

Article

The Optimal Mining Strategy of Proof of Stake Consensus in Peercoin Blockchain

Bolun Yang¹, Jiamin Hao^{2,3}, Yao Ma^{2,3} and Li Zhou^{2,3,*}

¹ School of Artificial Intelligence, Taiyuan University of Technology, Taiyuan 030024, China; bolunyang0206@163.com

² Shanxi Key Laboratory of Industrial Internet Security, Taiyuan University of Technology, Taiyuan 030024, China; 2024520654@link.tyut.edu.cn (J.H.); mayao@tyut.edu.cn (Y.M.)

³ College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China

* Correspondence: zhouli@tyut.edu.cn

Abstract

The integration of distributed data storage, P2P networks, consensus mechanisms, cryptography and other technologies, the application of blockchain technology has expanded from the initial financial field to many other areas, such as logistics and auditing. The consensus mechanism is the soul of blockchain technology, and it is of great significance to conduct a rigorous mathematical analysis. As far as we know, the Proof of Stake (PoS) consensus mechanism is only a qualitative description of the rich and the poor, the rich are richer, the poor are poorer, and there is no quantitative mathematical analysis. This paper presents a novel quantitative framework to quantitatively analyze the PoS consensus mechanism. Under the premise of not carrying out the attack, we use the expected reward and the reward ratio as the evaluation indicators, quantitatively analyze the optimal fund allocation strategy of the two parties game under the PoS consensus mechanism from the perspective of rich miners, and construct the reward function as the objective function. The inequality constrains the optimization problem and solves it using the Karush-Kuhn-Tucker condition. We consider the two schemes of assignment strategy and random strategy, and get the optimal fund allocation strategy. At the same time, it is compared with the general strategy to obtain the optimization effect of the optimal strategy. After that, we compare the situation in which both sides of the game use the optimal strategy. We found that for assignment strategy, the mining activity will not indicate that the rich are richer and the poor are poorer. However, for the random strategy, this will not happen. The random strategy is also the most common strategy in practice. We also use Markov decision process (MDP) to give the optimal strategy calculation method under the rational miner game, which is also applicable to the n -parties game. The work of this paper helps the blockchain developers to analyze the PoS consensus mechanism, and the adoption strategy of the assignment strategy and the random strategy can be used as the future research direction.



Academic Editors: Zijian Bao, Zhuangzhuang Zhang and Yunhao Ling

Received: 26 January 2026

Revised: 22 February 2026

Accepted: 23 February 2026

Published: 27 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: Proof of Stake; KKT condition; two parties game

1. Introduction

The blockchain technology represented by digital cryptocurrency has achieved significant developments in the financial field, such as Bitcoin [1] and Ethereum [2]. They mint new coins by mining to ensure that there are sufficient funds in the system. The mining process requires all nodes participating in the blockchain system to reach consensus

on transactions and generate effective blocks. The consensus mechanism is an algorithm that agrees on the order of the blockchain transactions in a period of time, ensuring that all honest nodes maintain a consistent blockchain view, which can solve the problem of how the blockchain achieves consistency in a distributed scenario. At present, the main consensus mechanism includes Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS).

Here we introduce the PoS consensus mechanism using PeerCoin v0.6 [3,4]. There are a large number of cryptocurrencies using the PoS consensus mechanism, such as NXT [5], quantum chain [6] and so on. At present, the development team has proposed a large number of mechanisms in order to prevent rich users from continuously minting coins, or users maliciously accumulating long-term age of coin to expect to launch malicious attacks, or to dig most of the new coins, making it difficult or impossible for other users to mint. As stated in Peercoin, the funds and rewards for successful mining can be used after being locked in 520 blocks, and the funds will be split evenly into two new addresses after mining; the maximum accumulated age of coin is specified as 90 days; set up a checkpoint mechanism to ensure that blockchain transaction records are not subject to large-scale tampering. In NXT, in order to prevent the accumulative attack of age of coin, NXT does not adopt the age of coin mechanism. The default one currency is a mining unit; the cumulative difficulty value of mining exists in each block; the funds after successful mining are locked 1440 block; the checkpoint mechanism stipulates that the check segment is 720 block.

The PoS consensus is related to the amount of funds owned by the user and the holding time. The mining uses the age of coin proposed in Bitcoin. The more age of coin you consume, the greater the probability of getting block credits. The normal operation of the blockchain network requires the participation of a large number of nodes. In order to ensure the activity of node participation, the blockchain will adopt different incentive mechanisms. Taking Peercoin as an example, the system provides three incentives: 1. A node that extracts a new block will receive a certain amount of system tokens; 2. A node that obtains the minting rights will receive transaction fees; 3. A token of age will reward interest. This article assumes that the reward is a constant. The miner uses the public key of the transaction initiator to verify the transaction signature, and the verification is passed into the transaction pool. The miners package the transactions into the block for a period of time, solve the computational problems required by the consensus mechanism, and then broadcast the information to the entire network for verification by others. If other miners pass the verification, they record in the blockchain and broadcast confirmation messages and prepare new mining.

Since the birth of Bitcoin, a large number of digital cryptocurrencies have used PoW consensus mechanisms such as Bitcoin and Litecoin. However, PoW consumes a lot of energy because it needs to perform hash operations continuously. In order to solve the problem of energy consumption of PoW, scholars have proposed a Proof of Stake (PoS) consensus mechanism, which uses the funds owned as a resource for mining. The more funds a miner has, the easier it is to obtain block credits. As far as we know, the PoS consensus mechanism is only a qualitative description of the rich are richer and the poor are poorer, and there is no quantitative mathematical analysis. This paper is the first paper to quantitatively analyze the PoS consensus mechanism.

In the PoS consensus mechanism, the miners' mining success rate per time step is related to the amount of funds invested. The more inputs, the easier it is to obtain block minting rights. However, if all funds are used in one time step, the miners may have a poor reward over a period of time due to the existence of a capital locked mechanism. Similarly, if there are too many funds, the investment per time step is relatively small, and

it is almost impossible for miners to obtain block minting rights. Therefore, in the PoS consensus mechanism, it is necessary to analyze the fund allocation plan for the miners to obtain the maximum reward given that the miners can use the total amount of funds and the miners do not perform the attack behavior.

In this paper, given the total amount of funds that miners can use and the miners do not perform the attack behavior, we analyze the miners' allocation plan for the funds, decide how many shares are allocated, and how much each is to make the profit efficiency of the mining activities. highest. It can also be simply understood as how to use the shortest time to achieve the expected reward target. The purpose of this work is to find the optimal fund allocation strategy from the perspective of wealthy miners, taking PeerCoin as an example. Taking the expected reward target and the efficiency of reward as the evaluation indicators, we quantitatively analyze the fund allocation strategy of the two games under the PoS consensus mechanism from the perspective of rich miners, and construct the non-equivalent constraint optimization problem with the reward function as the objective function. The Karush-Kuhn-Tucker condition is solved. We consider the two schemes of assignment strategy and random strategy and get the optimal fund allocation strategy under the PoS consensus. At the same time, it compares with the general strategy and compares the two optimal strategies. After that, we compare the situation in which both sides of the game use the optimal strategy to compare the situation of the rich and the poor. The random strategy is also the most common miner mining strategy in practice. We can also use KKT or MDP to give the optimal strategy calculation method under the rational miners game. At the same time, we give an analytical model of the n -square problem.

Through our work, demonstrated in Figure 1, we find that there is an optimal fund allocation strategy for both the assignment strategy and the random strategy, which can achieve the best reward efficiency. In the comparison with the general strategy, we get the improvement effect of the optimal strategy, and find that the optimization effect of the assignment strategy is small, and the optimization effect of the random strategy is large. Through experiments, it is also found that for the assignment strategy, the mining activities will not be richer and the poor will be poorer; but for random strategy, this will happen. By comparing the assignment strategy and the random strategy, it is found that the random strategy is better than the assignment strategy. However, when the number of funds is small, the assignment strategy is better than the random strategy. The specific strategy requires actual calculation.

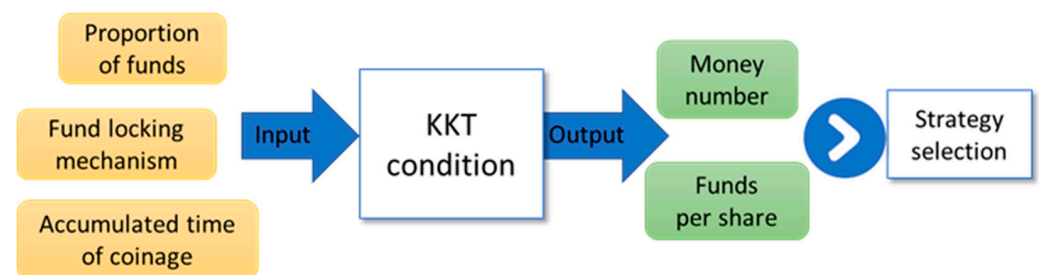


Figure 1. The Structure of work.

It is important to distinguish the scope of this study from the currently dominant Ethereum consensus mechanism (post-merge). While Ethereum utilizes a validator-based model with explicit slashing conditions and epoch-based finality, early PoS implementations like Peercoin rely on the concept of 'Coin Age'—a linear accumulation of time-weighting. Although Ethereum represents the majority of the current market capitalization, Coin Age-based protocols present a unique game-theoretical landscape where time itself

becomes a resource alongside capital. This paper focuses exclusively on the optimization problems within these Coin Age-based systems.

Contribution:

1. To the best of our knowledge, this paper presents a novel quantitative framework to quantify the PoS consensus mechanism and constructs a fund allocation framework based on inequality-constrained optimization.
2. Without considering the execution of the attack, we have found the optimal allocation of funds for the two mining strategies, and we have a certain degree of optimization. At the same time, the analytical model of the n -square game problem is given.
3. We have found that, for the assignment strategy, mining activities do not show that the rich are getting richer and the poor are poorer; but for random strategies, this happens.
4. The work of this paper helps the blockchain developers to analyze the PoS consensus mechanism, and the adoption strategy of the assignment strategy and the random strategy can be used as the future research direction. Some suggestions are given.

This article is described in the following order. The second chapter introduces the current work on the application mode of PoS and the quantitative analysis of blockchain. The third chapter details the problem and the required premise and its proof. We analyze the game and experimental results of both parties in fourth chapter; the fifth chapter is the conclusion of this paper.

2. Related Work

2.1. PoS Application Mode

2.1.1. Hybrid PoW/PoS Consensus

The hybrid mechanism uses a consensus mechanism combining PoW and PoS. PoW is used for initial fund allocation; PoS is used to ensure the security of transactions as a main method for generating trading blocks.

Hybrid PoW/PoS consensus is secure against long-range attacks provided there is enough hashing power in the network. Because of the PoW consensus, it can resist many attacks, such as bribe attacks. However, in order to solve the energy consumption problem of the PoW consensus, as Proof of Stake is usually introduced as an alternative to Proof of Work, combined solutions eventually abandon PoW completely or diminish its role.

2.1.2. Tendermint

Tendermint [7] is a proposed concept of a blockchain. In Tendermint, each block is cryptographically signed by validators. A validator is a user committing to his duty by locking his coins in a bonding transaction; the voting power of each validator is proportional to the amount of locked currency. After finishing his validator duty, a user unlocks his funds by posting an unbonding transaction; the funds are then unlocked with a certain delay. The block is considered valid if it is signed by validators holding at least $\frac{2}{3}$ of all voting power. Thus, a fork can occur only if there is a group of validators with $\frac{1}{3}$ total voting power signing both of competing chains. If there is a fork, validators signing multiple chains can be punished by a user publishing an evidence transaction containing a proof of their maliciousness, i.e., digital signatures of blocks at the same height on both chains. The evidence transaction destroys all locked funds of misbehaving validators. This logic protects against bribe attacks. However, it does not prevent long-range attacks: validators with $\frac{2}{3}$ of the voting power can conspire to publish a blockchain fork after all their funds are unlocked.

2.1.3. Casper

Casper is a conceptual Proof of Stake algorithm proposed by Ethereum [8]. Similar to Tendermint, Casper utilizes the concept of block validators. A validator bets on blocks by assigning each of them a probability; the sum of the probabilities for blocks at the same height must equal 100%. To select one block at each height, validators may need to perform several betting rounds until at least $\frac{2}{3}$ of validators bet on a certain block with a high probability. To resist majority attacks, Casper punishes deviations from the protocol by withholding generation rewards and locked funds from misbehaving validators.

2.2. Quantitative Analysis of Blockchain

As far as we know, there is currently little research work on the quantitative analysis of consensus mechanisms and incentive mechanisms. The following are some related works.

For the PoW consensus mechanism, Rosenfeld et al. [9,10] analyzed the double spending in Bitcoin, but they did not consider the optimal double spending strategy. Gervais et al. [11], referring to the study by Sapirshtein et al. [12], proposed a new quantitative model that analyzes the security and performance of the PoW consensus mechanism under the combined influence of consensus and network parameters. The MDP found the optimal attack strategy for double spending and selfish mining. Their research shows that under low block orphan rates, increasing block size has little effect on the ability of blockchains to resist selfish mining and double spending, and selfish mining is not always a reasonable strategy for attackers. A higher block reward for the system setting will be more resistant to double spending, and it is found that the set block size is 1 MB, and reducing the block interval to 1 min does not significantly affect security. However, the author did not consider the difficulty adjustment mechanism in the analysis process.

Eyal et al. [13] first proposed selfish mining, and selfish mining will cause the actual computing power of the network to decline, making 51% of attacks feasible. The results of the study show that under the implementation of selfish mining, when the attacker's computing power reaches 1/4 of the total power, the attacker and the honest miner have the same probability of successful mining; when the computing power reaches 1/3 of the total computing power, attacker can fully control the network. Nayak et al. [14] proposed Stubborn mining. Compared with the method of Eyal et al. [13], the above attacks will waste more of the power of honest miners, and the attack effect is better. And under this attack, Eclipse attack victims can also make a profit, so they do not actively defend. Because of the low enforceability of selfish mining, the withholding attack [15] may face the problem of "Prisoner's Dilemma", etc. Kwon et al. [16] combined the selfish mining and the withholding attack to propose a Fork After Withholding (FAW) attack. Experiments have shown that an attacker can at least get as much benefit as withholding attack, even reaching four times its benefit. And the two mine pools do not have a "prisoner's dilemma" when they perform FAW attacks on each other, giving clear results: a bigger pool will surely win! The author only considered two mine pools to attack each other, thus we can also consider the situation of more pools.

Carlsten et al. [17] first analyzed the security problems of Bitcoin reward distribution mechanism. They found that when the block reward is reduced to zero, the security and stability of the Bitcoin system will be greatly affected. Specifically, they designed a mining strategy simulator to simulate the miner's decision-making behavior. Based on the simulation of the mining process, the author analyzes the safety and stability of the system when the block reward falls to zero. At this point, there will be an increase in the orphan rate of the Bitcoin block, an increase in the fork probability, and an extension of the transaction confirmation time, which will make the 51% attack easier and even break the consensus of the blockchain.

Recent literature has extensively covered incentive mechanisms in Ethereum and BFT-style PoS protocols. These studies primarily focus on slashing risks and validator committee selection. In contrast, our work addresses the specific mechanics of Coin Age accumulation. Unlike the capital-locking models of Ethereum, Coin Age protocols introduce a continuous time-variable into the mining probability, requiring a distinct mathematical treatment for optimal resource allocation.

3. Premises

3.1. System Model and Assumptions

The Game Scope: A two-party zero-sum game between a rational miner (A) and the rest of the network (B).

Capital Constraints: Total funds normalized to 1.

Coin Age: Linear accumulation of coin age as per the Peercoin protocol.

Time Steps: Discrete time steps representing block intervals (approx. 10 min in Peercoin).

Compounding: We explicitly state that mining rewards are not immediately reinvested in the current optimization cycle to simplify the convex optimization problem.

3.2. PoS Mining Process

In PeerCoin, the blocks are divided into two forms, a POW block and a POS block, illustrated by Figure 2.

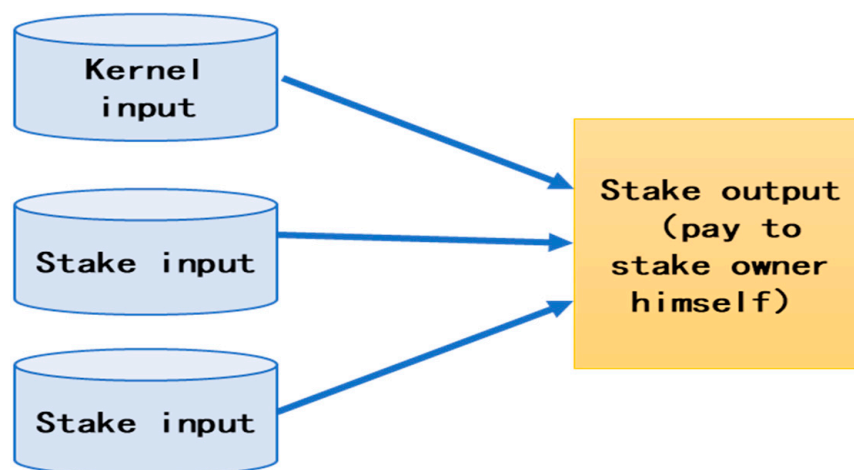


Figure 2. Structure of Proof of Stake (Coinstake) Transaction.

In this new type of block, POS is a special kind of transaction called coin stake. In the coin stake transaction, the block holder can consume his currency to get interest, while gaining a block for the network and coining with POS. The first input of the interest currency is called the kernel and needs to conform to a certain Hash target protocol. The PoS consensus is related to the amount of money the user has and the time of holding it. Here is a concept: age of coin, where $coinage = coin \times age$, Coin and age represent the number of funds and the time of the holding funds. The mining process is similar to that in PoW. The user in PoS needs to satisfy the formula for obtaining the block minting rights:

$$Hash(Prev.hash, A, t) < coinage \times Target \quad (1)$$

where *Prev.hash* is the hash value of previous block; A is the address of the user who mines the block; t is the timestamp; age of coin is the age of coin; and Target is the mining difficulty. The resulting PoS block has Randomness. The PoS random hash operation is done in a limited space, and the PoW is searched in an unrestricted space, so the PoS does

not require a large amount of energy consumption. The random hash target to be met by the equity kernel is based on the age of coin consumed in the core. Therefore, the more the consumption of the currency, the easier it is to comply with the target agreement.

Similarly, in order to maintain a constant rate of block generation, the PoS consensus also has a similar difficulty adjustment mechanism as the PoW consensus, but the PoS consensus difficulty has been changing throughout the mining process, with the main purpose of avoiding sudden fluctuations in mining. The calculation formula of the current difficulty target is

$$CT = \frac{PT \times [(nInterval - 1) \times nTS + 2 \times nAS]}{(nInterval + 1) \times nTS} \quad (2)$$

where $nInterval = \frac{nTTs}{nTS}$, nAS is the actual interval time of the previous two blocks, nTS is the target interval time of the two blocks, the target time space $nTTs = 7 \times 24 \times 60 \times 60$, and the difficulty target of the previous block is PT .

To ensure the validity of the optimization solution, we first analyze the properties of the objective function. The reward function, defined by the product of coin holding and time, exhibits convexity within the feasible region defined by our constraints. Consequently, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for establishing global optimality. We proceed by constructing the Lagrangian function to solve for the optimal strategy.

3.3. Mathematical Knowledge

3.3.1. KKT Condition

The KKT condition can be used to solve an optimization problem with inequality constraints. For a given function, its global minimum can be found on the specified constraint domain. For pure equality constraint optimization problems, the Lagrange multiplier method [18] can be used. The KKT condition is the generalization of the Lagrange multiplier method.

Consider a nonlinear optimization problem with m inequality constraints and l equality constraints:

$$\begin{aligned} \min_x & f(x) \\ \text{s.t.} & g_i(x) \leq 0, h_j(x) = 0 \end{aligned}$$

Under the condition that there are certain rules, solving this problem can have sufficient conditions and necessary conditions for the optimal solution to meet the KKT condition.

The sufficient and necessary conditions for KKT requirements [19]:

(1) Necessary conditions

Suppose that the objective function $f: R^n \rightarrow R$ and the constraint functions $g_i: R^n \rightarrow R$ and $h_j: R^n \rightarrow R$ are continuously differentiable at a point x^* . If x^* is a local optimum and the optimization problem satisfies some regularity conditions (see below), then there exist constants $\mu_i (i = 1, \dots, m)$ and $\lambda_j (j = 1, \dots, l)$, called KKT multipliers, such that

Stationarity

For maximizing $f(x)$:

$$\nabla f(x^*) = \sum_{i=1}^m \mu_i \nabla g_i(x^*) + \sum_{j=1}^l \lambda_j \nabla h_j(x^*),$$

For minimizing $f(x)$:

$$-\nabla f(x^*) = \sum_{i=1}^m \mu_i \nabla g_i(x^*) + \sum_{j=1}^l \lambda_j \nabla h_j(x^*),$$

Primal feasibility

$$g_i(x^*) \leq 0, \text{ for } i = 1, \dots, m.$$

$$h_j(x^*) = 0, \text{ for } j = 1, \dots, l.$$

Dual feasibility

$$\mu_i \geq 0, \text{ for } i = 1, \dots, m.$$

Complementary slackness

$$\mu_i g_i(x^*) = 0, \text{ for } i = 1, \dots, m.$$

In the particular case $\{m=0\}$, i.e., when there are no inequality constraints, the KKT conditions turn into the Lagrange conditions, and the KKT multipliers are called Lagrange multipliers.

(2) Sufficient conditions

For smooth, nonlinear optimization problems, a second-order sufficient condition is given as follows. Consider x^*, λ^*, μ^* that find a local minimum using the Karush-Kuhn-Tucker conditions above. With μ^* such that strict complementarity is held at x^* (i.e., all $\mu_i > 0$), then for all, $s \neq 0$ such that

$$\left[\frac{\partial g(x^*)}{\partial x}, \frac{\partial h(x^*)}{\partial x} \right]^T s = 0$$

(where the bracketed expression is a row vector), the following equation must hold:

$$s'^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) s \geq 0$$

If the above condition is strictly met, the function is a strict constrained local minimum.

The optimization problem formulated above involves both continuous variables (allocation amounts) and a discrete variable (number of splits, m). To ensure theoretical rigor, we adopt a two-step solution approach:

Convexity Analysis: For a fixed integer m , the objective function is strictly concave with respect to the allocation vector, and the constraints define a convex feasible set. This satisfies the regularity conditions required for the Karush-Kuhn-Tucker (KKT) conditions, guaranteeing that any local maximum found is a global maximum for that specific m .

Discrete Traversal: Since the feasible range of m is bounded in practice (due to minimum transaction limits), we employ a traversal algorithm to iterate through possible values of m . We solve the KKT conditions for each m and compare the resulting objective values to identify the global optimum.

3.3.2. Markov Decision Process

The Markov decision process can be used to solve continuous decision problems, defined as the following: MDP is usually represented by a five-element array $\langle S, A, R, T, \gamma \rangle$, where S represents the state space, A represents the action space, $R: S \times A \rightarrow \mathcal{R}$ represents the reward function, which is the mapping from the state action pair to the reward value, and $T: S \times A \times S \rightarrow [0, 1]$ is a state transition function, demonstrated by Figure 3.

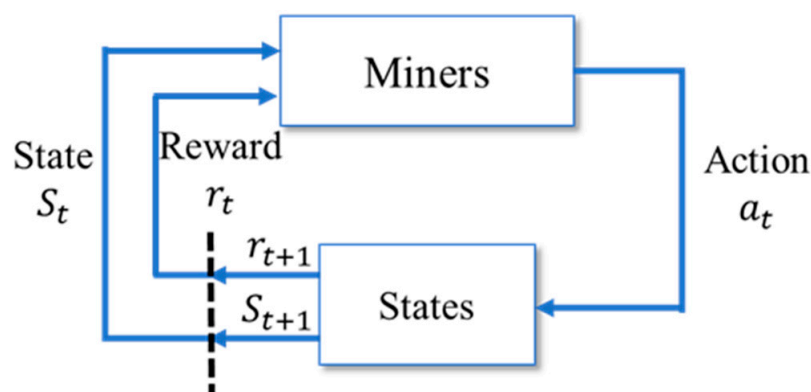


Figure 3. Markov decision process schematic diagram.

3.4. Parameters

The parameters involved in this article are:

- α : The proportion of funds owned by rich miners.
- T_1 : Rich miners' accumulated age.
- T_2 : Poor miners' accumulated age.
- α_i : When the rich miners participate in the mining of the i -th block, they invest the principal of the age of coin meet $\sum_{i=1}^n \alpha_i \leq \alpha$.
- n : The number of capital partitions of the poor miners' funds.
- r : The rewards for successful mining at each step.
- t : Time step.
- ER : Expected reward target.
- LT : The locked period specified by the fund lock mechanism.
- RT : The time required to reach the expected reward target.

Assignment strategy: refers to the miner dividing the funds according to the optimal strategy, but only using one fund at a time;

Random strategy: refers to every time the miner inputs all the funds that have been allocated, and randomly determines which funds are used;

General strategy: refers to the miners distributing the funds evenly, but only one shares at a time.

Rational miners and honest miners are miners who use optimal strategy and general strategy respectively.

3.5. Lemma

Lemma 1. In the PoS consensus mechanism, after the miners divide the money into multiples, the random strategy is not the same as the overall use of the undivided funds.

Proof. From Section 3.2, we simplify the problem, for example, finding a range of intervals $[0, 2a]$ and $[0, a]$ satisfying the condition between intervals $[0, b]$, where $a < b/2$. At this point, we perform a search for finding $[0, 2a]$ and two searches for finding $[0, a]$. Obviously, $P_1 = \frac{2a}{b}$. For each $[0, a]$ selection, $P_2^i = \frac{a}{b}$, $i = 1, 2$. Since the two-step operation is independent and identically distributed, we require that the requirement be met at least once, so

$$P_2 = P_2^1 + P_2^2 - P_2^1 \times P_2^2 = \frac{2a}{b} - \left(\frac{a}{b}\right)^2 < P_1 \quad (3)$$

Hence, the Lemma is established and the proof is completed. $\square$

This lemma can be understood intuitively. In the PoW mining consensus mechanism, the mining success rate of the small computing miners forming the mining pool is equal to the simple sum of the proportions of the computing power they have. This premise is to prove that this situation does not apply in PoS.

To rigorously interpret this analogy in a probabilistic framework, the ‘interval’ can be viewed as the sample space of valid hash outputs. The strategy of splitting funds essentially increases the frequency of independent Bernoulli trials within this sample space. Thus, the intuitive interval search corresponds mathematically to the union of independent probability events, confirming the distinct advantage of the splitting strategy.

Lemma 2. *If the rich miners want to get the most out of the mining revenue, the mining should be involved in mining as early as possible.*

Proof. Consider two parties game. A and B compete at a specific time step, If A wants to mine this block with a high probability, then A should pay at least as many age of coin as B in this time step, and the formula $\alpha_i(T_1 + t) \geq b(T_2 + t)$, where α_i, b are the number of funds invested by A and B in this step, and $(T_i + t)$, $i = 1, 2$ are the accumulated time of funds used by A and B respectively, assuming $T_1 > T_2$. It is possible to obtain $\alpha_i \geq \frac{b(T_2 + t)}{(T_1 + t)}$. Let $f = \frac{b(T_2 + t)}{(T_1 + t)}$, and calculate $\frac{\partial f}{\partial t} = \frac{b(T_1 - T_2)}{(T_1 + t)^2}$.

Since $T_1 > T_2$, $\frac{\partial f}{\partial t} > 0$, that is, α_i monotonically increases with t . Therefore, the sooner A participates in mining, the better. The proof is completed. $\square$

Lemma 3. *The work is analyzed from the perspective of the rich, and the upper bounds of the accumulated time of age of coin and the funds after successful mining are divided into two parts are not considered. However, this method is also applicable to consider the upper bound of accumulation time and divide funds. The way has certain universality.*

Lemma 4. *The mining reward is very small compared to the funds invested in mining, so we do not consider mining rewards for the next mining.*

4. Two Parties Game

In this chapter we analyze the two parties game. Assuming that the total capital is 1, the total funds owned by A and B are α and $1 - \alpha$, respectively, $\alpha \in (0.5, 1]$. The rational miners are defined as miners using the optimal allocation strategy, and the honest miners are using general strategy. The analysis body is A.

4.1. A Is Rational Miner and B Is Honest Miner

Assume that A's funds are divided into m shares, and the funds α are $\alpha_1, \alpha_2, \dots, \alpha_m$, $m \in \mathbb{Z}$, where m is unknown, and the accumulated age of the currency is T_1 . B's funds are divided into n shares on average, and the accumulated age of the currency is T_2 . Assume that the age of the coin has accumulated $T_1 > T_2$, and t is the time step.

4.1.1. Assignment Strategy

Here we analyze the fact that miners only invest one fund each time to participate in mining, and other funds are still accumulating age of coin. Our goal here is to maximize the reward of miners. For the mining success rate, we determine the proportion of funds invested by miners involved in mining.

We can calculate the success rate of A mining in the i -th step under the successful mining reward r .

$$P_i = \frac{\alpha_i [T_1 + (i-1)t]}{\alpha_i [T_1 + (i-1)t] + \frac{1-\alpha}{n} [T_2 + (i-1)t]} \quad (4)$$

The expected rewards per step is $R_i = P_i \times r$, where $i = 1, 2, \dots, m$. The total expected reward is

$$R = \sum_{i=1}^m R_i \quad (5)$$

To maximize the benefits of miners, we consider the inequality-constrained optimization problem:

$$\begin{cases} \max \sum_{i=1}^m R_i \\ \sum_{i=1}^m \alpha_i = \alpha \\ \alpha_i \geq 0 \quad i = 1, 2, \dots, m \end{cases} \quad (6)$$

Convert the objective function to a standard form:

$$\begin{cases} \min \left(-\sum_{i=1}^m R_i \right) \\ \sum_{i=1}^m \alpha_i - \alpha = 0 \\ -\alpha_i \leq 0 \quad i = 1, 2, \dots, m \\ \alpha_i - \alpha \leq 0 \quad i = 1, 2, \dots, m \end{cases} \quad (7)$$

This problem has inequality constraints, and we consider solving it using KKT condition. Next, we explain how the above optimization problem satisfies sufficient conditions and necessary conditions.

Use the following theorem: If $f(x) = af_1(x) + bf_2(x)$, where $a, b \geq 0$, if $f_1(x)$, $f_2(x)$ is a convex function on the convex set R_c , then $f(x)$ is a convex function of the set R_c . This theorem can be simply proved by definition of convex function. Obviously, it can be proved that R_i is a convex function, and the objective function $(-\sum_{i=1}^m R_i)$ can be obtained as a convex function by the above theorem. There is also a theorem in mathematics that “any local optimal solution of convex programming is a global optimal solution”.

Necessary condition:

Suppose $\sum_{i=1}^m R_i$, $\sum_{i=1}^m \alpha_i - \alpha$, α_i are all real set mappings, they are continuously differentiable at α^* ; if α^* is a local minimum point, then there will be a set of constant forms of the multiplier $\lambda \in \mathbb{R}$, $\mu_i \in \mathbb{R}^+$, $\sigma_i \in \mathbb{R}^+$ so that

$$\mu_i X = 0, \sigma_i Y = 0, i = 1, 2, \dots, m.$$

$$\frac{\partial R}{\partial \alpha_i} = 0, i = 1, 2, \dots, m$$

Sufficient condition:

Suppose the objective function $-\min \sum_{i=1}^m R_i$ and the constraint function $\sum_{i=1}^m \alpha_i - \alpha$, α_i is a convex function, then $\sum_{i=1}^m \alpha_i - \alpha$ is an affine function. Suppose there is a feasible point α^* , if there is a constant $\mu_i \geq 0$, $\sigma_i \geq 0$, so that

$$\mu_i X = 0, \sigma_i Y = 0, i = 1, 2, \dots, m.$$

$$\frac{\partial R}{\partial \alpha_i} = 0, i = 1, 2, \dots, m$$

Then, α^* is the global minimum point.

Next, we show that the optimization problem uses the solution process of KKT condition. Construct the objective function, including all equality constraint functions and inequality constraint function

$$L(\alpha_i, \lambda, \mu_i, \sigma_i) = -\sum_{i=1}^m R_i + \lambda \left[\sum_{i=1}^m \alpha_i - \alpha \right] + \mu_i(-\alpha_i) + \sigma_i(\alpha_i - \alpha) \quad (8)$$

where $\lambda \in \mathbb{R}$, $\mu_i \in \mathbb{R}^+$, $\sigma_i \in \mathbb{R}^+$.

Let $L(\alpha_i, \lambda, \mu_i, \sigma_i)$ derivate α_i and make it equal to 0, respectively, to obtain i equations.

$$\frac{\partial L}{\partial \alpha_i} = \frac{\frac{\alpha-1}{n}[T_1 + (i-1)t][T_2 + (i-1)t]}{\left\{ \alpha_i[T_1 + (i-1)t] + \frac{1-\alpha}{n}[T_2 + (i-1)t] \right\}^2} + \lambda - \mu_i + \sigma_i = 0 \quad (9)$$

$$\sum_{i=1}^m \alpha_i - \alpha = 0 \quad (10)$$

$$\mu_i(-\alpha_i) = 0 \quad (11)$$

$$\sigma_i(\alpha_i - \alpha) = 0 \quad (12)$$

where $i = 1, 2, \dots, m$. At this time, the above has a total of $3m + 1$ unknowns and $3m + 1$ equations, and the problem can be solved by the elimination method. It can be seen from the above analysis that the minimum value point obtained for a specific m is the global optimal value point. Here m is unknown; we can take the traversal approach, which is a hybrid optimization problem [20]. For a particular m , we can derive the optimal allocation scheme $\alpha_1, \alpha_2, \dots, \alpha_m$ of total funds α . The optimal expected reward is obtained by changing m .

The above KKT condition can obtain the maximum mining reward without considering the mining time, but the time for different rewards is different. We can imagine that the same reward is acquired by 10 days and 5 days, and it is obviously better to use the 5-day plan. We want to find the shortest time solution to achieve the desired reward target. At the same time, we consider the existence of a capital locked mechanism in order to better meet the actual situation. In order to prevent the rich from continuing to mint coins, it is difficult for miners with less capital to obtain block minting rights. Peercoin has a capital locked mechanism stipulates that after the miner successfully obtains the block minting rights, the principal and mining awards of the destroyed age of coin will be locked for 520 blocks before they can be used again. We can calculate the real time RT that reaches the expected reward target:

$$RT = \left\lfloor \frac{ER}{R} \right\rfloor \times LT + i \quad (13)$$

where

$$i = \min_i \left\{ \left(\left\lfloor \frac{ER}{R} \right\rfloor \times LT + \sum_{j=1}^i R_j \right) > ER \right\} \quad (14)$$

We can compare the different capital allocation strategy to achieve the expected reward target, as an indicator of the strengths and weaknesses of evaluation.

4.1.2. Random Strategy

But in reality, even if a miner divides the funds into multiples, they will still choose to put all the funds into mining in order to obtain more reward, and which capital can be used is random. Even more, some miners do not spend time arranging the funds used at each step but instead, all the inputs. Therefore, in order to better simulate the actual

situation, here we consider the random strategy. The number of funds and the number of funds $\alpha_1, \alpha_2, \dots, \alpha_m$ are obtained by assignment strategy in Section 4.1.1. According to Lemma 1), the rational miner mining success rate in each step is

$$P_1 = 1 - \prod_{i=1}^m \left(1 - \frac{\alpha_i T_1}{\alpha T_1 + \frac{1-\alpha}{n} T_2} \right) \quad (15)$$

$$P_i = 1 - P'_i \quad (16)$$

$$P'_i = \prod_{i=1}^{m-i+1} (1 - P''_i) \quad (17)$$

$$P''_i = \frac{\alpha_i (T_1 + (i-1)t)}{(\alpha - \sum_{j=1}^{i-1} \alpha_j) (T_1 + (i-1)t) + \frac{1-\alpha}{n} (T_2 + (i-1)t)} \quad (18)$$

where $i = 2, \dots, m$. Note, the cumulative elements P'_i in P_i are not necessarily the sequence order of the original α , because every time a block is successfully drilled, a random amount of funds will be used. Where P_i exists

$$\prod_{i=1}^{m-i+1} \left(1 - \frac{\alpha_i (T_1 + (i-1)t)}{(\alpha - \sum_{j=1}^{i-1} \alpha_j) (T_1 + (i-1)t) + \frac{1-\alpha}{n} (T_2 + (i-1)t)} \right) \quad (19)$$

it is to determine the probability that no funds will be used in this time step, via calculating

$$\max \sum_{i=1}^m P_i$$

to acquire the maximum mining reward under that which is not considered mining time. Similarly, we still specify an expected reward target ET to calculate the time RT required to reach the expected reward target, which is calculated in the same way as the assignment strategy.

4.1.3. Results

All experiments in this paper use the software MATLAB 2014a, and use KKT condition to solve the above-mentioned inequality-constrained optimization problem using the toolbox from MATLAB. Specifically, the 'Time Step' in our model is calibrated to the Peercoin protocol, where $t = 1$ represents a 10 min block interval.

Since we did not find the actual relevant data, we used the hypothetical data for analysis. First, we use the fixed variable method to prove that the choices of α , T_1 , T_2 , n , ER , LT have no effect on the results.

Figure 4 shows that the relationship between the number of capital partitions expected reward under $\alpha = 0.6$ and $\alpha = 0.8$ and other parameters are fixed. We can find that for different α , the course of curve is basically same, so different α has no effect on the experimental model.

Figure 5 shows the relationship between number of capital partitions and expected reward at $\alpha = 0.8$ and the different cumulative time of age of coin. The combination of T_1 and T_2 is (90,45), (90,90), (45,90) from top to bottom. We can see that the images are basically same, both appear monotonously rising, but have an upper bound. Figure 6 shows the relationship between number of capital partitions and expected reward under different n . From top to bottom, n is 30, 9, and 1. We can find that different n has no effect on trend of the image.

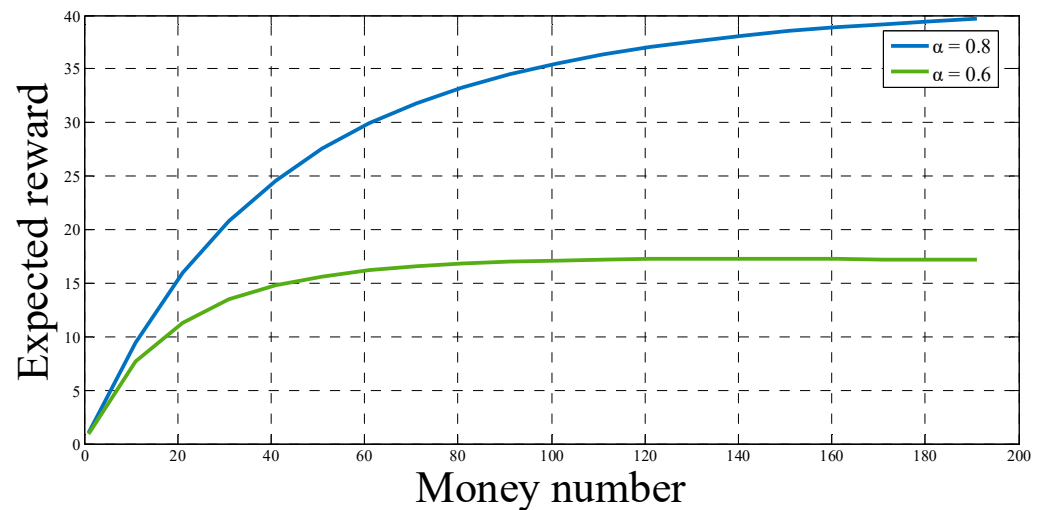


Figure 4. The relationship between the number of capital partitions expected reward under different α .

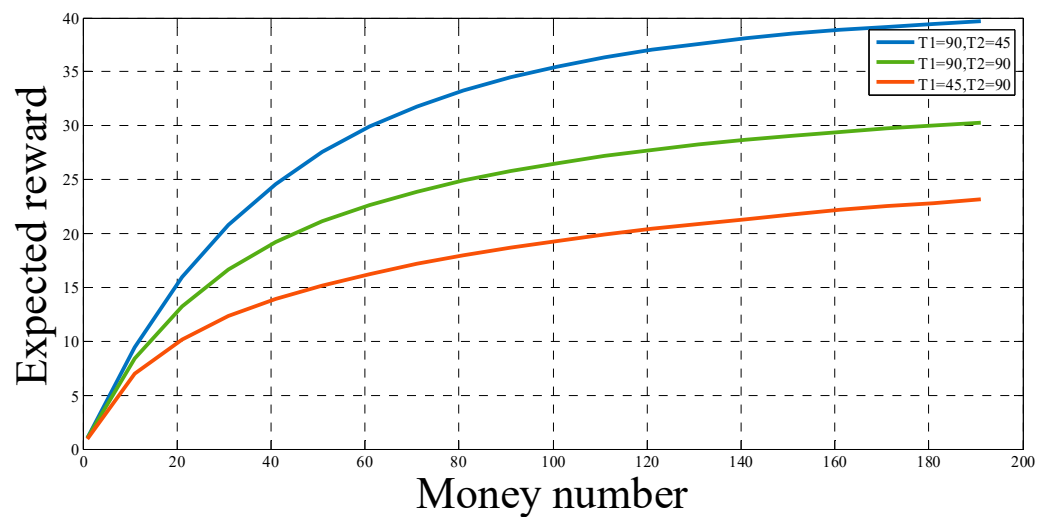


Figure 5. The relationship between number of capital partitions and expected reward under different T_1 and T_2 .

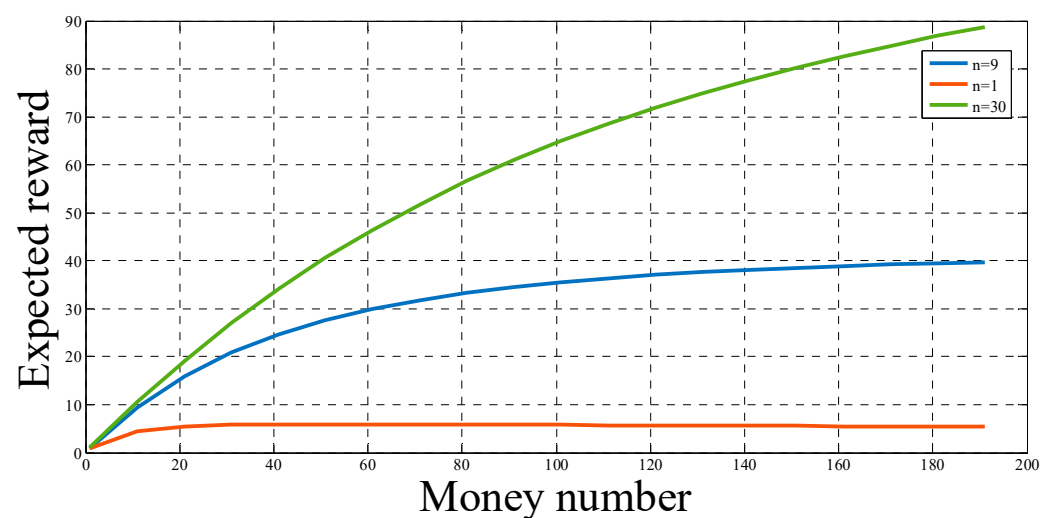


Figure 6. The relationship between number of capital partitions and expected reward under different n .

By the above analysis, the choices of α , T_1 , T_2 , n , ER , LT have no effect on the results. Therefore, we selected experimental parameters $\alpha = 0.8$, $T_1 = 90$, $T_2 = 45$, $n = 9$, $ER = 25$, and $LT = 20$ for analysis.

Assignment Strategy

For the first case, Figure 7 is a plot of the number of capital partitions and the optimal total expected reward at $\alpha = 0.8$. We can find that the curve is a single increasing function with a monotonic decreasing monotonic derivative. As the number of capital partitions increases, the expected reward also increases, but obviously the expected reward will converge to an upper bound, that is, the optimal reward has an upper limit. Because only one fund can be used per time step, different capital shares represent different mining time, and we can calculate the income by $\frac{\text{Expected reward}}{\text{time}}$. Figure 8 is an image of the relationship between $\frac{\text{Expected reward}}{\text{time}}$ and the number of capital partitions. We can find that the reward ratio shows a decreasing function of decreasing rate, so their capital yields are different.

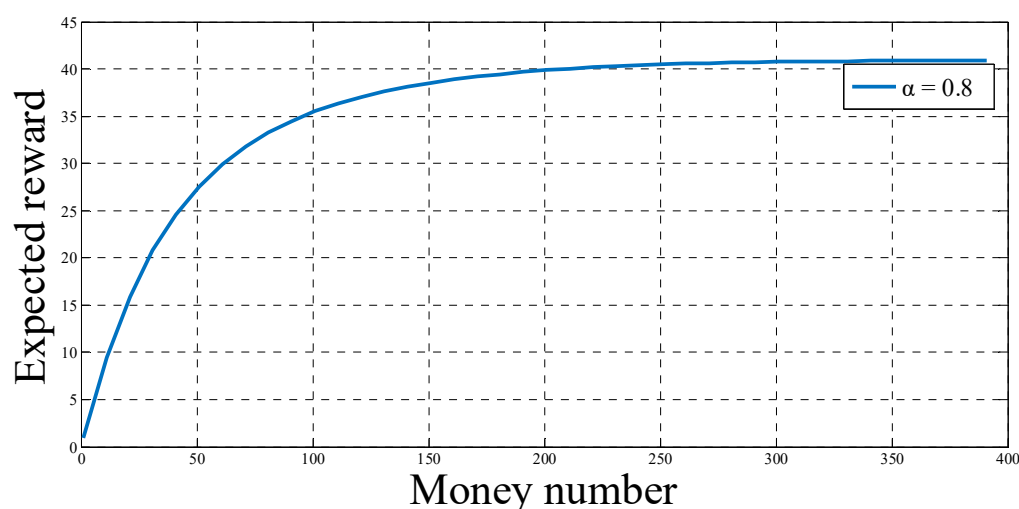


Figure 7. The relationship between number of capital partitions and optimal total expected reward.

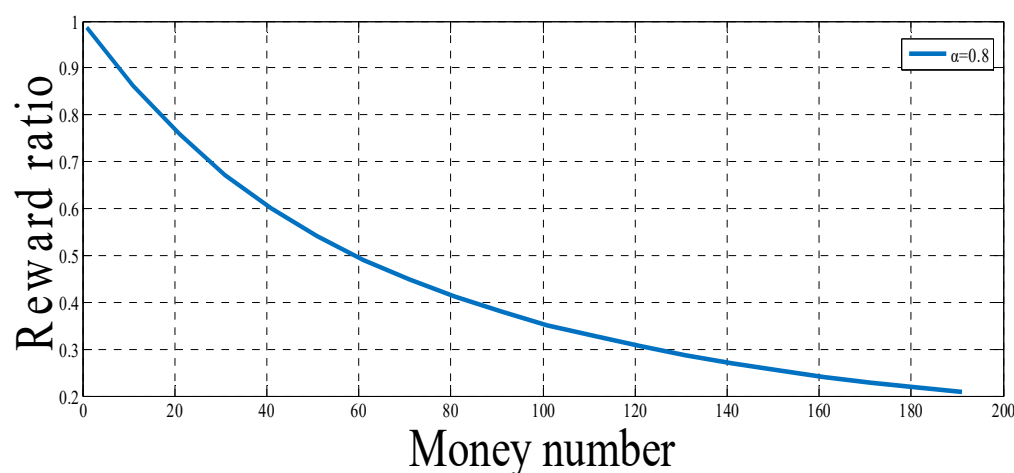


Figure 8. The relationship between number of capital partitions and reward ratio without locked mechanism.

It can be seen from the above that, due to the expected time of obtaining expected reward, the expected reward is not as large as possible. At the same time, in practice, a blockchain application such as PeerCoin is provided with a coin locked mechanism. Therefore, we have set the expected reward target under coin locked mechanism and

studied the length of time required for the rational miner to reach the reward target. The result is shown in Figure 9. We can see that there is an optimal strategy that minimizes the time required to achieve the expected target. Please note that in some cases, the number of capital partitions may be equal to the length of time the funds are locked, but not absolute. Figure 10 shows the relationship between number of capital partitions and reward ratio with locked mechanism, which obviously has the highest point, and can also prove that there is an optimal fund allocation strategy.

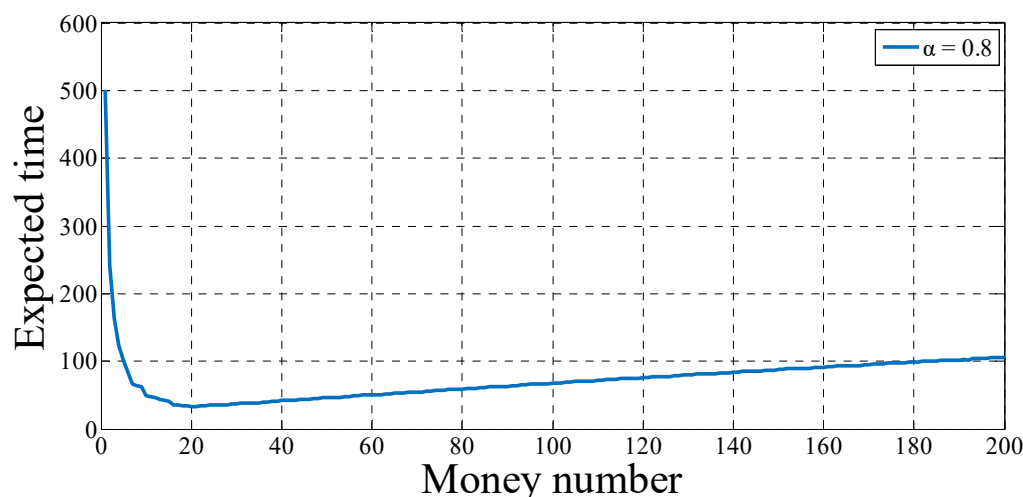


Figure 9. The relationship between number of capital partitions and expected time.

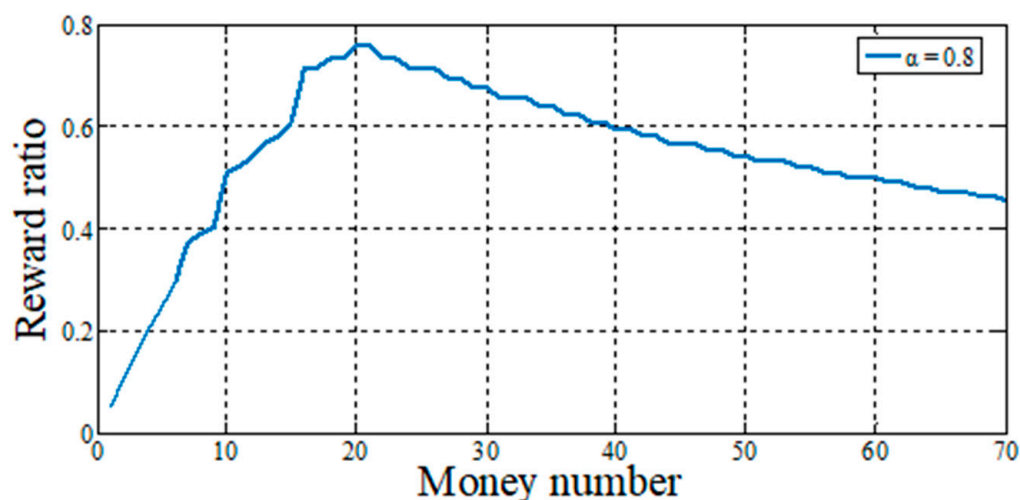


Figure 10. The relationship between number of capital partitions and reward ratio with locked mechanism.

In order to make the rich miners get more benefits, we consider the optimal strategy to analyze the amount of α to achieve the optimal reward. In Figure 11, we show the relationship between the reward ratio and the number of capital partitions in different α . We have considered three cases of $\alpha = 0.6$, $\alpha = 0.7$ and $\alpha = 0.8$, respectively. It can be found that the maximum reward ratio of different α corresponds to a different number of capital partitions, but it shows that under a certain relationship, the more funds, the greater the reward ratio.

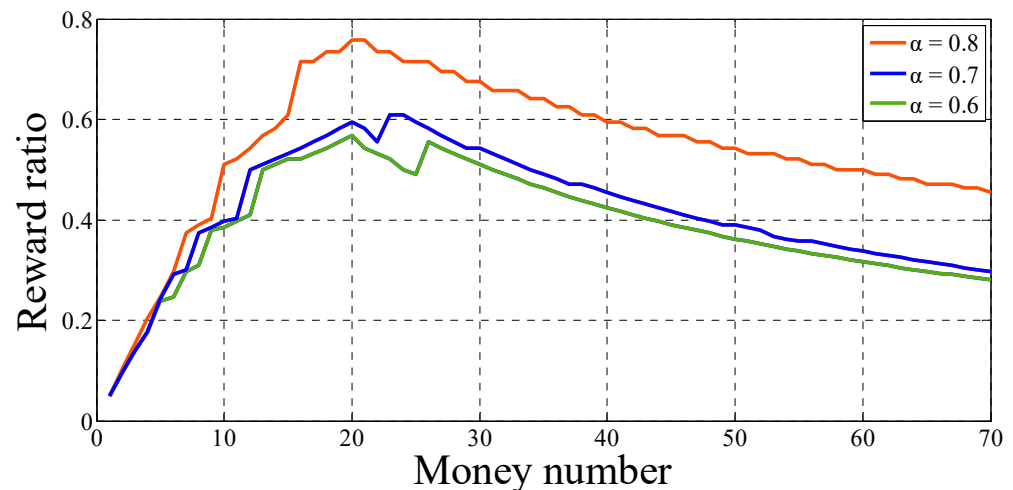


Figure 11. The relationship between reward ratio and number of capital partitions under different α .

Sensitivity analysis reveals that while variations in the parameter α scale the absolute magnitude of the expected rewards, they do not alter the structural properties of the optimal strategy. Specifically, as α increases, the gap between the optimal strategy and the baseline strategy widens monotonically, but the optimal number of splits m remains stable across a wide range of values. This suggests that the proposed strategy is robust against parameter fluctuations within reasonable protocol limits.

Random Strategy

Contrary to the intuitive assumption that a random strategy might inherently favor wealthier nodes due to their larger initial stake, our simulation reveals a more nuanced outcome. While wealthier nodes do obtain higher absolute rewards, their Return on Investment (ROI) under the random strategy fluctuates significantly and does not consistently exhibit a super-linear growth trend. This indicates that without an optimized allocation strategy, the ‘Rich get Richer’ phenomenon is not guaranteed solely by holding more capital. The optimal strategy derived in our model, however, demonstrates a clear efficiency advantage over the random approach.

For the random strategy, Figure 12 is the relationship between the number of capital partitions and the total expected reward. We can see that the reward curve basically shows a state of linear rise. Similarly, in considering the coin locked mechanism, in order to find a better allocation of reward ratio, we stipulate that the expected reward is 25, and the coin locked mechanism provides for 20 time steps. Figure 13 shows the relationship between the time required by rational miners and the number of capital partitions in order to achieve the set expected reward. Here we still consider the rational miners to have funds $\alpha = 0.8$.

Similarly, in order to make the rich miners gain greater benefits, we consider the optimal strategy to analyze the optimal reward ratio on different funds. In Figure 14, we show the relationship between the reward ratio and number of capital partitions in different α . We have considered three cases of $\alpha = 0.6$, $\alpha = 0.7$, and $\alpha = 0.8$, respectively. It can be found that the maximum capital yield of different α corresponds to the same number of capital partitions, and under a certain amount of funds, the more funds, the greater the reward ratio.

It is found by Figure 13 that under the 20-step time of the coin locked, the number of capital partitions is basically equal to the time of the coin locked. Figure 15 shows an image with $\alpha = 0.8$ and a coin locked time of 50. When the image shows the number of capital partitions $m \in [20, 40]$, the rational miner can reach the expected reward target as quickly

as possible. Therefore, the number of capital partitions is not necessarily equal to the time the coin locked, and needs to be calculated based on actual data.

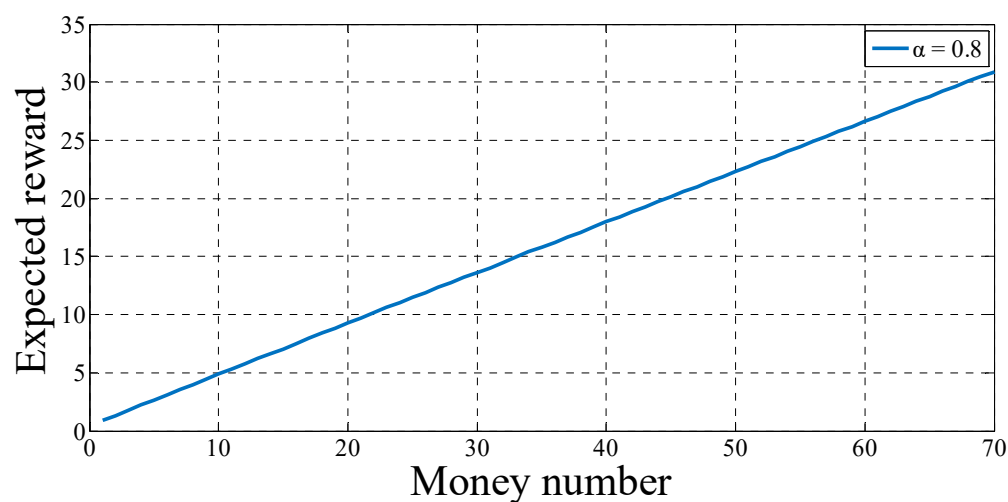


Figure 12. The relationship between number of capital partitions and total expected reward.

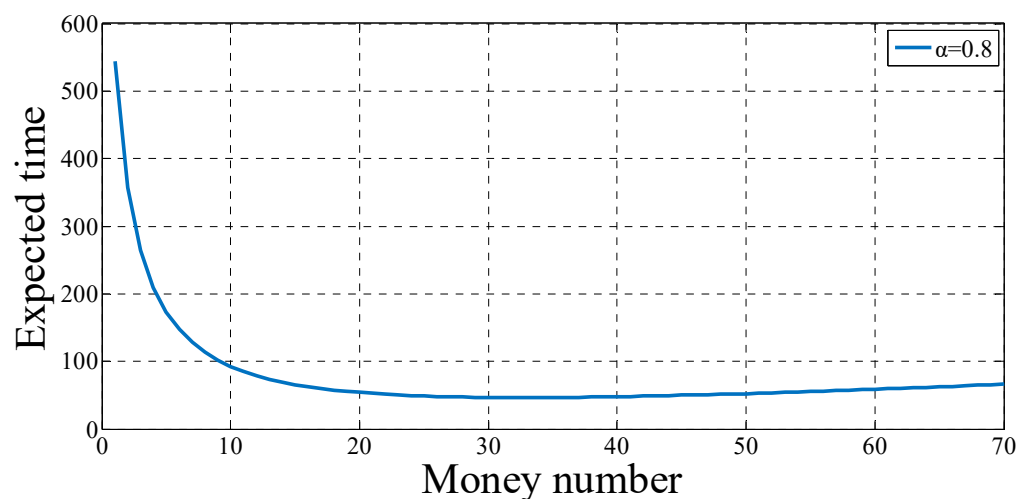


Figure 13. The relationship between number of capital partitions and expected time by rational miners.

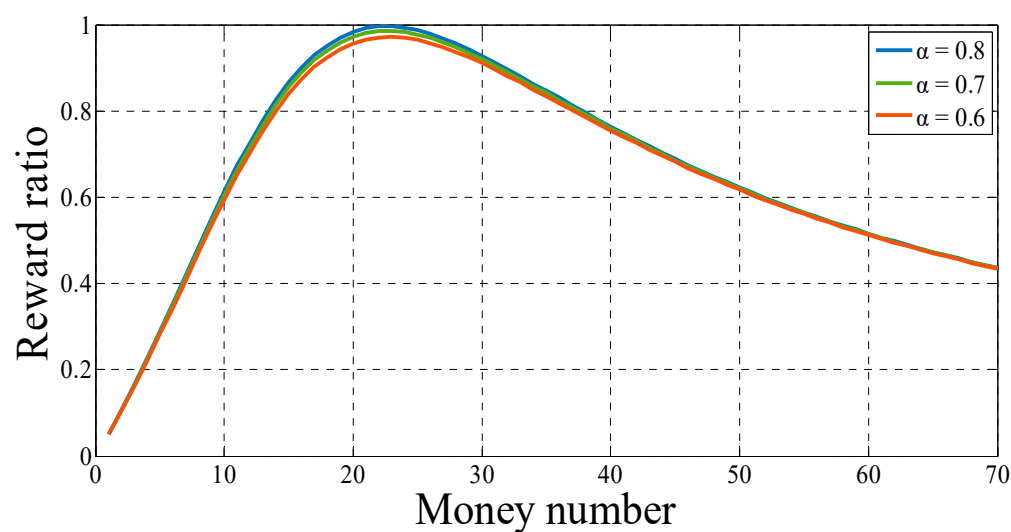


Figure 14. The relationship between number of shares and reward ratio with locked mechanism.

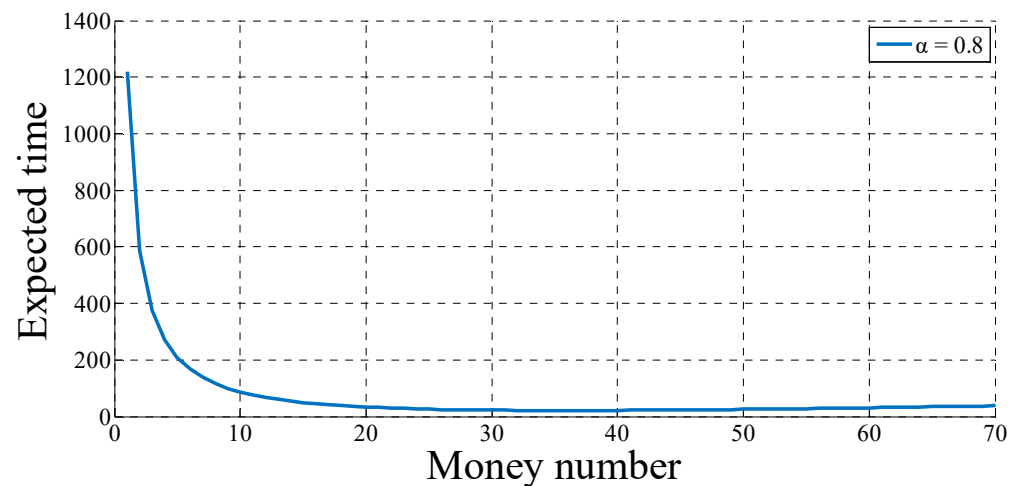


Figure 15. The relationship between number of capital partitions and expected time under the time of coin locked is 50.

4.1.4. Compared with General Strategy

From the above analysis, the optimal fund allocation scheme of the assignment strategy and the random strategy is obtained. In order to illustrate that the obtained strategy is beneficial to the rational miners, we compare the assignment strategy and random strategy with general strategy. Among them, in comparison with the assignment strategy, the general strategy is to divide the funds into multiples equally; in comparison with the random strategy, the general strategy is to use the funds as a whole.

Assignment Strategy

First, we analyze the situation where rational miners adopt the assignment strategy. We find that the optimal strategy obtained above is better than general strategy, but the reward improvement is not obvious, see Figure 16. Similarly, we compare the time of the two strategies to achieve the expected reward, which sets the expected reward target to 25, and the coin locked mechanism specifies a lock time of 20 time steps, see Figure 17. We can find that the image as a whole shows the optimal strategy and the general strategy to achieve the expected reward target time is basically equal.

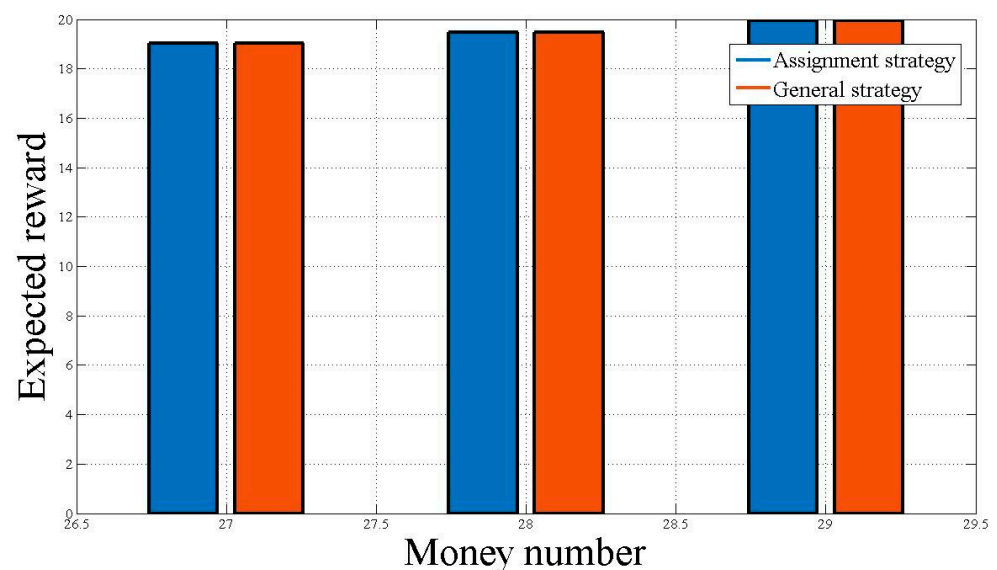


Figure 16. Comparison of the expected reward and number of capital partitions in the assignment strategy and general strategy.

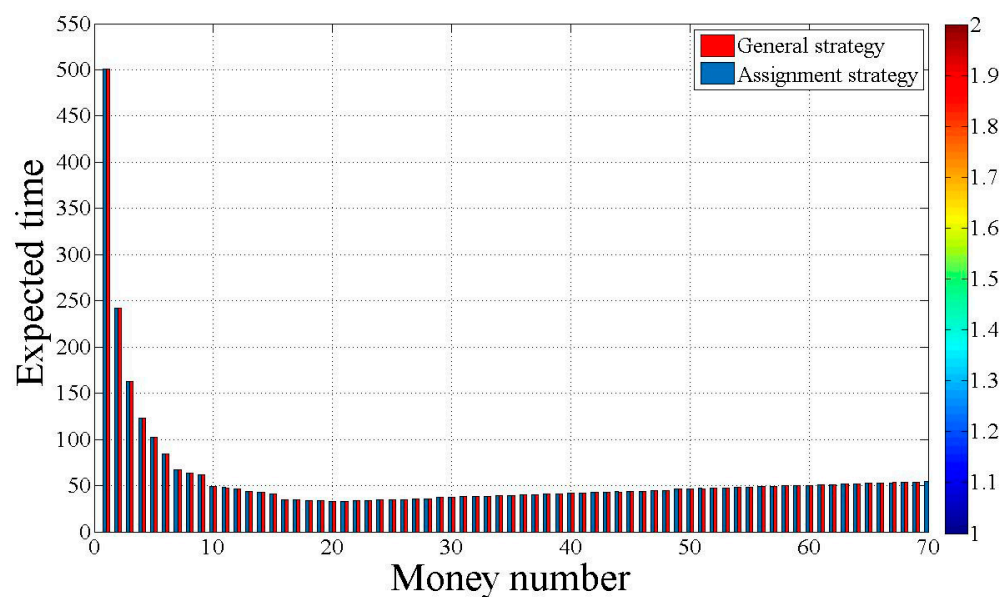


Figure 17. Comparison of reward time and number of capital partitions in the assignment strategy and general strategy.

Random Strategy

Next, we analyze the situation of rational miners using a random strategy. We can see from Figure 18 that the optimal strategy is far superior to the general strategy in terms of total expected reward. Figure 19 is an image of the time of the two strategies to achieve expected reward. We can see that the optimal strategy takes a shorter time than the general strategy to achieve the expected reward target. Obviously, the random strategy is superior to general strategy and assignment strategy.

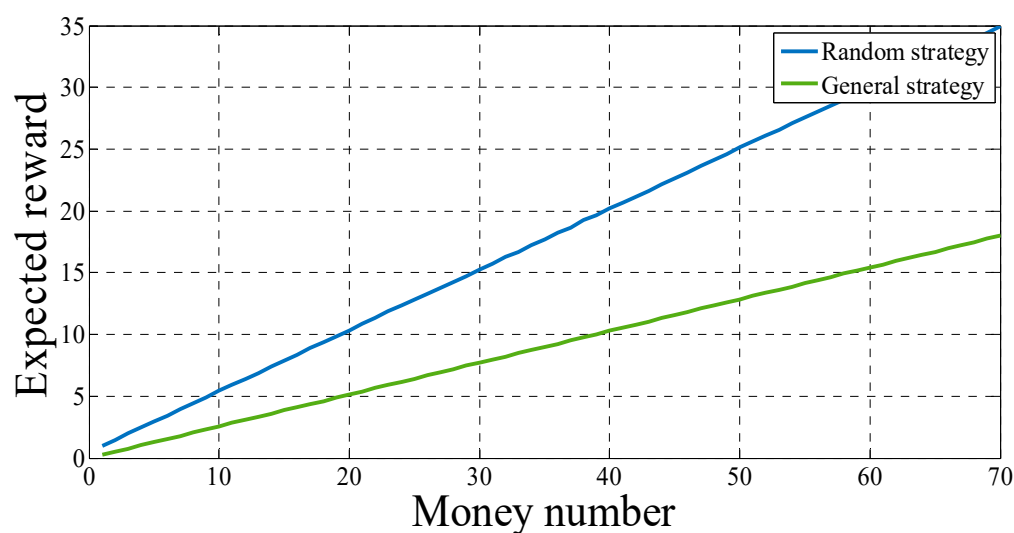


Figure 18. Comparison of the expected reward and number of capital partitions in the random strategy and general strategy.

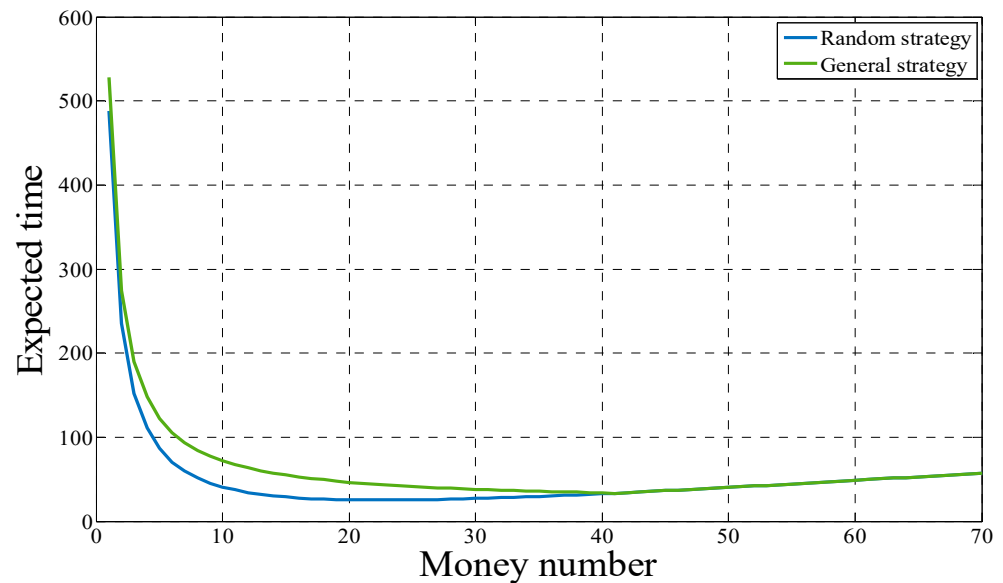


Figure 19. Comparison of the reward time and number of capital partitions in the random strategy and general strategy.

4.1.5. Compare Assignment Strategy and Random Strategy

Intuitively, the random strategy is better than assignment strategy, but we do not know the detailed relationship. Next, we compare assignment strategy with random strategy. Figure 20 is a comparison of the expected reward of the two with the number of capital partitions. We can find that the assignment strategy is obviously better than the random strategy when the number of capital partitions is small, and the maximum gap is reached when the number of capital partitions is about 30. This is because Lemma 1 proves that the success rate of each step of the random strategy is lower than the assignment strategy. Similarly, we also set the expected reward target to 25, and the capital locked mechanism specifies a lock duration of 20, which compares the two strategies. As shown in Figure 21. We can see that the image as a whole presents a random strategy with less time. However, when the number of capital partitions is small, the assignment strategy mining effect is better.

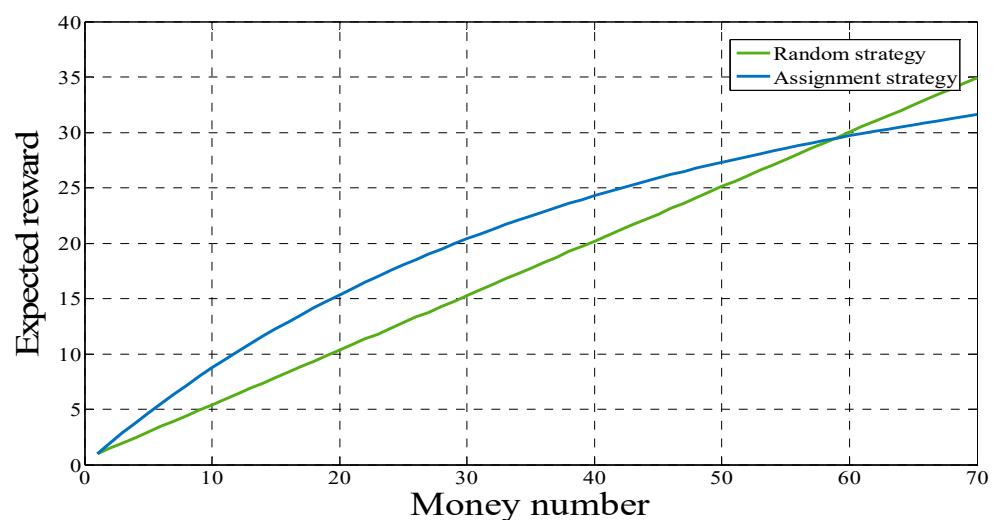


Figure 20. Comparison of the reward of assignment strategy and random strategy.

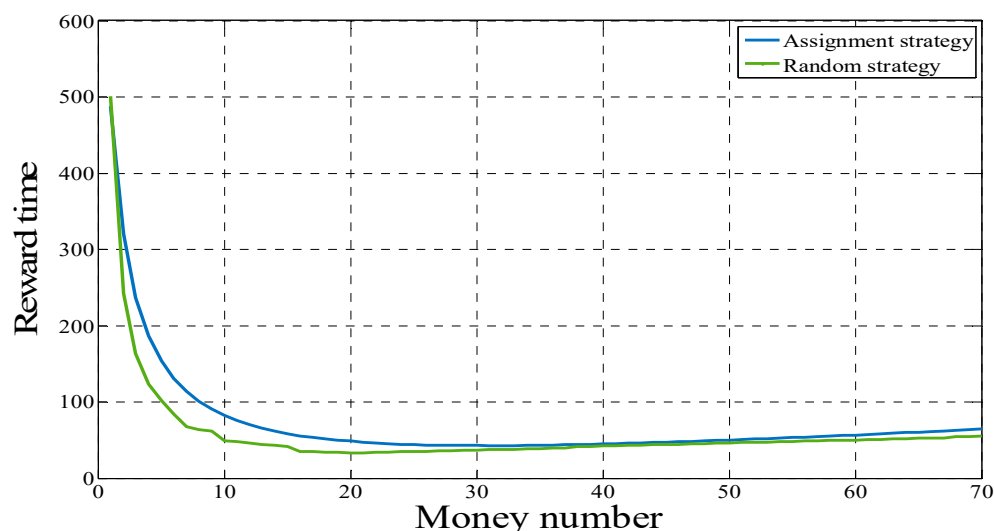


Figure 21. Comparison of the time of assignment strategy and random strategy.

In the case of considering the optimal strategy in the presence of the coin locked mechanism to complete the expected reward target, for the assignment strategy, it takes 33 time steps to reach the profit 25, but theoretically, 33 time step A has a funds value of 0.8 and can gain the 26.4 reward. The actual reward is less than the theoretical reward; for the random strategy, the time required for A to achieve the target reward is less than the theoretical time. For example, if B uses all funds, the optimal state where A gains 25 reward is required for 26.4075 time steps, but A can only earn 21.126 reward at this time theoretically, and the actual reward is greater than the theoretical reward. Therefore, for the assignment strategy, there will be no situation where the rich are richer and the poor are poorer, but this happens for random strategy.

4.2. A and B Are Rational Miners

Through the analysis in Section 4.1, we get the optimal fund allocation strategy of rational miners under the honesty miners, and find that the optimal strategies are better than the general strategy, and the mining effect using the random strategy is better. Next, we consider that both sides are rational miners in two different strategies. Here we consider two cases: the number of funds $A < B$ and $A = B$. First, let us first observe the results of both parties use the optimal strategy.

4.2.1. Assignment Strategy

Here we first analyze the use of assignment strategies by both parties. Here, we give the capital ratio α , and use the optimal fund allocation strategy solved by the KKT condition in Section 4.1.1 to determine the number of capital partitions m and $\alpha_1, \alpha_2, \dots, \alpha_m$. This method is used for both A and B to solve the allocation scheme. For each step A and B, the funds corresponding to the time step are used to participate in the mining. If there is no fund in the corresponding step, the mining is not involved.

We consider: (1) A and B funds are both 0.5; (2) A funds are 0.7, B funds are 0.3; (3) A funds are 0.3, B funds are 0.7, and other funds are allocated. We calculated through the above analysis model: When A's fund is 0.7, the actual reward of A is 16.3049, and the theoretical reward is 13.1602. When the fund is 0.3, the actual reward of A is 3.6951, the theoretical reward is 5.1074; when the fund is 0.5, A's actual reward was 11.6770, and the theoretical reward was 8.6430. The image is shown in Figure 20. We can find that the actual expected reward of A is lower than the theoretical expected reward, that is, A cannot achieve the expected mining effect.

4.2.2. Random Strategy

Here, we consider the case where both A and B use a random strategy. They all use the method of assignment strategy to determine the initial allocation of funds, that is, to determine the number of shares m and $\alpha_1, \alpha_2, \dots, \alpha_m$. From the perspective of A, the capital invested by B in each mining operation is $1-\alpha$.

We consider: (1) A and B funds are both 0.5; (2) A funds are 0.7, B funds are 0.3; (3) A funds are 0.3, B funds are 0.7, and other funds are allocated. We calculated that when A's funds are 0.7, A's actual reward is 16.3450, and the theoretical reward is 20.4384. When the funds are 0.3, A's actual reward is 7.7923 and the theoretical reward is 18.05576; when the funds are 0.5, A's actual reward is 12.1182 and the theoretical reward is 19.7272. See Figure 21. Obviously, the actual reward of A has a certain degree of decline compared with the theoretical reward, so A cannot achieve the expected mining effect. Moreover, there is an unexpected situation here. The actual reward of A and B are lower than the theoretical reward, and there is a vicious competition between the two parties, showed in Figures 22 and 23.

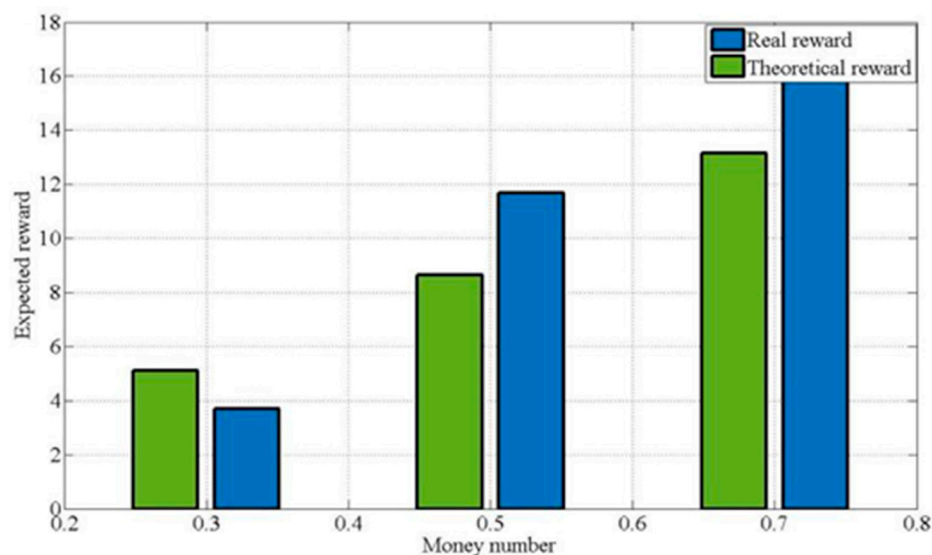


Figure 22. Comparison of actual reward and theoretical reward of A under assignment strategy.

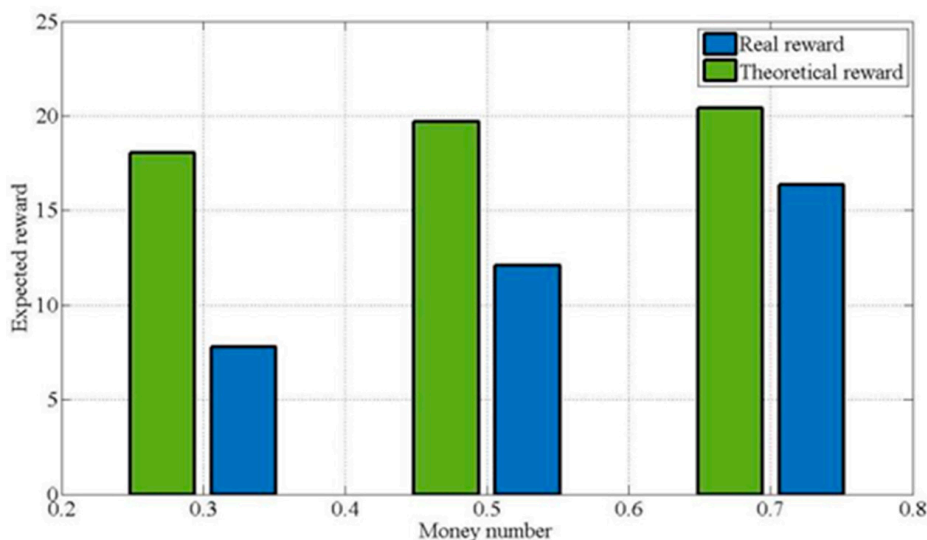


Figure 23. Comparison of actual reward and theoretical reward of A under random strategy.

4.3. The Optimal Strategy Under B Is Rational

It can be seen from the above analysis that for assignment strategy, there will be a game in the ancient Chinese classic story “Tianji Racing”. If A has less total capital than B, both A and B perform the above optimal strategy, and there will be A’s age of coin less than B’s for each time step, and the actual reward of A will not reach the expected reward. Therefore, we will analyze the strategy that A can use to obtain the maximum expected reward under the above-mentioned optimal strategy. The KKT condition is still applicable, we can also consider using MDP or Bellman-Ford algorithm [21] to solve, but the algorithm complexity is high; this method can also solve the n -parties game [22], which will not be described in detail here.

The MDP can be used to solve continuous decision problems. We can analyze that at each time step, the state space S is the remainder of the A funds $< \alpha_1, \alpha_2, \dots, \alpha_m >$; the action space A is how to choose among the remaining funds; for the transition probability P ,

$$P_A = \frac{\alpha_i[T_1 + (i-1)t]}{\alpha_i[T_1 + (i-1)t] + b_i[T_2 + (i-1)t]} \quad (20)$$

$$P_B = \frac{b_i[T_2 + (i-1)t]}{\alpha_i[T_1 + (i-1)t] + b_i[T_2 + (i-1)t]} \quad (21)$$

where $P_A + P_B = 1$, in reward matrix R , the reward matrix available for each time step of A and B is $R_A = (R, 0)$, $R_B = (0, R)$, and the depreciation rate γ is $\gamma = 1$. At this point, we construct $M = < S, A, P, R, \gamma >$ in order to maximize the expected reward of A. The expected reward of each step of A is $R_i = R * P_i$; then, the depreciation reward of A is expected to be $R_{A_total} = \sum_{i=1}^m (\gamma^{i-1} \times R_i)$.

However, for the random strategy, if A and B invest all of the funds each time, the actual reward of A and B will be lower than expected, and A and B have no other suitable separate strategy to increase their own profits. In the long run for A and B, cooperation may be the optimal strategy.

In the n -party game, A can use the analysis model of the two-party game to find the optimal fund allocation strategy. The difference is that the calculation formula of the mining success rate of i -th step is changed to $\frac{\alpha_i(T_1 + (i-1)t)}{\alpha_i(T_1 + (i-1)t) + \sum_{j=1}^{n-1} \beta_{ij}(T_{ij} + (i-1)t)}$, where α_i is the input of the i -th step the number of funds, β_{ij} is the number of funds invested by the j -th person in the remaining $n - 1$ people in the i -th step, and T_{ij} is the accumulated time of age of coin in the corresponding fund number. At this point we can still use the KKT condition to solve A’s optimal fund allocation strategy. Note that there is some randomness here, because the remaining $n - 1$ people are a combination of rational miners and honest miners, and their capital investment is uncertain. Similarly, the MDP and Bellman-Ford algorithm also apply to the n -party game.

4.4. Future Work

While this paper focuses on the rigorous quantitative analysis of the two-party game, the framework lays the foundation for multi-party interactions. The extension to an n -party scenario presents higher computational complexity, potentially requiring Markov decision processes or Reinforcement Learning approaches to solve for equilibrium states. We identify the modeling of n -player dynamic competition as a critical direction for future research, building upon the constraints and objective functions established in this work.

5. Conclusions

To the best of our knowledge, this article is the first paper to quantitatively analyze the PoS consensus mechanism. Under the premise of not performing attack behavior, we use the expected reward and the reward ratio as the evaluation indicators, and quantitatively analyze the fund allocation strategy of the two parties game under the PoS consensus mechanism from the perspective of the rich miners. We consider the assignment strategy and the random strategy. In both scenarios, the optimal fund allocation strategy under the PoS consensus was obtained. Comparing with the general strategy, we find that the assignment strategy is less optimized for general strategy and the random strategy is larger. At the same time, we find that for the assignment strategy, mining activities will not appear that the rich are richer and the poor are poorer; but for random strategy, this will happen. At the same time, the assignment strategy will better meet the needs of users, and the optimization strategy has limited effect on the optimization of general strategy. Developers can refer to Bitcoin's use of computing power to defend against Sybil attacks, or to pay a certain percentage of margin, limit the same equipment or IP running mining procedures, etc., as much as possible; the user cannot make multiple money and simultaneously invest in the mining process. The work of this paper helps the blockchain developers to analyze the PoS consensus mechanism, and the adoption model of the assignment strategy and the random strategy can be a future research direction.

Author Contributions: Methodology, B.Y.; Validation, J.H.; Formal analysis, Y.M.; Resources, L.Z.; Writing—original draft, B.Y.; Writing—review & editing, J.H., Y.M. and L.Z.; Visualization, B.Y., J.H. and Y.M.; Supervision, L.Z.; Project administration, Y.M.; Funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Program of Shanxi Province (No.202403021222070).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Scientific Research Publishing Inc.: Wuhan, China, 2008.
2. Wood, G. Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper. *Sci Res.* **2014**, *151*, 1–32.
3. King, S.; Nadal, S. PPcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. Self-Published Paper. 2012. Available online: <https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf> (accessed on 1 February 2026).
4. Bentov, I.; Gabizon, A.; Mizrahi, A. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2016.
5. Nxt Community. *Nxt Whitepaper, Revision 4—Nxt v1.2.2*; Nxt Community: Amsterdam, The Netherlands, 2014.
6. Dai, P.; Mahi, N.; Earls, J.; Norta, A. Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. 2017. Available online: <https://resources.cryptocompare.com/asset-management/140/1681209571371.pdf> (accessed on 1 February 2026).
7. Kwon, J. Tendermint: Consensus Without Mining. 2015. Available online: <http://tendermint.com/docs/tendermint.pdf> (accessed on 1 February 2026).
8. Zamfir, V. Introducing Casper “The Friendly Ghost”—Ethereum Blog. 2015. Available online: <https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/> (accessed on 1 February 2026).
9. Heilman, E.; Al Shenibr, L.; Baldimtsiy, F.; Scafuroz, A.; Goldberg, S. *TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub*; National Diabetes Services Scheme (NDSS): Sydney, Australia, 2017.
10. Ruffing, T.; Moreno-Sanchez, P.; Aniket, K. *P2P Mixing and Unlinkable Bitcoin Transactions*; National Diabetes Services Scheme (NDSS): Sydney, Australia, 2017.

11. Ruffing, T.; Moreno-Sanchez, P.; Aniket, K. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In Proceedings of the 19th European Symposium on Research in Computer Security, Wroclaw, Poland, 7–11 September 2014; pp. 345–364.
12. Ruffing, T.; Moreno-Sanchez, P. ValueShuffle: Mixing Confidential Transactions: Comprehensive Transaction Privacy for Bitcoin. In *Financial Cryptography and Data Security (FC), Proceedings of the 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, 7 April 2017*; International Financial Cryptography Association: Amsterdam, The Netherlands, 2017; pp. 133–154.
13. Eyal, I.; Sirer, E.G. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Financial Cryptography and Data Security (FC), Proceedings of the 2014 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Christ Church, Barbados, 7 March 2014*; International Financial Cryptography Association: Amsterdam, The Netherlands, 2014; pp. 436–454.
14. Nayak, K.; Kumar, S.; Miller, A.; Shi, E. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In Proceedings of the Security and Privacy Symposium (EuroS&P), Paris, France, 26–28 April 2016.
15. Courtois, N.T.; Bahack, L. On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency. *arXiv* **2014**, arXiv:1402.1718. [[CrossRef](#)]
16. Kwon, Y.; Kim, D.; Son, Y.; Vasserman, E.; Kim, Y. Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 30 October–3 November 2017; pp. 195–209.
17. Carlsten, M.; Kalodner, H.; Weinberg, S.M.; Narayanan, A. On the Instability of Bitcoin Without the Block Reward. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria, 24–28 October 2016; pp. 154–167.
18. Bertsekas, D.P. *Nonlinear Programming*; Athena Scientific: Belmont, CA, USA, 1999; pp. 191–276.
19. Karush-Khun-Tucker Conditions. Available online: https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions (accessed on 1 February 2026).
20. Wolsey, L.A. *Mixed Integer Programming*; Wiley Encyclopedia of Computer Science and Engineering: Hoboken, NJ, USA, 2007; pp. 1–10.
21. Bellman, R. On a routing problem. *Q. Appl. Math.* **1958**, *16*, 87–90. [[CrossRef](#)]
22. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 157–163.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.