Hopping-Proof and Fee-Free Pooled Mining in Blockchain

Hongwei Shi, Shengling Wang*, *Member, IEEE,* Qin Hu, Xiuzhen Cheng, *Fellow, IEEE,* Junshan Zhang, *Fellow, IEEE,* Jiguo Yu, *Senior Member, IEEE*

Abstract—The pool-hopping attack casts down the expected profits of both the mining pool and honest miners in Blockchain. The mainstream countermeasures, namely PPS (pay-per-share) and PPLNS (pay-per-last-N-share), can hedge pool hopping, but pose a risk to the pool as well as the cost to miners. In this study, we apply the zero-determinant (ZD) theory to design a novel pooled mining which offers an incentive mechanism for motivating non-memorial and memorial evolutionary miners not to switch in pools strategically. In short, our hopping-proof pooled mining has three unique features: 1) *fee-free*. No fee is charged if the miner does not hop. 2) *wide applicability*. It can be employed in both prepaid and postpaid mechanisms. 3) *fairness*. Even the pool can dominate the game with any miner, he has to cooperate when the miner does not hop among pools. The fairness of our scheme makes it have long-term sustainability. To the best of our knowledge, we are the first to propose a hopping-proof pooled mining with the above three natures simultaneously. Both theoretical and experimental analyses demonstrate the effectiveness of our scheme.

Index Terms—Pooled mining, pool-hopping attack, zero-determinant theory, incentive mechanism.

1 Introduction

Blockchain is the underlying fabric of mainstream cryptocurrency systems such as Bitcoin [1] and Ethereum [2]. These cryptocurrencies have obtained a phenomenal success, recognized as the *wave of future* [3] with a total market capitalization around 179.6B dollars at present. To realize a distributed and trustable consensus, a data structure called blockchain[1] is introduced in Blockchain system. It is a public ledger including a sequence of chained *blocks*, with each recording a set of digital transactions. Since anyone can participate in creating and verifying blocks, Blockchain system is open, leading to its vulnerability.

To deter attacks incurred by the openness of Blockchain system which essentially originates from its decentralized nature, a Proof-of-Work (PoW) [1] mechanism is employed, which allows network participants, i.e., miners, can approve new transactions only after mining a block successfully, implying that they need to solve cryptographic puzzles in the form of a hash computation with success. PoW undoubtedly increases the cost of malicious behavior, making many security attacks such as Sybil attack financially unaffordable. This is because 1) mining is actually a race where only the winner who solves the PoW task first can verify digital transactions, which needs a sufficient amount of computational power; 2) solving cryptographic puzzles is a probabilistic process, implying that no one would win the race with certainty even though it is computationally powerful.

In return for mining blocks successfully, miners are rewarded in proportion to the computational powers they invested. However, due to significant computational resources needed and probabilistic factors involved in the mining process, a solo miner has low expected revenue as well as volatility in the reward. For example, Bitcoin system now sets the difficulty of mining such that one block is generated every 10 minutes. Hence, a solo miner often has to wait 687 days in expectation to mine a block [4].

To tackle the above issue, solo miners join coalitions in the form of *mining pools*, gathering their computational powers to seek the solution of PoW puzzles and sharing the rewards proportionally to their contributions. This undoubtedly increases the chance of solving cryptographic puzzles successfully and makes the mining process more predictable. Hence, pooled mining can benefit miners from high payoffs and low variance in rewards. At present, nearly 80% of the computing power in Bitcoin and 60% of that in Ethereum belong to less than 8 and 3 mining pools, respectively.

The dominant position of pooled mining leads it to become a valuable target to be attacked. Many pools have an open trait, allowing any miner to join them through public Internet interfaces [5], which makes matters worse. Such a nature of openness makes pooled mining susceptible to attacks. There are mainly three kinds of security attacks

- *Hongwei Shi and Shengling Wang (Corresponding author) are with the School of Artificial Intelligence, Beijing Normal University, Beijing, China.*
  E-mail: hongweishi@mail.bnu.edu.cn and wangshengling@bnu.edu.cn
- *Qin Hu is with the Department of Computer and Information Science, Indiana University - Purdue University Indianapolis, IN, USA.*
  E-mail:qinhu@iu.edu
- *Xiuzhen Cheng is with the Department of Computer Science, The George Washington University, Washington DC, USA.*
  E-mail: cheng@gwu.edu
- *Junshan Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA.*
  E-mail: junshan.zhang@asu.edu
- *Jiguo Yu is with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China, with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China, and also with the School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China.*
  E-mail:jiguoyu@sina.com

1. In this paper, we use Blockchain to denote the technology while blockchain to indicate a chain of blocks.

in pooled mining: the selfish mining attack [6], [7], [8], the block withholding attack [5], [9], [10] and the pool-hopping attack [11]. The first two attacks can be well solved through the state-of-the-art approaches [7], [8], [9], [10], [12], and hence, we focus on the last one.

The pool-hopping attack was first proposed by Rosenfeld [11], in which the malicious miners strategically switch among the pools to obtain a higher payoff. This attack is cost-efficient and straightforward because of no more extra operations (e.g., keeping the block secret, dropping full proof of work or forking) needed. Studies have proved that the miner has no incentive to stay in a pool without pool hopping or redistributing the computing power [4], [9], [12]. This kind of greedy and opportunistic manner definitely casts down the mining power of a pool, resulting in its declined expected revenue. In addition, the pool-hopping attack also jeopardizes the interests of honest miners, who join in a pool continuously without switching to other pools. According to [11], the honest miners in the attacked pool will receive 43% less payoff in the worst-case theoretically, which is unfair for them.

However, little research has studied the pool-hopping attack. PPS and PPLNS [11] are pioneer countermeasures. Considering that the unbalanced distribution of reward over time makes room for miners' strategic hopping, the key idea of PPS and PPLNS is reducing the variance of reward in time series. Typically, in a PPS pool, a miner will be rewarded as long as she[2] submits a share (her contribution) to the pool, regardless of whether a block is mined successfully or not. PPLNS, one of the most prevailing reward mechanisms [13], drops the concept of "round", focusing on N shares submitted to the pool recently and distributing rewards according to the shares in proportion.

Essentially, the difference between PPS and PPLNS lies in that the former is driven by events while the latter is triggered by time. In detail, PPS rewards a miner once the event of receiving her share happens; PPLNS evaluates whether a miner should be awarded when the paying time arrives. The common feature of PPS and PPLNS is that they pay miners proportionally to their contribution, regardless of whether a block is mined successfully or not. Due to the uncertainty of mining results, the pool takes the full risk when no block is mined. Therefore, both PPS and PPLNS charge miners some fees to alleviate such a risk, which is critical to both of the pool and its members. The higher the fee, the higher the cost of the miner joining in the pool, and the smaller the motivation to mine and vice versa.

In a nutshell, the mainstream countermeasures to the pool-hopping attack, namely PPS and PPLNS, pose a risk to the pool as well as the cost to miners. Therefore, we propose a hopping-proof pooled mining with free fee in this paper, which can hedge pool hopping without any fee charged if the miner does not switch in pools strategically. The proposed pooled mining strategy has a wide scope of application since it can be employed in both prepaid and postpaid mechanisms. The former rewards once share is submitted, no matter whether there is a success mining or not; the latter awards only when the full cryptographic puzzle is solved.

It is challenging to realize the hopping-proof pooled mining without fee charged. The reasons behind the fact are: a) the strategic transferring among different pools is the instinctive demand of a miner. Especially when no fee is charged, costless hopping easily arouses miners to switch among pools; b) in the postpaid mode, mining risk is completely transferred from the pool to miners. In this situation, it is non-trivial to motivate miners to still work without hopping.

To tackle these challenges, we take advantage of the zero-determinant (ZD) theory to design an incentive mechanism for pooled mining, where cooperation (i.e., mining without hopping) is the dominant strategy of a rational miner in all situations. The ZD theory was first developed in [14] by Press and Dyson, in which the player who adopts the ZD strategy (i.e., the ZD adopter) can unilaterally set its adversary's utility no matter what strategy the adversary takes. The power of the ZD strategy endows the pool to dominate the game with any miner, rewarding her cooperation and punishing the defection, to lure the cooperation of the miner.

The main contributions of this paper can be summarized as follows:

- The interaction between the pool and any miner is formulated as an iterated prisoner's dilemma (IPD) game and the corresponding conditions are also identified. The generality of our model empowers the proposed pooled mining to have a wide scope of application, implying that it is suitable to both prepaid and postpaid mechanisms. When applied in a postpaid mechanism, the proposed pooled mining can incentivize miners to work without hopping while keeping the pool away from the risk of no block mined successfully.
- We investigate in detail whether the pool can be a ZD adopter and how he plays the ZD strategy. We draw a conclusion that the pool can unilaterally control the miner's payoff rather than his own one. The specific expected payoff of a miner that the pool can set is characterized.
- An incentive mechanism based on the ZD theory is proposed for motivating the non-memorial and memorial evolutionary miners to work without hopping. Specifically, the proposed mechanism empowers the pool to encourage the miner to behave cooperatively by increasing her short-term payoff without any additional payment in the long run.
- Both theoretical and experimental analyses demonstrate the effectiveness of the proposed incentive mechanism. More importantly, we find the proposed pooled mining is fair, implying that even the pool can dominate the game with any miner, he has to cooperate when the miner works collaboratively. The fairness of our scheme makes it have long-term sustainability.

The rest of the paper is organized as follows. Section 2 describes the formulation of our problem. The ZD strategy for the pool in an iterated prisoner's game is deduced in Section 3. Based on which, we propose an incentive

2. In this paper, we denote the pool as "he" and the miner as "she" for easy differentiation.

mechanism in light of the ZD theory in Section 4. We evaluate the mechanism both theoretically in Section 5 and experimentally in Section 6. The related literatures are listed in Section 7. Section 8 concludes our paper finally.

2 Game Formulation

In this section, we introduce our game model to formulate the interaction between the pool and the miner. Generally, we define the strategy space of each player as a dichotomous space, namely cooperation (c) and defection (d). In the PoW mining scenario, the pool is considered as a cooperator if he decides to pay the highest payoff to the miner; otherwise, he is regarded as a defector. On the other hand, the miner can devote herself wholeheartedly to the current pool by providing her total computational power to the pool without hopping, defined as cooperation, or contribute herself halfheartedly through offering partial computing ability or switching to other pools strategically, denoted by defection. We denote the actions of the pool and the miner as $x, y \in \{c, d\}$, respectively. Therefore, there are four possibilities of states in each round between the pool and the miner, i.e., $XY = (cc, cd, dc, dd)$, where X and Y denote the state of the pool and that of the miner, respectively. It is worth to note that the terminal of a mining round mentioned in our model can be defined as the time a block is mined successfully or the paying time similar to that in PPLNS. Hence, the proposed scheme can be applied in both prepaid and postpaid mechanisms.

Each state will correspond to specific payoffs for both players, which can be derived as follows:

- if both the pool and the miner are collaborative with the pool providing the highest payoff and the miner offering her entire computing power to the current pool, the payoffs of them are represented as K_p and K_m, respectively;
- when the miner defects while the pool cooperates, the miner will get an increase of $\sigma > 0$ based on her original payoff K_m, while the pool may obtain a decrease of $\pi > 0$ on K_p;
- in the case that the defective pool plays against a cooperative miner, the payoff of the pool increases by $\mu > 0$, while the miner receives a loss of $\rho > 0$;
- when both players behave maliciously, the payoffs of the pool and the miner are $K_p - \pi + \mu$ and $K_m + \sigma - \rho$, respectively.

Subsequently, the payoff vectors of the pool, denoted as $\mathbf{S}_p = (S_p^{xy})$, and the miner, denoted as $\mathbf{S}_m = (S_m^{xy})$, $x, y \in \{c, d\}$, can be presented as follows

$$\mathbf{S}_p = (S_p^{cc}, S_p^{cd}, S_p^{dc}, S_p^{dd}) = (K_p, K_p - \pi, K_p + \mu, K_p - \pi + \mu),$$

$$\mathbf{S}_m = (S_m^{cc}, S_m^{cd}, S_m^{dc}, S_m^{dd}) = (K_m, K_m + \sigma, K_m - \rho, K_m + \sigma - \rho),$$

which are also shown in Table 1.

Next, some insightful theorems are introduced to characterize the game in the following.

Theorem 2.1. *If $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, a prisoner's dilemma (PD) game can be modeled to depict the confrontation between the pool and the miner.*

TABLE 1
Payoff matrix of the pool and the miner

Pool \ Miner	Cooperation	Defection
Cooperation	K_p, K_m	$K_p - \pi, K_m + \sigma$
Defection	$K_p + \mu, K_m - \rho$	$K_p - \pi + \mu, K_m + \sigma - \rho$

Proof: To become a PD game, two fundamental conditions should be satisfied. In detail, 1) the stable state occurs when both players defect, i.e., $XY = dd$ is the Nash equilibrium; 2) mutual cooperation is the best outcome with respect to the social welfare, which means $XY = cc$ outperforms other states from an overall perspective.

The game between the pool and the miner satisfies the first condition. To be specific, if the miner is friendly, the pool will get a lower payoff as K_p when he cooperates than his payoff of $K_p + \mu$ when he defects; besides, if the pool challenges with a malicious miner, the payoff when he defects, i.e., $K_p - \pi + \mu$, is also larger than that of his cooperation, i.e., $K_p - \pi$. Thus, as a rational decision maker, the pool will always choose to defect rather than cooperation when facing an adversary with uncertain actions. With similar analysis, we can find the only feasible option for a rational miner is also to behave viciously. Accordingly, both the pool and the miner will select defection as the stable state. Therefore, the Nash equilibrium of this game comes to be $XY = dd$.

In order to investigate the second condition clearly, we denote the social welfare in each state as W_{cc}, W_{cd}, W_{dc} and W_{dd}. Thus, we have $W_{cc} = K_p + K_m$, $W_{cd} = K_p + K_m + \sigma - \pi$, $W_{dc} = K_p + K_m - \rho + \mu$, and $W_{dd} = K_p + K_m + \sigma + \mu - \rho - \pi$. Then the second condition is satisfied when $W_{cc} > W_{cd}, W_{cc} > W_{dc}, W_{cc} > W_{dd}$ hold. It is obvious that when $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, the above inequalities can be satisfied. Based on the analyses above, as self-regarding players, the pool and the miner will choose malicious behavior to maximize their payoffs, leading to mutual defection as the stable state in the game consequently. However, the most favorable outcome of the confrontation turns out to be mutual cooperation. Therefore, a PD game is formed when $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$. □

Notably, the miner may stay in the current pool for a long time without hopping to others. Hence, in this case, the PD game mentioned above can become an iterated one if some conditions are satisfied, which are summarized in the following theorem.

Theorem 2.2. *If $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, the confrontation between the pool and the miner can be modeled as an iterated prisoner's dilemma (IPD) game.*

Proof: A PD game becomes an iterated one when the payoff of any player's persistence on cooperation is larger than hopping between cooperation and defection. In other words, the inequalities below should hold

$$\begin{cases} 2K_p > K_p + \mu + K_p - \pi, \\ 2K_m > K_m + \sigma + K_m - \rho. \end{cases} \tag{1}$$

Hence, when $\pi > \mu, \rho > \sigma, \mu < \rho$ and $\sigma < \pi$, the game between the pool and the miner can be modeled as an IPD one. □

In light of the above analyses, we can find that the miner and the pool may be trapped into the iterated prisoner's dilemma, where the Nash equilibrium is far away from mutual cooperation, leading to low efficiency and distrust for Blockchain system in the long run. To tackle this problem, we employ the powerful ZD strategy to drive the players to cooperate so as to reach the win-win situation. As introduced in Section 1, the ZD adopter can unilaterally set its adversary's payoff no matter what strategy the adversary takes.

Aware of such an effective strategy, the pool is attracted to use the ZD strategy to resist a hopping miner. In this case, however, we are facing the following problems: *is the pool capable of being a ZD adopter? if yes, how does the ZD strategy work?* To address these questions, we conduct the following analyses.

3 ZD Strategy for the Pool

In this section, we examine whether the pool can play the ZD strategy, and if yes, how to achieve that. Firstly, a Markov game is established between the pool and the miner. As mentioned in Section 2, there are four possible game results, i.e., $XY = (cc, cd, dc, dd)$, in each round. We define the pool's mixed strategy as $\mathbf{p} = (p_1, p_2, p_3, p_4)$, where p_1 represents the probability of choosing cooperation in this round based on the previous outcome cc. Similarly, when the previous outcome is cd, dc or dd, the probability of the pool to cooperate in this round is p_2, p_3 or p_4. Accordingly, the probability of the pool being defective in each round is $(1-p_1, 1-p_2, 1-p_3, 1-p_4)$ corresponding to different game results in last round. Comparably, in the cases that the miner chooses to cooperate when cc, cd, dc or dd happens previously, her strategy can be denoted as $\mathbf{q} = (q_1, q_2, q_3, q_4)$, while the probability of defecting is $(1-q_1, 1-q_2, 1-q_3, 1-q_4)$.

With the above-defined strategies of the pool and the miner, the Markov matrix in each round can be derived as follow,

$$\mathbf{A} = \begin{bmatrix} p_1q_1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2q_2 & p_2(1-q_2) & (1-p_2)q_2 & (1-p_2)(1-q_2) \\ p_3q_3 & p_3(1-q_3) & (1-p_3)q_3 & (1-p_3)(1-q_3) \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4) \end{bmatrix},$$

where each element denotes the probability of state transition. For example, if the previous outcome is cc, combining the cooperation probabilities of the pool and the miner, i.e., p_1 and q_1, the probability of $XY = cc$ in this round is p_1q_1, so do other elements in $\mathbf{A}$.

Denote $\mathbf{v}$ as the stationary vector of matrix $\mathbf{A}$, then $\mathbf{v}^T\mathbf{A} = \mathbf{v}^T$ and $\mathbf{v}^T\mathbf{M} = \mathbf{0}$, where $\mathbf{M} = \mathbf{A} - \mathbf{I}$ ($\mathbf{I}$ is the identity matrix). According to the Cramer's rule, the equation $Adj(\mathbf{M})\mathbf{M} = det(\mathbf{M})\mathbf{I} = \mathbf{0}$ holds, where $Adj(\mathbf{M})$ and $det(\mathbf{M})$ represent the adjugate matrix and the determinant of $\mathbf{M}$. Subsequently, the equation above indicates that every row of $Adj(\mathbf{M})$ is in proportion to $\mathbf{v}$ [14]. Thus, if the dot product of $\mathbf{v}$ with any vector $\mathbf{f} = (f_1, f_2, f_3, f_4)^T$ is conducted, the determinate remains unchanged with some elementary column transformation, such as adding the first column to the second and the third columns. Thus, we have,

$$\mathbf{v}\cdot\mathbf{f} = D(\mathbf{p}, \mathbf{q}, \mathbf{f}) = det\begin{bmatrix} p_1q_1 - 1 & p_1 - 1 & q_1 - 1 & f_1 \\ p_2q_2 & p_2 - 1 & q_2 & f_2 \\ p_3q_3 & p_3 & q_3 - 1 & f_3 \\ p_4q_4 & p_4 & q_4 & f_4 \end{bmatrix}.$$

It is evident that the second column of the above determinant is only related to the pool's strategy. Based on this, the expected payoffs of the pool (S_p) and the miner (S_m) can be derived as

$$S_p = \frac{\mathbf{v}\cdot\mathbf{S}_p}{\mathbf{v}\cdot\mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_p)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})},$$

$$S_m = \frac{\mathbf{v}\cdot\mathbf{S}_m}{\mathbf{v}\cdot\mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_m)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}. \tag{2}$$

Hence, the linear relationship between the pool and the miner's expected payoffs holds as follows

$$\alpha S_p + \beta S_m + \gamma = \frac{D(\mathbf{p}, \mathbf{q}, \alpha\mathbf{S}_p + \beta\mathbf{S}_m + \gamma\mathbf{1})}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}, \tag{3}$$

where $\alpha,\ \beta,\ \gamma$ are coefficients.

Therefore, if the pool sets his strategy the same as $\alpha\mathbf{S}_p + \beta\mathbf{S}_m + \gamma\mathbf{1}$, the determinant in the numerator equals 0, because there exists two identical columns. In this case, $\alpha S_p + \beta S_m + \gamma = 0$, implying that a linear relation is established between the expected payoffs S_p and S_m, where the corresponding strategy is therefore called *Zero-Determinant Strategy*, denoted as $\hat{\mathbf{p}}$ below.

Specifically, when the pool sets $\hat{\mathbf{p}} = \beta\mathbf{S}_m + \gamma\mathbf{1}$ (i.e., $\alpha = 0$), the pool can control the miner's expected payoff independently as $S_m = -\frac{\gamma}{\beta}$; while when he exerts his strategy as $\hat{\mathbf{p}} = \alpha\mathbf{S}_p + \gamma\mathbf{1}$ by setting $\beta = 0$, he can set his own expected payoff at $S_p = -\frac{\gamma}{\alpha}$. The following theorem demonstrates the effectiveness of the ZD strategy adopted by the pool.

Theorem 3.1. *The pool can unilaterally control the miner's expected payoff as $S_m = \frac{(1-p_1)S_m^{dd} + p_4S_m^{cc}}{1-p_1+p_4}$, while he is not able to set his own expected payoff independently.*

Proof: Firstly, if the pool wants to control his adversary's expected payoff as $S_m = -\frac{\gamma}{\beta}$ by setting $\alpha = 0$, the specific ZD strategy of the pool should satisfy $\hat{\mathbf{p}} = \beta\mathbf{S_m} + \gamma\mathbf{1}$, according to which, we can deduce p_2 and p_3 with respect to p_1 and p_4,

$$\begin{cases} p_2 = \dfrac{p_1(S_m^{cd} - S_m^{dd}) - (1+p_4)(S_m^{cd} - S_m^{cc})}{S_m^{cc} - S_m^{dd}}, \\ p_3 = \dfrac{(1-p_1)(S_m^{dd} - S_m^{dc}) + p_4(S_m^{cc} - S_m^{dc})}{S_m^{cc} - S_m^{dd}}. \end{cases} \tag{4}$$

It is evident that p_2 and p_3 are meaningful as they belong to $[0, 1]$. Therefore, it is clear that being a ZD player, the pool can set the miner's expected payoff unilaterally. And the miner's expected payoff comes to be

$$S_m = -\frac{\gamma}{\beta} = \frac{(1-p_1)S_m^{dd} + p_4S_m^{cc}}{1-p_1+p_4}. \tag{5}$$

As (5) consisting of a weighted average of S_m^{cc} and S_m^{dd} with weights p_4 and $1-p_1$, we can conclude that the

expected payoff of the miner can be set in the range of $[S_m^{dd}, S_m^{cc}]$ by the pool's ZD strategy.

Secondly, when it comes to the case that the pool sets his own expected payoff, the ZD adopter's strategy should meet $\hat{\mathbf{p}} = \alpha \mathbf{S_p} + \gamma \mathbf{1}$ ($\beta = 0$). Using p_1 and p_4 to represent α and γ, we have

$$\begin{cases} \alpha = \dfrac{p_1 - p_4 - 1}{S_p^{cc} - S_p^{dd}}, \\ \gamma = \dfrac{(1-p_1)S_p^{dd} + p_4 S_p^{cc}}{S_p^{cc} - S_p^{dd}}. \end{cases} \tag{6}$$

And we can use p_1 and p_4 to describe p_2 and p_3 as

$$\begin{cases} p_2 = \dfrac{(1+p_4)(S_p^{cc} - S_p^{cd}) - p_1(S_p^{dd} - S_p^{cd})}{S_p^{cc} - S_p^{dd}}, \\ p_3 = \dfrac{-(1-p_1)(S_p^{dc} - S_p^{dd}) - p_4(S_p^{dc} - S_p^{cc})}{S_p^{cc} - S_p^{dd}}, \end{cases} \tag{7}$$

which indicates $p_2 \geq 1$ and $p_3 \leq 0$. Under this condition, the pool's strategy is feasible in only one case, i.e., $\hat{\mathbf{p}} = (1, 1, 0, 0)$, resulting in $\alpha = 0$ and $\gamma = 0$ according to (6). Thus, as a ZD player, the pool cannot control his payoff. □

4 Incentive Mechanism based on the ZD Strategy

In this section, we propose a ZD-based incentive mechanism for the pooled mining to hinder pool-hopping attacks. Theorem 3.1 reveals the capability of the pool as a ZD player to set the miner's expected payoff unilaterally. However, whether the pool can take advantage of such a capability to regulate the miner depends on her strategy. If the miner's strategy is irrelevant to her payoff, such as all-cooperation (ALLC, $\mathbf{q} = (1, 1, 1, 1)$), all-defection (ALLD, $\mathbf{q} = (0, 0, 0, 0)$), tit-for-tat (TFT, $\mathbf{q} = (1, 1, 0, 0)$), the pool cannot employ the ZD strategy to motivate the cooperative behavior of the miner. Hence, the proposed ZD-based incentive mechanism is suitable for the case that the strategy is laid down by the miner in light of her payoff. Win-stay-lose-shift (WSLS, $\mathbf{q} = (1, 0, 0, 1)$) and evolutionary strategies are typical payoff-driven examples.

A WSLS player will keep the same strategy as the previous round in which the outcome is good, that is so called "win-stay". Otherwise, it will adopt the strategy opposite to the one in the previous round, which is therefore named as "lose-shift". Hence, WSLS can be regarded as a particular case of the evolutionary strategy. In this work, we take the evolutionary strategy as the representative for further analysis, which can be categorized into two kinds: *non-memorial* and *memorial*. We introduce them in detail as follows.

4.1 Evolutionary strategies

The non-memorial evolutionary (**E**) strategy is featured by the fact that an **E** player may develop the strategy only based on its expected payoff. Specifically, as a rational player, if the cooperative behavior brings about a higher payoff than the defective one, the **E** player will choose to collaborate and vice versa. A typical non-memorial evolutionary strategy can be formulated as follow [15],

$$q^t(c|\mathbf{p}) = \frac{e^{\epsilon[E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})]}}{1 + e^{\epsilon[E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})]}}, \tag{8}$$

where $q^t(c|\mathbf{p})$ denotes the non-memorial **E** player's cooperation probability in round t based on the pool's strategy $\mathbf{p}$ and $\epsilon > 0$ is a scaling parameter. Besides, $E_m^t(c|\mathbf{p})$ and $E_m^t(d|\mathbf{p})$ represent the expected payoffs of the miner who acts cooperatively and defectively.

Different from the non-memorial evolutionary strategy, the memorial evolutionary strategy is associated with not only the expected payoff but also its strategy in the previous round, which we call it *memory*. That is to say, informed of the previous strategy and the expected payoff, the memorial **E** player may adjust its strategy more rationally.

Inspired by [16], we present the memorial evolutionary strategy as following: if the cooperation probabilities of the pool and the miner are denoted as p^t and q^t in round t, then the miner's cooperation probability q^{t+1} in the next round evolves as

$$q^{t+1} = q^t \cdot \frac{W_c^t}{E_m^t}, \tag{9}$$

where W_c^t indicates the expected payoff of the miner when she cooperates and E_m^t implies the expected payoff of the miner in round t. Accordingly, W_c^t and E_m^t can be calculated by

$$\begin{aligned} W_c^t &= p^t \cdot S_m^{cc} + (1 - p^t) \cdot S_m^{dc}, \\ E_m^t &= q^t \cdot W_c^t + (1 - q^t) \cdot W_d^t, \end{aligned} \tag{10}$$

where $W_d^t = p^t \cdot S_m^{cd} + (1 - p^t) \cdot S_m^{dd}$ is the miner's expected payoff when she defects.

4.2 ZD incentive mechanism

From equations (8) and (9), it is clear that if the miner obtains more payoff as a cooperative player, her cooperation probability will increase. That is to say, the miner is more likely to devote her computing power entirely to the pool without hopping if such an action brings about a higher payoff. Therefore, as a ZD player, the pool may reward the cooperation of a miner with a higher payoff while punishing her defection with the lower one. Based on this, we propose a ZD-based incentive mechanism for the pool to coerce the miner's collaborative action, thereby deterring the hopping behavior of the miner, which is detailed in the following.

As shown in Algorithm 1, in the first round, we offer the reward to each miner i ($i = 1, 2, ..., N$) proportionally to her contribution to the pool. The historical best computing power B_i is recorded as the initial computation power of each miner i, namely m_i^1 (Lines 1-4). In practice, whether a miner behaves cooperatively or defectively can not be deduced without any side information, since it is the private information of the miner. Hence, the pool has to differentiate a collaborate or defective miner based on the observation of the difference of computational powers between two continuous rounds. This requires the pool to record the computation power m_i^j of any miner i at the end of each round j (Line 7), so that the pool can obtain the difference of the devoted computational power of miner i between

Algorithm 1 The ZD-based incentive mechanism

Require:
The total number of iterations, M;
The number of miners, N;
The initial computation power of miner i, m_i^1;
The minimum and maximum payoffs that the pool can offer, L and H;

1: **for** $i = 1$ to N **do**
2: Calculate the initial reward according to $\frac{m_i^1}{\sum_{i=1}^{N} m_i^1} \cdot [H - L] + L$
3: $B_i = m_i^1$
4: **end for**
5: **for** $i = 1$ to N **do**
6: **for** $j = 2$ to M **do**
7: Update computation power m_i^j
8: $\Delta m_i^j = m_i^j - m_i^{j-1}$
9: **if** $\Delta m_i^j < 0$ **then**
10: Calculate $\mathbf{p}^j$ which makes $E_i^j = L$
11: **else if** $\Delta m_i^j = 0$ **then**
12: $\mathbf{p}^j = \mathbf{p}^{j-1}$ which makes $E_i^j = E_i^{j-1}$
13: **else if** $\Delta m_i^j > 0$ **then**
14: **if** $B_i < m_i^j$ **then**
15: $B_i = m_i^j$
16: **end if**
17: $y = (\frac{\Delta m_i^j}{B_i} + 1) \cdot E_i^{j-1}$
18: Calculate $\mathbf{p}^j$ which makes $E_i^j = H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}}$
19: **end if**
20: **end for**
21: **end for**

round $j-1$ and round j, i.e., $\Delta m_i^j = m_i^j - m_i^{j-1}$ (Line 8). If $\Delta m_i^j \geq 0$, miner i is considered to be a cooperative player and vice versa.

When $\Delta m_i^j < 0$, the miner splits her computing power into other pools[3], implying she is a pool-hopping attacker. Her payoff is therefore needed to be reduced in order to hinder such an attack. Under this situation, the pool will exert the ZD strategy, setting the attacker's payoff as the minimum one, i.e., L (Lines 9-10). If $\Delta m_i^j = 0$, the pool provides the same payoff to the miner as that in the last round (Lines 11-12). When $\Delta m_i^j > 0$, the pool would update B_i if needed (Lines 14-16). Since this case indicates the miner behaves more cooperatively, the pool will increase her payoff as $E_i^j = H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}}$, where $y = (\frac{\Delta m_i^j}{B_i} + 1) \cdot E_i^{j-1}$ and $\zeta > 0$ represents a scaling parameter (Line 17-18). It is worth to note that the more increment of computational power relative to B_i is, the higher reward the miner can obtain, which is up to the maximum payoff that the pool can offer, namely H.

5 Theoretical Analysis

In this section, we analyze the proposed incentive mechanism theoretically.

3. The situation where the miner is unavailable due to some reasons such as lacking of electricity is out of our consideration in this paper.

Theorem 5.1. *For any non-memorial evolutionary miner who is motivated by the ZD incentive mechanism, it is conceivable that the miner's cooperation probability will be maximized.*

Proof: To maximize $q^t(c|\mathbf{p})$ according to (8), we turn to prove that $E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})$ rises with the increase of game round t if the miner is a cooperative one. According to Algorithm 1, if any miner i behaves more cooperatively than the previous round, we have

$$\begin{aligned} E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p}) &= H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}} - L \\ &= H * \frac{e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}{1+e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}} - L. \end{aligned} \tag{11}$$

Since $(\frac{\Delta m_i^t}{B_i} + 1) \cdot R_i^{t-1}$ keeps raising because of the miner's collaborative behavior, $\frac{e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}{1+e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}$ becomes to one at last, leading $E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})$ equals to $H - L$ consequently. Hence, driven by the proposed ZD incentive mechanism, $q^t(c|\mathbf{p})$ can evolve to the maximum. □

Theorem 5.2. *For any memorial evolutionary miner who is motivated by the ZD incentive mechanism, her cooperation probability tends to 1 gradually.*

Proof: In light of (9), a memorial evolutionary miner can calculate her cooperation probability according to W_c^t and E_m^t, which can be deduced by (10). In practice, we use the cooperative frequencies f_p^t and f_m^t to approximate p^t and q^t. Specifically, f_p^t indicates the number of rounds the pool cooperates divided by the total number of rounds, while f_m^t denotes that of a miner.

Based on the ZD incentive mechanism, we consider the following two cases, where the miner chooses to cooperate or defect [17].

a) if the miner is considered as cooperative, the pool may reward her, resulting in $E_m^{t+1} \geq E_m^t$. In this case, with the increase of E_m^{t+1} and f_m^{t+1}, W_c^{t+1} turns to

$$W_c^{t+1} = \frac{E_m^{t+1} - (1 - f_m^{t+1}) W_d^{t+1}}{f_m^{t+1}}. \tag{12}$$

Hence, $\lim_{t \to +\infty} W_c^{t+1} = \frac{E_m^{t+1} - (1-f_m^t) W_d^{t+1}}{f_m^t} > W_c^t$ because of $W_d^{t+1} = W_d^t$.

b) when the miner is regarded as a defective miner, then we have $E_m^{t+1} \leq E_m^t$, and the decrease of E_m^{t+1} and f_m^{t+1} will lead to

$$W_d^{t+1} = \frac{E_m^{t+1} - f_m^{t+1} W_c^{t+1}}{1 - f_m^{t+1}}. \tag{13}$$

Comparably, $\lim_{t \to +\infty} W_d^{t+1} = \frac{E_m^{t+1} - f_m^t W_c^{t+1}}{1 - f_m^t} < W_d^t$ because of $W_c^{t+1} = W_c^t$.

To sum up, Case a) indicates that W_c^t increases and W_d^t remains unchanged and Case b) implies that W_d^t declines while W_c^t remains steady. Thus, $\exists T^* \in \mathbb{Z}^+$, such that $\forall t > T^*$, $W_c^t > W_d^t$ holds. Based on this, E_m^t can be derived as

$$\begin{aligned} E_m^t &= f_m^t W_c^t + (1 - f_m^t) W_d^t \\ &< f_m^t W_c^t + (1 - f_m^t) W_c^t = W_c^t. \end{aligned} \tag{14}$$

In light of (14), we can conclude that $q^{t+1} = q^t \frac{W_c^t}{E_m^t} \to 1$ with the increase of game round t. That is to say, the memorial evolutionary miner will gradually increase the cooperation probability to one eventually. □

Conclusively, the non-memorial and memorial evolutionary miner will be encouraged to behave cooperatively by the proposed ZD incentive mechanism in the end.

Another essential nature of the proposed incentive mechanism is that it can be employed into the prepaid mechanism as well as the postpaid mechanism, with the former rewards the miner when a share is submitted and the latter defines the terminal of a mining round as the time a block is mined successfully. Noteworthily, the ZD incentive mechanism is free-fee charged for miners in both prepaid and postpaid cases due to their wholehearted devotions. More importantly, in the postpaid mechanism, the proposed incentive mechanism can hinder pool hopping attackers without putting any risk on the pools since our mechanism enables the miners to mine wholeheartedly until a block is generated successfully.

Now that such a powerful strategy the pool can employ, he has an overwhelmingly dominant position compared with the miner, then *is the pool capable of getting a higher payoff greedily through defecting when the miner collaborates?* We use the following theorem as a response to the above concern.

Theorem 5.3. *When the miner chooses to cooperate, the only rational strategy of the pool who employs the ZD incentive mechanism is to collaborate.*

Proof: As demonstrated in Theorems 5.1 and 5.2, the miner will choose to contribute her maximum computational power into the pool because of the effectiveness of the proposed ZD incentive mechanism. In this case, the pool will provide the miner with the maximal payoff. Therefore, we will discuss what the ZD strategy is when the pool sets the expected payoff of the miner as the optimal value in the following.

According to Section 3, the miner's expected payoff can be set as $S_m = \frac{(1-p_1)S_m^{dd} + p_4 S_m^{cc}}{1-p_1+p_4}$, which belongs to $[S_m^{dd}, S_m^{cc}]$. Due to

$$\begin{aligned} \frac{\partial S_m}{\partial p_1} &= \frac{p_4(\rho-\sigma)}{(1-p_1+p_4)^2}, \\ \frac{\partial S_m}{\partial p_4} &= \frac{(1-p_1)(\rho-\sigma)}{(1-p_1+p_4)^2}, \end{aligned} \tag{15}$$

$\frac{\partial S_m}{\partial p_1} > 0$ and $\frac{\partial S_m}{\partial p_4} > 0$ because of $\rho > \sigma$ as indicated in Theorem 2.2, implying a monotonically increasing relationship between S_m and p_1, p_4. Hence, when $p_1 = 1$, $p_4 = 1$, the pool can maximize the miner's expected payoff. Furthermore, according to (4), if p_1 and p_4 are equivalent to 1, the only possible value of p_2 is 1 because p_2 should lie in $[0, 1]$ to be a probability, so as for p_3. That is to say, the pool can set $\mathbf{p} = (1, 1, 1, 1)$ to maximize the payoff of a miner.

In light of the above analysis, once the miner cooperates, the pool will set his ZD strategy as $\mathbf{p} = (1, 1, 1, 1)$ to maximize a collaborative miner's expected payoff. That is to say, whenever the miner cooperates, the pool will collaborate subsequently.

□

In summary, the pool will be collaborative in return if the miner offers her maximum computing power. Thus, the proposed ZD incentive mechanism is fair to both sides, which makes it be long-term sustainable. Such an aim is achieved via controlling the miner's short-term expected payoff by the pool. Then, *what are the players' actual payoffs over the long run?* This question can be answered by the following two theorems.

Theorem 5.4. *In the long run, the miner's actual payoff equals to K_m based on our proposed ZD incentive mechanism.*

Proof: a) For a non-memorial evolutionary miner, $\exists \tau \in \mathbb{Z}^+$, such that $\forall t \geq \tau$, q^t can be maximaized. That is to say, when $t \geq \tau$, the expected payoff of the miner is identical to K_m, which is the maximum payoff for a cooperative miner. In light of this, the actual payoff of the miner P_m^A can be derived as the average of the expected payoff E_m^i in each round i, where $i < \tau$ and the expected payoff K_m after round τ. Therefore, P_m^A can be written as:

$$P_m^A = \lim_{t\to\infty} \frac{\Sigma_{i=1}^{\tau-1} E_m^i + \Sigma_{i=\tau}^{t} K_m}{t} = K_m. \tag{16}$$

b) The actual payoff of a memorial evolutionary miner is

$$\begin{aligned} P_m^A &= \lim_{t\to\infty} \frac{W_c^t - (1-p^t)(K_m+\sigma-\rho)}{p^t} = \lim_{t\to\infty} W_c^t \\ &= \lim_{t\to\infty} \frac{E_m^t - (1-q^t)W_d^t}{q^t} = \lim_{t\to\infty} E_m^t = K_m. \end{aligned} \tag{17}$$

□

By inspecting Theorem 5.4, the miner will receive the actual payoff P_m^A as K_m over the long run. Then, *is it possible for the pool to own more payoff by greedy behavior?* This question can be resolved by the following theorem.

Theorem 5.5. *In the long run, the pool's actual payoff P_p^A is equivalent to K_p based on our proposed ZD incentive mechanism.*

Proof: According to Theorem 5.3, the pool will behave cooperatively to reward a collaborative miner, implying that $XY = cc$ is the stable state for the game. In such a case, $P_p^A = K_p$ holds according to Table 1. □

In light of Theorems 5.4 and 5.5, the pool and miner will obtain the actual payoffs as K_m and K_p, respectively. That is to say, neither the pool nor the miner can receive higher reward by noncooperative manner over the long run, which is quite fair for both sides.

6 Performance Evaluation

To testify the effectiveness of the ZD incentive mechanism proposed in Section 4, we conduct experimental simulations in this section. To be specific, we set the payoff vectors of the pool and the miner as $\mathbf{S}_p = (3, 0, 5, 2)$ and $\mathbf{S}_m = (3, 5, 0, 2)$, which is a typical example of the prisoner's dilemma. We also carry out the simulations with other parameter settings and derive the comparable results. So we omit to present those results to avoid redundancy. Note that each simulation is repeated 100 times to get the average value for statistical confidence.

In detail, if the pool is a ZD adopter competing with a miner who employs four classical strategies, i.e., ALLC, ALLD, TFT and WSLS, the miner's expected payoffs can be set at a fixed value as shown in Fig. 1. Taking the specific ZD strategy of the pool $\mathbf{p} = (0.9, 0.3, 0.8, 0.2)$ as an example,

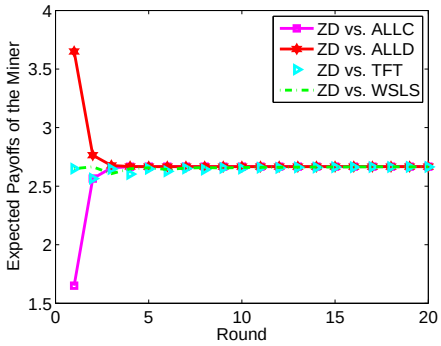


Fig. 1. The expected payoffs of the miner when she adopts ALLC, ALLD, TFT, WSLS strategies and the pool employs the ZD strategy.

no matter what strategies the miner employs, her expected payoff will finally become to a constant. That is to say, the adversary's outcome can be controlled unilaterally by the ZD adopter because of his effective strategy.

As mentioned in Section 4, the classical strategies ALLC, ALLD and TFT are out of our consideration because the strategies are irrelevant to the payoff of the player. Moreover, WSLS is regarded as a special evolutionary strategy. Hence, only the simulations of the evolutionary miners who compete with a ZD pool are included in this work, which are demonstrated as follows.

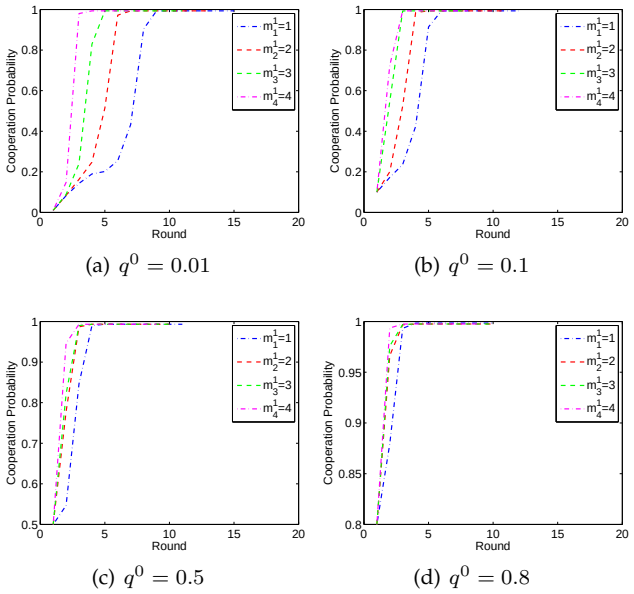
In our simulation, we assume there are four miners in a pool, whose initial computational powers are respectively $m_1^1 = 1, m_2^1 = 2, m_3^1 = 3, m_4^1 = 4$ [4]. Setting the original cooperation probabilities (CPs) $q^0 = 0.01, q^0 = 0.1, q^0 = 0.5$ and $q^0 = 0.8$, Figs. 2 and 3 respectively show how the CPs of the non-memorial evolutionary miners evolve according to the proposed ZD incentive mechanism when ϵ[5]$= 5$ and 8.

Through further observation of Figs. 2 and 3, we can conclude that the CPs of the non-memorial evolutionary miners converge to one with different speeds, which is mainly because of different initial computational investments and the scaling parameter ϵ. To be specific, a miner with a larger initial computing investment would be more inclined to accelerate the cooperation process due to the higher growth of payoff. Intuitively, a higher ϵ brings about a faster convergence speed of the CP according to (8).

Fig. 4 plots the CPs of a memorial evolutionary miner driven by the ZD-based incentive mechanism, where the CPs go up to 1 gradually with the initial values as $p^0 = q^0 = 0.01, 0.1, 0.5$, and 0.8. In detail, each subfigure shows that the CP of the miner with a small initial input converges slowly compared with other miners, even though they share the same initial cooperation probability. The reason may lie in that the miner with a smaller initial computing investment may get a relatively lower payoff in the beginning, leading a slow growth of the expected payoff. Thus, her CP would rise slower comparably. Moreover, considering the CPs of a miner with the same initial investment but having

4. The cases in which more miners exist in a ZD pool share the same conclusion, so we omit it for reducing repetition.

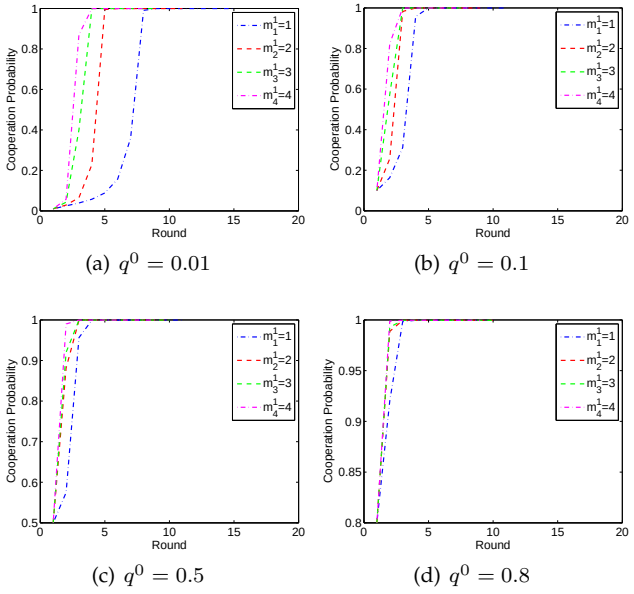
5. ϵ is set to be big enough here so that the maximum cooperation probability of a non-memorial evolutionary player (calculated by (8)), can approach to 1.



(a) $q^0 = 0.01$ (b) $q^0 = 0.1$

(c) $q^0 = 0.5$ (d) $q^0 = 0.8$

Fig. 2. The evolutions of the CPs of the non-memorial evolutionary miners when $\epsilon = 5$.



(a) $q^0 = 0.01$ (b) $q^0 = 0.1$

(c) $q^0 = 0.5$ (d) $q^0 = 0.8$

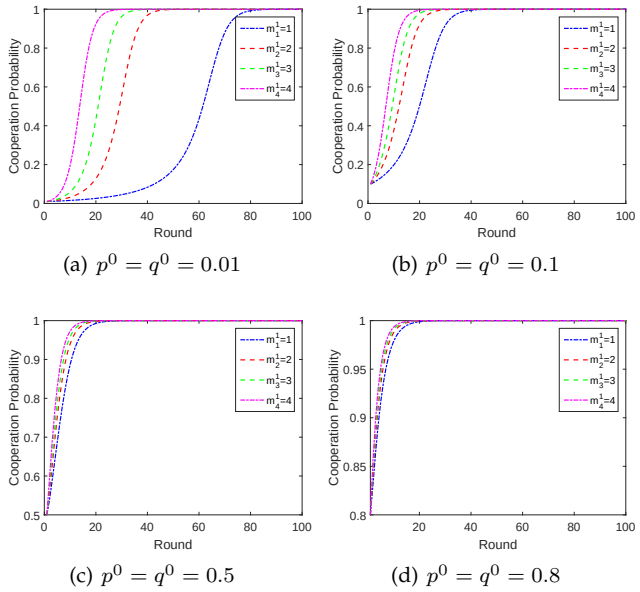
Fig. 3. The evolutions of the CPs of the non-memorial evolutionary miners when $\epsilon = 8$.

different initial cooperation probabilities, for example, the blue lines in subfigures (a)-(d), the result is that the higher the initial CP is, the faster it is converged to one, which is mainly caused by the *memory* we mentioned above in light of (9).

7 Related Work

At present, the researchers mainly focus on three kinds of security attacks in pooled mining: the selfish mining attack, the block withholding (BWH) attack and the pool-hopping attack.

In detail, a selfish mining attacker [18] keeps its mined block secret and intentionally forks the main blockchain.



(a) $p^0 = q^0 = 0.01$ (b) $p^0 = q^0 = 0.1$

(c) $p^0 = q^0 = 0.5$ (d) $p^0 = q^0 = 0.8$

Fig. 4. The evolutions of the CPs of the memorial evolutionary miners.

Specifically, the selfish miner mines on its private branch instead of working on the public chain as the honest miners. When the public ledger approaches it's private chain, the selfish miner advertises its concealed chain to the public, leading to wasting resources of the honest miners on resolving cryptopuzzles which ends up gaining no rewards. Several defense mechanisms have been proposed to block this selfish manner as well as its variants. For example, Saad *et al.* [7] developed a defense mechanism in the network-wide scope to detect and deter selfish miners; Zhang *et al.* [8] proposed a backward-compatible mechanism to defend selfish attacks.

The BWH attackers pretend to devote their computational capabilities into the target pool and then obtain payoffs. However, they send only partial proof of work, not full proof of work, resulting in reward reduction to other miners in the pool. This kind of attack was first proposed in [11], after which, Courtois *et al.* [19] summarized its concept and Eyal modeled the confrontation between the pools as a prisoner's dilemma in [5]. Specifically, in [5], a Nash equilibrium was established, where the rational pools would attack each other, resulting in a lose-lose situation. Besides, the pools are trapped into an iterative prisoner's dilemma, in which the pool chooses to attack or not is the so called miner's dilemma. Ongoing researches on avoiding this attack have proposed some efficient and cheap defense mechanisms. For example, Bag *et al.* in [9] proposed a generic scheme to counter BWH attacks via employing cryptographic commitment schemes, based on which, an implementation using hash function was presented as an alternative. Besides, Luu *et al.* [12] put forward a power splitting game for the miners so as to find a solution to fight back the BWH attacks. Additionally, Hu *et al.* [10] took advantages of the Zero-determinant theory to analyze the BHW attacks between two pools. Based on which, different conditions for the pools playing the ZD strategy individually and simultaneously have been demonstrated comprehensively.

We focus on the pool-hopping attack in this work. Pioneer countermeasures are PPS, PPLNS and their variants, including the Slush's method, maximum pay-per-share (MPPS), and pay-once-PPLNS. Detailedly, the pool manager can calculate the score of each share based on the exponential score function $s = e^{\frac{T}{c}}$, in which s represents the score of the share given in time T and c denotes the scaling parameter. Due to the share's score, the pool hopping behavior can be alleviated in mining pools by reducing the score of shares at the earlier stage of the round while increasing the score of shares later on. Such kind of score-based method is recognized as the Slush's method and has been applied in the mining pools such as Slushpool [20]. Besides, in the maximum pay-per-share method, two balances are kept for each miner, that is, a PPS balance and a proportional balance [11]. To be specific, if the miner offers a share, her PPS share balance is increased as if the pool is a PPS pool. When the pool generates a block, the proportional balances of the miners are increased as if they have joined a proportional pool. Based on which, the reward paid for each miner is the minimum between the PPS balance and the proportional balance. In pay-once-PPLNS, every share is rewarded at most once [11]. In other words, the share is deleted after it is paid, leading a higher probability to the elder shares to be paid for future blocks. If a share is partially paid, it will be deleted partially. However, theoretical analysis on the above mechanisms are lacking and their effectiveness in preventing pool-hopping attacks still remain an open issue [21].

8 Conclusion

In this paper, we propose a hopping-proof pooled mining with fee-free in Blockchain. To that aim, we formulate the interaction between the pool and any miner as an IPD game and identify the corresponding conditions. The generality of our model capacitates the proposed pooled mining to have wide applicability. Based on the model, we take advantage of the ZD theory to empower the pool can unilaterally control the miner's payoff, which can be used to motivate the cooperation of the non-memorial and memorial evolutionary miners through the proposed ZD incentive mechanism. Both theoretical and experimental analyses demonstrate the effectiveness of the ZD incentive mechanism. To the best of our knowledge, we are the first to propose a hopping-proof pooled mining with the natures of *fee-free, wide applicability and fairness* at the same time.

References

[1] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 305–320.

[4] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. Citeseer, 2015, pp. 919–927.

[5] I. Eyal, "The miner's dilemma," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 89–103.

[6] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 515–532.

[7] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 360–364.

[8] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Cryptographers Track at the RSA Conference*. Springer, 2017, pp. 277–292.

[9] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1967–1978, 2017.

[10] Q. Hu, S. Wang, and X. Cheng, "A game theoretic analysis on block withholding attacks using the zero-determinant strategy," in *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*. ACM, 2019, pp. 1–10.

[11] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *arXiv preprint arXiv:1112.4980*, 2011.

[12] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in *2015 IEEE 28th Computer Security Foundations Symposium*. IEEE, 2015, pp. 397–411.

[13] P. Chatzigiannis, F. Baldimtsi, I. Griva, and J. Li, "Diversification across mining pools: Optimal mining strategies under pow," *arXiv preprint arXiv:1905.04624*, 2019.

[14] W. H. Press and F. J. Dyson, "Iterated prisoners dilemma contains strategies that dominate any evolutionary opponent," *Proceedings of the National Academy of Sciences*, vol. 109, no. 26, pp. 10 409–10 413, 2012.

[15] H. P. Young, "The diffusion of innovations in social networks," *The economy as an evolving complex system III: Current perspectives and future directions*, vol. 267, p. 39, 2006.

[16] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 760–763, 2018.

[17] Q. Hu, S. Wang, L. Ma, R. Bie, and X. Cheng, "Anti-malicious crowdsourcing using the zero-determinant strategy," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 1137–1146.

[18] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[19] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *arXiv preprint arXiv:1402.1718*, 2014.

[20] "Slushpool," https://slushpool.com/home/, accessed June 25, 2019.

[21] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

Hongwei Shi received her B.S. degree in Computer Science from Beijing Normal University in 2018. Now she is pursuing her M.S. degree in Computer Science from Beijing Normal University. Her research interests include blockchain, game theory and combinatorial optimization.

Shengling Wang is a full professor in the School of Artificial Intelligence, Beijing Normal University. She received her Ph.D. in 2008 from Xian Jiaotong University. After that, she did her postdoctoral research in the Department of Computer Science and Technology, Tsinghua University. Then she worked as an assistant and associate professor from 2010 to 2013 in the Institute of Computing Technology of the Chinese Academy of Sciences. Her research interests include mobile/wireless networks, game theory, crowdsourcing.

Qin Hu received her Ph.D. degree in Computer Science from the George Washington University in 2019. She is currently an Assistant Professor in the department of Computer and Information Science, Indiana University - Purdue University Indianapolis. Her research interests include wireless and mobile security, crowdsourcing/crowdsensing and blockchain.

Xiuzhen Cheng [F] received her M.S. and Ph.D. degrees in computer science from the University of Minnesota Twin Cities in 2000 and 2002. She is a professor in the Department of Computer Science, George Washington University, Washington, DC. Her current research interests focus on privacy-aware computing, wireless and mobile security, dynamic spectrum access, mobile handset networking systems (mobile health and safety), cognitive radio networks, and algorithm design and analysis. She has served on the Editorial Boards of several technical publications and the Technical Program Committees of various professional conferences/workshops. She has also chaired several international conferences. She worked as a program director for the U.S. National Science Foundation (NSF) from April to October 2006 (full time), and from April 2008 to May 2010 (part time). She published more than 170 peer-reviewed papers.

Junshan Zhang received the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2000. In August 2000, he joined the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA, where he has been a Professor since 2010. His research interests fall in the general field of information networks and its intersections with power networks and social networks. His current research focuses on fundamental problems in information networks and energy networks, including modeling and optimization for smart grids, optimization/control of mobile social networks and cognitive radio networks, and privacy/security in information networks.

Jiguo Yu received the Ph.D. degree from Shandong University, in 2004. He became a Full Professor with the School of Computer Science, Qufu Normal University, Shandong, China, in 2007. He is currently a Full Professor with the Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), and a Professor with the School of Information Science and Engineering, Qufu Normal University. His research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. He is a member of ACM and a Senior Member of CCF.