Bitcoin Pooled Mining Reward Functions: Revisited

Namratha M[a], Kunwar Singh[b]

[a]Research Scholar, National Institute of Technology, Trichy - 620015
[b]Faculty of Computer Science and Engineering, National Institute of Technology, Trichy - 620015

ARTICLE INFO

Keywords:
Bitcoin
pooled-mining
reward systems
pool-hopping
MPPS
Slush
Geometric

ABSTRACT

Bitcoin was first introduced during October 2018 in a paper published by Satoshi Nakamoto. Decentralized consensus protocol in Bitcoin is executed through the process called mining. Currently, it is very difficult for individual miners to mine and they might take a timespan of few months or even few years depending on the complexity of the puzzle. This is the main reason for the miners to work in a collective manner by creating a group of miners called pool. As of now, mining pools are an inseparable part of bitcoin and other cryptocurrencies. In FC2016, Okke Schrijvers, Joseph Bonnaeu, Dan Boneh and Tim Roughgraden presented reward function properties for mining pool method which will show whether a mining method is incentive compatible or not. They have mentioned that it would be interesting to check whether score based pooled mining methods such as Slush method and Geometric method are incentive compatible or not. In this paper, the mathematical proof based on the properties given by Okke Schrijvers et.al has been derived to prove that the Slush method and Geometric method are not incentive compatible. The graphs generated from the results of the simulation showed clearly that the Slush method and Geometric method are not incentive compatible. The percentage of pool hopping allowed in Maximum Pay Per Share (MPPS) method is calculated. We also ensure that the pool never goes bankrupt at any point of time. We have also proved that the MPPS method is not hopping proof.

1. Introduction

Bitcoin was introduced in a paper written and published by Satoshi Nakamoto in October 2008 [1]. Bitcoin has proven to be one of the most popular cryptocurrency in the recent times. The entire functionality of Bitcoin is mainly based on the Decentralized consensus protocol. This protocol maintains a common global record which is called the ledger of all the transactions thus allowing transparency to all users. The protocol is also supported with individuals, referred to as miners, who add transaction to this global ledger by solving a cryptographic puzzle. This global ledger is commonly referred to as Blockchain. For this computational work, the miners are given an incentive in the form of newly generated Bitcoins and the transaction fees which is variable. Currently getting a single Bitcoin is very rewarding but it is very difficult for individual miners to mine and they might take a timespan of a few months or even few years depending on the complexity of the puzzle. This is the main reason for miners to work in a collective manner by creating a group of miners called Pool. All the miners in a group agree on a reward distribution system to divide a Bitcoin reward which is gained by the pool. Deciding a particular reward distribution algorithm to divide the block reward among all the miners of a pool is a challenging incentive design problem.

Mining pools are against the spirit of centralization. Andrew Miller et.al.[2] stated that the existence of mining groups is basically because of the limitation of the proof-of-work puzzle offered by Bitcoin. It provides an efficient technique for establishing cooperation in a group. Miller et.al. proposed a non-outsourceable puzzle that deter coalitions of Bitcoin miners. Pools are the most debated topic in the Bitcoin community as they present a system of domination. As of now they are an inseparable part of Bitcoin and other cryptocurrencies.

Meni Rosenfield [3] analyzed simple reward systems such as proportional method, score based methods such as Slush method, Geometric method, etc. Meni Rosenfield has shown that proportional method is prone to pool hopping . Pool hopping refers to mining in the pool only during good times and the miners leave the pool during bad times. Generally a pool-hopper does this to get more reward from the pool than the actual value contributed and hence increases the rewards at the expense of other miners. *Pool-hopping*, as the name indicates, is an insidious strategy where miners constantly hop into and out of various pools or might even opt for solo mining. The simple reward systems had a major drawback of pool hopping which was restricted in score based methods. It is not discussed in [3] and others [5,6,7] that how miners will be incentivized to report the full solution immediately. In FC2016, Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden had put forth three properties for a reward function for mining as a pool. These properties will help show whether a certain mining method is incentive compatible or not. Okke Schrijvers et.al.[4] have shown that proportional method is not incentive compatible. In [4], it is proved that Pay Per Share method is incentive compatible.

Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden[4] have mentioned that it will be interesting to check whether mining pool methods such as Slush method and Geometric method are incentive compatible or not. In this paper, it has been proved that the proportional method is not incentive compatible using the three properties of reward function. However, Pay Per Share method is found to be incentive compatible. In our work, we proved that the

ORCID(s):

score based methods such as Slush method and Geometric method are not incentive compatible. Simulation has been carried out and from the graphs generated, it was observerd that the above two methods behave exponentially. The simulation results bolster our conclusion that Slush method and Geometric method reward functions are not incentive compatible.In our work, we also devised a new model of reward systems that is incentive compatible. The benefits of pool hopping in MPPS method over PPS method is discussed. The percentage of pool hopping allowed in Maximum Pay Per Share (MPPS) method is calculated. We also ensure that the pool never goes bankrupt at any point of time.We have also proved that the MPPS method is not hopping proof.

2. Preliminaries

Business on the Internet has come to depend on electronic payments backed by the trusted third parties for the processing of transactions and their verification. The system works fine for almost all transactions but it undergoes the weakness of a trust-based model of requiring a trusted third- party for settling the disputes which can occur. An entirely non-reversible transaction is not possible in reality as sometimes financial institutions cannot settle all the disputes. The cost of intervention upsurges the transaction cost which limits the minimum transaction size and even eliminating the small amount transaction.

An Electronic Payment system, which depend on cryptographic proof of trust, performs transactions between any two willing parties without the need of any trusted third party. These transactions are computationally infeasible to converse which protects the seller from frauds, and a routine escrow mechanism can be employed to protect buyers.

BITCOIN

Bitcoin is an entirely decentralized electronic currency which does not rely on any central authority like banks. It was developed by Satoshi Nakamoto in 2008 and a paper was published on it. The paper discusses the notion of a purely peer-to-peer form of electronic currency which does not requires a trusted third party. He anticipated an electronic payment system which will totally rely on cryptographic proof instead of relying on a third party. This will allow any two parties to transact straight away between them without any middlemen. Bitcoin is a partly-anonymous mode of payment. It means that a user does not have to reveal his identity for performing a transaction with anyone in the world. He described the concept of hashing a transaction into a longest chain-like structure. And this notion of hash-based chain will make it easy for others in the network to verify the transactions. Many other cryptocurrencies have emerged by forking the same Bitcoin open source code.

POOLED MINING

The difficulty of mining (difficulty of solving the cryptographic puzzle) has increased to an extent where the chances of a miner, mining solo, getting rewarded is highly unlikely. Thus began the technique of pool-mining. To increase one's income or let us say to keep it steady, miners started working together as a group, called a pool. There is also a pool operator who maintains the pool and also charges fees for his services. So, when the pool finds a block with the collective power of all the miners, they are rewarded with Bitcoin which is divided among them. Now for the distribution of this reward many Pool Mining Methods have been devised like Proportional Method, Pay-per-Share Method, Pay-per-last 'N' Shares Method, Geometric Method, etc.

2.1. Reward Function Properties

Below are the three properties which are important for a reward function.These 3 properties were given in [3] and these properties will be used in this paper too to prove that the score based methods are not incentive compatible.

Property 1 (Incentive Compatibility):
A Reward Function is said to be incentive compatible where each and every miner submits the full solution immediately.

Next, we require the pool operator to pay the miners in such a way that it is proportional to the amount of work they have done. This is ensured by Property 2.

Property 2 (Proportional Payments):
A Reward Function must divide the block reward precisely and appropriately among all the miners of the pool after every round. And if this is not done properly then either the miners will be under loss or even the pool operator may have to pay for more than what he received.

Property 3 (Budget Balanced):
This property states that an incentive compatible reward function will never pay extra to the miners as compared to what has been received by the pool operator. It should be 1:1 i.e. the reward function will share the reward accurately and exactly among all the miners.

In [4], it also mentioned that for a reward function to be incentive compatible miners have to report the full solution immediately and the condition for this is as follows:

Lemma 1:
For a reward function 'R', a miner 'i' has an incentive to report full solution immediately if and only if the following condition holds good for all α_i (where i=1 to n), b_t, D and miner i:

$$\sum_{j=1}^{n} \alpha_j (R_i(b_t + e_j) - R_i(b_t)) \leq \frac{E_b[R_i(b)]}{D}$$

where α_i denotes the fraction of the reward ,
b_t the number of shares reported to the pool operator at time t,
e_i be the i^{th} standard basis vector which has 1 in its i^{th} component and 0 everywhere else,

and D is the difficulty.
Therefore, if a reward function statisfies the above equation, it is incentive compatible; else, not.

3. Our Work

Based on the incentive compatibility criteria discussed in the above section, the proofs for the Slush method and Geometric method are presented in this paper. This was also given as an open problem by the authors Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. They had proved using the above three properties of reward functions that the proportional method is not incentive compatible and Pay Per Share method is incentive compatible. **Based on this an incentive compatible model was devised**. It is interesting to show that the score based methods such as Slush method and Geometric method are not incentive compatible.
In this paper, based on the properties given in [4], we have mathematically shown that the score based methods such as Slush method and Geometric method are not incentive compatible. As a proof for incentive compatibility of a reward function, the miner should report full solutions immediately and the condition to be satisfied for that as given in [4] is:

$$\sum_{j=1}^{n} \alpha_j (R_i(b_t + e_j) - R_i(b_t)) \leq \frac{E_b(R_i(b)]}{D}$$

This condition is used in proving that the Slush method and Geometric method are not incentive compatible.

Since, the Lemma 1 is considering unordered history of transcripts but the score based methods such as Slush method and Geometric method need ordered shares, we first apply a sorting algorithm and order the shares based on the time at which they are submitted before proving the incentive compatibility. The first step of sorting is very essential in proving the incentive compatibilty of Slush method and Geometric method without which the Lemma 1 cannot be applied. Also, sorting has to be applied at the beginning of each round. So, we have an ordered list of shares at hand, as a result of the sorting, before proceeding with the proofs of incentive compatibility.

3.1. Slush method

Slush method is a score-based method aimed at lowering pool-hopping.It is based on proportional method except for the fact that a participant is rewarded at the end of a round in proportion to the score. The score value depends on the time when the share was submitted and the block reward is distributed among the miners in proportion to their score after every round. It is the first method which was basically developed to tackle the problem of pool hopping. Although some hopping is still possible, this method gives very high variance to intermittent miners. For reward distribution, we need Scoring Hash Rate which is the user's contribution to mining power of a pool. Suppose if your scoring hash rate is 3 percent of the pool then you will receive 3 percent of the block reward.
The Pool will calculate your reward by this formula:
r = B * (1 - P) * $\frac{s}{S}$
r = Your Reward
B = Block Value which is 12.5 BTC currently
P = Pool Fee which is 2 percent currently
s = Your Scoring Hash Rate
S = Pool Scoring Hash Rate

The hash rate is calculated as the the time instant from where a particular block is found and this calculation is used in the scoring of hash rates. They reflect some mining history before the actual block finding. Reward is completely autonomous from rounds. The most significant value here is the scoring hash rate. during the process of other miner connecting to the pool then the scoring hash rate will remain the same but pool's hash rate will increase proportionally. As a result, the miner shall receive a small part of the total reward, but more frequently, because the pool will be now more stronger. When some other miner disconnects from the pool, the opposite will happen. There is no major difference in the long run for the miners. But the effective hash rate is variable. To define a Reward Function in a more mathematical way when a block B is found at time t and its value is finally known to be b, the reward for each user i can be calculated as follows:
R_i(b) = B(1 - p) * $\frac{C_i(t(b))}{C(t(b))}$
Where
R_i(b) = Reward that will be given to user u
B = Reward given for a block
p = Pool Fees
C_i(t(b)) = User u's Score from time t(b)
C(t(b)) = Score of the whole pool
So the reward is distributed fairly among all the miners on the basis of their score. But this method also doesn't require a miner to submit his solution immediately. So it may or may not be incentive compatible.

The scoring function is given by C = exp(T/k) where C is the score and T is the time that has elapsed since the start of the round. k is some constant that determines the decay rate.

Since the scoring function is time-dependent, a participant may tend to delay reporting, for some number of shares, when a block is found. Doing so would increase his score exponentially thereby increasing his portion of reward. Intuitively, Slush method is not incentive compatible.

Theorem 1. *The reward function for Slush method is not incentive compatible.*

PROOF. *The reward function for the Slush's method is given by,*

$$R_i(b) = B(1-p) * \frac{C_i(t(b))}{C(t(b))}$$

and

$$R_i(b+ei) = B(1-p) * \frac{C_{i+1}(t(b+1))}{C(t(b+1))}$$

From Lemma 1 we know that the following condition needs to be satisfied for a reward function to be incentive compatible:

$$\sum_{j=1}^{n} \alpha_j * (R_i(b+e_j) - R_i(b)) \leq \frac{E_b[R_i(b)]}{D}$$

$$= \alpha_i * (B(1-p) * \frac{C_{i+1}(t(b))}{C(t(b))}) - (1-\alpha_i) * (B(1-p) * \frac{C_i(t(b))}{C(t(b))}$$

For the method to be incentive compatible, we know that,

$$\alpha_i * (B(1-p) * \frac{C_{i+1}(t(b))}{C(t(b))}) - (1-\alpha_i) * (B(1-p) * \frac{C_i(t(b))}{C(t(b))} > \alpha_i * (B(1-p) * \frac{C_i(t(b))}{C(t(b))}$$

From the above equation it is clear that this condition doesn't hold for all values of α_i and hence Slush's method is not incentive compatible.

For example, when α_i=0.25 the equation is violated and hence Slush method is not incentive compatible.
Also, we show that:
Considering left hand side.

$$\sum_{j=1}^{n} \alpha_j (R_i(b_t + e_j) - R_i(b_t)$$

$$= \alpha_r i(\frac{b_i + e^{T/C}}{N + e^{T/C}} - \frac{b_i}{N}) + (1-\alpha_i)(\frac{b_i}{N + e^{T/C}} - \frac{b_i}{N})$$
$$= \alpha_i(\frac{Ne^{T/C} - b_i e^{T/C}}{N(N + e^{T/C})}) + (1-\alpha_i)(\frac{-b_i e^{T/C}}{N(N + e^{T/C})})$$
$$= e^{T/C}(\frac{N\alpha_i - b_i}{N(N + e^{T/C})})$$
$$= e^{T/C}(\frac{\alpha_i - \frac{b_i}{N}}{N + e^{T/C}})$$

For incentive compatibility :

$$e^{T/C}(\frac{\alpha_i - \frac{b_i}{N}}{N + e^{T/C}}) \leq \frac{\alpha_i}{D}$$

$$= \frac{b_i}{N} \geq \alpha_i(1 - \frac{N + e^{T/C}}{De^{T/C}})$$

The above inequality is not guaranteed to be true always. There exist values of b_i, N such that condition is violated. Consider the case where $b_i = 20$ and N=1000 the LHS of the above equation turns out to be 20/1000 which is not greater than the RHS. In the RHS, considering the worst case where the complete exponent term turns out to be zero then we are left with α_i. Assuming value of α_i to be 0.25 as considered in the previous case it is very clear that LHS is not greater than or equal to RHS. (Since 20/1000 < 0.25). Hence the condition is violated for this specific case.

3.2. Geometric method

Geometric method is also a score based method which was designed to resist pool-hopping. It is designed in such a way that the expected payout per share submitted is always the same no matter when it was submitted. Geometric method is not one-hundred percent immune to pool-hopping. There are two fees: fixed fee and a variable fee or also known as score fee. Fixed fee(f) is the constant fraction of the reward taken by the operator.Variable fee(c) is similar to the decaying score assigned to the participants.

Let r be the decay factor and s be the counter to keep track of the score to be given for the next share submitted.S_k is the score of the k^{th} participant.

The geometric method algorithm which deals with difficulty changes is as follows:

1. Calculate p = $\frac{1}{difficulty}$ and choose f and c.

2. Compute r = $1 - p + \frac{p}{c}$. Assign the operator a score of $\frac{1}{r-1}$.

3. Set s=1 and S_k=0 at the start of every round

4. When a user submits a share: $S_k = S_k + spB$ and let $s = s * r$.

5. If the difficulty changes to a new value then $\frac{1}{difficulty}$ is now p^2:
Let $s = s * p^2/p$ and $r = 1 - p^2 + \frac{p^2}{c}$

Continue with step 4 for each submitted share henceforth.

6. When round ends, sum up all scores and divide the reward proportionally.

The operator claims a fixed fee of fB out of every block and a variable fee of *c(1-f)*B. So out of every block the operator gets a total of ($c + f$ - fB) on average and *(1-c)(1-f)*B is paid to the participants.The expected payout per share is *(1-c)(1-f)*pB. The sum of all scores can be evaluated easily using the formula of geometric progression,since each difficulty change creates a new progression.

Since the scoring function is time-dependent and also a variable fee is associated, a participant may tend to delay reporting for some number of shares when a block is found. Doing so would increase his score exponentially thereby increasing his portion of reward. Intuitively, Geometric method is not incentive compatible.

Theorem 2. *The reward function for Geometric method is not incentive compatible.*

PROOF. The reward function for the geometric method is given by,

$$R_i(b) = \frac{s_k}{\sum_{i=-\infty}^{N} r^{i-1} pB} * (1-f)B$$

and

$$R_i(b+e_i) = \frac{s_k}{\sum_{i=-\infty}^{N+1} r^{i-1} pB} * (1-f)B$$

$s_k + r^{N+1}pB > rs_k$

where r = 1 - p + $\frac{p}{c}$

and p = $\frac{1}{D}$

$s_k = s_k + spB$

and then s=sr

From Lemma 1 we know that the following condition needs to be satisfied for a reward function to be incentive compatible:

$$\sum_{j=1}^{n} \alpha_j(R_i(b_t+e_j) - R_i(b_t)) \leq \frac{E_b[R_i(b)]}{D}$$

For geometric method we have,

$s_k + r^{N+1}pB > [1 + p(\frac{1}{c} -1)]*s_k$

$s_k + r^{N+1}pB > s_k + s_k p(\frac{1}{c} -1)$

$r^{N+1}B > s_k p(\frac{1}{c} -1)$

$(1 + p(\frac{1}{c} -1))^{N+1}B > s_k(\frac{1}{c} -1)$

which can be approximated to,

$(N+1)p(\frac{1}{c} -1)B > s_k(\frac{1}{c} -1)$

$(N+1)pB > s_k$

$NpB + pB > s_k$

$NpB + pB > \frac{r^N-1}{r-1}$

$NpB + pB > \frac{(1+p(\frac{1}{c}-1))^N-1}{r-1}$

$NpB + pB > \frac{Np(\frac{1}{c}-1)-1}{1+p(\frac{1}{c}-1)-1}$

By expanding and simplifying the above terms on RHS we get,

$NpB + pB > N - \frac{1}{p(\frac{1}{c}-1)}$

$NpB > \frac{Np(\frac{1}{c}-1)-1}{p(\frac{1}{c}-1)}$

$NpB > \frac{Np(1-c)-1}{p(1-c)}$

Thus, NpB > N approximately if and only if p and B are greater than or equal to 1.

Hence, we have proved that $R_i(b+e_i)$ is not always greater than $R_i(b)$ and therefore, the miners need not report shares immediately. Since shares are not reported immediately in all cases, Geometric method is not incentive compatible.

For example, when B=0.5 the above equation $NpB > \frac{Np(1-c)-1}{p(1-c)}$ is not satisfied and violates. Hence, Geometric method is not incentive compatible.

3.3. Our proposed method

In the above sections, we have proved that Slush method and Geometric method are not incentive compatible. So it is desirable to devise a new reward function that is incentive compatible.
Now, let us define the reward function as follows:

$$R_i(b) = \frac{b_i}{D+\beta}$$

where b_i is the number of shares submitted by the miner
D is the difficulty of mining and
β is the total number of shares submitted in that round.

From Lemma 1 we know that the following condition needs to be satisfied for a reward function to be incentive compatible:

$$\sum_{j=1}^{n} \alpha_j(R_i(b_t+e_j) - R_i(b_t)) \leq \frac{E_b[R_i(b)]}{D}$$

For the new method devised we have,

LHS=$\frac{\alpha_i(b_i+1)}{D+\beta+1}$ - $\frac{\alpha_i(b_i)}{D+\beta}$ + (1-α_i) * $\frac{b_i-bi}{D+\beta}$

The last term in the above equation turns out to be zero and hence the equation reduces to,

$\frac{\alpha_i(b_i+1)}{D+\beta+1}$ - $\frac{\alpha_i(b_i)}{D+\beta}$

For incentive compatibility,

$\frac{\alpha_i(b_i+1)}{D+\beta+1}$ - $\frac{\alpha_i(b_i)}{D+\beta}$ <= $\frac{\alpha_i}{D}$

This implies,

$\frac{(b_i+1)}{D+\beta+1}$ - $\frac{(b_i)}{D+\beta}$ <= $\frac{1}{D}$

Simplifying we get,

$\frac{-\alpha_i+D+\beta}{(D+\beta+1)(D+\beta)}$ <= $\frac{1}{D}$

Let us now consider two cases,
Case 1: When D > β

Let D=2000, b_i=100, β =1000.

Substituting the assumed values in the above equation we get,

$\frac{-100+1000+2000}{3000*3001}$

This implies that,

$\frac{2900}{3000*3001} <= \frac{1}{2000}$

Simplifying the values we get,
0.0003 <= 0.0005

Case 2:When β > D

Let D=1000, b_i=100, β =2000.

Substituting the assumed values in the above equation we get,

$\frac{-100+2000+1000}{3000*3001}$

This implies that,

$\frac{2900}{3000*3001} <= \frac{1}{1000}$

Simplifying the values we get,
0.0003 <= 0.001

Case 1 and Case 2 show that the incentive compatibility condition is satisfied for both the cases.
Hence, we conclude that this reward function is incentive compatible.

3.4. Maximum Pay Per Share-MPPS

The Pay-Per-Share approach instantly rewards a participant when he submits a share. Each share accounts to exactly the expected value of each hash attempt given by (1-f)pB where f is the fixed fraction of block reward. The PPS system rewards the participants irrespective of how many blocks are found and the operator keeps the remaining block rewards. Hence, no losses are incurred due to pool hopping which is ineffectual in this method. However, this is a risky scheme for the pool operator and there exist possibilities for the pool balance to get exhausted.
Attempts have been made to make the PPS risk-free. One such proposal is the Maximum Pay Per Share (MPPS).
Here each participant bears two types of balances- PPS balance and proportional balance. The PPS balance of each participant is incremented whenever a share is submitted by him. The proportional balance of each participant increases whenever a block is found. The amount received by each participant is the minimum of his PPS balance and his proportional balance.
Since the total amount payed out is less than the total reward earned by the pool, the pool cannot go bankrupt. But still there are several defects in this scheme. The one under discussion here is the pool hopping in MPPS.

3.4.1. Pool Hopping in MPPS

The PPS balance is always less than or equal to the proportional balance. So the reward obtained by each party depends solely on his proportional balance as if it is a proportional pool.
The intuition behind how pool-hopping that occurs in a proportional system is based on the dependence of payout per share on N(total shares submitted in a round) i.e.

$$\text{payout per share} \propto \frac{1}{N}$$

So, as the length of a round grows, the value of each share depreciates. Eventually, it will be advantageous if the person submits his shares early in the round and then hops elsewhere. Thus, a pool-hopper can establish high proportional balance by mining till the number of shares submitted is 43.5% difficulty in that pool.

Each round is independent of the previous rounds. Supposing x_1 shares are submitted by a participant out of y_1 shares in round one and the pool receives reward at the end of every round. The contribution of shares by him is x_1/y_1. The pool-hopper can maximize his profit by leaving the pool once 43.5% of shares have been already submitted in this round.
In the next round, x_2 shares are submitted by the participant out of a total of y_2 shares. His contribution in this round would be x_2/y_2. In order to obtain maximum profit, he has to maximize his proportional balance which is $x_1/y_1 + x_2/y_2$. This is possible only when x_2/y_2 is maximized because x_1/y_1 is already maximized in the previous round. With this regard, the pool-hopper has to leave the pool once the number of shares submitted has reached 43.5% in this round.
So in a pool-hopper's prospect, fleeing the pool at every round once 43.5% shares are already submitted favours the hopper.

Suppose there is only one proportional pool. There are n rounds with the participant's contribution being I_i shares and the total number of shares in each round being N_i , i ∈ 1,2,....n with $N_i > I_i$.

$$Pr(N|N > I) = p(1-p)^{N_i - I_i - 1}$$

The expected payout for a share submitted in each round is:

$$E(B/N|N > I) = \sum_{N_i=I_i+1}^{\infty} \frac{p(1-p)^{N_i-I_i-1}B}{N_i}$$

The expected payout for n rounds is:

$$\sum_{N_1=I_1+1}^{\infty} \frac{p(1-p)^{N_1-I_1-1}B}{N_1} + \sum_{N_2=I_2+1}^{\infty} \frac{p(1-p)^{N_2-I_2-1}B}{N_2} + ... + \sum_{N_n=I_n+1}^{\infty} \frac{p(1-p)^{N_n-I_n-1}B}{N_n}$$

which becomes equal to fair average payout when
$I_1 = I_2 = ... = I_n = I$.
Therefore, the expected payout turns out to be

$$n \cdot \sum_{N=I+1}^{\infty} \frac{p(1-p)^{N-I-1}B}{N}$$

We can evaluate this expression using Hurwitz Lerch transcendent in integral form as:

$$E_1(x) = \int_1^{\infty} \frac{exp(-xt)}{t} dt$$

Assuming x=pI and y = pN, we get

$$E(B/N|y > x) \approx n \cdot \int_x^{\infty} \frac{p(1-p)^{\frac{(y-x)}{p}}B}{y} dy \approx n \cdot exp(x)E_1(x)pB$$

Let $exp(x)E_1(x)$ be f(x). f(x) denotes the amplification factor when xD shares have already been submitted. Considering the asymptotics of the function:

$$for \quad x \approx 0, \quad f(x) = -lnx - \gamma$$

$$for \quad x \approx \infty, \quad f(x) = \frac{1}{x+1}$$

It should be noted that we get f(x)=1 (solo mining) when $x_o = 0.435$ or 43.5 %. So the pool hopper has to mine in a proportional pool for $x < x_o$ with amplification f(x) and solo otherwise with amplification 1. The expected amplification for one round is :

$$E(\frac{B/N}{pB}) \approx \int_0^{x_o} exp(-x)f(x)dx + \int_{x_o}^{\infty} exp(-x)dx \approx 1.28149$$

So for *n* rounds , the pool-hopper can get 128 % of the normal payout times *n*, the number of rounds.
In this section pool hopping for MPPS has been considered. In [19], another type of atack such as coin hopping is described. MPPS is not completely free from coin hopping attacks.Since, MPPS is a pool mining technique, miners who belong to the same pool trust each other but this may result in coin-hopping. Generally, pool operates makes sure that no dishonest miners are allowed but still htere is a possibility of attack. But this becomes an attack only when the miners are frequently hopping switching between the coins.

3.5. Hurdles to pooled mining

Apart from pool hopping, another common type of attack vector in pooled mining is the block withholding attack [18]. With the current pool mining methods, miners can delay submitting a valid block or even hold it indefinitely.There are two types: sabotage and lie in wait. Sabotage is a case where the miner holds the share indefinitely and hence the pool operator is under loss but the miner has no benefits. The main drawback of pooled mining is that if a dishonest miner or an attacker takes control of the pool he/she will take away all the profit leaving behind all the pool miners under loss.Irrespective of which score based method: Slush method or Geometric method is used, both are subject to this type of attack vector since attack does not depend on the reward function used.

[20] talks about the non-outsourceable puzzles that deters coalitions in Bitcoin network. Coalitions can be in the form of hosted mining and pooled minings. A weak non-outsourceable puzzle deters mining pools while strong non-outsourceable puzzles prevents both hosted mining and mining pools. An non-outsourceable puzzle combined with suitable modifications to a reward function can thwart coalitions, making sure that individual mining is the most beneficial strategy.

4. Results and Discussion

For implementation C# Language with Microsoft Visual Studio 2015 has been used. A pool of 5 miners was created having different hashing power, different network latency from miner to pool server and tried to generate the graphs with their different. There are multiple classes to handle every element individually.

Say, Miner A with 10% of pool hashing power and is nearest to the pool
Miner B with 20% of pool hashing power and having a latency of 5 units
Miner C with 40% of pool hashing power and having a latency of 4 units
Miner D with 10% of pool hashing power and having a latency of 2 units
Miner E with 20% of pool hashing power and having the highest latency

On Execution the program performs the mining operation on the basis of miner's properties and generates a csv file having the information about the block creation, the reward gained by each miner and the pool.
Now this .csv file is used to create the graph with the help of MathPlot Library available in Python.

Simulation was carried out for different miners with different hashing powers and the graphs for the same are plotted. In all the below graphs, the X-axis indicates the number of blocks generated and the Y-axis indicates the reward generated for the particular miner. For a reward function to be incentive compatible, the graph should be a linear function parallel to X-axis that is the reward generated by the miner should be constant. But, from the below graphs we can infer that it is exponentially distributed and hence not incentive compatible.

The various miners reward to block generation for the Slush method have been illustrated in the below graphs.

Miner A Reward to Block Generation
Reward earned by Miner A with 10% hashing power.

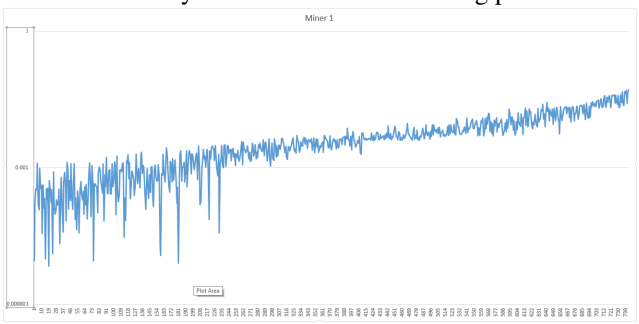

Fig 1: Miner A Reward to Block Ratio

Miner B Reward to Block Generation
Reward earned by Miner B with 20% hashing power.

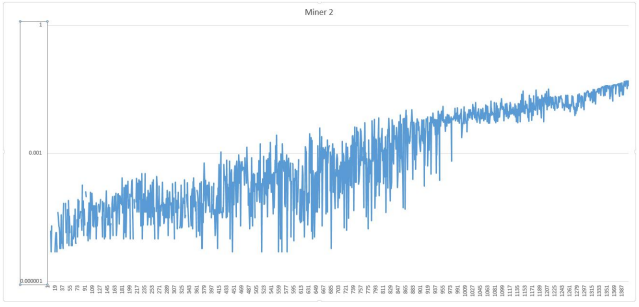

Fig 2: Miner B Reward to Block Ratio

Miner C Reward to Block Generation
Reward earned by Miner C with 40% of hashing power.

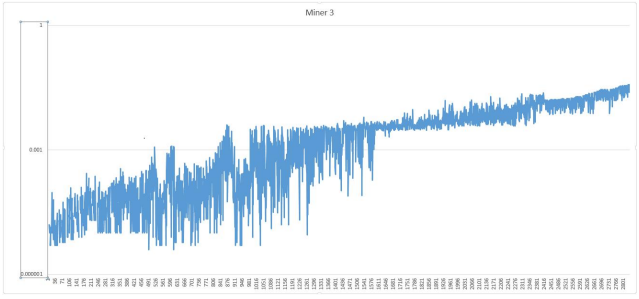

Fig 3: Miner C Reward to Block Ratio

Miner D Reward to Block Generation
Reward earned by Miner C with 20% of hashing power.

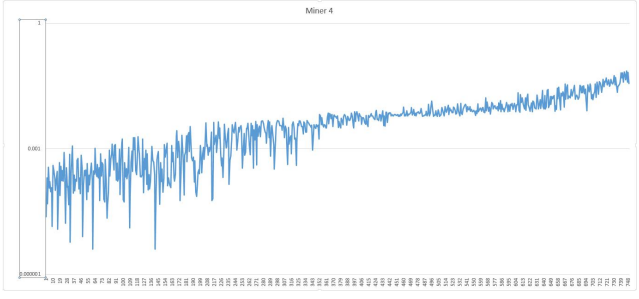

Fig 4: Miner D Reward to Block Ratio

Miner E Reward to Block Generation
Reward earned by Miner E with 20% of hashing power and highest latency.

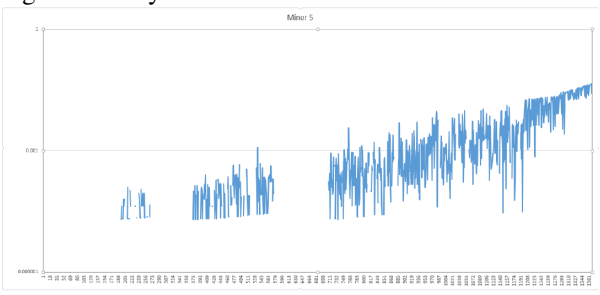

Fig 5: Miner E Reward to Block Ratio

The below figure i.e Fig 6 shows the overall Slush pool reward and when compared with Incentive Compatible Function, the reward per block is very less.

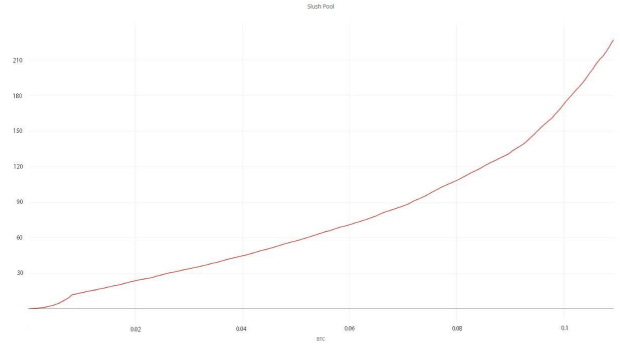

Fig 6: Pool Reward to Block Ratio

A reward function is said to be incentive compatible if it is a linear function but the above simulation results show an exponential behaviour. Hence, from simulation results also it is very clear that the Slush method and Geometric method are not incentive compatible.

5. Conclusion

In this paper,we have proved that the Slush method and Geometric method, for computing the reward function in pooled mining, are not incentive compatible. The mathematical proof based on the reward function and also simulation results, both confirm that the above methods are not incentive compatible.
Also, we have proved that MPPS method is not hopping proof.
This work could be extended to build a new incentive compatible model which will revisit the properties of incentive compatibility. Also, it would be interesting to check whether the score based methods such as Slush method and Geometric method are budget balanced or not.

References

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

[2] Andrew Miller, Ahmed Kosba, Jonathan Katz, Elaine Shi, Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Pages 680-691, October 2015.
[3] Meni Rosenfeld, Analysis of bitcoin pooled mining reward systems, Sept 2011.
[4] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden, Stanford University, Incentive Compatibility of Bitcoin Mining Pool Reward Functions, Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, Revised Selected Papers pp.477-498, February 22–26, 2016
[5] Ittay Eyal, The Miner's Dilemma: In IEEE Symposium on Security and Privacy,2015.
[6] Aron Laszka, Benjamin Johnson, and Jens Grossklags. When Bitcoin Mining Pools Run Dry: A Game-Theoretic Analysis of the Long-Term Impact of Attacks Between Mining Pools, In Workshop on Bitcoin Research, 2015.
[7] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of DDoS attacks against Bitcoin mining pools.In Workshop on Bitcoin Research, 2014.
[8] Ittay Eyal and Emin Gun Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Financial Cryptography, 2014.
[9] https://bitcoin.org/en/bitcoin-core/
[10] https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/
[11] Python pool hopper proxy. http://bitcointalk.org/?topic=26866.
[12] https://bitcointalk.org/?topic=18313.0.
[13] Bitcoin p2p virtual currency. http://www.bitcoin.org/.
[14] Raulo.Optimal pool abuse strategy, 2011. http://bitcoin.atspace.com/poolcheating.pdf.
[15] Bitcoin pooled mining. http://mining.bitcoin.cz/.
[16] Bitcoin and Cryptocurrency Technologies by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Princeton University Press, 2016.
[17] Meshkov, D., Chepurnoy, A., and Jansen, M. Revisiting Difficulty Control for Blockchain Systems. In Cryptocurrencies and Blockchain Tech. (2017), pp. 429–436.
[18] Loi Luu and Ratul Saha and Inian Parameshwaran and Prateek Saxena and Aquinas Hobor, "On Power Splitting Games in Distributed Computation: The Case of Bitcoin Pooled Mining", Cryptology ePrint Archive: Listing for 2015.
[19] https://docs.ergoplatform.com/ErgoPow.pdf
[20] "Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions" by Andrew Miller, Ahmed Kosba,Jonathan Katz,Elaine Shi, 2014