Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

CAPTURING MINER AND MINING POOL DECISIONS IN A BITCOIN BLOCKCHAIN NETWORK: A TWO-LAYER SIMULATION MODEL

Kejun Li
Yunan Liu
Hong Wan
Ling Zhang

Edward P. Fitts Department of Industrial and Systems Engineering
North Carolina State University
111 Lampe Drive
Raleigh, NC 27607, USA

ABSTRACT

Motivated by the growing interests in Bitcoin blockchain technology, we build a Monte-Carlo simulation model to study the miners' and mining pool managers' decisions in the Bitcoin blockchain network. Our simulation model aims to capture the dynamics of participants of these two different parties and how their decisions collectively affect the system dynamics. Given the limited amount of monetary budget and mining power capacity, individual miners decide on which mining pools to join and determine how much hashing power to invest. Mining pool managers need to determine how to appropriately allocate the mining reward and how to adjust the membership fee. In addition to the aforementioned miner and pool behavior, we also characterize the system-level dynamics of the blockchain in terms of mining difficulty level and total hashing power.

1 INTRODUCTION

Blockchain is a distributed database that maintains a dynamic list of records, secured against tampering and revision. As the first and most famous application of the technology, Bitcoin blockchain has attracted tremendous attention since proposed by Nakamoto et al. (2008). In the Bitcoin network, the decentralized community of individual participants replaces the trusted third party in traditional centralized systems. Contrary to conventional distributed systems that employ Byzantine fault tolerance consensus (Bracha and Toueg 1985; Correia et al. 2011), the Bitcoin system is considered revolutionary because it achieves a consensus of the ordering and confirmation of transactions among untrusted distributed participants via a *mining* game, which is referred to as the Nakamoto consensus. Specifically, participants of the mining game, also called *miners*, compete to be the first to solve a highly complex computation puzzle. The winner will confirm the next block of transactions, and more importantly, earn the corresponding payoff. The incentive mechanism of the system is designed as follows: winning miners of each mining game are rewarded with a certain amount of newly minted Bitcoin (12.5 Bitcoins) and transaction fees from general users as compensation.

Continuous participation in the mining game can be extremely energy-consuming, which incurs a nontrivial amount of costs. Nevertheless, there have been nearly $10,000$ active Bitcoin network nodes (participants) on a daily basis, since the bull run of Bitcoin price (nearly $\$20,000$) in late 2017 (Yeow 2020; Coin Dance contributors 2020). At the time of writing, the Bitcoin-USD rate is around $\$7,000$ (Blockchain.com contributors 2020), which consequently is attracting miners to conduct the daily operations of the system, such as minting new coins and recording transactions. It is well known that solo-mining

will no longer be able to sustain any profitable mining activities in the current days. In order to win mining games more steadily, profit-driven individual miners have conglomerated to form mining pools. Miners in the same mining pool collaborate with each other to compete in the mining game, and more importantly, share the mining reward (Rosenfeld 2011).

In the latest years, there are several papers using simulation models to evaluate various aspects of blockchain systems. Two main simulation methodologies are employed: discrete-event and agent-based models. In the first stream, Alharby and van Moorsel (2019) propose an event-driven model with transactions and emphasize on the block creation through *proof-of-work* (PoW); Aoki et al. (2019) involve events of block generation, block propagation, and message transmission/reception. In the second stream, Cocco and Marchesi (2016) reproduce the economy of the mining process with heterogeneous agents by including the Bitcoin transactions and price series; Kaligotla and Macal (2018) provide a generalized framework of modeling blockchain simulation by illustrating the essential agents and functioning of the system; Rosa et al. (2019) develop a security attack testing platform by exploiting *parallel and distributed simulation* (PADS) techniques with extended scalability.

To the best of our knowledge, we are the first to consider the individual budget constraint during the mining process. In addition, we embed more details of the process of block production through submitting shares within mining pools. Meanwhile, different share-based pool reward policies are also included. To investigate the miner behavior, two crucial decisions are involved, i.e., mining and pool selection. Aiming to provide a platform with comprehensive functionalities and configurations of the Bitcoin system, our proposed model also includes the membership fee adjustment and the adaptive difficulty mechanism.

Our simulation results generate some interesting results. First, our result shows that an initial oligopolistically distributed mining market does not eventually develop into a monopoly in the course of time. Next, we also observe that medium size mining pools can attract more individual miners than both small and large pools; this may provide some guidelines to the manager/operator of the emerging Bitcoin mining pool. Third, we validate that the dynamic adjustment of the mining difficulty is effective in terms of maintaining a stable block generation rate. Finally, we reveal an interesting relationship between the mining power allocation rule and the real-time difficulty level; such a relationship may help an individual miner to predict the dynamics of the total hashing rate of a new campaign by observing the change of difficulty level, in order to optimize her mining strategy.

The remainder of the paper is organized as follows. Section 2 describes model settings, assumptions and detailed algorithms. Section 3 gives input data and parameters, and presents experiment results. Finally, we provide concluding remarks and discuss future research opportunities in Section 4.

2 SIMULATION MODEL

2.1 Multi-Layer Model Scope

Our simulation model aims to explore what influences the collective and heterogeneous behaviors of miners and mining pools.

Layer 1: individual miner. The first and most fundamental layer of decision is individual miners. Because they participate in mining to gain profits, miners will make their mining decisions based on their expected profit. For each block added, the reward of the winner consists of two parts: transaction fees (roughly $1 \sim 2$ Bitcoins) and the Coinbase reward (currently 12.5 Bitcoins). The costs of mining include hardware costs and utility costs. The fixed cost of mining hardware ranges from \$50 to $\$10,500$ in the market. According to Morgan Stanley 2017 data, the total energy consumption of the Bitcoin network is equivalent to the total electricity supporting 2 million U.S. homes.

In addition to the mining decision (i.e., *to mine or not to mine*), miners will also decide which mining pool to join. Although the mining was first visioned to be performed by personal computers, over the years, individual and group miners have conglomerated to form mining pools, because mining pools provide a more steady income stream. On the other hand, mining pools charge membership fees, which is around

$1\% \sim 3\%$ of the total reward for the main-stream ones. Furthermore, depending on the reward-sharing policy, joining a mining pool affects the reward and cost structure for individual miners.

Layer 2: mining pool. Mining pool managers form the second layer in the Bitcoin blockchain network. We consider the case where the mining pool only serves as a centralized collaboration platform for miners, which is the direct opposite of the original Bitcoin design of a decentralized network. The primary objective in managing a mining pool is to make a profit, which consequently requires the manager to balance the incoming of new miners and the departure of old ones. New miners bring along hashing capacities, which increase the winning probabilities. However, should there exist any mining pool whose total hashing power is large enough to dominate the mining game, Bitcoin participants will inevitably question the credibility of the system, which may result in the abandonment of network supporters and eventually the collapse of the entire network. This would deprive the purpose and economic opportunities of a mining pool. Given Bitcoin system states and available hashing power, pool managers aim to set the proper membership fees and reward policies.

Metrics of Bitcoin system. We monitor the system-level dynamics, such as difficulty level and average block generation times, while individual miners and mining pool managers are making decisions on different levels. At the beginning of each 2-week period, the hashing difficulty level will be automatically adjusted; the goal of this is to maintain a steady block generation rate.

2.2 Model Assumptions

In our simulation model, players (i.e., individual miners and pool managers) conduct a campaign-repeated game. We set the maximum of campaigns to be w and the counter of campaigns to be W. Within each campaign, there are exactly $n = 2,016$ valid blocks generated. We denote by X^N the time to first generate the N^{th} valid block in a single campaign. By Satoshi's design, mining difficulty (i.e., a measure of how difficult it is to find a hash value below a given target) is updated at the end of the W^{th} campaign by

$$D^{W+1} = D^W \frac{600n}{\sum_{N=1}^{n} X^N} \tag{1}$$

with D^W representing the difficulty of the W^{th} campaign. The scaling is to maintain a nearly constant block generating rate (1 per 600 seconds on average) (Narayanan et al. 2016). The residual time of the current campaign is estimated by $\hat{T}(N) = 600(n-N)$. Let $\mathscr{I}$ and $\mathscr{J}$ denote the sets of miners and pools. Furthermore, we denote by $\mathscr{I}_\circ$ the set of "idle" miners, $\mathscr{I}_p$ the set of passive miners, and $\mathscr{I}_a = \mathscr{I} \setminus \mathscr{I}_p$ the set of active miners, which will be explained later.

Assumptions on miner behavior. From the perspective of miner $i \in \mathscr{I}$, she is characterized by a type vector $\theta_i = (b_i, c_i, \gamma_i, p_i)$, where b_i is the mining budget (\$) within a campaign (correspondingly, we define B_i to be her residual budget (\$) in the campaign), c_i is the mining cost (\$/hash), γ_i is the individual valuation parameter of Bitcoin, and p_i is her maximal mining power (hash/s). An individual miner's first decision is to mine or not to mine. Let q_i be binary variable, we write $h_i = q_i p_i$ as the mining power (hash/s) an individual miner spends in the mining game. The "idle" miner set is formally defined as $\mathscr{I}_\circ = \{i \in \mathscr{I} \mid q_i = 0\}$. We denote by $\hat{T}_i = B_i/(c_i h_i)$ the estimated residual time until exhausting the budget under the mining policy h_i, $\hat{T} = \min_i\{\hat{T}_i\}$, and $I_\circ = \arg\min_i\{\hat{T}_i\}$. When a miner is making a decision, she may face the following two scenarios.

(i) When a block is mined and broadcast to the whole network, a miner will decide whether to turn on her mining machine if it is profitable to participate in the mining of the next block in expectation.

(ii) In addition to Scenario (i), each "idle" miner $i \in \mathscr{I}_\circ$ will periodically check if a new block is released (this occurs once every Δ seconds). After l consecutive attempts with "negative" outcomes, the minor will, with probability $\eta(l)$, decide whether to run her mining machine. We assume $\eta(l)$ is increasing in l.

This monitoring mechanism mentioned in Scenario (ii) above can (a) maintain an acceptable fraction of open mining machines at all times, and (b) help an "idle" miner to closely track the trend of the overall hashing rate (regardless of other factors, such as Bitcoin market price, it can be a promising opportunity to participate in mining game when the overall hashing rate is low). If a miner decides to participate in mining the next block, she will execute the mining decision with probability

$$\beta_i = \min\left\{\frac{B_i}{\hat{T}(N)c_i p_i}, 1\right\}.$$

A miner turns on the machine if her residual budget can cover the estimated mining expenditure (i.e., $\hat{T}(N)c_i p_i$) until the end of the campaign. Because all budgets will be refilled at the beginning of a new campaign, miners can be more aggressive in executing mining decisions towards the end of the current campaign. For a miner i, the time to generate the N^{th} block (i.e., X_i^N) after updating difficulty is exponentially distributed (Narayanan et al. 2016) with rate

$$\mu_i = \frac{h_i}{D^W/D^\circ},$$

where D° is the minimal difficulty. Since we also include the pool hopping decision, we use $P_i \in \mathscr{J}$ for miner i's pool index. Moreover, we assume that miner i's valuation of a Bitcoin V_i follows a distribution $F_V(\cdot;\Gamma,\gamma_i)$, parameterized by the exogenous market valuation Γ and her own valuation factor γ_i. In particular, $V_i = \tilde{V}_i \mathbf{1}_{\{\tilde{V}_i \geq 0\}}$ and $\tilde{V}_i \sim \mathscr{N}(\Gamma, \gamma_i^2)$, where Γ and γ_i are estimated by historical prices of Bitcoin (Table 3).

Following Salimitari et al. (2017), we apply *prospect theory* to model the loss and risk aversion nature of miners during the pool hopping process. Suppose a miner with expected profit x joins a pool with mining power share y, her utility is given by

$$U\left(x, y; \lambda_i, \phi_i, \omega_i, \rho_i\right) = V\left(x; \lambda_i, \phi_i\right) \cdot W\left(x, y; \omega_i, \rho_i\right). \tag{2}$$

The value function $V\left(x; \lambda, \phi\right)$ characterizes the *reflection effect*. In particular, it has the form of

$$V(x; \lambda_i, \phi_i) := \begin{cases} x^{\phi_i} & \text{if } x \geq 0 \\ -\lambda_i(-x)^{\phi_i} & \text{otherwise} \end{cases}$$

with parameters $\lambda_i > 1$ and $0 < \phi_i < 1$. Moreover, to include the effect of the mining power distribution among pools, the weight function $W(x, y; \omega_i, \rho_i)$ is used to account for the *certainty effect*. In particular,

$$W(x, y; \omega_i, \rho_i) := \begin{cases} y^{\rho_i}\left[y^{\rho_i} + (1-y)^{\rho_i}\right]^{-1/\rho_i} & \text{if } x \geq 0 \\ y^{\omega_i}\left[y^{\omega_i} + (1-y)^{\omega_i}\right]^{-1/\omega_i} & \text{otherwise} \end{cases}$$

with parameters $0.5 \leq \omega_i < \rho_i < 1$.

Assumptions on pool policies. We denote by F_j^W the membership fee of the W^{th} campaign for the j^{th} mining pool from set $\mathscr{J}$ (i.e., a proportion of the total reward set by the pool manager). We will update the membership fee based on *shares*. A share is a partial solution to the original puzzle of generating a valid block, which is corresponding to a lower difficulty. The number of shares submitted can be used to measure the computational power a miner/pool possesses. We assume that $F_j^1 = F_j^2 \sim \mathscr{U}(a, b)$, and

$$F_j^{W+1} = F_j^W + \alpha_j \mathbf{1}_{\left\{S_j^W / S_j^{W-1} > 1+\varepsilon\right\}} - \alpha_j \mathbf{1}_{\left\{S_j^W / S_j^{W-1} < 1-\varepsilon\right\}}, \quad W \geq 2, \tag{3}$$

where S_j^W counts the total shares produced by the j^{th} pool within the W^{th} campaign, the constant $\varepsilon \in (0,1)$ controls the sensitivity in the change of share numbers, and α_j is the step size for the membership fee adjustment. If the production of shares in a specific pool changes significantly, which leads to a notable fluctuation of its mining power, the manager will take action to adjust the membership fee. In the simulation, we apply two share-based mining pool reward policies (Rosenfeld 2011).

- *Proportional system* (PROP): at the end of every round (i.e., the time between one block successfully mined by the pool to the next), the pool manager will distribute the block reward among miners, in direct proportion to the number of shares they submitted during this round.
- *Pay per last N shares* (PPLNS): instead of using the total number of shares in a round, the pool manager focuses on the last "N" shares.

To implement these two policies, we introduce the following additional parameters: the difficulty shadow factor of the j^{th} pool δ_j; the current difficulty to generate a valid share for the miners in the j^{th} pool $D_j^W = D^W/\delta_j$; the number of shares the i^{th} miner needs to get a valid block S_i^b, which is a geometric random variable with parameter $1/\delta_{P_i}$; the miner's inter-share times $\{X_{ik}^s\}_{k=1,\cdots,S_i^b}$, which are independent exponential random variables with rate

$$\nu_i = \frac{h_i}{D_{P_i}^W/D^\circ} = \delta_{P_i}\mu_i.$$

System level metrics. On the system level, we record the following metrics:

- X^N: the time first generating the N^{th} valid block in a campaign, $X^N = \min_i\{X_i^N\}$;
- I^N: the index of the miner first finds the N^{th} valid block in a campaign, $I^N = \arg\min_i\{X_i^N\}$;
- J^N: the index of the mining pool that miner I^N comes from, $J^N = P_{I^N}$;
- bd_j^W: the proportion of blocks found by the j^{th} pool in the W^{th} campaign;
- md_j^W: the proportion of miners of the j^{th} pool in the W^{th} campaign;
- pd_j^W: the proportion of mining power capacity of the j^{th} pool in the W^{th} campaign;
- $hd_j(\tau)$: the proportion of mining power committed in the j^{th} pool at system time τ.

Other assumptions. Furthermore, we list other assumptions as follows.

- The individual valuation of Bitcoin V_i is independent.
- Considering a significant portion of hashing power belongs to the pool manager herself in practice, we introduce the passive miner set $\mathscr{I}_p$ (Cong et al. 2019): passive miners stick to the same mining pool instead of hopping periodically. We assume that a miner is passive with probability π_p.
- Pool hopping with probability: at the end of each campaign, a miner prioritizes pools and joins the k^{th} one with probability hp_k.
- Miners pool-hop at the end of each campaign, and the hopping time window is negligible.
- To avoid the collapse of the trustworthiness of the Bitcoin system caused by the over-centralization of the hashing power, each mining pool sets an upper threshold UB of pool capacity.
- We ignore transaction fees included in each block.

2.3 Simulation Algorithms

In this subsection, we describe the detailed simulation algorithms. The main structure of the simulation is illustrated in Figure 1. We describe the following *event list*: the time point of the next block generation t_A, the time point of the next block check t_B, and the time point of the next miner to run out her budget t_C. The system state is updated by the *next-event time advance* approach in Algorithm 1.

Suppose miner i turns on her machine, the individual winning probability is

$$\mathbb{P}_i = \frac{\tilde{\mu}_i}{\tilde{\mu}_i + \hat{\mu}}, \quad \text{where} \quad \tilde{\mu}_i = \frac{p_i}{D^W/D^\circ} \quad \text{and} \quad \hat{\mu} = \frac{N}{\sum_{k=1}^N X^k} \tag{4}$$

are the individual exponential rate when turning on her machine and the estimated overall mining rate, respectively. Then we can use $\mathbb{P}_i$ to update the expected profit of the miner in Step 7 of Algorithm 1.

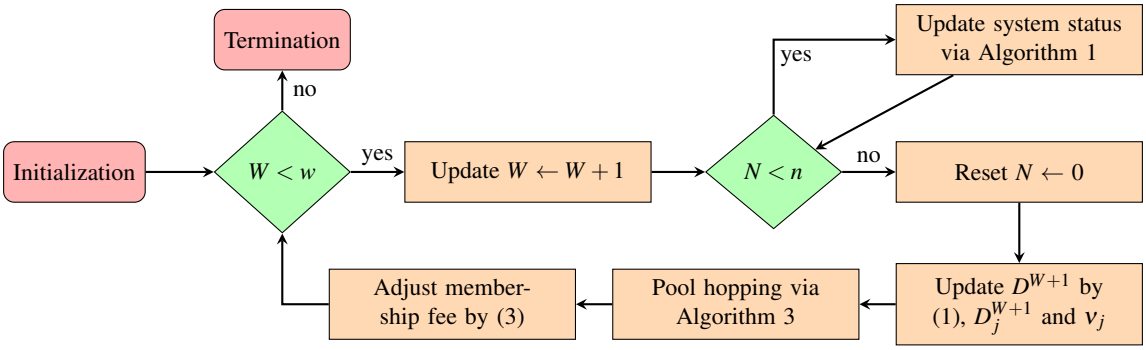


Figure 1: Main routine of simulation

Algorithm 1 Subroutine: update system status

switch $t \leftarrow \min\{t_A, t_B, t_C\}$ **do**
  case $t_A = t$ ▷ The next event is block generation
    Reset $l \leftarrow 0$, update $N \leftarrow N+1$.
    if $N = n$ **then** reset $B_i \leftarrow b_i$, $i \in \mathscr{I}$; **else** update $B_i \leftarrow B_i - c_i h_i (t_A - \tau)$, $i \in \mathscr{I}$. **end if**
    Update $\beta_i \leftarrow \min\left\{\frac{B_i}{\hat{T}(N) c_i p_i}, 1\right\}$, $i \in \mathscr{I}$; $\tau \leftarrow t_A$.
    Distribute block reward within pool J^N.
    Update $\mathbb{P}_i$ by (4), $\mathbb{E}[R_i] \leftarrow 12.5(1-F_{P_i})V_i\mathbb{P}_i - 600 c_i p_i$, $i \in \mathscr{I}$.
    if $\mathbb{E}[R_i] > 0$ **then** update $q_i \leftarrow 1$ w.p. β_i; $q_i \leftarrow 0$ w.p. $(1-\beta_i)$. **end if**
    Update $v_i \leftarrow \frac{q_i p_i}{D^c_{P_i}/D^m}$, $i \in \mathscr{I}$.
    Update $\hat{T}_i \leftarrow \frac{B_i}{c_i h_i}$, $i \in \mathscr{I}$; $\hat{T} \leftarrow \min_i\{\hat{T}_i\}$, $I_\circ \leftarrow \arg\min_i\{\hat{T}_i\}$.
    Update X_i^{N+1}, $i \in \mathscr{I}$. ▷ See details in Algorithm 2
    Update $X^{N+1} \leftarrow \min_{i: X_i^{N+1} \le \hat{T}_i}\{X_i^{N+1}\}$, $I^{N+1} \leftarrow \arg\min_{i: X_i^{N+1} \le \hat{T}_i}\{X_i^{N+1}\}$, $J^{N+1} \leftarrow P_{I^{N+1}}$.
    Update $t_A \leftarrow \tau + X^{N+1}$, $t_B \leftarrow \tau + \Delta$.
  end case
  case $t_B = t$ ▷ The next event is block check
    Update $l \leftarrow l+1$, $\mathscr{I}_\circ \leftarrow \{i \in \mathscr{I} \mid q_i = 0\}$.
    Update $B_i \leftarrow B_i - c_i h_i (t_B - \tau)$, $\beta_i \leftarrow \min\left\{\frac{B_i}{\hat{T}(N) c_i p_i}, 1\right\}$, $i \in \mathscr{I}$; $\tau \leftarrow t_B$.
    Update $q_i \leftarrow 1$ w.p. $\beta_i \eta(l)$; $q_i \leftarrow 0$ w.p. $(1-\beta_i\eta(l))$, $i \in \mathscr{I}_\circ$. Update $v_i \leftarrow \frac{q_i p_i}{D^c_{P_i}/D^m}$, $i \in \mathscr{I}_\circ$.
    Update $\hat{T}_i \leftarrow \frac{B_i}{c_i h_i}$, $i \in \mathscr{I}$; $\hat{T} \leftarrow \min_i\{\hat{T}_i\}$, $I_\circ \leftarrow \arg\min_i\{\hat{T}_i\}$.
    Update X_i^{N+1}, $i \in \mathscr{I}_\circ$. ▷ See details in Algorithm 2
    Update $X^{N+1} \leftarrow \min_{i: X_i^{N+1} \le \hat{T}_i}\{X_i^{N+1}\}$, $I^{N+1} \leftarrow \arg\min_{i: X_i^{N+1} \le \hat{T}_i}\{X_i^{N+1}\}$, $J^{N+1} \leftarrow P_{I^{N+1}}$.
    Update $t_A \leftarrow \tau + X^{N+1}$, $t_B \leftarrow \tau + \Delta$.
  end case
  case $t_C = t$ ▷ The next event is one miner runs out her budget
    Update $B_i \leftarrow B_i - c_i h_i (t_C - \tau)$, $\beta_i \leftarrow \min\left\{\frac{B_i}{\hat{T}(N) c_i p_i}, 1\right\}$, $i \in \mathscr{I}$; $\tau \leftarrow t_C$.
    Update $q_{I_\circ} \leftarrow 0$, $v_{I_\circ} \leftarrow 0$.
    Update $\hat{T}_i \leftarrow \frac{B_i}{c_i h_i}$, $i \in \mathscr{I}$; $\hat{T} \leftarrow \min_i\{\hat{T}_i\}$, $I_\circ \leftarrow \arg\min_i\{\hat{T}_i\}$.
  end case
  Update $t_C \leftarrow \tau + \hat{T}$.
end switch

We next describe how miner i produces a valid block by submitting shares via Algorithm 2.

Algorithm 2 Subroutine: generate individual inter-block time via inter-share time

1: Generate $S_i^b \sim \text{Geo}(1/\delta_{P_i})$.
2: Generate $\{X_{ik}^s\}_{k=1,\cdots,S_i^b} \sim \exp(\nu_i)$.
3: Update the first inter-share time $X_{i1}^s \leftarrow X_{i1}^s + l\Delta$.
4: Update $X_i^{N+1} \leftarrow \sum_{k=1}^{S_i^b} X_{ik}^s$.

We finally discuss the pool hopping decision in Algorithm 3. Suppose miner i joins pool j and a new valid block is solved by the specific pool, the average Coinbase reward share of the miner is estimated by

$$\mathbb{P}_{ij} = S_i^W / S_j^W \mathbf{1}_{\{j=P_i\}} + \frac{S_i^W / S_{P_i}^W bd_{P_i}^W}{bd_j^W + S_i^W / S_{P_i}^W bd_{P_i}^W} \mathbf{1}_{\{j\neq P_i\}}, \tag{5}$$

where S_i^W is the counter of shares generated by the i^{th} miner in the W^{th} campaign (the definition is similar to S_j^W). The proportion of shares yielded by the miner in the current pool is $S_i^W / S_{P_i}^W$, approximating the proportion of her hashing power within the pool. The proportion of blocks produced by the j^{th} pool during the W^{th} campaign is bd_j^W, approximating the proportion of hashing power occupied by the pool. Then we can use $\mathbb{P}_{ij}$ to update the expected profit of the miner i to join pool j in Step 1 of Algorithm 3.

Algorithm 3 Subroutine: pool hopping

1: Update $\mathbb{P}_{ij}$ by (5), $\mathbb{E}[R_{ij}] \leftarrow 12.5(1-F_{P_i})V_i\mathbb{P}_{ij} - 600c_ip_i$, $i \in \mathscr{I}_a$, $j \in \mathscr{J}$.
2: Update $u_{ij} \leftarrow U\left(\mathbb{E}[R_{ij}], bd_W^j; \lambda_i, \phi_i, \omega_i, \rho_i\right)$ by (2), $i \in \mathscr{I}_a$, $j \in \mathscr{J}$.
3: Order $\left\{u_{ij}\right\}_{j\in\mathscr{J}}$ to get the hopping priority $\{j_1^i, j_2^i, \cdots, j_k^i, \cdots, j_K^i\}$ such that $u_{ij_1^i} \geq u_{ij_2^i} \geq \cdots \geq u_{ij_k^i} \geq \cdots \geq u_{ij_K^i}$, where $K = |\mathscr{J}|$, $i \in \mathscr{I}_a$.
4: Update $P_i \leftarrow j_k^i$ w.p. hp_k, $i \in \mathscr{I}_a$.

3 NUMERICAL EXPERIMENTS

3.1 Input Data and Parameters

Mining machines. Rauchs (2020) summarizes more than 80 different SHA-256 mining equipments, among which, we list 5 popular ASICs with distinct hashing power in Table 1. The mining cost of each machine is calculated based on the electricity price of 0.05 \$/kWh. For details, see Rauchs (2020) and references therein. In our simulation, we assume a miner purchases her mining machine from Table 1 with equal probabilities (see Table 3).

Table 1: Typical ASICs in mining market.

	SHA-256 Mining Equipment	Estimated Quantity	Hashing power (Th/s)	Efficiency (J/Gh)	Cost (\$/Th)
1	MicroBT Whatsminer 10S	67,455	55	0.064	8.84E−07
2	Bitfily Snow Panther B1+	390,400	25	0.086	1.19E−06
3	Bitmain Antminer T9	1,015,040	13	0.126	1.75E−06
4	Canaan AvalonMiner 741	89,826	7	0.158	2.19E−06
5	Bitmain Antminer S7	129,610	5	0.273	3.80E−06

Mining power distribution. There are in total 156,695 valid blocks generated over the time span of February 2016 – January 2019. Wang et al. (2019) summarizes the distribution of valid blocks generated

from top Bitcoin mining pools, which is the estimator of the mining power distribution over the network. We consider the top 9 pools and group all other minor pools and solo miners into the 10 category, see in Table 2. This shows that oligopoly indeed exists in the Bitcoin mining game: several players control a large proportion of the hashing power (Cong et al. 2019), but none of them can dominate the entire mining market solely (i.e., exceed the 50% threshold). Our model assumes that the i^{th} miner's pool index P_i is initialized by a discrete probability distribution estimated by the proportions in Table 2.

Table 2: An overview of top 9 Bitcoin mining pools and others.

	1	2	3	4	5
Mining Pool	AntPool	F2Pool	BTC.com	ViaBTC	SlushPool
# of Blocks	27,026	19,282	17,488	12,100	12,002
Percent	17.2%	12.3%	11.2%	7.7%	7.7%

	6	7	8	9	10
Mining Pool	BTC.TOP	BTCC	BitFury	BW.COM	Others
# of Blocks	11,256	10,586	8,754	7,315	30,886
Percent	7.2%	6.8%	5.6%	4.7%	19.7%

Input parameters are listed in Table 3.

Table 3: Summary of input parameter design.

	Parameter	Description
System	$r = 40$	Number of simulation replications
	$w = 15$	Number of campaigns in each replication
	$n = 2,016$	Number of blocks in each campaign
	$\Delta = 600$	Period of block check
	$\eta(l) = 0.35^{6-l}\mathbf{1}_{\{1\le l\le 6\}} + \mathbf{1}_{\{l>6\}}$	Probability of deciding to mine after l "negative" checks in a row
	$\Gamma = \$8,807.71$	Market valuation parameter of Bitcoin
Miner	$\lvert\mathscr{I}\rvert = 300$	Number of miners
	$\pi_p = 20\%$	Initialization probability to generate the passive set $\mathscr{I}_p$
	$\gamma_i = \gamma = \$1,490.16$	Homogeneous individual valuation parameter of Bitcoin
	(p_i, c_i) generated by $\mathscr{U}\{1, 5\}$	Maximal mining power and the corresponding mining cost
	$\xi_i = 200\%, 100\%, 50\%$ equally likely	Scaling factor of budget
	$b_i = 2,016 \cdot 600 \cdot p_i \cdot c_i \cdot \xi_i$	Mining budget of a single campaign
	λ_i generated by $\mathscr{U}(1, 2)$	Loss aversion parameter of value function in prospect theory
	ϕ_i generated by $\mathscr{U}(0, 1)$	Risk aversion parameter of value function in prospect theory
	ω_i generated by $\mathscr{U}(0.5, 1)$	Parameter of weight function for loss in prospect theory
	ρ_i generated by $\mathscr{U}(\omega_i, 1)$	Parameter of weight function for gain in prospect theory
Pool	$\lvert\mathscr{J}\rvert = 10$	Number of mining pools
	$a = 0.01$	Lower boundary of the uniform distribution to initialize the membership fee
	$b = 0.03$	Upper boundary of the uniform distribution to initialize the membership fee
	$\alpha_j = \alpha = 0.002$	Step size of the membership fee adjustment
	$\varepsilon = 0.002$	Sensitivity parameter of the membership fee adjustment
	$\delta_j = \delta = 1,000$	Homogeneous difficulty shadow factor
	$s\delta_j = 2\delta_j$	Window size parameter "N" of PPLNS policy

Continued on next page

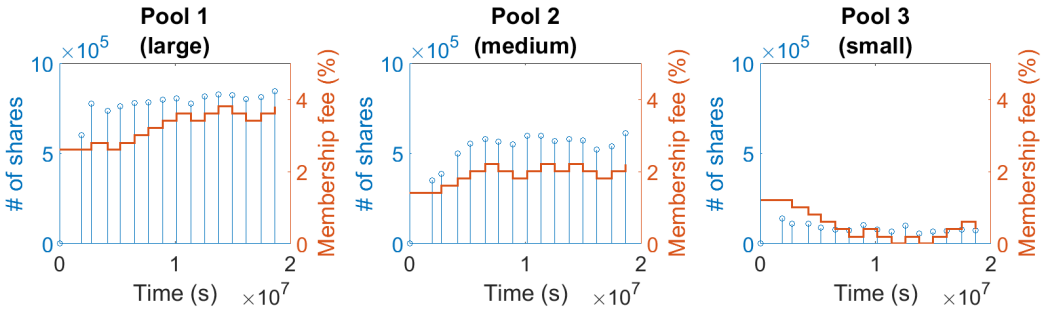
Table 3 – continued from previous page

	Parameter		Description
Pool	$hp_k = \begin{cases} 0.8 & \text{if } k = 1 \\ 0.8(1 - \sum_{g=1}^{k-1} hp_g) & \text{if } 2 \le k \le K-1 \\ 1 - \sum_{g=1}^{k-1} hp_g & \text{if } k = K \end{cases}$		Hopping probability to the k^{th} most profitable pool
	$UB = 0.40$		Upper threshold of pool capacity

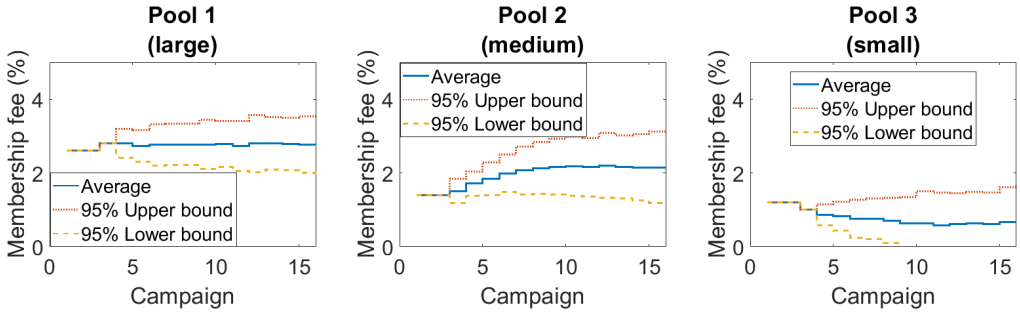
3.2 Experiment Results

We now present our simulation results and discuss how they can be used to generate useful insights. Figure 2 depicts the dynamics of shares and membership fees from three representative pools, having large, medium and small mining capacities. Figure 3 illustrates the dynamics of mining difficulty and the histogram of inter-block times. Figure 4 shows the dynamics of the total hashing rate of the whole system.

Mining pools. Given that pool managers apply the simple policy as in (3) for membership fee adjustment, it is not surprising to see that the membership fees and the number of shares are coping with each other (Figure 2a). Furthermore, in the real Bitcoin mining game, the mining power implementation is unobservable among players. Even in the same pool, the hashing rate is not transparent to the pool manager or individual miners. The number of valid shares submitted can be an effective estimator to measure the true hashing rate within the pool. So we conclude that the mining power dynamics in a pool could be inferred by the membership fee change. Because mining power is not equally distributed over the pools (Table 2), pools 1 and 2 own larger proportions of mining power, who yield the majority production of shares. Nevertheless, as time evolves, we do not observe any monopolistic structure. Contributing factors of this result are the finiteness of the pool capacity and the existence of passive miners.



(a) The dynamics of shares generated in each campaign and membership fees (a single replication).

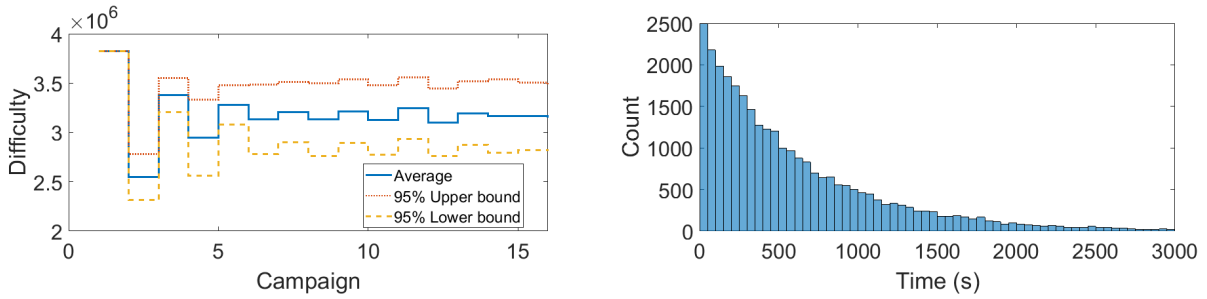


(b) The dynamics of membership fees (average and 95% C.I. of 40 replications).

Figure 2: Simulation of 3 types of pools: large, medium and small.

From Figure 2b, we can observe three different membership fee dynamics: (i) a steady curve in large size pools (e.g., pool 1), (ii) an increasing trend in medium size pools (e.g., pool 2), and (iii) a decreasing trend in small size pools (e.g., pool 3). As mentioned before, the increase and decrease of the mining power occupied by each pool can be related to the corresponding changes of the membership fee. We give some explanations: the expansion of the large size pool is limited by the capacity threshold as well as the under-reaction of miners (modeled by prospect theory) during the pool selection process; on the other hand, the medium size pool has a higher potential to attract individual miners. This observation may be able to provide guidelines for some consortium interested in setting up a new mining pool: sufficient initial mining power is essential to attract individual miners; however, the pool size cannot be too large, which prevents the system's mining power from being over-centralized and the blockchain network from collapsing due to miners' loss of trust.

System metrics. Figure 3a shows the convergence of the Bitcoin mining difficulty level as campaigns evolve. It validates the effectiveness of the adaptive difficulty mechanism designed by Satoshi. The block mining rate depends on the difficulty level and mining power committed by miners, which are updated every $2,016$ blocks and fluctuated according to individual mining decisions in real time, respectively. As a result, new valid blocks occur according to a *nonhomogeneous* Poisson process (Bowden et al. 2018). Nevertheless, the histogram reported in Figure 3b is similar to an exponential distribution. Hence, the new block arrival rate function $\mu(t)$ is steady thanks to the bi-weekly difficulty adjustment mechanism. On the other hand, we recognize that the estimated mean of inter-block time is 616.61 with standard deviation 643.49, which is slightly greater than 600, the idealistic value designed by Satoshi. This may be attributed to the delay in updating the difficulty, see Kraft (2016), Meshkov et al. (2017), Garay et al. (2017).



(a) The dynamics of Bitcoin mining difficulty level (average and 95% C.I. of 40 replications).

(b) The histogram of inter-block time (a single replication).

Figure 3: Bitcoin mining difficulty level and overall hash rate.

From Figure 4, we observe an interesting relationship between the mining power allocation rule and the difficulty level dynamics: (i) if the current difficulty decreases significantly from the previous one, the overall hashing rate exhibits an upward spike when approaching the end of the current campaign; (ii) conversely, there is a downward spike towards the end of the current campaign. The reason for the first scenario is that the current difficulty level is too high for miners with respect to their mining capacities and budgets. Even though the mining activity is profitable, some miners have already exhausted their budgets. On the other hand, the current difficulty level in the second scenario is relatively low so that miners will be more risk-seeking (i.e., increasing β_i's) to execute mining decisions, in order to spend all residual budgets by the end of the campaign. This finding may help miners to predict the future allocation of the overall mining power by taking advantage of the dynamics of the mining difficulty, and to select the optimal timing to begin mining. For example, if a miner has a lower budget, she could mine more actively at the beginning of a new campaign in Scenario (i); she should not turn on her machine idle until the end of a new campaign in Scenario (ii).

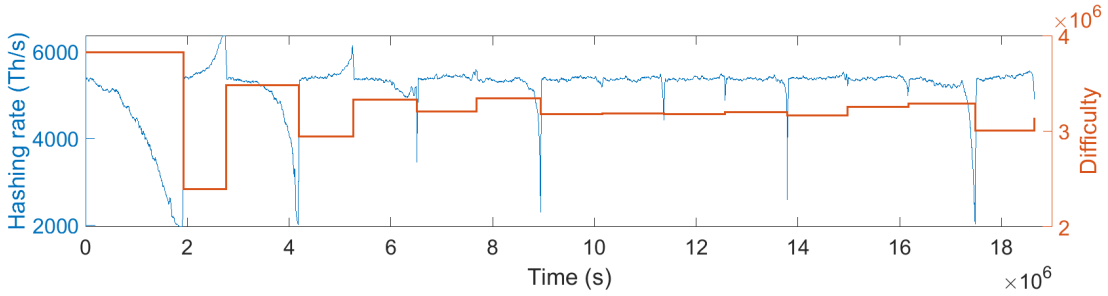


Figure 4: The dynamics of overall hashing rate (100-point moving average, one of replications).

4 CONCLUSION AND DISCUSSION

Summary. We develop a discrete-event Monte-Carlo simulation model to study the behavior of individual miners and mining pool managers, with the objective of testing different mining and pool managing policies. Compared with previous works, our model is more flexible and practical, because it includes realistic features including hashing rate, mining cost, monetary budget, Bitcoin market price, mining pool reward policies, and membership fees. Our simulation results may provide useful insights for individual miners and pool managers in the realistic mining game. We first validate the effectiveness of the current adaptive difficulty recalculation algorithm to approach the ideal block generation rate designed by Satoshi. Another interesting finding here is that medium pools may have a greater growth potential than small and large ones. Furthermore, the relationship between the overall hashing power implementation and difficulty level may help individual miners to make more profitable mining decisions by choosing the appropriate timing to mine.

Future works. We next plan to extend our simulation model to include various attacks and coping strategies in the Bitcoin network, such as double-spending attack, selfish-mining attack, block withholding attack and fork after withholding attack. Meanwhile, we may also consider Black Swan events, e.g., the skyrocket or plummet of Bitcoin price and the outbreak of epidemic diseases.

REFERENCES

Alharby, M., and A. van Moorsel. 2019. "Blocksim: a simulation framework for blockchain systems". *ACM SIGMETRICS Performance Evaluation Review* 46(3):135–138.

Aoki, Y., K. Otsuki, T. Kaneko, R. Banno, and K. Shudo. 2019. "SimBlock: A blockchain network simulator". In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 325–329. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Blockchain.com contributors 2020. "BTC to USD: Bitcoin to US Dollar Market Price - Blockchain". https://www.blockchain.com/charts/market-price, accessed 30th May.

Bowden, R., H. P. Keeler, A. E. Krzesinski, and P. G. Taylor. 2018. "Block arrivals in the Bitcoin blockchain". *arXiv preprint arXiv:1801.07447*.

Bracha, G., and S. Toueg. 1985. "Asynchronous consensus and broadcast protocols". *Journal of the ACM (JACM)* 32(4):824–840.

Cocco, L., and M. Marchesi. 2016. "Modeling and Simulation of the Economics of Mining in the Bitcoin Market". *PloS one* 11(10).

Coin Dance contributors 2020. "Coin Dance — Bitcoin Nodes Summary". https://coin.dance/nodes, accessed0 30th May.

Cong, L. W., Z. He, and J. Li. 2019. "Decentralized mining in centralized pools". Technical report, National Bureau of Economic Research.

Correia, M., G. S. Veronese, N. F. Neves, and P. Verissimo. 2011. "Byzantine consensus in asynchronous message-passing systems: a survey.". *IJCCBS* 2(2):141–161.

Garay, J., A. Kiayias, and N. Leonardos. 2017. "The bitcoin backbone protocol with chains of variable difficulty". In *Annual International Cryptology Conference*, 291–323. Springer.

Kaligotla, C., and C. M. Macal. 2018. "A generalized agent based framework for modeling a blockchain system". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, S. J. A. Skoogh, and B. Johansson, 1001–1012. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kraft, D. 2016. "Difficulty control for blockchain-based consensus systems". *Peer-to-Peer Networking and Applications* 9(2):397–413.

Meshkov, D., A. Chepurnoy, and M. Jansen. 2017. "Short paper: revisiting difficulty control for blockchain systems". In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, 429–436. Springer.

Nakamoto, Satoshi and others 2008. "Bitcoin: A peer-to-peer electronic cash system.(2008)".

Narayanan, A., J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. 2016. *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton University Press.

Rauchs, Michel 2020. "Cambridge Bitcoin Electricity Consumption Index (CBECI)". https://www.cbeci.org/, accessed 30th May.

Rosa, E., G. D'Angelo, and S. Ferretti. 2019. "Agent-based Simulation of Blockchains". In *Asian Simulation Conference*, 115–126. Springer.

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

Salimitari, M., M. Chatterjee, M. Yuksel, and E. Pasiliao. 2017. "Profit maximization for bitcoin pool mining: A prospect theoretic approach". In *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, 267–274. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wang, C., X. Chu, and Q. Yang. 2019. "Measurement and analysis of the bitcoin networks: A view from mining pools". *arXiv preprint arXiv:1902.07549*.

Yeow, Addy 2020. "Global Bitcoin Nodes Distribution - Bitnodes". https://bitnodes.io/, accessed 30th May.

AUTHOR BIOGRAPHIES

KEJUN LI is a Ph.D. student in the Department of Industrial and Systems Engineering at North Carolina State University. His research interests include applied probability, simulation, game theory, and blockchain technologies. His e-mail address is kli15@ncsu.edu.

YUNAN LIU is an associate professor in the Department of Industrial and Systems Engineering at North Carolina State University. He earned his Ph.D. in Operations Research from Columbia University. His research interests include queueing theory, stochastic modeling, simulation, applied probability, online learning, and optimal control, with applications to call centers, healthcare, transportation, and blockchain systems. His email address is yliu48@ncsu.edu. His website is https://yunanliu.wordpress.ncsu.edu/

HONG WAN is an associate professor in the Department of Industrial and Systems Engineering at North Carolina State University. She received her Ph.D. in industrial engineering and management sciences from Northwestern University. Her research focuses on the areas of simulation data analysis, complex system simulation, and blockchain modeling, mechanism design, and application. She is the director of the ISE blockchain lab, and serves as the editor in chief of Journal of Blockchain Research and the associate editor of ACM TOMACS. She is a member of INFORMS, IISE, POMS and IEEE blockchain society. Her email address is hwan4@ncsu.edu and her website is https://www.ise.ncsu.edu/people/hwan4/.

LING ZHANG is a last year Ph.D. student in the Department of Industrial and Systems Engineering at North Carolina State University. His research interests include queueing theory, stochastic modeling, optimal control theory and simulation, with applications in transportation, healthcare, and distributed consensus systems. His e-mail address is lzhang42@ncsu.edu.