2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

Multi-miner's Cooperative Evolution Method of Bitcoin Pool Based on Temporal Difference Leaning Method

Wei Ou
Department of Electronic and Information Engineering
Hunan University of Science and Engineering
Yongzhou, China
ouwei1978430@163.com

Mingwei Deng
Department of Electronic and Information Engineering Hunan University of Science and Engineering
Yongzhou, China
sherdan@126.com

Entao Luo
Department of Electronic and Information Engineering
Hunan University of Science and Engineering
Yongzhou, China
cs_entaoluo@csu.edu.cn

Wei Shi
School of Information Technology
Carleton University
Ottawa, Canada
wei.shi@carleton.ca

Zhiyuan Tan
School of Computing Edinburgh Napier University Edinburgh,
United Kingdom
z.tan@napier.ac.uk

Md Zakirul Alam Bhuiyan
Department of Computer and Information Sciences
Fordham University
New York, America

***Abstract*—Proof of Work (PoW) is used to provide a consensus mechanism for Bitcoin. In this mechanism, the process of generating a new block in the blockchain is referred to as mining. Such process is intentionally designed to be resource-intensive and time consuming so that the rate of block generation remains steady. A single participant, called a miner usually has limited computation power to produce PoWs. This leads miners to form a mining pool, where miners aggregate their computing power and share the rewards. However, a phenomenon raises in such a mining pool activity, where miners attack each other. Consequently, this results in a decrease in total rewards received from the mining pool. To address the abovementioned problem, we build a multi-miner model for forming a mining pool. We further propose a method to improve the cooperation-probability of miners in the pool by introducing a Zero-Determinant strategy and a Temporal Difference learning method (TD(λ)). Experimental simulation results show that the proposed method can effectively promote the cooperation among miners, therefore, increase the rewards received from the formed mining pool.**

Keywords—Bitcoin, Blockchain, Temporal Difference Learning Method, Zero-Determinant Strategy, Block withholding Attack

I. INTRODUCTION

Bitcoin [1] is currently the most successful blockchain application in the world, which possess the largest number of users, the largest system, and the most stable transactions. It provides a novel consensus mechanism, called Proof of Work (PoW), which directly led to the birth of blockchain technology. PoW describes a secure accounting system that solves the Byzantine Problem by introducing a hash rate (the computing power in Bitcoin network) competition of distributed nodes to ensure data consistency and consensus. The nodes who participate in the competition are called miners, and the process of computing is called mining. As the Bitcoin system becomes larger, the possibility of a single miner mining successfully becomes smaller. In order to get more rewards, the mining pool, where several miners aggregate their own hash rate and share the overall rewards, has appeared.

Studies have shown that in an open pool, miners can implement block withholding attack to increase their own profits. From the perspective of game theory based on hypothesis of rational man, all miners will eventually choose to attack each other, but they will receive less revenue than not* attacking each other. This is the dilemma of the miners under PoW. We can compare it with the classic Prisoner's Dilemma [2] in game theory, which represents such a situation that the optimal strategy for the individual is not the overall optimal strategy. Therefore, we can analyze and optimize the miner's dilemma from the perspective of game theory.

Zero-Determinant Strategies (ZD strategies) is one of the hot trends in current game theory research. Originating from the papers published by Press and Dyson [5] , they indicates that there is such a strategy in the iterative prisoner's dilemma: a single prisoner is able to unilaterally pin his opponent's payoff or to enforce a liner relationship between his own payoff and his opponent's payoff. On the basis of Press and Dyson, Pan et al applied the ZD strategy in a multi-player game, and proved that the ZD strategy can enforce the linear relationship between the payoffs of us and all our opponents [6] . This brings great inspiration to the writing of this paper.

This paper build a multi-miner game model and uses the multi-ZD strategy to optimize Bitcoin pool, achieving the full cooperation of miners, thus increasing the overall rewards of pool. In order to acquire the best strategy of the game and converge the overall cooperation probability of pool to 1 within the shortest number of iterations, we treat each pool as an agent, using the Temporal Difference learning method to predict next round payoffs, and choose strategy of next round by comparing them, meanwhile changing the cooperation probability of pool.

II. RELATED WORK

A. Block withholding attack

Mining pool is consisted of pool manager and several miners. The pool manager joins the Bitcoin system as a singer miner. Instead of searching the specific nonce value, he outsources the work to the miners. Every miner in the pool will be assigned partial nonce value search task, which called partial proof of work. And the pool manager evaluates miners' effort by estimating each miner's power with partial proof of work that be submitted. Once the specific nonce value, which

* Supported by the construct program of applied characteristic discipline in Hunan University of Science and Engineering

978-1-7281-2980-8/19/$31.00 ©2019 IEEE
DOI 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00131

called full proof of work, is generated by any miner in the pool, this miner sends it to the pool manager. Subsequently, the pool manager publishes the nonce value to the Bitcoin system. Finally, the pool manager receives the full revenue of the block and distributed it fairly according to the miners' power [7] .

Since most of the pool is open, allowing any miners to join, therefore any pool could infiltrate another by sending his miners to implement block withholding attack [8] . The concept of block withholding attack is that the attacker joins the pool, only sends partial proof of work to the pool manager but discards full proof of work. Due to the partial proof of work the attack delivered, pool manager considers that he is an honest miner and estimate his power. Therefore, the attacker enjoys the rewards of pool but does not actually contribute, which resulting in lower revenue of all miner in the pool including himself.

B. Prisoner's dilemma

Prisoner's dilemma was first proposed by Kuh et al [9] . In prisoner's dilemma, two agents have to choose cooperation (C) or betrayal (D) in the circumstance of not knowing the information of his opponent. Both parties get the payoffs R when they all cooperate, and get the payoffs P when they all betrayed; One party cooperates, the other party betrayed, the betrayer gets the maximum payoffs T, and the cooperator gets the minimum payoffs S. The parameters satisfy the condition T > R > P > S. Under this circumstance, the best strategy of each agent is betrayal. However, if two agents betray each other together, they will both get the payoffs P, which is less than they both cooperate. How to solve the dilemma is the main problem in prisoner's dilemma research.

C. Zero-Determinant strategies

For prisoner's dilemma, many scholars proposed many strategies: WSLS (win stay, lost shift), TFT (tit-for-tat), GTFT (generous tit-for-tat), no-memory full cooperation strategy and full betrayal strategy. However, none of these strategies can determine the opponent's payoffs unilaterally until Press and Dyson first proposed Zero-Determinant Strategies in 2012, which can not only unilaterally determine the opponent's payoffs, also ensure that our payoffs are multiple of opponents' payoffs, thus achieving the purpose of extortion. Its advantages have been widely concerned by many scholars.

D. Temporal difference leaning method

Temporal-Difference learning method that based on linear value function can be traced back to 1988. Linear Temporal Difference (LTD) and TD(λ) method were first proposed from Sutton. In Sutton's paper, The Temporal-Difference learning is used as a multi-steps predictive learning method based on Markov chain to solve the policy evaluation or value function prediction problem of the smoothing Markov decision process, So far, scholars have been studying and improving such learning method. This paper uses the TD(λ) [10] algorithm to predict next round payoffs in the miners' game, and then compares payoffs of different strategy to choose a more profitable strategy.

III. MULTI-MINER GAME

A. Miner's dilemma in multi-miner game

In Bitcoin's pool, miner's calculation task for the specific nonce value assigned by the pool manager would consume a certain amount of hash rate, assuming that the resource consumed by this part is $c(c>0)$. If multiple miners choose to cooperate in mining, the probability of finding the specific nonce value will increase greatly, that is, the expected payoffs of each miner will be greater than they mining alone. We assume that if miners mining cooperatively, the expected payoffs of the pool will be expanded r times, and r is a value greater than one. The pool manager will distribute the mining revenue according to the hash rate of each miners. For miners who implement block withholding attacks, the pool manager will also distribute the revenue according to their hash rate.

Due to the complexity of the multi-party game, we will only discuss the fact that each miner in the pool has equal hash rate in this paper. Assuming the total hash rate of the pool with N miners is N, then the hash rate of each miner can be expressed as 1. Therefore, we can simply use the miner's hash rate to represent the miner's assigned payoffs. When a miner chooses to cooperate, he needs to consume a certain amount of hash rate, which is represented here by c as we defined above. Also, when multiple miners cooperate together, the overall payoffs of the pool will expand. We use r to denote the revenue expansion coefficient. However, there is a problem: with the increase of cooperative miners, the possibility of finding the full of proof will increase. That means the value of r should also increase with the number of cooperators and the growth curve should gradually become flatter. To solve the problem, we define $r=\ln(k+b)$, where k is number of cooperators and b is a constant.

Due to the high dimension of the payoffs constituted by N miners, it is hard to show a payoffs table to find the existing condition of multi-miner's dilemma. But from the definitions above, we still could find it.

According to the definition, for a miner in a pool with N miners, we can express his payoffs of cooperation and attack as:

- Cooperation:

$$\frac{(n+1)\ln(n+1+b)}{\mathrm{N}}-c$$

- Attack:

$$\frac{n\ln(n+b)}{\mathrm{N}}$$

Where n denotes the number of cooperators among his opponents. According to prisoner's dilemma, our miner is in such a scenario that the best strategy of he is attack whether his opponents cooperate or attack. In multi-miner game, which scenario is existed when:

$$\frac{n\ln(n+b)}{\mathrm{N}}>\frac{(n+1)\ln(n+1+b)}{\mathrm{N}}-c \qquad (1)$$

Solving (1), we have the feasible region for multi-miner miner:

$$\mathrm{N}>\frac{\ln\frac{(n+1+b)^{n+1}}{(n+b)^{n}}}{c} \qquad (2)$$

Where b and c are two constants, which means that the feasible region of N changes only as n changes. And analysis of (2) shows that the right part of the inequation is an increasing function of n, in this way we can find a certain region of N if we substitute the maximum n, which is $n_{\max} = \mathrm{N} - 1$ based on our assumption. Then we have a new inequation:

$$\mathrm{N} > \frac{\ln \frac{(\mathrm{N}+b)^{\mathrm{N}}}{(\mathrm{N}-1+b)^{\mathrm{N}-1}}}{c} \tag{3}$$

Through a simple transformation, the inequation will become:

$$e^{c\mathrm{N}} > \mathrm{N} + b \tag{4}$$

Next, we construct two function: $\mathrm{f}_1(\mathrm{N})=\mathrm{e}^{c\mathrm{N}}$ and $\mathrm{f}_2(\mathrm{N}) = \mathrm{N} + b$, where $0 < c < 1$ and $b \geq e - 1$ according to our definition. Finally, we could graph these two functions to find the feasible region in Fig. 1. From Fig. 1, it's easy to know that there is $\mathrm{N}_i \in (0,+\infty)$ meet $e^{c\mathrm{N}_i} = \mathrm{N}_i + b$, and $e^{c\mathrm{N}} > \mathrm{N} + b$ when $\mathrm{N} > \mathrm{N}_i$. $\mathrm{N} > \mathrm{N}_i$ is exactly the feasible region of N.

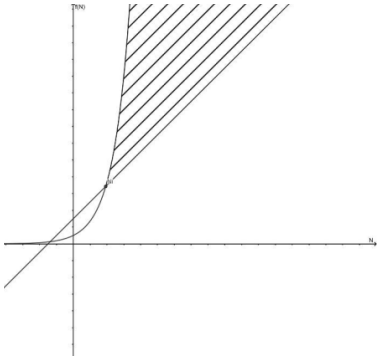

Fig. 1 Feasible region of N

B. Multi-miner model

Assuming there are N miners in the pool, miners cannot communicate with each other, and they independently decide whether to implement the block withholding attack. At this time, the action space of game is:

$$B = [\mathrm{C}, \mathrm{A}]$$

Next, we assume that the current round of miner's action is determined only by the state of previous round, then the repeated game between N miners can be regarded as a Markov chain.

In such a game, each round will have 2^{N} possible state outcomes. For example, when $\mathrm{N} = 3$, the game state space can be expressed as $S = [\mathrm{CCC}, \mathrm{CCA}, \mathrm{CAC}, \mathrm{CAA}, \mathrm{ACC}, \mathrm{ACA}, \mathrm{AAC}, \mathrm{AAA}]$. If N is too large, the game state space will be difficult to give a concrete expression. Therefore, we use s_i to refer to the specific game state outcome:

$$S = [s_1, \cdots, s_i, \cdots, s_{2\mathrm{N}}]$$

Also, for any miner in the pool, he will have a strategy vector:

$$p^x = \left[p_1^x, \cdots, p_i^x, \cdots, p_{2\mathrm{N}}^x\right]^T$$

Where p_i^x indicates the probability that an arbitrary miner chooses cooperation in the current round in the case where the final game outcome of the previous round is s_i. We take three-miners as an example, the strategy vector of an arbitrary miner in a pool with three miners, let's say miner 1's strategy vector, can be expressed as $p^1 = \left[p_1^1, p_2^1, p_3^1, p_4^1, p_5^1, p_6^1, p_7^1, p_8^1\right]^T$.

To clarify the state of p^x, we can further define the expression of p^x. In a pool game with N miners, participants can be divided into us, let's say miner1, and the opponents, which means those $\mathrm{N}-1$ other miners. The action of miner 1 and the action of his all opponents constitute a game state. If we know how many opponents have chosen to cooperate in the previous round, we can express the state of the previous round in the strategy vector p^x. Therefore, we define $p_{\mathrm{C},n}$ (or $p_{\mathrm{A},n}$) to indicate the probability that miner 1 cooperate in current round when it cooperates (or attack) while n opponents cooperate in the last round. This way, miner 1's strategy vector can be expressed as:

$$p^1 = \begin{bmatrix} p_{\mathrm{C},0}^1, \cdots, p_{\mathrm{C},n}^1, \cdots p_{\mathrm{C},n}^1, \cdots, p_{\mathrm{C},\mathrm{N}-1}^1, \\ p_{\mathrm{A},0}^1, \cdots, p_{\mathrm{A},n}^1, \cdots p_{\mathrm{A},n}^1, \cdots, p_{\mathrm{A},\mathrm{N}-1}^1 \end{bmatrix}^T$$

For example, in a pool with three miners, miner 1's strategy vector can be expressed as:

$$p^1 = \left[p_{\mathrm{C},2}^1, p_{\mathrm{C},1}^1, p_{\mathrm{C},1}^1, p_{\mathrm{C},0}^1, p_{\mathrm{A},2}^1, p_{\mathrm{A},1}^1, p_{\mathrm{A},1}^1, p_{\mathrm{A},0}^1\right]^T$$

Since there are 2^{N} outcomes of game state in the pool with N miners, we will have a payoff vector with 2^{N} corresponding sub-elements for each miner in the pool. Let's consider the miner's payoff vector:

$$u^x = \left[u_1^x, \cdots, u_i^x, \cdots, u_{2^{\mathrm{N}}}^x\right]^T$$

Where $x \in [1, \mathrm{N}]$. Recalling the definitions in last section, we can obtain the expression of u_i^x:

$$u_i^x = \frac{r\left[n(i) + h_i^x\right]}{\mathrm{N}} - h_i^x c$$

Where $n(i)$ denotes the number of cooperators among opponents in the game state s_i; h_i^x indicates our action in the game state s_i, if we choose Cooperate, then $h_i^x = 1$, otherwise, $h_i^x = 0$; Similarly, given the pool with three miners, the payoff vector of miner 1 is:

$$u^1 = \left[r - c, \frac{2r}{3} - c, \frac{2r}{3} - c, \frac{r}{3} - c, \frac{2r}{3}, \frac{r}{3}, \frac{r}{3}, 0\right]^T$$

Noticing that we use a fixed value of r rather than $r=\ln(k+b)$ for convenient calculation latter. This will not affect our derivation.

Next, we define the Markov state transition matrix of multi-miner game:

$$\mathrm{M}=\left[M_{i,j}\right]_{2^{\mathrm{N}}\times 2^{\mathrm{N}}}$$

Where $M_{i,j}$ refers to the transition probability of the pool moving from state i to state j (i and j are the indexes of old and new states respectively). Subsequently, according to the definition of the Markov state matrix, $M_{i,j}$ can be calculated by the following equation:

$$M_{i,j}=\prod_{x=1}^{\mathrm{N}} m^x$$

Where x represents any miners in the pool, further defined：

$$m^x=\begin{cases}\left(p^x_{\mathrm{C},n(i)}\right)^{h^x_j}\left(1-p^x_{\mathrm{C},n(i)}\right)^{1-h^x_j}, \textit{if miner x chooses cooperate in state i};\\ \left(p^x_{\mathrm{A},n(i)}\right)^{h^x_j}\left(1-p^x_{\mathrm{A},n(i)}\right)^{1-h^x_j}, \textit{if miner x chooses attack in state i};\end{cases}$$

Here $n(i)$ is the number of cooperators among opponents in state i; h^x_j indicates the action of us in state j. If we choose cooperate, then $h^x_j=1$, otherwise, $h^x_j=0$.

IV. Application of Multi Zero-Determinant Strategies in Multi-Miner Game

A. Multi zero determinant strategy

If M is a regular state transition matrix, it must have a unique stationary vector V^T. Here, we take the stationary vector V^T whose eigenvalue is 1, and get:

$$V^T\cdot\mathrm{M}=V^T \tag{5}$$

Now we define that:

$$\mathrm{M}'=\mathrm{M}-\mathrm{I} \tag{6}$$

Which is:

$$M'_{i,j}=\prod_{x=1}^{\mathrm{N}} m^x-\delta_{i,j} \tag{7}$$

Where $\delta_{i,j}$ is Kronecker delta, the specific expression is:

$$\delta_{i,j}=\begin{cases}0, if\ i\neq j;\\ 1, \text{if}\ i\neq j;\end{cases}$$

Next we do some elementary column operations for M′ [6], then we can separate the joint probabilities, leaving one column solely controlled under the strategy of miner x. We define the strategy of miner x after separated as $\tilde{p}^x$:

$$\tilde{p}^x=\begin{bmatrix}-1+p_{\mathrm{C},0},\cdots,-1+p_{\mathrm{C},n},\cdots,-1+p_{\mathrm{C},n},\cdots,\\ -1+p_{\mathrm{C,N-1}},p_{\mathrm{A},0},\cdots,p_{\mathrm{A,n}},\cdots,p_{\mathrm{A},n},\cdots,p_{\mathrm{A,N-1}}\end{bmatrix}$$

We can apply the Cramer rule to M′:

$$\mathrm{Adj}(\mathrm{M}')\mathrm{M}'=\det(\mathrm{M}')\mathrm{I}=0 \tag{8}$$

Meanwhile, combined with (5) and (6), we obtain:

$$V^T\cdot\mathrm{M}'=0 \tag{9}$$

Comparing (8) and (9), it is easy to know that each row of $\mathrm{Adj}(\mathrm{M}')$ is proportional to the stationary vector V^T. Therefore, for an arbitrary payoff vector u^x, there is:

$$V^T\cdot u^x=\sum_{j=1}^{\mathrm{N}}\mathrm{Adj}(\mathrm{M}')_{ij}u^x_j=a\sum_{j=1}^{\mathrm{N}} m'_{j,i}u^x_j \tag{10}$$

Where $m'_{i,j}$ denotes the minor of $M'_{j,i}$. From (10), we obtain a determinant of the matrix which by replacing the i^{th} column of the matrix M′ with the payoff vector u^x. Assuming that i is the last column:

$$V^T\cdot u^x=\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},u^x\right) \tag{11}$$

Where $\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},u^x\right)$ is a determinant of $2^{\mathrm{N}}\times 2^{\mathrm{N}}$ matrix. We give the $V^T\cdot u$ of the three-miner game in Fig 2.

Using (11), we can do a Laplace expansion on the last column of $V^T\cdot u^x$ (i.e. u^x) to find the long-term expected payoffs E^x of any miner in the pool, thus we have the formula of expected payoffs:

$$\begin{aligned}E^x&=\frac{V^T\cdot u^x}{V^T\cdot 1}\\&=\frac{\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},u^x\right)}{\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},1\right)}\end{aligned} \tag{12}$$

Where 1 is an all-one vector introduced for normalization.

Analysis of (12) shows that the expected payoffs of miner is linearly related to his payoff vector. A linear combination of all miners' expected payoffs is obtained, and expressed as:

$$\begin{aligned}&\sum_{x=1}^{\mathrm{N}}\alpha_x E^x+\alpha_0\\&=\frac{\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},\sum_{x=1}^{\mathrm{N}}\alpha_x u^x+\alpha_0 1\right)}{\det\left(p^1,\cdots,p^x,\cdots,p^{\mathrm{N}},1\right)}\end{aligned} \tag{13}$$

Recalling the transformation of matrix M′, we know that there is a column $\tilde{p}^x$ that unilaterally determined by a specific miner. Also, we call the specific miner miner 1. If miner 1 sets his own $\tilde{p}^1=\sum_{x=1}^{\mathrm{N}}\alpha_x u^x+\alpha_0 1$, he can unilaterally make the determinant of (13) vanish and enforce a linear relationship between all miners' expected payoffs:

$$\sum_{x=1}^{\mathrm{N}}\alpha_x E^x+\alpha_0=0 \tag{14}$$

$$V^T\cdot u=\mathrm{D}\left(p^1,p^2,p^3,u\right)=\det\begin{bmatrix}
-1+p^1_{\mathrm{C},2}p^2_{\mathrm{C},2}p^3_{\mathrm{C},2} & -1+p^1_{\mathrm{C},2}p^2_{\mathrm{C},2} & -1+p^1_{\mathrm{C},2}p^3_{\mathrm{C},2} & -1+p^1_{\mathrm{C},2} & -1+p^2_{\mathrm{C},2}p^3_{\mathrm{C},2} & -1+p^2_{\mathrm{C},2} & -1+p^3_{\mathrm{C},2} & u_1\\
p^1_{\mathrm{C},1}p^2_{\mathrm{C},1}p^3_{\mathrm{A},2} & -1+p^1_{\mathrm{C},1}p^2_{\mathrm{C},1} & p^1_{\mathrm{C},1}p^3_{\mathrm{A},2} & -1+p^1_{\mathrm{C},1} & p^2_{\mathrm{C},1}p^3_{\mathrm{A},2} & -1+p^2_{\mathrm{C},1} & p^3_{\mathrm{A},2} & u_2\\
p^1_{\mathrm{C},1}p^2_{\mathrm{A},2}p^3_{\mathrm{C},1} & p^1_{\mathrm{C},1}p^2_{\mathrm{A},2} & -1+p^1_{\mathrm{C},1}p^3_{\mathrm{C},1} & -1+p^1_{\mathrm{C},1} & p^2_{\mathrm{A},2}p^3_{\mathrm{C},1} & p^2_{\mathrm{A},2} & -1+p^3_{\mathrm{C},1} & u_3\\
p^1_{\mathrm{C},0}p^2_{\mathrm{A},1}p^3_{\mathrm{A},1} & p^1_{\mathrm{C},0}p^2_{\mathrm{A},1} & p^1_{\mathrm{C},0}p^3_{\mathrm{A},1} & -1+p^1_{\mathrm{C},0} & p^2_{\mathrm{A},1}p^3_{\mathrm{A},1} & p^2_{\mathrm{A},1} & p^3_{\mathrm{A},1} & u_4\\
p^1_{\mathrm{A},2}p^2_{\mathrm{C},1}p^3_{\mathrm{C},1} & p^1_{\mathrm{A},2}p^2_{\mathrm{C},1} & p^1_{\mathrm{A},2}p^3_{\mathrm{C},1} & p^1_{\mathrm{A},2} & -1+p^2_{\mathrm{C},1}p^3_{\mathrm{C},1} & -1+p^2_{\mathrm{C},1} & -1+p^3_{\mathrm{C},1} & u_5\\
p^1_{\mathrm{A},1}p^2_{\mathrm{C},0}p^3_{\mathrm{A},1} & p^1_{\mathrm{A},1}p^2_{\mathrm{C},0} & p^1_{\mathrm{A},1}p^3_{\mathrm{A},1} & p^1_{\mathrm{A},1} & p^2_{\mathrm{C},0}p^3_{\mathrm{A},1} & -1+p^2_{\mathrm{C},0} & p^3_{\mathrm{A},1} & u_6\\
p^1_{\mathrm{A},1}p^2_{\mathrm{A},1}p^3_{\mathrm{C},0} & p^1_{\mathrm{A},1}p^2_{\mathrm{A},1} & p^1_{\mathrm{A},1}p^3_{\mathrm{C},0} & p^1_{\mathrm{A},1} & p^2_{\mathrm{A},1}p^3_{\mathrm{C},0} & p^2_{\mathrm{A},1} & -1+p^3_{\mathrm{C},0} & u_7\\
p^1_{\mathrm{A},0}p^2_{\mathrm{A},0}p^3_{\mathrm{A},0} & p^1_{\mathrm{A},0}p^2_{\mathrm{A},0} & p^1_{\mathrm{A},0}p^3_{\mathrm{A},0} & p^1_{\mathrm{A},0} & p^2_{\mathrm{A},0}p^3_{\mathrm{A},0} & p^2_{\mathrm{A},0} & p^3_{\mathrm{A},0} & u_8
\end{bmatrix}$$

Fig. 2 $V^T\cdot u$ of the three-miner game

At this time, miner 1's strategy vector $\tilde{p}^1$ is exactly the zero determinant strategy under the multi-miner game.

B. Extortion strategy

In the extortion strategy, the purpose of miner 1 is to use a certain strategy to make his payoffs is χ times sum of his opponents' payoffs. We set miner 1 to implement such a strategy:

$$p^1=\Phi\left[\left(u^1-P1\right)-\chi\sum_{x=2}^{\mathrm{N}}\left(u^x-P1\right)\right] \tag{15}$$

Where P denotes the payoffs when all the miners in the pool choose attack. Recall the miner's payoff vector u_i^x, it is clear that under the full attack state, the payoffs of each miner are 0, therefore (15) can be written as:

$$p^1=\Phi\left[u^1-\chi\sum_{x=2}^{\mathrm{N}}u^x\right] \tag{16}$$

From the strategy vector in (16), we can obtain the relation between the expected payoffs of miner 1 and all his opponents:

$$\Phi\left[E^1-\chi\sum_{x=2}^{\mathrm{N}}E^x\right]=0 \tag{17}$$

After a simple transformation:

$$E^1=\chi\sum_{x=2}^{\mathrm{N}}E^x \tag{18}$$

According to (18), if miner 1 uses the extortion strategy, he can unilaterally set the extortion factor χ to control that his expected payoffs are χ times the total expected payoffs of his opponents. Under this linear relationship, the best strategy of each miner in the pool will be full cooperation, thus we solve the multi-miner's dilemma.

V. Temporal Difference learning method with Zero-Determinant strategy in multi-miner game

Below we combine the Temporal Difference learning method with the Zero-Determinant extortion strategy to make the pool fully cooperative.

According to (18), the extortion factor χ have the following definition: multiples of our expected payoffs and our opponents' sum . If we set a fixed value χ, which can guarantee that we get higher payoffs, but this is not conducive to achieve the full cooperation state in pool. Based on this consideration, we set χ as a dynamic extortion factor:

$\chi=\dfrac{10}{P_{\mathrm{C}}}$, where P_{C} denotes the cooperation probability of all miners in the pool, and $P_{\mathrm{C}}\in[0,1]$. Which means the pool will achieve full cooperation state if $P_{\mathrm{C}}=1$. When P_{C} is small, we improve χ to guarantee that we can obtain high payoffs; When P_{C} is large, we lower χ, forcing the pool to achieve a full cooperation state; Once the pool is in full cooperation state, that is, P_{C} converges to 1, the extortion factor χ will evolve into a constant and continue to maintain the full cooperation state of the pool.

In reality, there are numerous miners in the pool, and the number of state space and expected payoffs of the pool will increase linearly with the number of miners, which makes it extremely difficult to give an explicit expression of the expected payoff formula. Therefore, we use the pool in full cooperation state, which means $p^x=1$ (and $x\in[2,\mathrm{N}]$) in the pool, to predict the pool's payoffs and to simulate the evolution trend of P_{C}, we give the expected payoff formula of us and opponents:

$$E^1=\frac{\det\left(1,\cdots,1,\cdots,1,u^1\right)}{\det\left(1,\cdots,1,\cdots,1,1\right)}$$

$$\sum_{x=2}^{\mathrm{N}}E^x=\frac{\det\left(1,\cdots,1,\cdots,1,\sum_{x=2}^{\mathrm{N}}u^x\right)}{\det\left(1,\cdots,1,\cdots,1,1\right)}$$

In t -round, we use $E_{adp}(t)$ represents our payoffs, $E_{opp}(t)$ represents opponents' payoffs. Meanwhile, our expected payoffs of cooperation and attack are represented by $E_{coo}(t)$ and $E_{att}(t)$ respectively. Therefore, in $t+1$ round, our expected payoff formula can be expressed as:

$$E_{coo}(t+1)=V_{\mathrm{C}}(t+1)+E^1$$

$$E_{att}(t+1)=V_{\mathrm{C}}(t+1)+\sum_{x=2}^{\mathrm{N}}E^x$$

$$V_{\mathrm{C}}(t+1)=\sum_{i=1}^{t}P_{\mathrm{C}}^{t-i}\left(E_{adp}(i)-E_{opp}(i)\right)$$

$$V_{\mathrm{A}}(t+1)=\sum_{i=1}^{t}P_{\mathrm{A}}^{t-i}\left(E_{adp}(i)-E_{opp}(i)\right)$$

Where the cooperation probability P_C and the attack probability P_A are both $x \in [0,1]$.

The strategy of next round is determined by comparing the expected payoffs of attack and cooperation:

1) If $E_{coo}(t+1) > E_{att}(t+1)$, miner chooses to cooperate. Meanwhile, the cooperation probability of next round $P_C(t+1) = P_C(t) + F(P_C(t+1))$ and the attack probability of next round $P_A(t+1) = P_A(t) - F(P_A(t+1))$.

2) If $E_{coo}(t+1) < E_{att}(t+1)$, miner chooses to attack. Meanwhile, the cooperation probability of next round $P_C(t+1) = P_C(t) - F(P_C(t+1))$ and the attack probability of next round $P_A(t+1) = P_A(t) + F(P_A(t+1))$.

3) If $E_{coo}(t+1) = E_{att}(t+1)$, miner chooses to cooperate. Meanwhile, the cooperation probability of next round $P_C(t+1) = P_C(t) + F(P_C(t+1))$ and the attack probability of next round remains the same.

Where Fermi function $F(\varepsilon)$ is expressed as

$$F(\varepsilon(t+1)) = \frac{1}{1+\exp\left[\frac{\varepsilon(t)-\varepsilon(t-1)}{k}\right]}$$

The extortion factor χ vary with P_C during the algorithm iteration process, which will affect the expected payoffs. Based on the hypothesis of rational man, the opponents will realize that cooperation is the optimal strategy and choose to cooperate in subsequent rounds. With enough iterations, P_C will eventually converge to 1, thus achieve full cooperation state.

VI. Simulation and Experiment

To test the validity of our application, our experiment simulates the evolution of cooperation probability under scale-free network environment in pool with three miners. Furthermore, we design control experiment, setting the initial cooperation probability of pool as 0.1, 0.3, 0.5, 0.7, 0.9 respectively to test the iteration rounds required to converge to 1 under different initial cooperation probability. This paper takes the data from first 40 rounds, predict the cooperation payoffs and attack payoffs of each rounds and change the cooperation probability. Subsequently, we compared rounds required to converge to 1 of our strategy with the adaptive strategy.

We show the first 20 rounds of cooperation probability evolution in Fig. 3. As the Fig. 3 shown, the cooperation probability of pool has an overall increasing tendency with the increasement of iteration rounds. Meanwhile, the smaller the initial probability of cooperation, the fewer rounds required to converge to1. After 6 rounds, all cooperation probabilities converge to 1.

We show the comparation of rounds required to converge between our strategy and adaptive strategy in Fig. 4. No matter what the initial cooperation probability is, rounds required of our strategy are all 1-3 rounds less than the adaptive strategy, which means our strategy has better performance than adaptive strategy.

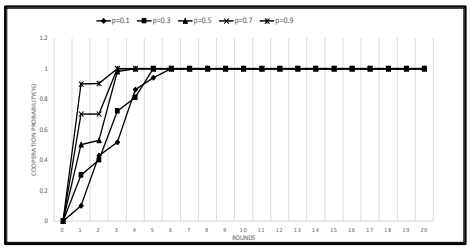

Fig. 3 Evolution of cooperation probability

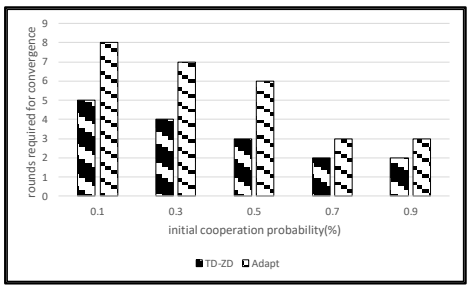

Fig. 4 Comparison of TD algorithm with ZD strategy and adaptive strategy on rounds required to converge cooperation probability to 1

We show the evolution of payoffs with different initial cooperation probability in Fig. 5-9. As shown, the cooperation payoffs are always higher than attack payoffs, therefor a rational miner will always choose to cooperate. Note that the cooperation payoffs actually did not converge until the cooperation probability converged if improve data accuracy. And the attack payoffs have different convergence value under different initial cooperation probability. Fig. 6 eventually converge in the 139th round. Fig. 7 eventually converge in 35th rounds. The initial cooperation probability of Fig. 8 is 0.5, and it converge in 11th rounds. Fig. 9 and Fig. 10 converge in 9th and 5th rounds respectively. Note that the bigger initial cooperation probability, the less rounds required to converge to a fixed value. Besides, with the convergence of cooperation probability, the revenue of whole pool will increase, and the performance of Bitcoin system will increase, too.

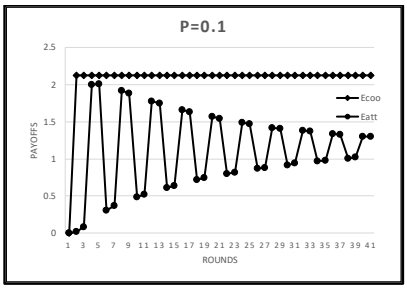

Fig. 5 Evolution of payoffs with initial cooperation probability is 0.1

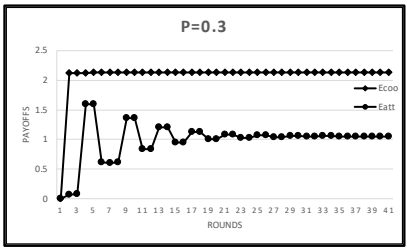

Fig. 6 Evolution of payoffs with initial cooperation probability is 0.3

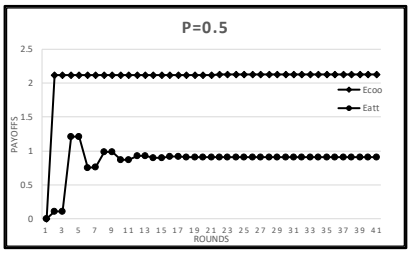

Fig. 7 Evolution of payoffs with initial cooperation probability is 0.5

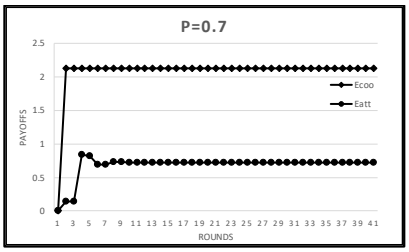

Fig. 8 Evolution of payoffs with initial cooperation probability is 0.7

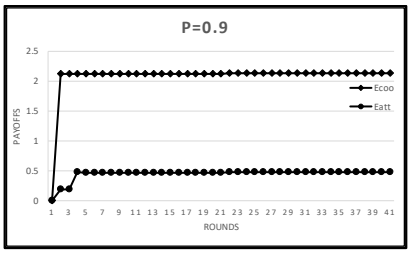

Fig. 9 Evolution of payoffs with initial cooperation probability is 0.9

VII. Conclusion and Futurework

In this paper, we build a multi-miner game model for a given mining pool based on PoW. We provide a new solution to the problem of block withholding attack in Bitcoin pool by introducing a Zero-Determinant strategy and a temporal difference learning method. Specifically, to analyze the game situation between miners, we regard the game between miners as an iterative prisoner's dilemma. We build a game model with multi-miners, and use the Zero-Determinant strategy to extort the opponents by extortion factor χ. Subsequently, we use the temporal difference learning method to predict the payoffs of the next round by using the formula that we obtained in the derivation of Zero-Determinant strategy. We then compare the payoffs of different strategies and choose the one returning larger payoffs to implement the next round. A full cooperation state of pool is achieved while cooperation probability P_C and attack probability P_A eventually converge to 1 after multiple iterations.

Furthermore, the game between the pool members can also be regarded as a multi-party game model. An attacker can use its own miners to infiltrate other pool members and implement block withholding attack on other pool members. In general, the effective hash rate of the victim pool is uncharged, but its total revenue is distributed among more miners (including its own miners and infiltrating miners). The attacker's hash rate is reduced, however it earns additional payoffs by infiltrating other pool members. As our future research, we plan to analyze the game between the pool members and build a multi-pool game model, trying to optimize the model by using a multi-party game strategy as well as reinforcement learning algorithms to improve the cooperation probability between the pool members.

References

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

[2] Tang Chang-Bing, et al. "Game dilemma analysis and optimization of PoW consensus algorithm. " Acta Automatic Sinica 43.9 (2017): 1520-1531.

[3] Rong, Zhihai, Zhi-Xi Wu, and Guanrong Chen. "Coevolution of strategy-selection time scale and cooperation in spatial prisoner's dilemma game." EPL (Europhysics Letters) 102.6 (2013): 68005.

[4] Hofbauer, Josef, and Karl Sigmund. Evolutionary games and population dynamics. Cambridge university press, 1998.

[5] Press, William H., and Freeman J. Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent." Proceedings of the National Academy of Sciences 109.26 (2012): 10409-10413.

[6] Pan, Liming, et al. "Zero-determinant strategies in iterated public goods game." Scientific reports 5 (2015): 13096.

[7] Eyal, Ittay. "The miner's dilemma." Security and Privacy (SP), 3.2015 IEEE Symposium on. IEEE, 2015.

[8] Rosenfeld, Meni. "Analysis of bitcoin pooled mining reward systems." arXiv preprint arXiv:1112.4980 (2011).

[9] Gale, David, Harold W. Kuhn, and Albert W. Tucker. "On symmetric games." Contributions to the Theory of Games 1 (1950): 81-87.Tang, Changbing, Ang Li, and Xiang Li. "When Reputation Enforces Evolutionary Cooperation in Unreliable MANETs."IEEE Trans. Cybernetics 45.10 (2015): 2190-2201.

[10] Fan Li, et al. "A method of cooperative evolution for Blockchain mining pool based on Adaptive Zero-Determinant strategy." Journal of Computer Applications. (2018): 0-0.