \ge \frac{\sum_{j\in S} k_j}{n_0 - 1}$;

2) $\sum_{j\in S} h_j^{ne} = \frac{(|S|-1)P}{\sum_{j\in S} k_j}$;

3) for $i \in S$, $\sum_{j\in S\setminus\{i\}} h_j^{ne} = \frac{(|S|-1)^2 P k_i}{\left(\sum_{j\in S} k_j\right)^2}$

For computing *NE*, there are two cases:

①For $i \notin S$: It is obvious that $k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} = k_i \sum_{j\in S} h_j^{ne}$. Using 1) and 2), we get $k_i \sum_{j\in S} h_j^{ne} \ge P$. According to (3.5), we have $\beta(h_{-i}^{ne}) = 0$. So, it is the best response strategy given h_{-i}^{ne} for $\notin S$.

② For $i \in S$: From the Line 3 of Algorithm 1, we get $(i-1)\,k_i < \sum_{j=1}^{i} k_j$. Then we have

$(|S|-1)k_i = (i-1)k_i + (|S|-i)k_i < \sum_{j=1}^{i} k_j + \sum_{j=i+1}^{n} k_j = \sum_{j=1}^{n} k_j$.

Thus, $k_i < \frac{\sum_{j=1}^{n} k_j}{|S|-1}$.

Furthermore, using 3) we have

$$k_i \sum_{j\in M\setminus\{i\}} h_j^{ne} = k_i \sum_{j\in S\setminus\{i\}} h_j^{ne} = k_i \frac{(|S|-1)^2 P k_i}{\left(\sum_{j\in S} k_j\right)^2} = P \frac{(|S|-1)^2 k_i^2}{\left(\sum_{j\in S} k_j\right)^2}$$

$$< P\frac{(|S|-1)^2\left(\frac{\sum_{j\in S}k_j}{|S|-1}\right)^2}{\left(\sum_{j\in S}k_j\right)^2} = P.$$

Thus, $k_i < \frac{P}{\sum_{j\in M\setminus\{i\}}h_j^{ne}}$. According to (3.5), we have

$$\beta(h_{-i}^{ne}) = \sqrt{\frac{P\sum_{j\in M\setminus\{i\}}h_j^{ne}}{k_i}} - \sum_{j\in M\setminus\{i\}}h_j^{ne} = \frac{(|S|-1)P}{\sum_{j\in S}h_j} - \frac{(|S|-1)^2Ph_i}{\left(\sum_{j\in S}h_j\right)^2} = h_i^{ne}$$

In summary of ① and ②, h^{ne} is an *NE* of *HD* game. ■

Theorem 2. *The NE computed by Algorithm 1 is unique.*

PROOF: First, we assume that there exists one miner $i \in M$ whose $h_i' \neq h_i^{ne}$, but it also satisfies $u_i(h_i', h_{-i}^{ne}) \geq u_i(h_i, h_{-i}^{ne})$ for any $h_i > 0$. We consider the following two cases:

①If $i \notin S$, There must have $h_i' > 0$ becasue $h_i' \neq h_i^{ne}$ and h_i^{ne} =0. However, it cannot change the truth that $k_i < \frac{k_i+\sum_{j\in S}k_j}{|S|}$, which means that $k_i\sum_{j\in S}h_j^{ne} \geq P$ (See the proof of Theorem 1). So, its h_i' have to be 0 in order to avoid a deficit. $h_i' = 0$ is contradict with $h_i' > 0$.

②If $i \in S$, reminding that (2.1) is a concave function, and reaches the maximum when $h_i = h_i^{ne}$. So, $u_i(h_i', h_{-i}^{ne}) < u_i(h_i^{ne}, h_{-i}^{ne})$. This contradict with $u_i(h_i', h_{-i}^{ne}) \geq u_i(h_i, h_{-i}^{ne})$ for any $h_i > 0$.

In summary of ① and ②, there is no any miner $i \in M$ whose $h_i' \neq h_i^{ne}$, and it also satisfies $u_i(h_i', h_{-i}^{ne}) \geq u_i(h_i, h_{-i}^{ne})$ for any $h_i > 0$. So, The *NE* in theorem 1 is unique. ■

3.2 Payment Determination

According to the above analysis, the platform, which is the leader in the *Mining* game, knows that there exists a unique *NE* for the miner for any given value of *P*. Hence the platform can maximize its utility by setting the optimal value of *P*. Substituting h^{ne} into (2.2), we have

$$u_0 = \frac{X}{A+X} - P \tag{3.6}$$

where $X = \sum_{j\in S}\frac{(|S|-1)P}{\sum_{j\in S}k_j}(1 - \frac{(|S|-1)k_i}{\sum_{j\in S}k_j})$.

$X' = \frac{\partial X}{\partial P} = \sum_{j\in S}\frac{(|S|-1)}{\sum_{j\in S}k_j}(1 - \frac{(|S|-1)k_i}{\sum_{j\in S}k_j})$. Obviously, X' is a constant. We use Y to represent X'.

Theorem 3. *There exists a unique Stackelberg Equilibrium* (P^*, h^{ne}) *in the Mining game, where* P^* *is the unique value of P to maximize the utility of the platform given in (3.6) over* $P \in [0, \infty)$.

PROOF: We have

$$\frac{\partial u_0}{\partial P} = \frac{AY}{(A+X)^2} - 1, \tag{3.7}$$

$$\frac{\partial^2 u_0}{\partial P^2} = -\frac{2AY^2}{(A+X)^3} < 0. \quad (3.8)$$

Therefore the utility u_0 defined in (3.6) is a strictly concave function of P, for any $P \in [0, \infty)$. Since the value of u_0 in (3.6) is 0 if $P = 0$, and goes to $-\infty$ when P goes to ∞, it has a unique maximum value P^* that can be effectively computed using Newton's method [9]. ■

In the following, we present the analysis, demonstrating that *Mining* game can achieve the desired properties.

Theorem 4. *Mining game is individually rational, profitable, and has unique Stackelberg Equilibrium.*

PROOF: For the individual rationality, any miner i can set $h_i = 0$ to make $u_i = 0$ according to (2.1). Since the miners in *Mining* game always maximize their utilities, we have $u_i \geq 0$. For the profitability, the pool can always set $P = 0$ to get $u_0 = 0$ according to (2.2) (In this case, all miners should set $h_i = 0$ according to (3.5)). Since the platform in *Mining* game always maximizes its utility, we have $u_0 \geq 0$. The uniqueness of *Stackelberg Equilibrium* has been proved in Theorem 3. ■

4 Incentive Mechanism for Private Cost Model

Although the incentive mechanism of payment determinable achieves maximum utility for both platform side and miner side, it cannot be applied to the situation when the pool platform does not know the cost of all miners. To address this problem, we model the mining process as a reverse auction. In the auction, each miner submits a bid including its private cost to the platform first. Since we consider the miners are selfish, the miners may achieve more utility by submitting a false cost. To prevent such strategy behavior, the designed auction should be strategy-proof. Moreover, to avoid the deficit of platform, the designed auction should achieve the property of budget feasibility.

In this section, we use the budget feasible reverse auction to solve the *BFRO* problem defined in (2.6). First, we give the definitions of marginal value and non-decreasing submodular.

Definition 4 (Marginal Value). When a set of miners S have already taken part in the mining pool, the marginal value of a miner i is the increased reward of platform in exception caused by adding i into S. The marginal value of miner i for miner set S is defined as:

$$V_i(S) = V(S \cup \{i\}) - V(S) = R\frac{\sum_{j \in S \cup \{i\}} h_j}{\sum_{j \in S \cup \{i\}} h_j + A} - R\frac{\sum_{j \in S} h_j}{\sum_{j \in S} h_j + A}.$$

Definition 5 (Non-decreasing Submodular). Function $V: 2^{|n|} \to R_+$ is nondecreasing submodular if $V(S \cup \{i\}) - V(S) \geq V(T \cup \{i\}) - V(T)$ and $V(S) \leq V(T)$ for $\forall S \subseteq T$.

Next, we show that our reward function defined in (2.4) is a non-negative non-decreasing submodular function.

Theorem 5. The reward function $V(S)$ is a non-negative non-decreasing submodular function.

PROOF: According to Definition 5, we need to show that $V(S \cup \{i\}) - V(S) \geq V(T \cup \{i\}) - V(T)$, for any $S \subseteq T \subseteq M$ and $i \in M \backslash T$. Considering $V(S) = R\frac{\sum_{i \in S} h_i}{\sum_{i \in S} h_i + A}$, we rewrite $V(S)$ as $V(x) = R\frac{x}{x+A}$. It is obvious to see that $V''(x) < 0$. So $V'(x_0 + \epsilon_1) < V'(x_0)$ for $\forall x_0, \epsilon_1 > 0$, and

$$V(x_0 + \epsilon_1 + \epsilon_2) - V(x_0 + \epsilon_1) < V(x_0 + \epsilon_2) - V(x_0) \text{ for } \forall \epsilon_2 > 0. \quad (3.9)$$

Let $x_s = \sum_{j \in S} h_j$, $x_T = \sum_{j \in T} h_j$, $x_\Delta = \sum_{j \in T \backslash S} h_j$, we have $x_T = \sum_{j \in S} h_j + \sum_{j \in T \backslash S} h_j = x_s + x_\Delta$. Thus, $x_{S \cup \{i\}} = x_s + h_i$ and $x_{T \cup \{i\}} = x_T + h_i = x_s + x_\Delta + h_i$. According to (3.9), we have $V(x_s + x_\Delta + h_i) - V(x_s + x_\Delta) < V(x_s + h_i) - V(x_s)$, i.e., $V(S \cup \{i\}) - V(S) \geq V(T \cup \{i\}) - V(T)$ for $\forall S \subseteq T$.

It is also obvious to see that $f'(x) > 0$, i.e., $f(x_0 + \epsilon_1) > f(x_0)$ for $\forall x_0, \epsilon_1 > 0$. So $f(x_s + x_\Delta) > f(x_s)$, i.e., $V(S) \leq V(T)$ for $\forall S \subseteq T$. The nonnegativity of $V(S)$ is obvious. ■

Since the reward function is a non-negative non-decreasing submodular function. The *BFRO* problem is a budget feasible submodular maximization problem actually. We apply the budget feasible reverse auction proposed by Singer [14], which has been proved to satisfy the computational efficiency, individual rationality, budget feasibility, truthfulness, and constant approximation.

The budget feasible reverse auction consists of winner selection phase and payment determination phase. Illustrated in Algorithm 2, the budget feasible reverse auction selects the winners using a greedy approach, and determines the payment through proportional share allocation rule.

Algorithm 2: Budget Feasible Reverse Auction

Input: user set M, budget B, bid profile $\mathbf{E} = (E_1, E_2, \ldots, E_n)$
Output: winner set S, payment profile $\mathbf{p} = (p_1, p_2, \ldots, p_n)$
// Winner Selection
$S \leftarrow \emptyset$; $i \leftarrow \arg max_{i\prime \in M} \frac{V_{i\prime}(S)}{b_{i\prime}}$;
while $b_i \leq \frac{V_i(S)B}{V(S \cup \{i\})}$ **do**
  $S \leftarrow S \cup \{i\}$;
  $i \leftarrow \arg max_{i\prime \in M \backslash S} \frac{V_{i\prime}(S)}{b_{i\prime}}$;
end while
// Payment Determination
foreach $i \in M$ **do** $p_i \leftarrow 0$;
foreach $i \in S$ **do**
  $M' \leftarrow M \backslash \{i\}$; $S' \leftarrow \emptyset$;
  $i_j \leftarrow \arg max_{i\prime \in M\prime} \frac{V_{i\prime}(S\prime)}{b_{i\prime}}$;
  while $b_{i_j} \leq \frac{V_{i_j}(S\prime)B}{V(S\prime \cup \{i_j\})}$ **do**
    $i_j \leftarrow \arg max_{i\prime \in M\prime \backslash S\prime} \frac{V_{i\prime}(S\prime)}{b_{i\prime}}$;
    $p_i \leftarrow max\{p_i, min\{\frac{V_i(S\prime)B}{V(S' \cup \{i\})}, \frac{V_i(S\prime)b_{i_j}}{V_{i_j}(S\prime)}\}\}$;
    $S' \leftarrow S' \cup \{i_j\}$;
  end while
end for

In the winner selection phase, the miners are sorted according to the unit marginal value, which is defined as $\frac{V_i(S)}{b_i}$ for any miner $i \in M$. Let $S \leftarrow \emptyset$ initially, and in each iteration of the winner selection phase, we select the miner with maximum unit marginal value over the unselected miner set $M \backslash S$ as the winner until $b_i > \frac{V_i(S)B}{V(S \cup \{i\})}$.

In payment determination phase, for each winner $i \in S$, we execute the winner selection phase over $M \backslash \{i\}$, and denote the winner set as S'. For each miner $i_j \in S'$, we compute the maximum value of $min\{\frac{V_i(S')B}{V(S' \cup \{i\})}, \frac{V_i(S')b_{i_j}}{V_{i_j}(S')}\}$ as the payment to each winner $i \in S$.

We can obtain the following theorem according to [14] straightforwardly.

Theorem 6. The incentive mechanism for private cost model is computationally efficient, individually rational, truthful, budget feasible, and constant approximate.

5 Performance Evaluation

5.1 Evaluation of Public Cost Model

Since we only care about the proportion of reward to the miners, the reward of a valid block is inessential to our model. We set the block reward as $R = 100$. The default number of miners in the pool is 100. We assume the unit cost of each miner subjects to two different distributions: normal distribution and uniform distribution with $\mu = 4.0788 \times 10^{-12}$, which can be estimated from the miners in [10].

First, we test variance σ of two distributions to make the evaluation meaningful. Fig. 3 shows that when σ is larger than 10^{-1} under normal distribution, the size of |S| will be small, and it is approximately equal to the minimum value of 2. Note that $|S| \geq 2$ is the condition to make Algorithm 1 effective. Through the similar tests, σ should not larger than 10^{-23} under uniform distribution.

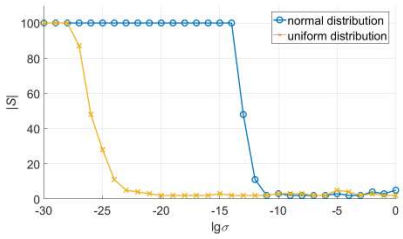

Fig. 3. Number of selected miners with different σ

To explore the influence of σ further, we pick some meaningful value of σ, and measure the hash quantity and utility of miners. Note that the miners are sorted based on unit cost in the nondecreasing order. The results are shown in Fig. 4, Fig. 5, Fig. 6, and Fig. 7.

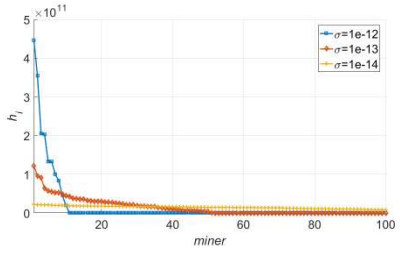

Fig. 4. Hash quantity of each miner under normal distribution

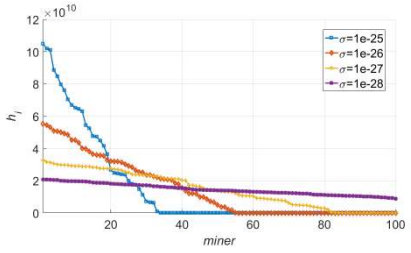

Fig. 5. Hash quantity of each miner under uniform distribution

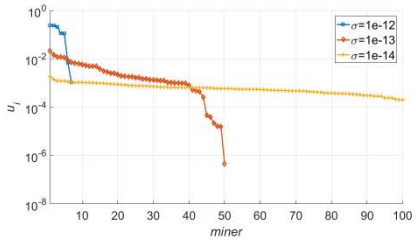

Fig. 6. Utility of each miner under normal distribution (blank areas represent $u_i = 0$)

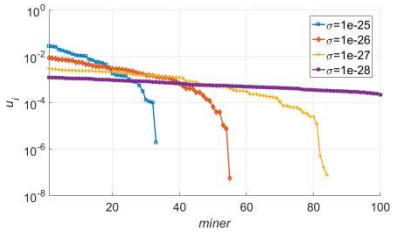

Fig. 7. Utility of each miner under uniform distribution (blank areas represent $u_i = 0$)

First, we can see from Fig. 4 and Fig. 5, the larger σ is, the wider the range of hash quantity is. This is because the strategies in *NE* taken by mines largely depend on their unit cost based on Theorem 1.

Moreover, as shown in Fig. 4 and Fig. 5, the miners with low cost are willing to provide more hash to the pool since the *NE* computed by Algorithm 1 is a decreasing function with the unit cost.

From Fig. 6 and 7, we can see that the miners who contribute more hash will be paid more. This is because the payment received by miner is proportional to its hash quantity provided. Note that each miner is with nonnegative utility in all cases, clarifying the desirable profitability of individual rationality.

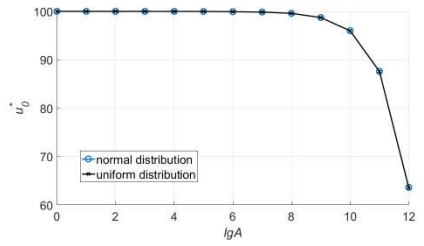

Fig. 8. Utility of platform with different total hash power in Bitcoin network

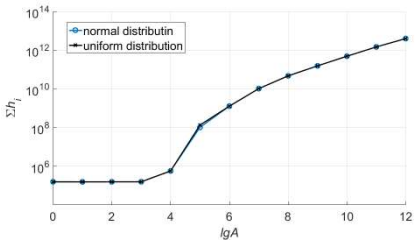

Fig. 9. Total hash quantity in the pool with different total hash power in Bitcoin network

In Fig. 8, we can see that the utility of platform under two distributions are equal exactly no matter how much the σ is (as long as σ is in the meaningful domain). As shown in Fig. 9, the total hash provided under two distributions are also equal exactly.

We can see form Fig. 8 that the platform almost gets the expected utility of 100 when A is low. When A becomes larger, the expected utility of the platform decreases acutely. This is because when A is small, the pool has high probability to win the block and gets the block reward. On the other side, when A is large, the probability to win the block becomes small, and the expected utility of platform will decrease when A goes up.

According to Fig. 9, when A is low, the platform only collects a little hash quantity from miners. When A becomes larger, the platform collects more hash quantity from miners. This is because when A is small, the pool can control the Bitcoin network. This means that the pool only needs to pay a little money to incentive miners to work for it. When A is large, the pool has to pay more money to incentive miners to provide more hash to compete with other pools.

Moreover, we note that the expected utility of the platform is nonnegative in all cases, validating the desirable profitability of profitability.

5.2 Evaluation of Private Cost Model

As same as subsection 5.1, we set the block reward as $R = 100$. The default number of miners in the pool is 100. We assume the unit cost of each miner subjects to normal distribution with $\mu = 4.0788 \times 10^{-12}$ and $\sigma = 1 \times 10^{-11}$. The hash quantity submitted by the miners subjects to normal distribution with $\mu_h = 1 \times 10^{r}$ and $\sigma_h = 1 \times 10^{r+1}$, where r is an adjustable parameter.

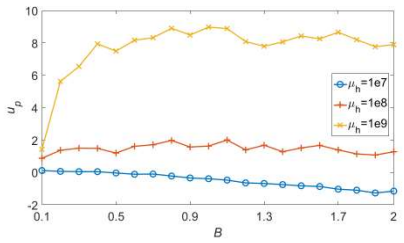

Fig. 10. Utility of platform with different budgets

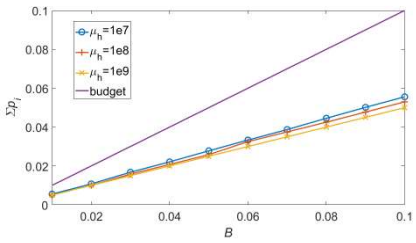

Fig. 11. Total payment with different budgets

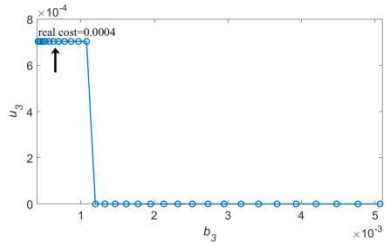

Fig. 12. Truthfulness of miner 3

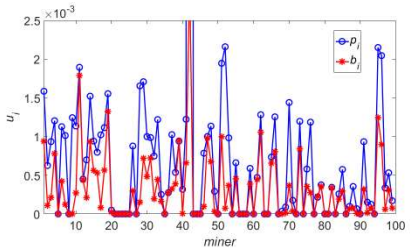

Fig. 13. Utility of each miner

Fig. 10 shows the utility of platform with different budgets. We can see that the platform will obtain more utility when the miners provide more hash quantity. Note that the utility of platform does not always increase with budget. This is because although large budget may make the pool get more hash quantity, it also makes pool pay more to miners. Fig. 11 shows that the total payment always increases with the budget. The total payment never exceeds the budget, validating the desirable profitability of budget feasible.

Then, we choose the 3rd miner from all the 100 miners. Fig. 12 shows that if miner 3 submits a bid, which is different from its real cost (0.0004), he will not increase its utility. Thus the 3rd miner has no motivation to submit a false cost, clarifying the desirable profitability of truthfulness. From Fig. 13, we can see that for each miner i, the payment of i is always larger than its cost, clarifying the desirable profitability of individual rationality.

6 Related Work

Since launched in 2009, Bitcoin has received lots of attention in the research community. Rosenfeld *et al*. [5] describe the various scoring systems used to calculate rewards of participants in Bitcoin pooled mining, and explain the problems each were designed to solve and analyze their respective advantages and disadvantages. Schrijvers *et al*. [6] introduce a game-theoretic model for reward functions in Bitcoin mining pools. They define a precise condition for incentive compatibility to ensure miners' strategy choices optimize the welfare of the pool as a whole. Lewenberg *et al*. [7] use cooperative game theoretic tools to analyze how pool members may share rewards. However, they do not take miners' cost into consideration.

Some other work analyzes cryptocurrency security in realistic settings, which take the cost into consideration. For example, Tsabary *et al.* [16] propose a system, called the gap game, which takes into account all elements of expenses and rewards. Although they introduce the mining cost, they do not design incentive mechanism for mining pool.

Taking into cost into consideration for mining pool is important because there are many selfish strategies of miners. These selfish strategies may deviate the pool manager's real intention, and even are harmful to the whole Bitcoin system if there is no incentive mechanism to stimulate the rational miners. Some of the selfish strategies can be selfish mining [11], bride attack [17], sybil attack [18], and withholding attack [19], etc.

Overall, there is no off-the-shelf incentive mechanism designed in the literature for the mining pool system to stimulate the strategic users. Hence, it is urgent to study how the cost influences the behavior of miners and pool manager.

7 Conclusion

In this paper, we have presented two models for pool mining system with rational miners: public cost model and private cost model. We have modeled the mining pro-

cess in public cost model as *Stackelberg* game, called *Mining* game, at which the pool platform is the leader, and the miners in the pool are the followers. We have shown how to compute the unique *Stackelberg Equilibrium*. For the private cost model, we have formulated the *Budget Feasible Reward Optimization* problem to maximize the reward function under the budget constraint. we have shown that the *BFRO* problem is a budget feasible submodular maximization problem, and designed a budget feasible reverse auction, which is computationally efficient, individually rational, truthful, budget feasible, and constant approximate.

Acknowledgements This research was partly funded by the National Natural Science Foundation of China (No. 61872193).

References

1. Liu Y, Xu C, Zhan Y, et al. Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach. *Computer Networks*, vol.129, pp.399-409, (2017).
2. Jia L, Xu Y, Sun Y, et al. A game-theoretic learning approach for anti-jamming dynamic spectrum access in dense wireless networks. *IEEE Transactions on Vehicular Technology*, vol.68, no.2, pp.1646-1656, (2018).
3. Xu J, Xiang J, Yang D. Incentive mechanisms for time window dependent tasks in mobile crowdsensing. *IEEE Transactions on Wireless Communications*, vol.14, no.11, pp.6353-6364, (2015).
4. Xu J, Rao Z, Xu L, et al. Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities. *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2019.2911512, (2019).
5. Rosenfeld, Meni. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980* (2011).
6. Schrijvers, Okke, et al. Incentive compatibility of bitcoin mining pool reward functions. *International Conference on Financial Cryptography and Data Security*. Springer, Berlin, Heidelberg (2016).
7. Lewenberg, Yoad, et al. Bitcoin mining pools: A cooperative game theoretic analysis. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems* (2015).
8. Gao L, Xu Y, and Wang X, Map: Multi-auctioneer progressive auction for dynamic spectrum access. *IEEE Transactions on Mobile Computing*, 10(8):1144 –1161, August (2011).
9. S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, (2004).
10. Btcfans Homepage, http://mining.btcfans.com/, last accessed 2019/4/21.
11. Eyal I, Sirer E G. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, vol.61, no.7, pp.95-102 (2018).
12. Kiayias A, Koutsoupias E, Kyropoulou M, et al. Blockchain mining games. *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, pp.365-382 (2016).
13. Eyal I, Gencer A E, Sirer E G, et al. Bitcoin-ng: A scalable blockchain protocol. *13th Symposium on Networked Systems Design and Implementation*, pp.45-59 (2016).

14. Singer, Yaron. Budget feasible mechanisms. *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010.
15. Xue G, Xu J, Wu H, Lu W, Xu L, Incentive mechanism for bitcoin mining pool based on Stackelberg game. *Proceedings of the 2nd International Conference on Science of Cyber Security,* Springer, Nanjing, (2019).
16. Tsabary, Itay, and Ittay Eyal. The gap game. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018.
17. Bonneau, Joseph, et al. Why buy when you can rent? bribery attacks on bitcoin consensus. (2016).
18. Bissias, George, et al. Sybil-resistant mixing for bitcoin. *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 2014.
19. Courtois, Nicolas T, and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718* (2014).
20. BitcoinWiki. 2019. Supply. https://en.wikipedia.org/wiki/Bitcoin#Mining