A New Theoretical Framework of Pyramid Markov Processes for Blockchain Selfish Mining

Quanlin Li, Yanxia Chang, Xiaole Wu, Guoqing Zhang

School of Economics and Management, Beijing University of Technology, Beijing 100124, China
liquanlin@tsinghua.edu.cn

School of Management, Fudan University, Shanghai 200433, China
wuxiaole@fudan.edu.cn (✉)

Department of Mechanical, Automotive & Materials Engineering, University of Windsor, Windsor, Ontario, Canada
gzhang@uwindsor.ca

Abstract. In this paper, we provide a new theoretical framework of pyramid Markov processes to solve some open and fundamental problems of blockchain selfish mining under a rigorous mathematical setting. We first describe a more general model of blockchain selfish mining with both a two-block leading competitive criterion and a new economic incentive mechanism. Then we establish a pyramid Markov process and show that it is irreducible and positive recurrent, and its stationary probability vector is matrix-geometric with an explicitly representable rate matrix. Also, we use the stationary probability vector to study the influence of orphan blocks on the waste of computing resource. Next, we set up a pyramid Markov reward process to investigate the long-run average mining profits of the honest and dishonest mining pools, respectively. As a by-product, we build one-dimensional Markov reward processes and provide some new interesting interpretation on the Markov chain and the revenue analysis reported in the seminal work by Eyal and Sirer (2014). Note that the pyramid Markov (reward) processes can open up a new avenue in the study of blockchain selfish mining. Thus we hope that the methodology and results developed in this paper shed light on the blockchain selfish mining such that a series of promising research can be developed potentially.

Keywords: Blockchain, Proof of Work, selfish mining, main chain, pyramid Markov process, pyramid Markov reward process, phase-type distribution, Matrix-geometric solution

1. Introduction

Bitcoin has received tremendous attention as the first fully decentralized distributed cryptocurrency since its advent by Satoshi Nakamoto in 2008. Blockchain is used to securely record a public shared ledger of Bitcoin payment transactions among Internet users in an open P2P network. In the past decade, we have witnessed the explosive growth of blockchain, as exemplified by Bitcoin (2008), Ethereum and smart contract (2013), Hyperledger Fabric (2015), Libra (Facebook 2019), among others. On the one hand, the great success of blockchain is based on solving the cryptographic puzzle by means of the brute force, namely Proof of Work (PoW). Note that the PoW is a most frequently used consensus protocol (or algorithm). Readers may refer to Garay et al. (2015), Bastiaan (2015), Tromp (2015), Garay et al. (2017 2020), Kiayias et al. (2016 2020), Kiayias and Zindros (2019), Karakostas and Kiayias (2021), and the references therein. On the other hand, an economic incentive mechanism in Bitcoin is designed for a lot of miners according to the shared proportion of their computing powers in solving the cryptographic puzzle. Based on this, Nakamoto (2008) showed a key blockchain characteristic: The fairness of PoW, i.e., as long as no more than 50% of the total computing power follows the PoW, the probability that a honest miner can earn the total revenue (i.e., the block rewards and the transaction fees) is proportional to his computing power.
In 2014, a seminal work was reported by Eyal and Sirer (2018) (nearly at the same time, Bahack (2013)), in which Eyal and Sirer first introduced an important concept: selfish mining. Then they set up a Markov chain to analyze the competitive behavior of the selfish mining. Furthermore, they provide a revenue analysis of the blockchain system with selfish mining. Based on this, they reported an interesting finding: The fairness of Bitcoin PoW can be destroyed by the selfish mining attacks, i.e., the dishonest (or selfish) mining pool secretly withholds blocks not to broadcast in the open P2P network if the number of blocks mined by the selfish mining pool remains ahead of the honest mining pool. In this case, the dishonest mining pool continues to mine on the top of the selfish-mining block branch such that the dishonest mining pool can earn more than his fair amount of mining profit. Since then, the Markov chain method of Eyal and Sirer (2018) has been widely applied in the literature on the selfish mining, for example, stubborn mining by Nayak et al. (2016), Wang et al. (2019) and Liu et al. (2020); Ethereum (and smart contract) by Feng and Niu (2019); multiple mining pools (or miners) by Liu et al. (2018) and Jain (2019); multi-stage blockchain by Chang et al. (2019); among others. However, so far no one has wondered or explained whether the Markov chain method of Eyal and Sirer (2018) makes sense. This motivates us in this paper to set up a pyramid Markov process to discuss the Markov chain method of Eyal and Sirer (2018).

To further understand the selfish mining, this paper introduces a new economic incentive mechanism by means of two practical factors: The efficiency-increased ratio, and the mining rate of jumping miners from the honest mining pool to the dishonest mining pool (the jumping’s mining rate, in short). Note that the efficiency-increased ratio is used to measure the improved degree of mining efficiency of the dishonest mining pool, while the jumping’s mining rate denotes the social reputation and influence of the dishonest mining pool. Furthermore, to describe the block-pegging process clearly, we propose a two-block leading competitive criterion. We design the third practical factor: A block-detained probability sequence to express the block-pegging decision-making of the dishonest mining pool. To our best knowledge, this paper is the first to add the above three practical factors of competitive advantage into the selfish mining scenario. In addition, we find a key phenomenon from the interaction effect between the block-pegging and mining processes: No block is mined during the block-pegging process, that is, the block-pegging and mining processes must be separated from two connected time intervals. This is because each block is generated by means of finding a nonce through solving a cryptographic puzzle of using all the foregoing information of that blockchain in front of this block. On the contrary, if the block-pegging and mining processes are not separated, then the cryptographic puzzle can be possibly chaotic for the cryptographic puzzle design in a block by block mining environment.

For our more general model of blockchain selfish mining, we set up a pyramid Markov process to express the mining competition between the dishonest and honest mining pools, as seen in the two block branches forked at the common tree root. Note that a key factor of forming the pyramid Markov process is our important finding that no block is mined during the block-pegging process. The pyramid Markov process seems fairly complicated. Fortunately, we can easily show that the pyramid Markov process is irreducible and positive recurrent, and its stationary probability is matrix-geometric with an explicitly representable rate matrix. Note that our matrix-geometric solution is simpler than that in Neuts (1981) for Markov chains of GI/M/1
type, thus we can provide a unified theoretical framework of pyramid Markov processes, which is applicable to a wide range of blockchain selfish mining. In addition, from the pyramid Markov process, we build a new one-dimensional Markov model to further deal with the mining processes of the honest and dishonest mining pools. Also, under no network latency, we provide some new interesting interpretation on the Markov chain (see Figure 1) and the revenue analysis (see (1) to (3)) given in Eyal and Sirer (2018). Specifically, we show that the Markov chain method of Eyal and Sirer (2018) is a rough approximation, and there is no valid mathematical theory behind it.

It is necessary and useful to design the block-leading number (i.e., the difference of two block numbers in the two block branches forked at the common tree root). Eyal and Sirer (2018) used a two-block leading competitive criterion; while Göbel et al. (2016) studied a one-block leading competitive criterion. In this paper, we focus on the two-block leading competitive criterion and set up a pyramid Markov process. From the pyramid Markov process, it is easy to see that our technique can easily extend and generalize to a $K$-block leading competitive criterion. This motivates us to find an optimal number $K^*$ from a revenue management perspective such that the $K^*$-block leading competitive criterion can be adaptively used in the selfish mining scenario.

From the two-dimensional Markov processes, our paper is closely related to Göbel et al. (2016). However, our pyramid Markov process is distinguished from that in Göbel et al. (2016) in the following aspects. First, we study the two-block leading competitive criterion, while Göbel et al. (2016) discussed the one-block leading competitive criterion. This leads to the different starting states of pegging blocks, as seen in the pyramid Markov process depicted in Figure 4. Second, we assume that the block-pegging and mining processes must be separated from two connected time intervals, such that the two-dimensional Markov process is of pyramid type; while in Göbel et al. (2016), these mining processes can still continue during the block-pegging period. This leads to that the two-dimensional Markov process of Göbel et al. (2016) is more complicated and difficult than our pyramid Markov process. Thus, it is a key finding that the block-pegging and mining processes can not synchronously go ahead. Third, our pyramid Markov process is matrix-geometric; while the stationary probability vector of Göbel et al. (2016) is more general and complicated, although Göbel et al. (2016) and Javier and Fralix (2020) provided two effective methods to deal with the stationary probability vector. Fourth, we provide three practical and useful factors (e.g., the efficiency-increased ratio, the jumping’s mining rate, and the block-detained probability sequence) to support the competitive advantages of the dishonest mining pool; while there is no competitive element in Göbel et al. (2016) such that, when the network latency disappears, all the selfish miners become honest. Therefore, their simple assumptions would not be reasonable and enough to support the study of blockchain selfish mining.

More realistically, it is interesting to consider the influence of various orphan blocks on the waste of computing resource. So far, there have been some qualitative analysis for the influence of orphan blocks, but the quantitative research is limited. For the qualitative analysis, readers may refer to Carlsten et al. (2016), Velner et al. (2017), Stifter et al. (2019), Saad et al. (2019), Awe et al. (2020) and others. To our best knowledge, our pyramid Markov process is the first effective quantitative technique which is used to analyze the orphan blocks in the blockchain. To this end, we use the stationary probability vector of the pyramid Markov process to provide a detailed analysis for the
performance measures of the blockchain selfish mining model with a lot of orphan blocks.

From an economic analysis perspective, we set up a pyramid Markov reward process to study the revenue analysis of our more general model of blockchain selfish mining. Based on this, we express the long-run average mining profits of the honest and dishonest mining pools, respectively. Also, we find that the two long-run average mining profits are multivariate linear in some key parameters of the blockchain system. This enables us to find a simple sufficient condition under which the blockchain can operate normally. Moreover, we can measure the mining efficiency of the dishonest mining pool through comparing with the honest mining pool. Therefore, we develop a unified theoretical framework of pyramid Markov reward processes to open a new avenue, which can support many potential promising research of blockchain selfish mining.

In summary, the main contributions of this paper are listed as follows:

1. We describe a more general model of blockchain selfish mining with a two-block leading competitive criterion and a new economic incentive mechanism that captures three key factors: The efficiency-increased ratio, the jumping's mining rate, and a block-detained probability sequence. (Section 3)
2. We set up a pyramid Markov process to express the dynamics of the selfish mining, and show that the pyramid Markov process is irreducible and positive recurrent, and the stationary probability vector is matrix-geometric with an explicitly representable rate matrix. (Section 4). Also, we use the stationary probability vector to provide an effective method to analyze the influence of various orphan blocks on the waste of computing resource. (Section 5)
3. We establish a pyramid Markov reward process to investigate the long-run average mining profits of the honest and dishonest mining pools, respectively. Also, we show that the long-run average mining profits are multivariate linear in some key parameters. This leads to a sufficient condition under which the blockchain can operate normally. Moreover, we can measure the mining efficiency of the dishonest mining pool. (Section 6)
4. We study the transient mining profits in the time interval $[0, t)$ by means of the PH distribution of infinite sizes and the associated PH renewal process. To our best knowledge, this paper is the first to establish the transient mining profits in the study of blockchain selfish mining. (Section 7)
5. We build a new one-dimensional Markov model to further deal with the mining processes of the honest and dishonest mining pools. (Section 8). Further, we discuss a special case without network latency. Fortunately, by using the one-dimensional Markov model, we can provide some new interesting interpretation on the Markov chain and the revenue analysis of Eyal and Sirer (2018). We show that the Markov chain method of Eyal and Sirer (2018) is a rough approximation, and there is no valid mathematical theory behind it. (Section 9)

In addition, we provide the relevant literature in Section 2. In Section 10, we use some numerical examples to verify the correctness and computability of our theoretical results. In Section 11, we provide some useful concluding remarks. Finally, an appendix provides three proofs for two theorems and one lemma.

2. Literature Review

In this section, we summarize the literature on the selfish mining of blockchain. At the
same time, we relate our paper to three lines of research by means of a simple classification of these literature.

We first explain the reason why the selfish mining pools can be formed and developed rapidly. When a block is managed by a miner who can provide the nonce, the miner receives two parts of profits: The block reward and the transaction fee. On the one hand, under such a profit incentive, each miner is willing to compete in producing and broadcasting a block in an open P2P network, and hope to peg the block on the blockchain such that he can receive the profit (including the block reward and the transaction fee) as much as possible. If the electricity price is low and the mining profit is high, then more and more people would like to join the mining process. On the other hand, since the total number of generating blocks is controlled at an average rate of one every ten minutes, the probability of individual miner generating and pegging a block becomes lower and lower, as the number of miners increases. This greatly increases the mining risk of each individual miner. In this situation, some miners willingly form a selfish mining pool. So far, developing such selfish mining pools has become increasingly widespread, e.g., see Lee and Kim (2019). Figure 1 introduces and compares the best and biggest Bitcoin mining pools in the world in 2020.

Now, we discuss that our paper is related to three lines of research on the blockchain selfish mining, in which some available methodologies and results are also listed in details.

The first research stream started with discussions on the selfish mining attacks, which can destroy the fairness of Bitcoin PoW. For the blockchain with two mining pools, Eyal and Sirer (2018) set up a Markov chain to express the dynamic of the selfish mining attacks, and also designed a fairly simple revenue analysis for observing how the selfish mining attacks can influence on the profit allocation between the honest and dishonest mining pools. After then, following the Markov chain method of Eyal and Sirer (2018), some researchers extended and generalized such a similar discussion for some attack strategies of blockchain, among which important examples include stubborn mining by Nayak et al. (2016), Wang et al. (2019) and Liu et al. (2020);
Ethereum (and smart contract) by Feng and Niu (2019); multiple mining pools (or miners) by Leelavimolsilp et al. (2018 2019), Liu et al. (2018) and Jain (2019); multi-stage blockchain by Chang et al. (2019); no block reward by Carlsten et al. (2016); power adjusting by Gao et al. (2019); extending Eyal and Sirer’s Markov chain by Gervais et al. (2015), Mwale (2016), Moustapha (2018), Mulser (2018), Bai et al. (2019), Marmolejo-Cossío et al. (2019), Dong et al. (2019), Lee and Kim (2018) and Liu et al. (2019). Different from the previous literature, this paper provides another interesting research perspective: Does the Markov chain method of Eyal and Sirer (2018) make sense? For this purpose, we develop a new theoretical framework of pyramid Markov processes, and show that the Markov chain method of Eyal and Sirer (2018) is a rough approximation, and there is no valid mathematical theory behind it. Göbel et al. (2016) set up a two-dimensional Markov chain to study the selfish mining attacks. Javier and Fralix (2020) further provided a new computational method for analyzing the stationary probability vector of the two-dimensional Markov chain given in Göbel et al. (2016). Our paper is closely related to Göbel et al. (2016) from the two-dimensional structure of Markov chains. Distinctively, we find that the block-generating and block-pegging processes must be separated from two connected time intervals, thus we can establish a pyramid Markov process which is simpler than that of Göbel et al. (2016), and show that our stationary probability vector is matrix-geometric with an explicitly representable rate matrix.

On the other hand, to dynamically optimize the selfish mining strategies, the Markov decision processes have been applied successfully. Important examples include Sapirshtein et al. (2016), Sompolinsky and Zohar (2015), Gervais et al. (2016), Wüst (2016) and Gupta (2020).

Our paper is related to the second research stream of blockchain selfish mining, while their mathematical analysis has still been relatively weak or more difficult up to now. This greatly motivates us to open up potentially interesting research on the different directions, including applications of our pyramid Markov processes. In what follows, we classify the literature into seven different research directions:

(a) Generalizations of selfish mining strategies. Important examples include the subversive miner strategy by Courtois and Bahack (2014); and the stubborn mining strategy by Nayak et al. (2016). In addition, Lee and Kim (2019) developed a detective mining by means of the information of miners.

(b) The puzzle difficulty. Note that the Bitcoin miners competitively solve a PoW puzzle to find the nonce. Once such a nonce is found, a Bitcoin block can immediately be generated and pegged on the blockchain. See Nakamoto (2008), Narayanan et al. (2016), Miller et al. (2015), Göbel et al. (2016), Kraft (2016) and Davidson and Diamond (2020). The difficulty level used to discover a new block can be constantly adjusted such that, on average, one block is expected to be discovered every 10 minutes. Readers may refer to, such as block arrivals by Göbel et al. (2016); zeroblock by Solat and Potop-Butucaru (2016ab); smartpool by Luu et al. (2017); difficulty control by Fullmer and Morse (2018); pooled mining by Lee and Kim (2018); unfairness of blockchain by Guerraoui and Wang (2018); multi-stage blockchain by Chang et al. (2019); bobtail by Bissias and Levine (2020); and so on. Note that, when the difficulty level is changed continuously, the

The probability theory of blockchain selfish mining is further developed. On the one hand, Grunspan and Pérez-Marco (2018abc 2019) applied martingale to analyze the profitability of selfish mining, stubborn mining, trailing mining, and Ethereum. Albrecher and Goffard (2020) studied the profitability of blockchain selfish mining by means of the theory of ruin.
Markov process of blockchain selfish mining is nonhomogeneous, thus transient analysis of such a Markov process is required but more difficult and challenging.

(c) The forked structure. When there exist more than one mining pools in blockchain (see Leelavimolsilp et al. (2019), Lee and Kim (2018), Liu et al. (2018) and Jain (2019)), it is possible that two or more blocks with the same preceding block (or parent block) can be produced and form block branches forked at the preceding block. This leads to a forked structure starting from the parent block, as the length (or height) of blockchain increases. Readers may refer to Section 4 of Eyal and Sirer (2018) for some explanatory examples. To enhance the scalability and transaction throughput of blockchain, Sompolinsky and Zohar (2015) and Lewenberg et al. (2015) generalized the simple forked tree (i.e., several block branches forked at a common tree root) to a more general tree with multiple branching points (called GHOST), and further to a Direct Acyclic Graph (abbreviated DAG), e.g., see Kiayias and Panagiotakos (2017), Lee (2018) and Choi et al. (2018). Typically, it is interesting and challenging to study the blockchain selfish mining with multiple mining pools whose forked structure is either tree, GHOST or DAG. In this case, our pyramid Markov processes are related to the fluid and diffusion approximations in order to be able to deal with the multidimensional stochastic systems of blockchain selfish mining.

(d) The orphan blocks and uncle blocks. In the blockchain selfish mining, the orphan blocks and uncle blocks can lead to the waste of computing resource by means of the longest chain or the heaviest path. Readers may refer to the orphan blocks by Carlsten et al. (2016), Velner et al. (2017), Stifter et al. (2019), Saad et al. (2019) and Awe et al. (2020); and the uncle blocks (in Ethereum) by Gervais et al. (2016), Ritz and Zugenmaier (2018), Feng and Niu (2019), Werner et al. (2019), Wang et al. (2019), Chang et al. (2019) and Liu et al. (2020). It is worthwhile to note that our pyramid Markov processes are successful and applicable in the study of orphan blocks and uncle blocks.

(e) Detection of selfish mining. In a blockchain, it is possible to have some different classes of attacks, such as 51% attack, selfish mining attack, stalker attack, eclipse attack, physical attack and so forth. In this case, an interesting topic is how to detect the different attacks effectively, and to be able to put forward some defensive measures for the attacks. Important examples include Liu et al. (2019), Saad et al. (2019), Lee and Kim (2019) and Chicarino et al. (2020).

(f) Defense against selfish mining. The selfish mining attack is a fundamental challenge in the study of blockchain, since it breaks the fairness of blockchain (see Guerraoui and Wang (2018)) and poses potential threats to the decentralized structure. Thus it is very important to defense against and prevent the selfish mining. Readers may refer to, such as Zhang (2015), Solat and Potop-Butucaru (2016ab) and Zhang and Preneel (2017).

(g) The double spending attacks. The double spending attacks have a stronger relation with the selfish mining. Note that the block-pegging time can increase due to the network delay (see Decker and Wattenhofer (2013)), thus the released blocks will not be confirmed immediately. In this situation, the selfish mining attacks can make the double spending attacks, e.g., see Karame et al. (2012), Rosenfeld (2014) and Javarone and Wright (2018) and the references therein.

Other research is also mentioned here, for example, the data and experiment analysis by Wright (2017), Wright and Savanah (2017), Eijkel and Fehnker (2019) and Kedzia et al. (2020).

Finally, our work is related to the third research stream in the study of consensus pro-
tocols. So far, the consensus protocols (or algorithms) have become the core of blockchain development. Normally, the consensus protocol is used to determine the generation, storage and validation of data, it sets up operations mode of blockchain and determines performance and security of blockchain. Now, the frequently used consensus protocols include two different types: A consensus protocol based on Byzantine fault tolerance algorithm (BFT), and another consensus protocol based on PoW / Proof of Stake (PoS). The BFT consensus protocol is usually used in smaller private chains or alliance chains; while the PoW/PoS consensus protocol is suitable for large-scale public chains. Furthermore, following BFT, PoW and PoS, so far over fifty consensus protocols have been developed with advantage of special needs. Readers may refer to survey papers by, for example, Bissias et al. (2016), Cachin and Vukolic (2017), Nguyen and Kim (2018), Wang et al. (2019), Xia et al. (2020), Bano et al. (2019), Natoli et al. (2019), Huang et al. (2021) and Fan et al. (2020). Recently, it is interesting and challenging to develop the theory of Markov processes (more generally, probability theory), including our pyramid Markov processes, in the study of consensus protocols.

3. Model Description

In this section, we describe a more general model of blockchain selfish mining with two different mining pools: The dishonest mining pool, and the honest mining pool (i.e., a virtual pool with all the honest miners). Also, the blockchain selfish mining is controlled by not only a two-block leading competitive criterion but also an economic incentive mechanism that captures three types of key parameters: The efficiency-increased ratio $\mathbb{E}$, the jumping’s mining rate $\gamma$, and the block-detained probability sequence $\{p_k : k = 2, 3, 4, \cdots\}$. In addition, we introduce some mathematical notation used in our later study.

To avoid the 51% attack, we assume that the computing power of the honest mining pool is more than half of the total computing power of the blockchain system, that is, the honest miners are in the majority while the dishonest miners are in the minority. Note that the honest mining pool follows the two-block leading competitive criterion, while the dishonest mining pool follows the selfish mining attack strategy, which is a modification of the two-block leading competitive criterion through using the block-detained probability sequence $\{p_k : k = 2, 3, 4, \cdots\}$. See Figure 2 for an intuitive understanding.

In the blockchain selfish mining, the honest and dishonest mining pools compete fiercely in finding the nonce (i.e., the nonce is given by means of solving the cryptographic puzzle) to generate the blocks, and they publish the blocks to make two block branches forked at a common tree root (parent block). For the two honest and dishonest mining pools under the two-block leading competitive criterion, by
observing the two block branches forked at the common tree root, we define the main chain as the longer block branch; while another shorter block branch is called the chain of orphan blocks.

In the blockchain system, every external transaction first needs to be checked by paying a certain handling expense (called the transaction fee), and it is sent to the transaction pool. Then some transactions are randomly taken from the transaction pool to generate a block, e.g., see Li et al. (2018 2019) for an intuitive interpretation. On the other hand, the transactions of an orphan block are returned to the transaction pool but no additional transaction fee is required again. For convenience of analysis, we assume that the transaction pool always has a large enough capacity.

In what follows, we provide some model descriptions for the blockchain selfish mining as follows.

1) The block-generating processes: We assume that the blocks mined by the dishonest and honest mining pools have formed two block branches forked at a common tree root, and the growths of the two block branches are two Poisson processes with block-generating rates $\alpha > 0$ and $\beta > 0$, where $\alpha = \tilde{\alpha}(1 + \Re)$, $\Re \geq 0$ is the efficiency-increased ratio of the dishonest mining pool; and $\tilde{\alpha} > 0$ is regarded as a net mining rate when all the miners of the dishonest mining pool become honest, $\beta$ is a net mining rate of the honest mining pool.

2) The jumping’s mining rate: The dishonest mining pool has more mining advantages than the honest mining pool, so it can attract some honest miners to jump into the dishonest mining pool. Let $\gamma$ be the net mining rate of such jumping miners. Then after jumping, the block-generating rates of the dishonest and honest mining pools are given by $(\tilde{\alpha} + \gamma)(1 + \Re)$ (i.e., $\alpha + \gamma(1 + \Re)$) and $\beta - \gamma$, respectively.

To overcome the 51% attacks, it is necessary to limit the jumping’s mining rate $\gamma$. Let $\tilde{\alpha} + \gamma < (1/2)(\tilde{\alpha} + \beta)$ and $\beta - \gamma > (1/2)(\tilde{\alpha} + \beta)$. Then

$$0 \leq \gamma < \frac{1}{2}(\beta - \tilde{\alpha}) \quad (1)$$

From a practical economic perspective, the two key parameters $\Re$ and $\gamma$ are well related to the dishonest mining pool’s operations management level, social reputation and influence, the reward of each excavated block, and so on. Based on this, this paper is clearly different from those previous works in the literature. Figure 3 depicts some relations between the honest and dishonest mining pools.

3) The block-pegging time of the main chain: The network latency always results
in some delay for each block pegged on the blockchain. Thus the block-pegging time is used to express the network latency, and it is a time interval that begins at the broadcast time of a block until the block is pegged on the blockchain. Currently, the block-pegging time contains two different parts: The block broadcast time, and the verification time by various miners. Also, the block broadcast time is always longer than the verification time, because the verification time is very short.

Once a main chain is formed, then the mining process is terminated immediately, and the whole main chain is pegged on the blockchain. We assume that the block-pegging time is independent and identically distributed (i.i.d) and exponential with mean $1/\mu$.

Note that no new block can be generated during the block-pegging process of the main chain. Once the main chain is pegged on the blockchain, then a new mining competition immediately starts on two new block branches forked at another common tree root.

4) The return time of the orphan block: Since an orphan block cannot be pegged on the blockchain, it has to return to the transaction pool. We assume that the return time of each orphan block is exponential with mean $1/\mu$.

5) A two-block leading competitive criterion:

(a) The main chain by the honest mining pool. Once the honest chain of blocks is two blocks ahead of the dishonest chain of blocks, then the honest chain of blocks is the main chain. In this case, the forked process of two block branches ends immediately, and the whole main chain is pegged on the blockchain. At the same time, all the blocks of the dishonest chain immediately become orphan blocks, and all of them need to be returned to the transaction pool without re-paying any new fee.

(b) The main chain by the dishonest mining pool. Once the dishonest chain of blocks is $k$ blocks ahead of the honest chain of blocks for $k \geq 2$, then the dishonest chain of blocks is taken as the main chain. Based on this, two different cases are considered as follows:

\( b-1 \) With probability $p_k$, the whole main chain is pegged on the blockchain, and the forked process of two block branches ends immediately. Also, all the blocks of the honest chain immediately become orphan blocks, each of which is returned to the transaction pool.

\( b-2 \) With probability $1 - p_k$, the dishonest mining pool does not broadcast and peg the main chain such that it continues to mine more blocks to lengthen the main chain. In addition, to be able to peg the main chain on the blockchain finally, we assume that $\lim_{n \to \infty} p_n = 1$.

6) The reward of a block generated and pegged on the blockchain: Once a block is generated and pegged on the blockchain, then the mining pool of generating this block can obtain an appropriate amount of reward (or compensation) from two different parts:

(a) A block reward $r_B$ by the blockchain system. When a block is generated and pegged on the blockchain, the mining pool receives a certain amount of block reward $r_B$.

(b) An average total transaction fee $r_F$ in the block. It is possible that the transactions and their number contained in the blocks are different from each other. For the sake of simplicity, we assume the average total transaction fee for a block is $r_F$.

Obviously, $r_B + r_F$ is the total reward received by a mining pool who generates and pegs the block on the blockchain.

Note that the two mining pools do not receive any block reward and transaction fee from an orphan block.

7) The mining cost: The mining cost of the blockchain system contains two parts:

(a) The electric charge. Let $c_E$ be the electric price per unit of net mining rate and per unit time. Then the electric costs per unit
time by the dishonest and honest mining pools are given by $c_E (\bar{\alpha} + \gamma)$ and $c_E (\beta - \gamma)$, respectively.

(b) The administrative fee. Let $c_A$ be the administrative price per unit of net mining rate and per unit time. Then the administrative costs per unit time by the dishonest and honest mining pools are given by $c_A (\bar{\alpha} + \gamma) (1 + \mathcal{R})$ and $c_A (\beta - \gamma)$, respectively.

Independence: We assume that all the random variables defined above are independent of each other.

Remark 1. To support the selfish mining behavior effectively, the efficiency-increased ratio $\mathcal{R}$ measures the improvement of management ability of the dishonest mining pool, the block-detained probability sequence $\{p_k, k = 2, 3, 4, \ldots\}$ denotes the unfair competition strategy used by the dishonest mining pool, and the jumping’s mining rate $\gamma$ reflects the social influence of the dishonest mining pool in the world.

Remark 2. If $\lim_{n \to \infty} p_n < 1$, then with probability $1 - \lim_{n \to \infty} p_n$, the dishonest mining pool does not broadcast the mined blocks so that some main chains will become the chain of orphan blocks. This case is clearly impractical from the mining purpose. Thus this paper will not consider the case with $\lim_{n \to \infty} p_n < 1$.

Remark 3. (a) Note that such a block reward is halved after every 210,000 blocks are mined. The block reward was reduced from its initial level of 50 Bitcoins to 25 Bitcoins in November 2012, further reduced to 12.5 Bitcoins in July 2016, and then again in July 2020 to its current level of 6.25 Bitcoins. Finally, the block reward will be reduced 32 more times before eventually reaching zero sometime around 2140. (b) As $r_B$ decreases after every period of four years, designing a sufficient large transaction fee $r_F$ will be the main economic source to maintain the mining operations of a blockchain system.

4. A Pyramid Markov Process

In this section, we set up a new pyramid Markov process to analyze the blockchain selfish mining with a two-block leading competitive criterion and a new economic incentive mechanism. Note that all the key factors or parameters designed in the blockchain selfish mining are well related to the physical and dynamic structure of the pyramid Markov process. In particular, we show that the pyramid Markov process is always irreducible and positive recurrent, and that the stationary probability vector is matrix-geometric with an explicitly representable rate matrix.

In the two block branches forked at a common tree root, we denote by $I(t)$ and $J(t)$ the numbers of blocks mined by the honest and dishonest mining pools at time $t$, respectively. It is clear that $\{ (I(t), J(t)) : t \geq 0 \}$ is a continuous-time Markov process whose state space is given by

$$
\Omega = \Omega_0 \cup \bigcup_{k=0}^{\infty} \Omega_k
$$

where

Level $0 : \Omega_0 = \{(0, 0)\}$
Level $1 : \Omega_1 = \{(0, 1), (1, 0), (0, 2), (2, 0), \ldots\}$
Level $k \geq 2 : \Omega_k = \{ (k, k-2), (k, k-1), (k, k), (k, k+1), (k, k+2), \ldots \}$

Based on this, the state transition relation of the Markov process $\{(I(t), J(t)) : t \geq 0\}$ is depicted in Figure 4. Note that the Markov process $\{(I(t), J(t)) : t \geq 0\}$ is of pyramid type, thus it is also called a pyramid Markov process (note that such a pyramid form can become clearer in the case with multiple mining pools).
Remark 4. State \((0, 0)\) plays a key role in setting up the pyramid Markov process. In fact, State \((0, 0)\) describes the tree root as the starting point of a fork attack, see Figure 4 for more details. If the pyramid Markov process enters State \((0, 0)\), then the fork attack ends immediately, and the main chain is pegged on the blockchain.

By using Figure 4, the infinitesimal generator of the Markov process \(\{(I(t), J(t)) : t \geq 0\}\) is given by

\[
Q = \begin{pmatrix}
Q_{0,0} & Q_{0,1} & Q_{0,2} \\
Q_{1,0} & Q_{1,1} & Q_{1,2} \\
B & B & A \\
\vdots & \vdots & \vdots
d\end{pmatrix}
\]

\[
Q_{0,0} = \begin{pmatrix}
0 & 0 & 0 \\
\mu p_2 & 0 & 0 \\
\mu p_3 & 0 & 0 \\
\mu p_4 & 0 & 0 \\
\vdots & \vdots & \vdots
d\end{pmatrix}
\]

\[
Q_{0,1} = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots
d\end{pmatrix}
\]

where \(a = (\bar{\alpha} + \gamma)(1 + \mathcal{R})\), \(b = \beta - \gamma\), \(\xi_k = a (1 - p_k) + b + \mu p_k\) for \(k \geq 2\),

\[
Q_{0,0} = -(a + b), \quad Q_{0,0} = (a, 0, 0, \cdots) \\
Q_{0,1} = (b, 0, 0, \cdots)
\]
In the above matrices, we only give the non-zero elements, while all the zero elements are empty with an easy understanding from the context of the state space \( \Omega \).

The following theorem describes a key stability characteristic of the pyramid Markov process \( Q \).

**Theorem 1** The pyramid Markov process \( Q \) is always irreducible and positive recurrent.

**Proof.** Please see (a) Proof of Theorem 1 in the appendix.

Note that the pyramid Markov process \( Q \) is always irreducible and positive recurrent. For \( i, j = 0, 1, 2, \cdots \), we define the probabilities

\[
p_{i,j}(t) = P\{I(t) = i, J(t) = j\}
\]

and

\[
\pi_{i,j} = \lim_{t \to +\infty} p_{i,j}(t)
\]

We write

\[
\pi = (\pi_0, \pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots)
\]

where by corresponding to the state space \( \Omega \),

\[
\pi_0 = \pi_{0,0}
\]

\[
\pi_0 = (\pi_{0,1}, \pi_{0,2}, \pi_{0,3}, \pi_{0,4}, \cdots)
\]

\[
\pi_1 = (\pi_{1,0}, \pi_{1,1}, \pi_{1,2}, \pi_{1,3}, \pi_{1,4}, \cdots)
\]

\[
\pi_k = (\pi_{k,k-1}, \pi_{k,k-1}, \pi_{k,k}, \pi_{k,k+1}, \pi_{k,k+1}, \pi_{k,k+2}, \cdots), k \geq 2
\]

To express the stationary probability vector \( \pi = (\pi_0, \pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots) \), we need to compute some inverse matrices for a class of upper triangular matrices of infinite size, for example, the matrices \( Q_{0,0}, Q_{1,1} \) and \( A \) given in (2).
The following lemma provides the inverse of an upper triangular matrix of infinite size, which is useful in our later study.

**Lemma 1** Let $d_k \neq 0$ for $k = 0, 1, 2, \ldots$. Then the upper triangular matrix

$$D = \begin{pmatrix} d_0 & f_0 \\ d_1 & f_1 \\ d_2 & f_2 \\ \vdots & \vdots \end{pmatrix}$$

is invertible, and there exists a unique inverse matrix as follows:

$$D^{-1} = \begin{pmatrix} \frac{1}{d_0} - \frac{f_0}{d_0 d_1} & \frac{f_0 f_1}{d_0 d_1 d_2} & \frac{f_0 f_1 f_2}{d_0 d_1 d_2 d_3} & \cdots \\ \frac{1}{d_1} & -\frac{f_1}{d_1 d_2} & \frac{f_1 f_2}{d_1 d_2 d_3} & \cdots \\ \frac{1}{d_2} & -\frac{f_2}{d_2 d_3} & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$

whose $(k, k + l)$th element is given by

$$D^{-1}_{k,k+l} = (-1)^l \frac{f_k f_{k+1} f_{k+2} \cdots f_{k+l-1}}{d_k d_{k+1} d_{k+2} \cdots d_{k+l-1} d_{k+l}}$$

for $k = 0, 1, 2, \ldots$, and $l = 0, 1, 2, 3, \ldots$.

**Proof.** Note that $d_k \neq 0$ for $k = 0, 1, 2, \ldots$, the upper triangular matrix $D$ is invertible. Let

$$X = \begin{pmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} & \cdots \\ x_{1,1} & x_{1,2} & x_{1,3} & \cdots \\ x_{2,2} & x_{2,3} & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$

Then the invertible upper triangular matrix $D$ must have the unique upper triangular inverse matrix. Then by using $XD = I$, we obtain that for $k = 0, 1, 2, 3, \ldots$, and $l = 0, 1, 2, 3, \ldots$,

$$x_{k,k+l} = (-1)^l \frac{f_k f_{k+1} f_{k+2} \cdots f_{k+l-1}}{d_k d_{k+1} d_{k+2} \cdots d_{k+l-1} d_{k+l}}$$

This proof is completed.

By using Lemma 1, it is easy to explicitly express the inverse matrices of the matrices $Q_{0,0}, Q_{1,1}$ and $A$. Here, we omit their details.

The following lemma indicates that each element of the matrix $(I - R)^{-1}$ of infinite size is finite. This result is necessary and useful in our later analysis.

**Lemma 2** Let $R = C (-A)^{-1}$ and $(I - R)^{-1} = \sum_{k=0}^{\infty} R^k$. Then each element of the matrix $(I - R)^{-1}$ of infinite size is finite.

**Proof.** Please see (b) Proof of Lemma 2 in the appendix.

The following theorem provides an explicit expression for the stationary probability vector $\pi = (\pi_0, \pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots)$ by means of a new matrix-geometric solution with two rate matrices

$$R = C (-A)^{-1}$$

and

$$\tilde{R} = Q_{1,2} (-A)^{-1}$$

**Theorem 2** The stationary probability vector $\pi = (\pi_0, \pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots)$ of the pyramid Markov process $Q$ is matrix-geometric, given by

$$\pi_k = \pi_0 \tilde{R} R^{k-2}, \quad k = 2, 3, 4, \cdots$$

and $\pi_0$ and $\pi_1$ are uniquely determined by means of solving the following system of linear equations

$$\begin{pmatrix} Q_{0,0} + Q_{0,0} (-Q_{0,0})^{-1} Q_{0,0} \\ Q_{1,0} + \tilde{R} (I - R)^{-1} B \\ Q_{0,1} + Q_{0,0} (-Q_{0,0})^{-1} Q_{0,1} \\ Q_{1,1} \end{pmatrix} = 0$$

(6)
with the normalized condition

\[ \pi_0 \left[ 1 + Q_{0,0} (-Q_{0,0})^{-1} e \right] + \pi_1 \left[ \tilde{R} (I - R)^{-1} \right] e = 1 \]

(7)

**Proof.** From \( \pi Q = 0 \) and \( \pi e = 1 \), it is easy to see that

\[
\begin{aligned}
\pi_0 Q_{0,0} + \pi_0 Q_{0,0} + \pi_1 Q_{1,0} + \sum_{k=2}^{\infty} \pi_k B &= 0 \\
\pi_0 Q_{0,0} + \pi_0 Q_{0,0} &= 0 \\
\pi_0 Q_{0,1} + \pi_0 Q_{0,1} + \pi_1 Q_{1,1} &= 0 \\
\pi_1 Q_{1,2} + \pi_2 A &= 0 \\
\pi_k C + \pi_{k+1} A &= 0, \quad k \geq 2
\end{aligned}
\]

(8)

From the second equation of (8), it is easy to see that

\[ \pi_0 = \pi_0 Q_{0,0} (-Q_{0,0})^{-1} \]

and from the fourth and fifth equations of (8), it is easy to check that for \( k = 2, 3, 4, 5, \ldots \)

\[ \pi_k = \pi_1 \tilde{R} R^{k-2} \]

Also, by using the first and third equations of (8), the boundary equation (6) is obtained. Finally, the normalized condition (7) is given easily by means of \( \pi_0 + \sum_{k=0}^{\infty} \pi_k e = 1 \). This proof is completed.

**Remark 5.** Although the pyramid Markov process \( Q \) of blockchain selfish mining is more complicated, we obtain that the stationary probability vector is matrix-geometric, in which the two rate matrices can be explicitly expressed as \( R = C (-A)^{-1} \) and \( \tilde{R} = Q_{1,2} (-A)^{-1} \). Obviously, our matrix-geometric solution is different from that in Neuts (1981) for the Markov chains of GI/M/1 type whose rate matrix \( R \) only has a numerical solution by means of the nonlinear equation: \( \sum_{k=0}^{\infty} R^k A_k = 0 \).

In the remainder of this section, we simply explain three useful special cases.

**Case 1:** \( p_2 = 1 \). In this case, the main chain with two leading blocks by the dishonest mining pool is pegged on the blockchain with probability one. The dishonest mining pool cannot continue to mine more leading blocks.

Thus there do not exist all the states \((k, k + l)\) for \( k = 0, 1, 2, \ldots \) and \( l = 3, 4, 5, \ldots \), and the infinitesimal generator of the pyramid Markov process is simplified greatly.

**Case 2:** \( 0 \leq p_i < 1 \) for \( 2 \leq i \leq n_0 - 1 \) and \( p_{n_0} = 1 \). In this case, the pyramid Markov process does not have all the states \((k, k + l)\) for \( k = 0, 1, 2, \ldots \) and \( l = n_0 + 1, n_0 + 2, n_0 + 3, \ldots \). Also, the sizes of the two rate matrices: \( R = C (-A)^{-1} \) and \( \tilde{R} = Q_{1,2} (-A)^{-1} \) are finite, so that the computation of the two rate matrices is ordinary.

**Case 3:** \( p_k = 1 \) for \( k = 1 \), and the two mining pools follow the one-block leading competitive criterion. In this case, this becomes the model given by Gőbel et al. (2016), which is modified by means of our block-pegging rule: No new block can be generated during the block-pegging process of the main chain.

5. **Analysis of Orphan Blocks**

In this section, by using the stationary probability vector of the pyramid Markov process, we analyze some interesting performance measures of the main chain and the chain of orphan blocks. To our best knowledge, this is the first one to develop an effective quantitative method in the study of orphan blocks.

Once the honest and dishonest mining pools simultaneously begin a mining process, it is seen from Figure 2 that the selfish mining can result in two block branches forked at a common tree root, in which one chain comes from the honest mining pool, while the other from the dishonest mining pool. Also, the two block branches forked at the common tree root are terminated immediately once the main chain begins to peg on the blockchain.

If the main chain is taken by the honest mining pool, then all the blocks on the other chain by the dishonest mining pool become orphan blocks. On the contrary, if the main chain comes from the dishonest mining pool, then all the blocks on the other chain by the
honest mining pool are orphan blocks. Obviously, the main chain and the chain of orphan blocks end at the same moment that the main chain is confirmed.

Note that the main chain is taken by the honest mining pool at state \( (k, k - 2) \) with probability \( \pi_{k,k-2} \) for \( k = 2, 3, 4, \ldots \); while the main chain is taken by the dishonest mining pool at state \( (k, k + l) \) with probability \( \pi_{k,k+l} \) for \( k = 0, 1, 2, \ldots, l = 2, 3, 4, \ldots \). In addition, all the other states can not result in such a main chain. It is easy to see that \( \sum_{k=2}^{\infty} \pi_{k,k-2} \) and \( \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} \pi_{k,k+l} \) are the probabilities that the main chain is taken by the honest and dishonest mining pools, respectively.

Let \( P_H \) and \( P_D \) be the probabilities that the main chain is taken by the honest and dishonest mining pools, respectively. Then

\[
P_H = \frac{\sum_{k=2}^{\infty} \pi_{k,k-2}}{\sum_{k=2}^{\infty} \pi_{k,k-2} + \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} \pi_{k,k+l}}
\]

\[
P_D = \frac{\sum_{k=0}^{\infty} \sum_{l=2}^{\infty} \pi_{k,k+l}}{\sum_{k=2}^{\infty} \pi_{k,k-2} + \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} \pi_{k,k+l}}
\]

(a) The average stationary lengths of the two chains

Between the two block branches forked at the common tree root, one is taken as the main chain, while the other is regarded as the chain of orphan blocks.

Let \( L_M \) and \( L_O \) be the average stationary lengths of the main chain and the chain of orphan blocks, respectively.

The following theorem provides expressions for \( L_M \) and \( L_O \) by using the stationary probability vector of the pyramid Markov process \( Q \).

**Theorem 3** (a) The average stationary length of the main chain is given by

\[
L_M = P_H \sum_{k=2}^{\infty} k \pi_{k,k-2} + P_D \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l}
\]

(b) The average stationary length of the chain of orphan blocks is given by

\[
L_O = P_H \sum_{k=2}^{\infty} (k - 2) \pi_{k,k-2} + P_D \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} k \pi_{k,k+l}
\]

**Proof.** We only prove (a), while (b) can be proved similarly.

Let \( N_M \) be the stationary length of the main chain. Also, we introduce two events as follows:

\[
E_H = \{ \text{The main chain is taken by the honest mining pool} \}
\]

and

\[
E_D = \{ \text{The main chain is taken by the dishonest mining pool} \}
\]

Then \( P_H = P \{ E_H \} \) and \( P_D = P \{ E_D \} \). By applying the law of total probability, we have

\[
L_M = E[N_M] = E[N_M \mid E_H] P \{ E_H \} + E[N_M \mid E_D] P \{ E_D \}
\]

Note that \( E[N_M \mid E_H] = \sum_{k=2}^{\infty} k \pi_{k,k-2} \), since the stationary length \( N_M \) of the honest main chain is \( k \) with probability \( \pi_{k,k-2} \); \( E[N_M \mid E_D] = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l} \), because the stationary length of the dishonest main chain is \( k + l \) with probability \( \pi_{k,k+l} \). Thus it follows from (11) that

\[
L_M = P_H \sum_{k=2}^{\infty} k \pi_{k,k-2} + P_D \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l}
\]

This completes the proof. □
(b) The blockchain pegging rate and the orphan block removal rate

Let $Y_M$ and $Y_O$ be the stationary block-pegging rate of the main chain, and the stationary orphan block removal rate of the chain of orphan blocks, respectively. Then the following theorem provides an expressions for $Y_M$ and $Y_O$.

**Theorem 4** (a) The stationary block-pegging rate of the main chain is given by

$$Y_M = \mu \left[ P_H \sum_{k=0}^{\infty} k \pi_{k,k-2} + P_D \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) p_l \pi_{k,k+l} \right]$$

(b) The stationary orphan block removal rate of the chain of orphan blocks is given by

$$Y_O = \mu \left[ P_H \sum_{k=2}^{\infty} (k - 2) \pi_{k,k-2} + P_D \sum_{k=2}^{\infty} \sum_{l=2}^{\infty} k \pi_{k,k+l} \right]$$

Proof. The proof is similar to that in Theorem 3, while the only difference between both of them comes from the following two points:

**Point one:** In the main chain (resp. the chain of orphan blocks), each of the blocks that is submitted to the blockchain (resp. the transaction pool) has an exponential network delay time with parameter $\mu$. Thus the $k$ blocks can have the exponential network delay time with parameter $k \mu$.

**Point two:** When the main chain is taken by the dishonest mining pool, it is a key to observe at state $(k, k + l)$ that with probability $p_l$, the main chain is published on the blockchain; while with probability $1 - p_l$, the main chain is detained to continue mining more blocks such that it is not broadcasted in the blockchain network. In this case, the dishonest mining pool hopes to obtain more mining profit through winning on mining more blocks. On the contrary, the stationary orphan block removal probability of the chain of orphan blocks at state $(k, k + l)$ is $\pi_{k,k+l}$ because the orphan blocks are removed to the transaction pool with probability one. This completes the proof.

Now, we introduce two useful ratios of the blockchain selfish mining, which are necessary and useful in design of blockchain.

We define the ratio of two average stationary lengths as

$$\phi = \frac{L_O}{L_M}$$

and the stationary ratio of block removal and pegging rates as

$$\psi = \frac{Y_O}{Y_M}$$

It is easy to see from $0 \leq p_l \leq 1$ that $0 < \phi < \psi < 1$. This shows that the main chain may not immediately be pegged on the blockchain due to the blockchain selfish mining.

In the remainder of this section, we develop some local performance measures of the blockchain system, and discuss their monotonous properties for the jumping’s mining rate $\gamma$ and the efficiency-increased ratio $\mathcal{R}$. Note that the monotonous properties are useful in our later study.

Let $L_M^{(H)}$ and $L_M^{(D)}$ be the average stationary lengths of the main chain by the honest and dishonest mining pools, respectively; and $L_O^{(H)}$ and $L_O^{(D)}$ the average stationary lengths of the...
chain of orphan blocks by the honest and dishonest mining pools, respectively. We write
\[ \Lambda = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l} \]

The following corollary shows that the blockchain selfish mining can make a larger waste of computing resources.

**Corollary 1** For the blockchain selfish mining, we have
(a) \( 0 < L_{(M)}^{(H)} - L_{(O)}^{(H)} < 2 \),
(b) \( 0 < L_{(M)}^{(D)} - L_{(O)}^{(D)} < \Lambda \), and
(c) \( 0 < L_{M} - L_{O} < 2P_{H} + \Lambda P_{D} \).

**Proof.** We only prove (b) and (c), while (a) can be proved similarly.

It is easy to check that
\[ L_{(M)}^{(D)} = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l} \]
and
\[ L_{(O)}^{(D)} = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} l \pi_{k,k+l} \]

Thus we obtain
\[ L_{(M)}^{(D)} - L_{(O)}^{(D)} = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} l \pi_{k,k+l} > 0 \]

and
\[ L_{M}^{(D)} - L_{O}^{(D)} = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} l \pi_{k,k+l} \]

< \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \pi_{k,k+l} = \Lambda

Now, we prove (c). Since
\[ L_{M} = P_{H}L_{M}^{(H)} + P_{D}L_{M}^{(D)} \]
and
\[ L_{O} = P_{H}L_{O}^{(H)} + P_{D}L_{O}^{(D)} \]

we obtain
\[ 0 < L_{M} - L_{O} = P_{H} \left( L_{M}^{(H)} - L_{O}^{(H)} \right) + P_{D} \left( L_{M}^{(D)} - L_{O}^{(D)} \right) \]
\[ < 2P_{H} + \Lambda P_{D} \]

This completes the proof.

**Corollary 2** In the blockchain selfish mining, we have
(a) the two average stationary lengths \( L_{M}^{(H)} \) and \( L_{O}^{(H)} \) decrease as the jumping’s mining rate \( \gamma \) increases;
(b) the two average stationary lengths \( L_{M}^{(D)} \) and \( L_{O}^{(D)} \) increase as the jumping’s mining rate \( \gamma \) increases.

**Proof.** We only prove (a), while (b) can be proved similarly.

If decreasing the mining rate of the honest mining pool and simultaneously increasing the mining rate of the dishonest mining pool, then the probability \( \pi_{k,k-2} \) decreases for \( k \geq 2 \).

As the jumping’s mining rate \( \gamma \) increases, the mining rate \( \beta - \gamma \) of the honest mining pool decreases, and simultaneously the mining rate \((\tilde{\alpha} + \gamma)(1 + R)\) of the dishonest mining pool increases. Since
\[ L_{M}^{(H)} = \sum_{k=2}^{\infty} k \pi_{k,k-2} \]
and
\[ L_{O}^{(H)} = \sum_{k=2}^{\infty} (k - 2) \pi_{k,k-2} \]

it is easy to see that the two average stationary lengths \( L_{M}^{(H)} \) and \( L_{O}^{(H)} \) decrease as the jumping’s mining rate \( \gamma \) increases. This completes the proof.

**Corollary 3** In the blockchain selfish mining, we have
(a) the two average stationary lengths \( L_{M}^{(H)} \) and \( L_{O}^{(H)} \) decrease as the efficiency-increased ratio \( R \) increases; and
(b) the two average stationary lengths \( L_{M}^{(D)} \) and \( L_{O}^{(D)} \) increase as the efficiency-increased ratio \( R \) increases.

**Proof.** We only prove (a), while (b) can be proved similarly.

We write
\[ h(R) = \frac{\beta - \gamma}{\beta - \gamma + (\tilde{\alpha} + \gamma)(1 + R)} \]
and \[
\tilde{d} (\mathcal{R}) = \frac{(\tilde{\alpha} + \gamma) (1 + \mathcal{R})}{\beta - \gamma + (\tilde{\alpha} + \gamma) (1 + \mathcal{R})} = \frac{\tilde{\alpha} + \gamma}{\tilde{\alpha} + \gamma + \frac{\beta - \gamma}{1 + \mathcal{R}}.}
\]

It is easy to check that \( h (\mathcal{R}) \) decreases and \( d (\mathcal{R}) \) increases as the efficiency-increased ratio \( \mathcal{R} \) increases.

When decreasing the relative mining rate \( h (\mathcal{R}) \) of the honest mining pool and simultaneously increasing the relative mining rate \( d (\mathcal{R}) \) of the dishonest mining pool, the probability \( \pi_{k,k-2} \) can decrease. Thus, the two average stationary lengths \( L_M^{(H)} \) and \( L_O^{(H)} \) decrease as the efficiency-increased ratio \( \mathcal{R} \) increases. This completes the proof. ■

Although Corollaries 2 and 3 provide some monotonous properties for the local performance measures (e.g., the average stationary lengths \( L_M^{(H)} \) and \( L_O^{(H)} \), and \( L_M^{(D)} \) and \( L_O^{(D)} \)), there do not exist such a monotonicity for the total performance measures (e.g., \( L_M \) and \( L_O \); \( \Upsilon_M \) and \( \Upsilon_O \); and \( \phi \) and \( \psi \)). This can be observed in Figure 10 and 11 of Section 10.

6. Markov Reward Processes

In this section, we set up a pyramid Markov reward process to evaluate the long-run average mining profits of the honest and dishonest mining pools, respectively. Note that our results apply the Markov reward process to provide a more complete and practical mining profit than that in Eyal and Sirer (2018).

In the pyramid Markov process \{\((I (t), J (t)) : t \geq 0\)\}, let \( f_H (I (t), J (t)) \) and \( f_D (I (t), J (t)) \) be the reward functions received by the honest and dishonest mining pools at time \( t \), respectively. Then the average mining profits in the time interval \([0, t)\) of the honest and dishonest mining pools are respectively given by

\[
R_H (t) = E \left[ \frac{1}{t} \int_0^t f_H (I (t), J (t)) \, dt \right]
\]
and
\[
R_D (t) = E \left[ \frac{1}{t} \int_0^t f_D (I (t), J (t)) \, dt \right]
\]

Since the pyramid Markov process \{\((I (t), J (t)) : t \geq 0\)\} is irreducible and positive recurrent, by using Chapter 10 of Li (2010), the long-run average mining profits of the honest and dishonest mining pools are respectively given by

\[
R_H = \lim_{t \to +\infty} R_H (t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \pi_{i,j} f_H (i, j)
\]
\[
+ \sum_{i=2}^{\infty} \sum_{j=1}^{i-2} \pi_{i,j} f_H (i, j)
\]

(16)

and

\[
R_D = \lim_{t \to +\infty} R_D (t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \pi_{i,j} f_D (i, j)
\]
\[
+ \sum_{i=2}^{\infty} \sum_{j=1}^{i-2} \pi_{i,j} f_D (i, j)
\]

(17)

Note that the stationary probability vector \( \pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots) \) is given in Theorem 2, it is easy to see from (16) and (17) that we first need to express the two reward functions \( f_H (i, j) \) and \( f_D (i, j) \) for \( i, j = 0, 1, 2, 3, \cdots \). From the model description of Section 3, we can obtain

\[
f_H (i, j) = \begin{cases} 
k\mu (r_B + r_F) - (c_E + c_A) (\beta - \gamma), & \text{for } i = k, j = k - 2, k \geq 2 \\
- (c_E + c_A) (\beta - \gamma), & \text{for all the other states}
\end{cases}
\]
\[ f_D(i,j) = \begin{cases} 
(k + l) \mu p_l (r_B + r_F) - (\bar{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})], & \text{for } i = k, j = k + l, k \geq 0, l \geq 2 \\
(\bar{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})], & \text{for all the other states} 
\end{cases} \]

Now, we introduce the Hadamard product of two vectors \( W = (w_1, w_2, w_3, \cdots) \) and \( V = (v_1, v_2, v_3, \cdots) \) as follows:

\[ W \odot V = (w_1 v_1, w_2 v_2, w_3 v_3, \cdots) \]

It is easy to check that

\[ W \odot V_1 + W \odot V_2 = W \odot (V_1 + V_2) \]

Let \( \mathbf{e}_1 = (1, 0, 0, 0, \cdots) \).

The following theorem provides an explicit expression for the long-run average mining profit of the honest mining pool.

**Theorem 5** The long-run average mining profit of the honest mining pool is given by

\[
R_H = \mu (r_B + r_F) \left\{ e_1 \odot \right. \\
\left. \left\{ \pi_1 \mathbf{R} \left[ (I - \mathbf{R})^{-1} + (I - \mathbf{R})^{-2} \right] \right\} \mathbf{e} \right. \\
\left. - (c_E + c_A) (\beta - \gamma) \right. 
\]

\[ (18) \]

**Proof.** Note that

\[
R_H = \sum_{i=0}^{1} \sum_{j=0}^{\infty} \pi_{i,j} f_{H} (i,j) + \sum_{i=2}^{\infty} \sum_{j=1}^{\infty} \pi_{i,j} f_{H} (i,j) \\
= \mu (r_B + r_F) \sum_{k=2}^{\infty} k \pi_{k,k-2} \\
- (c_E + c_A) (\beta - \gamma) \left( \sum_{i=0}^{1} \sum_{j=0}^{\infty} + \sum_{i=2}^{\infty} \sum_{j=1}^{\infty} \right) \pi_{i,j} \\
\]

since \( \left( \sum_{i=0}^{1} \sum_{j=0}^{\infty} + \sum_{i=2}^{\infty} \sum_{j=1}^{\infty} \right) \pi_{i,j} = 1 \) and

\[
\sum_{k=2}^{\infty} k \pi_{k,k-2} = \sum_{k=2}^{\infty} (e_1 \odot k \pi_k) e \\
= \left\{ e_1 \odot \left( \pi_1 \mathbf{R} \sum_{k=2}^{\infty} k \mathbf{R}^{k-2} \right) \right\} \mathbf{e} \\
= e_1 \odot \left\{ \pi_1 \mathbf{R} \left[ (I - \mathbf{R})^{-1} + (I - \mathbf{R})^{-2} \right] \right\} \mathbf{e} \\
\]

we obtain

\[
R_H = \mu (r_B + r_F) \left\{ e_1 \odot \right. \\
\left. \left\{ \pi_1 \mathbf{R} \left[ (I - \mathbf{R})^{-1} + (I - \mathbf{R})^{-2} \right] \right\} \mathbf{e} \right. \\
\left. - (c_E + c_A) (\beta - \gamma) \right. 
\]

This completes the proof.

**Corollary 4** In the blockchain selfish mining, we have

(a) the long-run average mining profit \( R_H \) of the honest mining pool decreases as the jumping’s mining rate \( \gamma \) increases;

(b) the long-run average mining profit \( R_H \) of the honest mining pool decreases as the efficiency-increased ratio \( \mathbb{R} \) increases.

**Proof.** Note that

\[
R_H = \mu (r_B + r_F) \sum_{k=2}^{\infty} k \pi_{k,k-2} \\
- (c_E + c_A) (\beta - \gamma) 
\]

as the jumping’s mining rate \( \gamma \) (or the efficiency-increased ratio \( \mathbb{R} \)) increases, it is easy to see from Corollaries 2 and 3 that the long-run average mining profit \( R_H \) of the honest mining pool decreases. This completes the proof.

Now, we consider the long-run average mining profit of the dishonest mining pool, which is a bit more complicated than that of the long-run average mining profit of the honest mining pool.

Let

\[
\mathbf{p}_0 = (0, 2p_2, 3p_3, 4p_4, 5p_5, \cdots) \\
\mathbf{p}_1 = (0, 0, 0, 3p_2, 4p_3, 5p_4, 6p_5, \cdots) \\
\]

and for \( k \geq 2 \),

\[
\mathbf{p}_k = \{0, 0, 0, (k + 2) p_2, (k + 3) p_3, \cdots \} \\
\]

\[ (k + 4) p_4, (k + 5) p_5, \cdots \} \]
The following theorem provides an explicit expression for the long-run average mining profit of the dishonest mining pool.

**Theorem 6** The long-run average mining profit of the dishonest mining pool is given by

\[
\mathbb{R}_D = \mu (r_B + r_F) \left\{ \mathbb{P}_0 \odot \left[ \pi_0 Q_{0,0} ( -Q_{0,0} )^{-1} \right] \right. \\
+ \mathbb{P}_1 \odot \pi_1 \\
+ \sum_{k=2}^{\infty} \mathbb{P}_k \odot \left( \pi_1 \mathbb{R}_D^{k-2} \right) \left\} e \\
- (\tilde{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})] \right. \\
× \left. \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{l=2}^{\infty} \sum_{j=l-2}^{\infty} \pi_{i,j} f_D (i, j) \right) \right. \\
\left. \left. (R_B + r_F) \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} p_l (k + l) \pi_{k,k+l} \right. \\
- \left. (\tilde{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})] \right) \\
\times \left. \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{l=2}^{\infty} \sum_{j=l-2}^{\infty} \pi_{i,j} \right) \right. \\
this gives the desired result. We complete the proof. 

**Proof.** We only need to note that

\[
\mathbb{R}_D = \mu (r_B + r_F) \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} p_l (k + l) \pi_{k,k+l} \\
- (\tilde{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})] \\
\]

By using Corollaries 2 and 3, it is easy to give our desired result. We complete the proof. 

The following theorem provides a sufficient condition under which the blockchain system can operate normally.

**Theorem 7** In the blockchain selfish mining, the long-run average mining profits \( \mathbb{R}_H \) and \( \mathbb{R}_D \) are multivariate linear in the three key parameters: \( r_B \), \( r_F \), \( c_E \) and \( c_A \).

The following theorem provides a sufficient condition under which the blockchain system can operate normally.

**Theorem 8** In the blockchain selfish mining, there exists a minimal positive number \( \mathbb{V} \) such that for \( r_B + r_F > \mathbb{V} \), the blockchain system can operate normally.

**Proof.** For the blockchain selfish mining, to guarantee the normal operations of blockchain, we need to satisfy two basic conditions: \( \mathbb{R}_H > 0 \) and \( \mathbb{R}_D > 0 \). It follows from (19) and (21) that \( \mathbb{R}_H > 0 \) and \( \mathbb{R}_D > 0 \) if

\[
(r_B + r_F) > \frac{(c_E + c_A) (\beta - \gamma)}{\mu \sum_{k=2}^{\infty} k \pi_{k,k-2}} \\
\]

and

\[
(r_B + r_F) > \frac{(\tilde{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})]}{\mu \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} p_l (k + l) \pi_{k,k+l}} \\
\]

Let

\[
\mathbb{V} = \max \left\{ \frac{(c_E + c_A) (\beta - \gamma)}{\mu \sum_{k=2}^{\infty} k \pi_{k,k-2}}, \right. \\
\left. \frac{(\tilde{\alpha} + \gamma) [c_E + c_A (1 + \mathbb{R})]}{\mu \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} p_l (k + l) \pi_{k,k+l}} \right\} \\
\]
Thus it is clear that if $r_B + r_F > V$, then $R_H > 0$ and $R_D > 0$. This completes the proof. ■

In the remainder of this section, we further provide a long-run economic ratio of the dishonest mining pool over the honest mining pool. We focus on the economic ratio using the per unit net mining rate. Define

$$J = \frac{1}{\alpha + \gamma} \frac{R_D}{R_H} = \frac{\beta - \gamma}{\beta - \gamma} \frac{R_D}{R_H}$$

(22)

Obviously, the economic ratio $J$ measures the mining advantage of the dishonest mining pool.

The following theorem provides a useful approximate evaluation for the long-run economic ratio if $r_B + r_F$ is significantly larger than the costs.

Theorem 9 In the blockchain selfish mining, if $\max\{c_E, c_A\} / (r_B + r_F) \approx 0$, then there exists a positive constant $C$ such that $J \approx C$.

Proof. It follows from (19), (21) and (22) that for $\max\{c_E, c_A\} / (r_B + r_F) \approx 0$,

$$J = \frac{\beta - \gamma}{\alpha + \gamma} \frac{R_D}{R_H} \approx \frac{\beta - \gamma}{\alpha + \gamma} \frac{1}{\alpha + \gamma} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} p_l (k + l) \pi_{k,k+l}$$

$$= \frac{\beta - \gamma}{\alpha + \gamma} \frac{1}{\alpha + \gamma} \sum_{k=2}^{\infty} k \pi_{k,k-2}$$

Let

$$C = \frac{\beta - \gamma}{\alpha + \gamma} \frac{1}{\alpha + \gamma} \sum_{k=2}^{\infty} k \pi_{k,k-2}$$

This completes the proof. ■

In addition, we consider the long-run block-pegging rate ratio of the two mining pools. We define

$$\tau = \frac{1}{\alpha + \gamma} \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + l) \mu p_l \pi_{k,k+l}$$

$$= \frac{1}{\beta - \gamma} \sum_{k=2}^{\infty} k \mu \pi_{k,k-2}$$

$$= \frac{\beta - \gamma}{\alpha + \gamma} \frac{1}{\alpha + \gamma} \sum_{k=2}^{\infty} k \pi_{k,k-2}$$

The following corollary provides an interesting monotonicity for both the long-run economic ratio $J$ and the long-run block-pegging rate ratio $\tau$ with respect to the efficiency-increased ratio $R$. The proof is easy by means of (b) of Corollaries 4 and 5 and is omitted here.

Corollary 6 In the blockchain selfish mining, each of the two long-run ratios $J$ and $\tau$ increases, as the efficiency-increased ratio $R$ increases.

Remark 6. By establishing the pyramid Markov reward process, this paper investigates the long-run average mining profits of the honest and dishonest mining pools, respectively. Further, we show that the long-run average mining profits are multivariate linear in the three key parameters: $r_B + r_F$, $c_E$ and $c_A$. Based on this, we can measure the mining efficiency of the dishonest mining pool. Thus, our method of pyramid Markov reward processes can effectively improve the revenue analysis given in (3) of Eyal and Sirer (2018).

7. The Mining Profits in the Time Interval $[0, t]$

In this section, we compute the mining profits in the time interval $[0, t)$ of the honest and dishonest mining pools, respectively. This is an interesting but difficult topic due to a high computational complexity and fewer available results of the transient solution of Markov processes, e.g., see Li (2010). To this end, we develop a new effective method that applies the PH distribution of infinite size and the associated PH renewal process to determine the expected numbers that the main chains by the
honest mining pool and the dishonest mining pool are pegged on the blockchain, respectively.

For simplicity of calculation, we only consider the stable case of the blockchain system, that is, the initial probability vector of the Markov process \( \{ (l(t), f(t)) : t \geq 0 \} \) is observed at its states following the stationary state probability distribution \( \pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots) \).

To compute the expected number of generating the main chains by the honest (resp. dishonest) mining pool, it is a key to observe that state \((0,0)\) is generated by either the honest mining pool or the dishonest mining pool. In this case, we write state \((0,0)\) by the honest mining pool (resp. the dishonest mining pool) as an absorbing state \(\Delta\) of the Markov process \( \{ (l(t), f(t)) : t \geq 0 \} \). The case that state \((0,0)\) by the honest mining pool is regarded as an absorbing state is illustrated in Figure 5; while the other case that state \((0,0)\) by the dishonest mining pool is regarded as an absorbing state is illustrated in Figure 6.

In what follows we consider the honest and dishonest mining processes, both of which have a common initial probability vector \( \pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \cdots) \) at time 0, and show that each of their first passage times to the absorbing state \(\Delta\) follows a PH distribution of infinite size, e.g., see Chapter 8 of Li (2010) and Chapter 2 of Neuts (1981).

(a) The honest mining pool

Now, we compute the expected number of generating the main chains by the honest mining pool. It is key to compute the first passage time that the honest mining process enters the absorbing state \(\Delta\) for the first time. Clearly, the honest mining process is a continuous-time Markov process whose state transition relation of such a process is depicted in Figure 5.

From Figure 5, we write the infinitesimal generator of the honest mining process with the absorbing state \(\Delta\) as

\[
Q_H = \begin{pmatrix}
0 & 0 \\
T_0 & T
\end{pmatrix}
\]

where

\[
T = \begin{pmatrix}
Q_{0,0} & Q_{0,1} & Q_{0,2} \\
Q_{1,0} & Q_{1,1} & Q_{1,2} \\
\vdots & \vdots & \ddots
\end{pmatrix}
\]

\[
T^0 = \begin{pmatrix}
0 \\
0 \\
\vdots
\end{pmatrix}
\]

and all the other block-elements are the same as those in (2).

When the main chain by the honest mining pool is pegged on the blockchain, we denote by \(\chi_H\) the time duration starting from the initial mining moment 0 until the main chain by the honest mining pool is pegged on the blockchain for the first time. We call \(\chi_H\) the first generated-pegged time of the main chain by the honest mining pool.

**Theorem 10** The first generated-pegged time \(\chi_H\) of the main chain by the honest mining pool is of phase type of infinite size with irreducible representation \((\pi, T)\). At the same time, we have

\[
E[\chi_H] = -\pi T^{-1}_{\max} e
\]

where \(T^{-1}_{\max}\) is the maximal non-positive inverse matrix of the matrix \(T\) of infinite sizes.
Proof. Please see (c) Proof of Theorem 10 in the appendix.

When the mining difficulty level is not adjusted, it is easy to see that the number $N_H(t)$ of generating the main chain by the honest mining pool in the time interval $[0, t)$ is a renewal process whose interarrival times are i.i.d. and the first generated-pegged time is $H$. Since the first generated-pegged time $H$ is of phase type of infinite size with irreducible representation $(\pi, T)$, its associated renewal process is a PH renewal process of infinite size. Therefore, $N_H(t)$ is also the number of renewals of the PH renewal process in the time interval $(0, t)$. Let $J(t)$ be the phase of the PH renewal process at time $t$. It is clear that $J(t) \in \{0, 1, 2, \cdots \}$. We write

$$P_{i,j}(n, t) = P \left\{ N_H(t) = n, J(t) = j \mid N_H(0) = 0, J(0) = i \right\}$$

for $n \geq 0$ and $i, j = 0, 1, 2, \cdots$. We denote by $P(n, t)$ a matrix of infinite size with the $(i, j)$-th element $P_{i,j}(n, t)$. It is clear from Neuts (1981) and Li (2010) that the matrix sequence \{P(n, t)\} satisfies the Chapman-Kolmogorov differential equations as follows:

$$\frac{d}{dt} P(0, t) = TP(0, t)$$
$$\frac{d}{dt} P(n, t) = TP(n, t) + T^0 \pi P(n - 1, t), \quad n \geq 1$$

with $P(n, 0) = \delta_{n0}I$.

Let $P^*(z, t) = \sum_{n=0}^{\infty} z^n P(n, t)$. Then

$$P^*(z, t) = \exp \left\{ (T + z T^0 \pi) t \right\}, \quad t \geq 0.$$

Let $Q^*_H = T + T^0 \pi$. Note that the Markov process $Q^*_H$ is irreducible and positive recurrent, and thus there must exist the stationary probability vector $\nu$ such that $\nu Q^*_H = 0$ and $\nu e = 1$. In this case, we have

$$E[N_H(t)] = \frac{d}{dz} P^*(z, t) e|_{z=1}$$
$$= \nu T^0 \pi t \exp(Q^*_H t) e$$
$$= \frac{t}{E[(\chi_H)]} \pi \exp(Q^*_H t)$$
Figure 6 The State Transition Relation of the Pyramid Markov Process $Q_D$

since $\nu T^0 = 1/E[\chi_H]$. 

Now, we compute the stationary average number $\Psi_H$ of blocks in the main chain by the honest mining pool. It follows from Figure 5 that

$$\Psi_H = \sum_{k=2}^{\infty} k \pi_{k,k-2}$$

Based on the above analysis, the mining profit in the time interval $[0,t)$ of the honest mining pool is given by

$$R_H(t) = E[N_H(t)] \Psi_H (r_B + r_F) - t (c_E + c_A) (\beta - \gamma)$$

$$= \frac{t \Psi_H (r_B + r_F)}{E[\chi_H]} \pi \exp(Q_H t) e$$

$$- t (c_E + c_A) (\beta - \gamma)$$

(b) The dishonest mining pool

By a similar analysis to that in (a), we can discuss the mining profit in the time interval $[0,t)$ of the dishonest mining pool. Here, we provide a simple outline of such an analysis.

Our aim is to compute the expected number of generating the main chains by the dishonest mining pool. To this end, it is key to compute the first passage time that the dishonest mining process enters the absorbing state $\Delta$ for the first time. The dishonest mining process is a continuous-time pyramid Markov process whose state transition relation is depicted in Figure 6.

From Figure 6, the infinitesimal generator of the pyramid Markov process with an absorbing state $(0,0)$ is

$$Q_D = \begin{pmatrix} 0 & 0 \\ S^0 & S \end{pmatrix}$$

where

$$S = \begin{pmatrix} Q_{0,0} & Q_{0,1} & Q_{0,2} \\ 0 & Q_{1,1} & Q_{1,2} \\ \vdots & \vdots & \ddots \end{pmatrix}$$
The first generated-pegged time of the main chain by the dishonest mining pool is pegged on the blockchain, we de-
scribe the blockchain selfish mining. It follows from Figure 6 that

\[ \Psi_D = \sum_{k=0}^{\infty} \sum_{l=2}^{\infty} (k + 1) p l \pi_{k,k+l} \]  

(26)

Therefore, the mining profit in the time interval [0, t) of the dishonest mining pool is given by

\[ R_H(t) = t (\bar{\alpha} + \gamma) [c_E + c_A (1 + H)] - \frac{t\Psi_D (r_B + r_F)}{E[\chi_D]} - \pi \exp(Q_D^0 t) e \]

(27)

where \( Q_D^0 = S + S^0 \pi \).

8. A One-Dimensional Markov Model

In this section, we provide a new one-dimensional Markov model to further deal with the mining processes of the two honest and dishonest mining pools. To this end, we establish a new continuous-time Markov process to discuss the blockchain selfish mining.

Let \( N_H(t) \) and \( N_D(t) \) denote the numbers of blocks mined by the honest and dishonest mining pools in the time interval [0, t), respectively. We write \( N(t) = N_D(t) - N_H(t) \). It is easy to see that \( \{ N(t) : t \geq 0 \} \) is a one-dimensional Markov process whose state space
is \( E = \{(0,0), -2, -1, 0, 1, 2, \cdots \} \), and its state transition relations are depicted in Figure 7.

By using Figure 7, the infinitesimal generator of the Markov process \( \{N(t): t \geq 0\} \) is given by

\[
QE = \begin{pmatrix}
-(a + b) & 0 & b & 0 & a \\
\mu & -(a + \mu) & a & 0 & 0 \\
0 & b & -(a + b) & a & 0 \\
0 & b & -(a + b) & a & 0 \\
\mu p_2 & b & -\xi_2 & a(1 - p_2) & \\
\mu p_3 & b & -\xi_3 & a(1 - p_3) & \\
\vdots & \ddots & \ddots & \ddots & \\
\end{pmatrix}
\]

Now, we use the mean-drift method to study the stability of the Markov process \( QE \). For \( k \geq 5 \), it is easy to see from the infinitesimal generator \( QE \) that the downward rate from level \( k \) to levels \( k - 1 \) and 0 is given by \( r_{\text{High} \to \text{Low}}^{(k)} = b + \mu p_{k-3} \); while the upward rate from level \( k \) to levels \( k + 1 \) is given by \( r_{\text{Low} \to \text{High}}^{(k)} = a(1 - p_{k-3}) \).

Note that \( \lim_{k \to \infty} p_k = 1 \), and we obtain that \( \lim_{k \to \infty} r_{\text{High} \to \text{Low}}^{(k)} = b + \mu > 0 \) and \( \lim_{k \to \infty} r_{\text{Low} \to \text{High}}^{(k)} = 0 \). This gives

\[
\lim_{k \to \infty} r_{\text{High} \to \text{Low}}^{(k)} > \lim_{k \to \infty} r_{\text{Low} \to \text{High}}^{(k)}
\]

Based on this, there exists a big enough positive integer \( n_0 \) such that \( r_{\text{High} \to \text{Low}}^{(k)} > r_{\text{Low} \to \text{High}}^{(k)} \) for each \( k > n_0 \). Therefore, by using the mean drift method, it is easy to check that the Markov process \( QE \) is irreducible and positive recurrent.

If the Markov process \( QE \) is irreducible and positive recurrent, then there must exist one unique stationary probability vector \( \Psi = (\psi_0, \psi_{-2}, \psi_{-1}, \psi_0, \psi_{-1}, \psi_{-2}, \psi_3, \cdots) \), which satisfies the system of linear equations: \( \Psi Q_E = 0 \) and \( \Psi e = 1 \).

Now, we provide a matrix-analytic method to compute the stationary probability vector \( \Psi \) from the system of linear equations: \( \Psi Q_E = 0 \) and \( \Psi e = 1 \). To do this, we write the infinitesimal generator \( QE \) as the standard structured form of the matrix-analytic method as follows:

\[
QE = \begin{pmatrix}
B_1 & B_0 \\
B_2 & A_1^{(1)} & A_0^{(1)} \\
\mu p_4 & A_2^{(1)} & A_0^{(2)} \\
\mu p_5 & A_2^{(2)} & A_1^{(2)} & A_0^{(3)} \\
\vdots & \ddots & \ddots & \ddots & \\
\end{pmatrix}
\]

where

\[
B_1 = \begin{pmatrix}
-(a + b) & 0 & b & 0 & a \\
\mu & -(a + \mu) & a & 0 & 0 \\
0 & b & -(a + b) & a & 0 \\
0 & b & -(a + b) & a & 0 \\
\mu p_2 & & & & \\
\end{pmatrix}
\]
\[ B_0 = (0, 0, 0, 0, 0, a (1 - p_2)) \]
\[ B_2 = (\mu p_3, 0, 0, 0, 0, b) \]
\[ A_1^{(i)} = -\xi_3, \quad A_0^{(i)} = a (1 - p_3) \]

for \( k \geq 2 \)
\[ A_1^{(k)} = -\xi_{k+2}, \quad A_0^{(k)} = a (1 - p_{k+2}), \quad A_2^{(k)} = b \]

Let the sequence of numbers \( \{R_k : k \geq 2\} \) be the minimal positive solution to the system of quadratic equations. For \( k \geq 2 \),
\[ A_0^{(k-1)} + R_k A_1^{(k)} + R_k R_{k+1} A_2^{(k+1)} = 0 \]
or
\[ a (1 - p_{k+1}) - \xi_{k+2} R_k + b R_k R_{k+1} = 0 \quad (27) \]

Once the sequence of numbers \( \{R_k : k \geq 2\} \) is obtained numerically, we further solve the system of linear equations (28). Based on this, we obtain the stationary probability vector \( \Psi = (\psi_{0,0}, \psi_{-2}, \psi_{-1}, \psi_0, \psi_1, \psi_2, \psi_3, \cdots) \).

**Remark 7.** It is a key to solve the system of nonlinear equation (27). To this end, some effective algorithms were developed in Bright and Taylor (1995 1997), and also Liu et al. (2020) provided some practical examples of such a computation.

By using the sequence of numbers \( \{R_k : k \geq 2\} \), it follows from Section 1.3 in Chapter one of Li (2010) that
\[ \psi_{k+2} = \psi_3 R_k R_{k-1} R_{k-2} \cdots R_2, \quad k \geq 1 \]
and the seven positive numbers \( \psi_{(0,0)}, \psi_{-2}, \psi_{-1}, \psi_0, \psi_1, \psi_2 \) and \( \psi_3 \) uniquely satisfy the following system of linear equations:

\[
\begin{align*}
-(a + b) \psi_{(0,0)} + \mu \psi_{-2} + \mu p_2 \psi_2 + \mu \left( p_3 + \sum_{k=2}^{\infty} p_{k+2} R_k R_{k-1} R_{k-2} \cdots R_2 \right) \psi_3 &= 0 \\
-(a + b) \psi_{-2} + b \psi_{-1} &= 0 \\
b \psi_{(0,0)} + a \psi_{-2} - (a + b) \psi_{-1} + b \psi_0 &= 0 \\
a \psi_{-1} - (a + b) \psi_0 + b \psi_1 &= 0 \\
a \psi_{(0,0)} + a \psi_0 - (a + b) \psi_1 + b \psi_2 &= 0 \\
a \psi_1 - \xi_2 \psi_2 + b \psi_3 &= 0 \\
\psi_{(0,0)} + \psi_{-2} + \psi_{-1} + \psi_0 + \psi_1 + \psi_2 + \left( 1 + \sum_{k=2}^{\infty} p_{k+2} R_k R_{k-1} R_{k-2} \cdots R_2 \right) \psi_3 &= 1 
\end{align*}
\]

**Remark 8.** Since the Markov process \( Q_E \) is not a birth-death process due to the key state \((0,0)\) with infinitely many rates of entry, there does not exist any explicit expression for the stationary probability vector. Despite of this, it still has the structure of a birth-death process just beyond the six boundary states: \((0,0), -2, -1, 0, 1, \) and \(2\). Based on this, we can apply the matrix-product solution to express and further numerically compute the stationary probability vector of the Markov process \( Q_E \).

In what follows we show how to use the stationary probability vector \( \Psi \) to study the expected mining profits of the blockchain selfish mining.

For the Markov process \( Q_E \), we denote by \( R_H \) and \( R_D \) the long-run average profits of the honest and dishonest mining pools, respectively. Then
\[
R_H = \psi_{-2} \mu (r_B + r_F) \Psi_H - \left( \psi_{0,0} + \sum_{i=-2}^{\infty} \psi_i \right) (c_E + c_A) (\beta - \gamma) \\
= \psi_{-2} \mu (r_B + r_F) \Psi_H - (c_E + c_A) (\beta - \gamma)
\]
Finally, we provide a long-run economic ratio of the dishonest mining pool over the honest mining pool by using per unit net mining rate as follows:

\[ \mathfrak{J} = \frac{1}{\tilde{\alpha} + \gamma} \rho_D \frac{\rho_D}{\beta - \gamma} \rho_H = \frac{\beta - \gamma}{\tilde{\alpha} + \gamma} \frac{\rho_D}{\beta - \gamma} \rho_H \]  

(30)

We can numerically indicate that the long-run ratio \( \mathfrak{J} \) increases as the jumping’s mining rate \( \gamma \) (or the efficiency-increased ratio \( \mathfrak{R} \)) increases.

9. No Network Latency

In this section, we further discuss a special case without network latency of the one-dimensional Markov model discussed in Section 8.

When the network latency is very short so that it can be ignored, we assume that \( \mu = +\infty \) or \( 1/\mu = 0 \). In this case, Figure 8 provides a simple state transition relation of the Markov process \( \{N(t) : t \geq 0\} \).

From Figure 8, it is easy to check that the infinitesimal generator of the Markov process \( \{N(t) : t \geq 0\} \) is given by

\[ \begin{align*}
\rho_1 &= \frac{\psi_{-2}}{\psi_{-2} + \sum_{k=2}^{\infty} \psi_k} \\
\rho_2 &= \frac{\sum_{k=2}^{\infty} \psi_k}{\psi_{-2} + \sum_{k=2}^{\infty} \psi_k}
\end{align*} \]
Obviously, the Markov process \( Q_{\mu=\infty} \) is irreducible and positive recurrent. Let \( \Psi^{(\infty)} = (\psi_{0,0}, \psi_{-1}, \psi_{0}, \psi_{1}, \psi_{2}, \ldots) \) be the stationary probability vector of the Markov process \( Q_{\mu=\infty} \).

Let the sequence of numbers
\[
\{R_{k}^{(\infty)} : k = 1, 2, 3, \ldots\}
\]
be the minimal positive solution to the system of nonlinear equations
\[
a (1 - p_{k+1}) - R_{k}^{(\infty)} (a + b) + R_{k+1}^{(\infty)} b = 0,
\]
where, \( k = 1, 2, 3, \ldots \).

Then from the system of linear equations
\[
\Psi^{(\infty)} Q_{\mu=\infty} = 0 \quad \text{and} \quad \Psi^{(\infty)} e = 1,
\]
we obtain that for \( k = 2, 3, 4, \ldots \),
\[
\Psi_{k}^{(\infty)} = \Psi_{1}^{(\infty)} R_{k-1}^{(\infty)} R_{k-2}^{(\infty)} \cdots R_{2}^{(\infty)} R_{1}^{(\infty)}
\]
and \( \Psi_{0,0}, \Psi_{-1}, \Psi_{0}, \Psi_{1}, \Psi_{2}, \ldots \) uniquely satisfy the following system of linear equations
\[
\begin{align*}
-\psi_{0,0}^{(\infty)} (a + b) + \psi_{-1}^{(\infty)} b + \psi_{1}^{(\infty)} (a p_{2} + \sum_{k=2}^{\infty} a p_{k+1} R_{k-1}^{(\infty)} R_{k-2}^{(\infty)} \cdots R_{2}^{(\infty)} R_{1}^{(\infty)}) &= 0, \\
\psi_{0,0}^{(\infty)} b - \psi_{-1}^{(\infty)} (a + b) + \psi_{0}^{(\infty)} b &= 0, \\
\psi_{-1}^{(\infty)} a - \psi_{0}^{(\infty)} (a + b) + \psi_{1}^{(\infty)} b &= 0, \\
\psi_{0,0}^{(\infty)} a + \psi_{-1}^{(\infty)} (a + b) + \psi_{0}^{(\infty)} R_{1}^{(\infty)} b &= 0, \\
\psi_{0,0}^{(\infty)} + \psi_{-1}^{(\infty)} + \psi_{0}^{(\infty)} + \psi_{1}^{(\infty)} (1 + \sum_{k=2}^{\infty} R_{k-1}^{(\infty)} R_{k-2}^{(\infty)} \cdots R_{2}^{(\infty)} R_{1}^{(\infty)}) &= 1
\end{align*}
\]
Similarly, we can solve the system of nonlinear equations (31) by means of the effective algorithms developed in Bright and Taylor (1995 1997).

For the the Markov process \( Q_{\mu=\infty} \), we denote by \( R_{H}^{(\infty)} \) and \( R_{D}^{(\infty)} \) the long-run average mining profits of the honest and dishonest mining pools, respectively. Then
\[
R_{H}^{(\infty)} = b \psi_{-1}^{(\infty)} \cdot 2 (r_{B} + r_{F}) \Psi_{H} - (c_{E} + c_{A} (\beta - \gamma)) R_{D}^{(\infty)}
\]
\[
= a \sum_{k=1}^{\infty} \psi_{k}^{(\infty)} \cdot p_{k+1} (k + 1) \times (r_{B} + r_{F}) \Psi_{D} - (\bar{a} + \bar{\gamma}) [c_{E} + c_{A} (1 + \Re)]
\]
where \( \Psi_{H} \) and \( \Psi_{D} \) are given in (24) and (26), respectively. Thus, the total long-run average mining profit of the blockchain system is given by
\[
R^{(\infty)} = \rho_{1} R_{H}^{(\infty)} + \rho_{2} R_{D}^{(\infty)}
\]
\[
= \rho_{1} b \psi_{-1}^{(\infty)} \cdot 2 (r_{B} + r_{F}) \Psi_{H} + \rho_{2} a \sum_{k=1}^{\infty} \psi_{k}^{(\infty)} \cdot p_{k+1} (k + 1)
\]
\[
\times (r_{B} + r_{F}) \Psi_{D} - \rho_{1} (c_{E} + c_{A} (\beta - \gamma)) - \rho_{2} (\bar{a} + \bar{\gamma}) [c_{E} + c_{A} (1 + \Re)]
\]
(32)
Figure 9 A Markov Process with $\mu = +\infty$ and $p_2 = 1$ is Related to Figure 1 of Eyal and Sirer (2018)

where

$$\tilde{\rho}_1 = \frac{\psi^{(\infty)}_{-1}}{\psi^{(\infty)}_{-1} + \sum_{k=1}^{\infty} \psi^{(\infty)}_k}$$

$$\tilde{\rho}_2 = \frac{\sum_{k=1}^{\infty} \psi^{(\infty)}_k}{\psi^{(\infty)}_{-1} + \sum_{k=1}^{\infty} \psi^{(\infty)}_k}$$

Figure 14 shows how $R^{(\infty)}_H$ and $R^{(\infty)}_D$ depend on the jumping’s mining rate $\gamma$ and the efficiency-increased ratio $\Re$.

In what follows we first discuss a special case with $p_2 = 1$. Then we provide a detailed comparison between our work and Eyal and Sirer (2018). It is necessary and useful for understanding the Markov chain method of Eyal and Sirer (2018).

When $p_2 = 1$, Figure 9 depicts the state transition relation of the Markov process $Q_{\mu=+\infty}$. It is easy to see that the Markov process $Q_{\mu=+\infty}$ can not arrive at state $k$ for $k = 2, 3, 4, \cdots$. By using Figure 9, the infinitesimal generator of the Markov process $Q_{\mu=+\infty}$ is given by

$$Q_{\mu=+\infty} = \begin{pmatrix}
-(a + b) & b & 0 & a \\
0 & -(a + b) & a & 0 \\
0 & b & -(a + b) & a \\
a & 0 & b & -(a + b)
\end{pmatrix}$$

Since the Markov process $Q_{\mu=+\infty}$ is irreducible and positive recurrent, its stationary probability vector is given by

$$\Psi^{(1)} = \begin{pmatrix}
\psi^{(1)}_{0,1} & \psi^{(1)}_{0,2} & \psi^{(1)}_{0,3} & \psi^{(1)}_{0,4} \\
\frac{a^2 + b^2}{2(a + b)^2} & \frac{b}{2(a + b)^2} & \frac{2}{a} & a \\
\frac{ab}{(a + b)^2} & \frac{a}{2(a + b)^2} & \frac{2}{a} & a \\
a & 0 & b & -(a + b)
\end{pmatrix}$$

Further, we obtain

$$R^{(1)} = \tilde{\rho}_1 R^{(1)}_H + \tilde{\rho}_2 R^{(1)}_D$$

$$= \tilde{\rho}_1 b \psi^{(1)}_{-1} \cdot 2(r_B + r_F) \Psi_H$$

$$+ \tilde{\rho}_2 a \psi^{(1)}_1 \cdot 2(r_B + r_F) \Psi_D$$

$$- \tilde{\rho}_1 (c_E + c_A) (B - \gamma)$$

$$- \tilde{\rho}_2 (c_E + \gamma) [c_E + c_A (1 + \Re)]$$

where $\Psi_H$ and $\Psi_D$ are given in (24) and (26),
respectively, and
\[
\bar{p}_1 = \frac{\psi_{-1}^{(1)}}{\psi_{-1}^{(1)} + \psi_{1}^{(1)}} = \frac{b}{a + b}
\]
\[
\bar{p}_2 = \frac{\psi_{1}^{(1)}}{\psi_{-1}^{(1)} + \psi_{1}^{(1)}} = \frac{a}{a + b}
\]

Now, we provide a long-run economic ratio of the dishonest mining pool over the honest mining pool by using per unit net mining rate. We define
\[
\mathcal{S}^{(1)} = \frac{1}{\bar{p} - \gamma} R_D = \frac{\beta - \gamma}{\alpha + \gamma} R_D
\]
\[
1 - \gamma \frac{1}{2} R_H
\]

We can numerically show that the long-run ratio \(\mathcal{S}^{(1)}\) increases as the efficiency-increased ratio \(\mathcal{R}\) or the jumping’s mining rate \(\gamma\) increases.

In the remainder of this section, by comparing our Figure 9 with Figure 1 of Eyal and Sirer (2018), we provide some useful remarks for understanding the Markov chain (Figure 1) and revenue analysis ((1) to (3)) of Eyal and Sirer (2018) as follows:

(a) Do there exist the two boundary states 0 and 0’?

In Figure 1 of Eyal and Sirer (2018), the two boundary states 0 and 0’ were introduced in a strange way which is not easy to understand. From Figure 9 in this paper, we establish a one-dimensional Markov process with state (0, 0), and it is observed that our state (0, 0) corresponds to state 0 of Eyal and Sirer (2018); while our state 0 becomes state 0’ of Eyal and Sirer (2018). However, we do not know in any way how to set up states 0 and 0’ of Eyal and Sirer (2018) by using the theory of Markov processes.

(b) Can Eyal and Sirer’s parameter \(\gamma\) play a necessary role in the revenue analysis?

In Figure 1 of Eyal and Sirer (2018), it is seen that 1 is the state transition probability from state 0’ to state 0. However, the probability 1 is decomposed into three parts (or separated path-probabilities): \(\alpha\), \(\gamma(1 -\alpha)\) and \((1 - \gamma)(1 -\alpha)\) by using \(1 = \alpha + \gamma(1 -\alpha) + (1 - \gamma)(1 -\alpha)\). Although Eyal and Sirer’s parameter \(\gamma\) is introduced, the Markov chain (Figure 1 of Eyal and Sirer (2018)) is independent of Eyal and Sirer’s parameter \(\gamma\). For example, the stationary probability vector has nothing to do with Eyal and Sirer’s parameter \(\gamma\). Furthermore, Eyal and Sirer’s parameter \(\gamma\) is used in the revenue analysis, see (1), (2) and (3) of Eyal and Sirer (2018). Clearly, their revenue computation is based on the law of total probability, in which from Eyal and Sirer (2018), its event probabilities are determined by using the Markov chain (Figure 1, and its event revenue is derived in (1) and (2) by means of discussing those reward cases from (a) to (h). Since the stationary probability vector of the Markov chain (Figure 1 of Eyal and Sirer (2018)) is independent of Eyal and Sirer’s parameter \(\gamma\), the selfish mining pool’s revenue function ((1) to (3) of Eyal and Sirer (2018)) can linearly depend on Eyal and Sirer’s parameter \(\gamma\). From Sections 8 and 9 of this paper, it is easy to see that the selfish mining pool’s revenue function ((3) of Eyal and Sirer (2018)) is not based on a rigorous mathematical calculation by using the Markov reward processes. Therefore, it is easy to see that Eyal and Sirer’s parameter \(\gamma\) will not play any role in the revenue analysis of the blockchain selfish mining under a rigorous mathematical setting.

(c) States \(k\) for \(k \geq 2\) can not exist in Figure 1 of Eyal and Sirer (2018).

From Figure 9 of this paper, it is easy to see that there do not exist states 2, 3, 4, ···, unless introducing a block-detained probability sequence \(\{p_k : k = 2, 3, 4, ···\}\). Therefore, there can not exist the Markov chain (Figure 1 of Eyal and Sirer (2018)) in the blockchain selfish mining. Based on this, we explain the realistic background that the Markov chain of Eyal and Sirer (2018) can be related to.

In brief, one of our main findings demonstrates that the Markov chain and the revenue
Li et al.: A New Theoretical Framework of Pyramid Markov Processes for Blockchain Selfish Mining

analysis in Eyal and Sirer (2018) should be confused. This paper provides some new insights on improving the Markov chain method of Eyal and Sirer (2018), and those works following Eyal and Sirer (2018) in the literature.

10. Numerical Examples

In this section, we use some numerical examples to verify our theoretical results, and indicate how performance measures of our more general model of blockchain selfish mining depend on some key parameters of blockchain.

Our numerical experiments are classified into three parts: (a) The orphan blocks, (b) the long-run average mining profits, and (c) one-dimensional Markov model without network latency.

Part one: The orphan blocks

To study the influence of orphan blocks, we take some parameters: \( \tilde{\alpha} = 10, \beta = 28, \) the block-pegging rate \( \mu = 3. \) Let the jumping’s mining rate \( \gamma \in [5, 8.5], \) and the efficiency-increased ratio \( \mathcal{R} \in 0.5, 0.7, 0.9. \)

For the average stationary lengths \( L_M \) and \( L_O \) of the main chain and the chain of orphan blocks, Figure 10 shows that \( L_M \) and \( L_O \) are not monotonous for \( \gamma \) or \( \mathcal{R}, \) and they start to decrease and then increase as \( \gamma \) or \( \mathcal{R} \) increases. In fact, such a theoretical analysis is given in the end of Section 5.

Now, we observe the ratio \( \phi \) of the two average stationary lengths, and the stationary ratio \( \psi \) of the two block removing and pegging rates. From Figure 11, it is seen that \( 0 < \phi < \psi < 1. \) Also, \( \phi \) and \( \psi \) begin to increase and then decrease, as \( \gamma \) or \( \mathcal{R} \) increases. Thus \( \phi \) and \( \psi \) are not monotonous in \( \gamma \) or \( \mathcal{R}. \)

Part two: The long-run average mining profits

To discuss the two long-run average mining profits given in Section 6, we take some parameters: \( \tilde{\alpha} = 10, \beta = 28, \) the block-pegging rate \( \mu = 3, r_B = 0.5, r_F = 0.5, c_E = 0.5, c_A = 0.5. \) Let the jumping’s mining rate \( \gamma \in [5, 8.5], \) and the efficiency-increased ratio \( \mathcal{R} = 0.5, 0.7, 0.9. \)

From the left half of Figure 12, it is seen that the long-run average mining profit \( \mathcal{R}_H \) of the honest mining pool decreases as the jumping’s mining rate \( \gamma \) increases, and it also decreases as the efficiency-increased ratio \( \mathcal{R} \) increases.

From the right half of Figure 12, it is observed that the long-run average mining profit \( \mathcal{R}_D \) of the dishonest mining pool increases as the jumping’s mining rate \( \gamma \) increases, and it also decreases as the efficiency-increased ratio \( \mathcal{R} \) increases.

From the left half of Figure 13, it is seen that the long-run economic ratio \( \mathcal{J} \) of the dishonest mining pool over the honest mining pool, and the long-run block-pegging rate ratio \( \mathcal{T} \) of the dishonest mining pool over the honest mining pool. The two long-run ratios are necessary and useful in the study of blockchain selfish mining.

From the left half of Figure 13, it is seen that...
Figure 11 $\phi$ and $\psi$ vs. $\gamma$ for Three Different Values of $\mathcal{R}$

Figure 12 $R_H$ and $R_D$ vs. $\gamma$ for Three Different $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$

Figure 13 $I$ and $\tau$ vs. $\gamma$ for Three Different $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$
that $I > 1$. Also, $I$ increases as the jumping’s mining rate $\gamma$ increases, and it also increases as the efficiency-increased ratio $\mathcal{R}$ increases.

From the right half of Figure 13, it is seen that $\tau > 1$. Moreover, $\tau$ increases as the jumping’s mining rate $\gamma$ increases, and it also increases as the efficiency-increased ratio $\mathcal{R}$ increases.

Corollary 6 shows that each of the two long-run ratios $I$ and $\tau$ increases, as the efficiency-increased ratio $\mathcal{R}$ increases. This is the same as our numerical experiment. However, we cannot prove that the two long-run ratios $I$ and $\tau$ are monotonically increasing in the jumping’s mining rate $\gamma$, because $I$ and $\tau$ have a complicated relation with $\gamma$. While it is interesting that our numerical experiment indicates such increasing monotonicity. That is, the bigger the selfish mining pool, the more mining profit each selfish miner is obtained. This is a key incentive such that the selfish mining pool becomes bigger and bigger.

Part three: One-dimensional Markov model without network latency

In the one-dimensional Markov model without network latency given in Section 9, we take some parameters: $\bar{\alpha} = 10, \beta = 28$, the block-pegging rate $\mu = 3, r_B = 15, r_F = 3, c_E \equiv 3, c_A \equiv 1$. Let the jumping’s mining rate $\gamma \in [0.5, 8]$, and the efficiency-increased ratio $\mathcal{R} = 0.5, 0.7, 0.9$.

We first analyze the long-run average mining profit $R_H^{(oo)}$ of the honest mining pool. From the left half of Figure 14, it is seen that $R_H^{(oo)}$ decreases as the jumping’s mining rate $\gamma$ increases, and it also decreases as the efficiency-increased ratio $\mathcal{R}$ increases.

Then we discuss the long-run average mining profit $R_D^{(oo)}$ of the dishonest mining pool. From the right half of Figure 14, it is seen that $R_D^{(oo)}$ increases as the jumping’s mining rate $\gamma$ increases, and it also increases as the efficiency-increased ratio $\mathcal{R}$ increases.

Note that the two long-run average mining profits $R_H$ and $R_H^{(oo)}$ have a similar monotonicity. So are $R_D$ and $R_D^{(oo)}$.

11. Concluding Remarks

In this paper, we provide a new theoretical framework of pyramid Markov processes in the study of blockchain selfish mining. We describe a more general model of blockchain selfish mining with both a two-block leading competitive criterion and a new economic incentive, both of which are expressed by means of our three key parameters: The block-detained probability sequence, the efficiency-increased ratio and the jumping’s mining rate. For such selfish mining, we establish a pyramid Markov process, and show that the pyramid Markov process is irreducible and positive recurrent, and its stationary probability vector is matrix-geometric with an explicitly representable rate matrix. Also, we use the stationary probability vector to analyze the influence of orphan blocks on the waste of computing resource. Furthermore, we set up a pyramid Markov reward process to investigate the long-run average mining profits of the honest and dishonest mining pools, respectively. Based on this, we can measure the mining efficiency of the dishonest mining pool through comparing with the honest mining pool. As a by-product, we build the one-dimensional Markov models when the system states are taken as the difference of block numbers on the two forked branches at the common tree root. For a special case without network latency, one of our main findings demonstrates that the Markov chain method in Eyal and Sirer (2018) should be incorrect, and thus the results by following the Markov chain method of Eyal and Sirer (2018) in the literature may not be true as well. Finally, we use some numerical examples to verify our theoretical results.

Note that the pyramid Markov (reward) processes open a new avenue to the study of blockchain selfish mining. We hope that
the methodology and results developed in this paper can shed light on the blockchain selfish mining and lead to a series of potentially promising research. We will continue our future research in the following directions:

– Considering the blockchain selfish mining under a changing difficulty level of PoW puzzle, which is described as a Markovian arrival process, or a transient periodic point process.

– Analyzing the case that a part of the main chain by the dishonest mining pool is pegged on the blockchain, while the other part of the main chain is left to support the next round of competition between two new block branches forked at a common tree root. Note that the other part of the main chain by the dishonest mining pool is possible to become the orphan blocks if its subsequent new branch is no longer ahead of that by the honest mining pool. The significant risk should be considered when the dishonest mining pool decides whether to peg the whole main chain or peg a part of it. It is an interesting topic to find an optimal policy how much of the main chain by the dishonest mining pool is pegged on the blockchain.

– Discussing the case with a $K$-block leading competitive criterion for $K = 1, 2, 3, 4, 5, \ldots$, and optimizing the positive integer $K$ to maximize the mining efficiency and/or the long-run average mining profit.

– Setting up a pyramid block-structure Markov process for Ethereum, and developing an effective algorithm for computing the matrix-analytic solution. Using the stationary probability vector to analyze the influence of orphan and uncle blocks on the waste of computing resource, and further investigate the long-run average mining profit of Ethereum.

– Developing the fluid and diffusion approximation for analyzing the blockchain selfish mining with multiple mining pools, providing the stability conditions of the multidimensional blockchain systems, and establishing the long-run average mining profits of the multiple mining pools.

– Further developing stochastic optimization and dynamic control of the blockchain selfish mining, for example, Markov decision processes, stochastic game, and evolutionary game in the study of blockchain selfish mining.

### Appendix A

This appendix provides three proofs for Theorem 1, Lemma 2 and Theorem 10, respectively. Our purpose is to increase the readability of the main paper.

**Proof of Theorem 1.** It is easy to see from Figure 4 that the pyramid Markov process $Q$ is irreducible, since for any two states $(n_1, n_2)$ and $(m_1, m_2)$, there must exist a state transition path such that the pyramid Markov process $Q$ can arrive at state $(m_1, m_2)$ from state $(n_1, n_2)$.
To prove the stability, it is a little bit complicated. Here, we use a double mean drift method for dealing with not only the sub-Markov process on each level but also the whole Markov process, e.g., see Chapter 3 of Li (2010).

We first prove the sub-Markov process only observed within Level \( k \) is positive recurrent for \( k = 0, 1, 2, \cdots \). From Figure 15, our analysis is to consider the following two different cases:

**Case one:** The sub-Markov process \( \{ J(t), I(t) = 0 : t \geq 0 \} \) observed within Level 0. Here, we only discuss the case with Level 0, while the case with Level 1 can be dealt with similarly.

It is easy to see from (a) of Figure 15 that the sub-Markov process \( \{ J(t), I(t) = 0 : t \geq 0 \} \) observed within Level 0 has the state space \( \Omega_{\text{Level } 0} = \{ \Delta, 1, 2, 3, \cdots \} \), and its infinitesimal generator is given by

\[
Q_{\text{Level } 0} = \begin{pmatrix}
-a & a & 0 & 0 & \cdots \\
0 & -a & a & 0 & \cdots \\
\mu p_2 & -[\mu p_2 + a(1-p_2)] & a(1-p_2) & 0 & \cdots \\
\mu p_3 & -[\mu p_3 + a(1-p_3)] & a(1-p_3) & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \ddots
\end{pmatrix}
\]
Here, we apply the mean drift method to consider the stability of the sub-Markov process $Q_{\text{Level } 0}$. To this end, it is observed on state $k$ that the mean drift rate of moving left to state $\Delta$ is given by $k \cdot \mu p_k$, where $k$ denotes a state transition from state $k$ to state $\Delta$; while the mean drift rate of moving right to state $k + 1$ is given by $1 \cdot a (1 - p_k)$, where $1$ denotes a state transition from state $k$ to state $k + 1$. Note that $\lim_{k \to \infty} p_k = 1$, it is clear that $\lim_{k \to \infty} k \cdot \mu p_k = \lim_{k \to \infty} k \mu = \infty$ and $\lim_{k \to \infty} 1 \cdot a (1 - p_k) = 0$. This shows that there exists a sufficient large positive integer $K$ such that for any $n > K$,

$$n \cdot \mu p_n > 1 \cdot a (1 - p_n)$$

that is, for any $n > K$, the mean drift rate of moving from state $n$ to state $\Delta$ is bigger than the mean drift rate of moving from state $n$ to state $n + 1$. Therefore, the sub-Markov process $Q_{\text{Level } 0}$ is stable.

Similarly, the sub-Markov process $Q_{\text{Level } 1}$ can be proved to be stable by means of (b) of Figure 15.

**Case two:** The sub-Markov process $\{J(t), I(t) = k : t \geq 0\}$ observed within Level $k$ for $k = 2, 3, 4, \cdots$. Here, we only discuss a case with Level $k$. It is easy to see from (c) of Figure 15 that the sub-Markov process $\{J(t), I(t) = k : t \geq 0\}$ observed within Level $k$ has the state space $\Omega_{\text{Level } k} = \{\Delta, k - 2, k - 1, k, k + 1, k + 2, k + 3, \cdots\}$, and its infinitesimal generator is given by

$$Q_{\text{Level } k} = \begin{pmatrix}
-a & a & 0 & 0 & 0 & 0 \\
\mu & -a & a & 0 & 0 & 0 \\
0 & -a & a & 0 & 0 & 0 \\
0 & 0 & -a & a & 0 & 0 \\
0 & 0 & 0 & -a & a & 0 \\
0 & 0 & 0 & 0 & -a & a \\
\mu p_2 & \mu p_3 & \mu p_4 & \mu p_5 & \eta_{2} & \delta_{2} \\
\eta_{1} & \eta_{3} & \eta_{4} & \eta_{5} & \cdots & \cdots \end{pmatrix}$$

where $\eta_k = \mu p_k + a (1 - p_k)$ and $\delta_k = a (1 - p_k)$ for $k = 2, 3, 4, \cdots$.

By using a similar analysis of the mean drift method to that in the sub-Markov process $Q_{\text{Level } 0}$, we can indicate that the sub-Markov process $Q_{\text{Level } k}$ is positive recurrent for $k = 2, 3, 4, \cdots$.

In what follows, we compute the mean drift rates of the pyramid Markov process $Q$ on Level $k$ for a large positive integer $k$.

Corresponding to the state space $\Omega_{\text{Level } k} = \{\Delta, k - 2, k - 1, k, k + 1, k + 2, k + 3, \cdots\}$, we write the stationary probability vector of the sub-Markov process $Q_{\text{Level } k}$ as

$$\theta = (\theta_{\Delta}, \theta_{-2}, \theta_{-1}, \theta_{0}, \theta_{1}, \theta_{2}, \theta_{3}, \cdots)$$

Through solving the system of linear equations $\theta Q_{\text{Level } k} = 0$ and $\theta e = 1$, we obtain that for $k = 2, 3, 4, \cdots$,

$$\theta_{k} = \frac{\delta_{k-1} \delta_{k-2} \delta_{k-3} \cdots \delta_{2a}}{\eta_{k} \eta_{k-1} \eta_{k-2} \cdots \eta_{3} \eta_{2}} \frac{a}{a + \mu} \theta_{\Delta}$$

and

$$\theta_{1} = \theta_{0} = \theta_{-1} = \theta_{-2} = \frac{a}{a + \mu} \theta_{\Delta}$$

and

$$\theta_{\Delta} = \frac{1}{1 + \left(4 + \frac{b_{k-1} b_{k-2} b_{k-3} \cdots b_{5a}}{\eta_{k} \eta_{k-1} \eta_{k-2} \cdots \eta_{3} \eta_{2}}\right) \frac{a}{a + \mu}}$$
Observing state $\Delta$ of the sub-Markov process $Q_{\text{Level } k}$, let $\tilde{\theta} = (\theta_{-2}, \theta_{-1}, \theta_0, \theta_1, \theta_2, \theta_3, \cdots)$. Then it is easy to see from Figure 4 that for the pyramid Markov process $Q$ on Level $k$, the mean drift rate of moving left to state $\Delta$ (see Level $\tilde{0}$) is given by
\[
(k + 1) \cdot \tilde{\theta} Be = (k + 1) \left(\mu \theta_{-2} + \mu \sum_{k=2}^{\infty} \theta_k p_k\right) > \mu \theta_{-2} (k + 1)
\]
while the mean drift rate of moving right to Level $k + 1$ is given by
\[
1 \cdot \tilde{\theta} Ce = b \sum_{k=-1}^{\infty} \theta_k < b
\]
Since
\[
\lim_{k \to \infty} (k + 1) \cdot \tilde{\theta} Be = \lim_{k \to \infty} \mu \theta_{-2} (k + 1) = \infty > b > 1 \cdot \tilde{\theta} Ce
\]
there exists a sufficient large positive integer $K$ such that for any $n > K$,
\[
(n + 1) \cdot \tilde{\theta} Be > 1 \cdot \tilde{\theta} Ce
\]
This shows that in the pyramid Markov process $Q$ on Level $n$ for any $n > K$, the mean drift rate of moving left to state $\Delta$ (see Level $\tilde{0}$) is bigger than the mean drift rate of moving right to Level $n + 1$.

Based on the above discussion, for the pyramid Markov process $Q$, we obtain two basic results: (i) The sub-Markov process $Q_{\text{Level } k}$ is stable for $k = 0, 1, 2, 3, 4, \cdots$ (ii) For the pyramid Markov process $Q$ on Level $n$ with any $n > K$, the mean drift rate of moving left to state $\Delta$ (see Level $\tilde{0}$) is bigger than the mean drift rate of moving right to Level $n + 1$. Therefore, it follows from Chapter 3 of Li (2010) that the pyramid Markov process $Q$ must be positive recurrent. This completes the proof. ■

**Proof of Lemma 2.** Note that
\[
(I - R)^{-1} = \left[I - C (-A)^{-1}\right]^{-1} = [-(A + C)]^{-1} (-A)
\]
where
\[
A + C = \begin{pmatrix}
-a & a & 0 \\
-b & -a & 0 \\
0 & 0 & -a
\end{pmatrix}
\]
which is the infinitesimal generator of an irreducible birth-death process, having
\[-(A + C) e = (\mu, 0, 0, 0, \mu p_2, \mu p_3, \mu p_4, \cdots)^\top\]
By using Section 3 in Chapter 1 of Li (2010), the LU-type RG-factorization is given by
\[
A + C = (I - R_{UL}) U (I - G_L)
\]
where
\[
U = \text{diag}(U_0, U_1, U_2, U_3, \cdots)
\]
This proof is completed.

Note that the real number sequence \{R_k : k \geq 0\} is the minimal positive solution to the system of nonlinear equations

\[
R_1 R_0 b - R_0 (a + b) + a = 0 \\
R_2 R_1 b - R_1 (a + b) + a = 0 \\
R_3 R_2 b - R_2 (a + b) + a = 0 \\
R_{k+1} R_k b - R_k \xi_{k-1} + a = 0, \quad k \geq 3
\]

while the real number sequence \{G_l : l \geq 1\} is the minimal positive solution to the system of nonlinear equations

\[
b - G_1 (a + b) + G_2 G_1 a = 0 \\
b - G_2 (a + b) + G_3 G_2 a = 0 \\
b - G_3 (a + b) + G_4 G_3 a = 0 \\
b - G_k \xi_{k-2} + G_{k+1} G_k a = 0, \quad k \geq 4
\]

Furthermore, the real number sequence \{U_k : k \geq 0\} is given by

\[
U_0 = - (a + \mu) + R_0 b = - (a + \mu) + a G_1 \\
U_1 = - (a + b) + R_1 b = - (a + b) + a G_2 \\
U_2 = - (a + b) + R_2 b = - (a + b) + a G_3 \\
U_3 = - (a + b) + R_3 b = - (a + b) + a G_4 \\
U_k = - \xi_{k-2} + R_k b = - \xi_{k-2} + a G_{k+1}, \quad k \geq 4
\]

Thus, we obtain

\[
(I - R)^{-1} = [- (A + C)]^{-1} (-A) \\
= (I - G_L)^{-1} \text{diag} (-U_0^{-1}, -U_1^{-1}, -U_2^{-1}, -U_3^{-1}, \cdots) (I - R_U)^{-1} (-A)
\]

Following Theorem 1.2 in Chapter 1 of Li (2010), some matrix computation indicates that each element of the matrix \((I - R)^{-1}\) is finite. This proof is completed.
we obtain

\[
(I - R_L)^{-1} = \begin{pmatrix}
    I & R_0 & R_0R & R_0R^2 & \cdots \\
    I & R & R^2 & \cdots \\
    I & R & \cdots \\
    \vdots & \vdots & \ddots
\end{pmatrix}
\]

\[
U_D^{-1} = \text{diag}(U_0^{-1}, U_1^{-1}, U_2^{-1}, U_3^{-1}, \cdots)
\]

\[
(I - G_L)^{-1} = \begin{pmatrix}
    I & G_1 & I \\
    G_2 & I \\
    G_3 & I \\
    \vdots & \ddots
\end{pmatrix}
\]

This gives

\[
T_{\text{max}}^{-1} = \begin{pmatrix}
    I & G_1 & I \\
    G_2 & I \\
    G_3 & I \\
    \vdots & \ddots
\end{pmatrix}
\]

\[
\begin{pmatrix}
    U_0^{-1} & U^{-1} \\
    U^{-1} & U^{-1} \\
    U^{-1} & U^{-1} \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
    I & R_0 & R_0R & R_0R^2 & R_0R^3 & \cdots \\
    I & R & R^2 & R^3 & \cdots \\
    I & R & R^2 & I & R & \cdots \\
    \vdots & \vdots & \ddots
\end{pmatrix}
\]

which can be computed easily. Therefore, the first generated-pegged time \( \lambda_H \) of the main chain by the honest mining pool is of phase type of infinite size with the irreducible representation \((\pi, T)\), because the Markov process \( T + T^0 \pi \) is irreducible. This completes the proof.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant No. 2020AAA0103801. Quanlin Li is supported by the National Natural Science Foundation of China under Grant Nos. 71671158 and 71932002, and the Beijing Social Science Foundation Research Base Project under Grant No. 19JDGLA004. Xiaole Wu is supported by the National Natural Science Foundation of China under Grant No. 72025102.

References

Albrecher H, Goffard P O (2020). On the profitability of selfish blockchain mining under consideration of ruin. HAL Id: hal-02649025: 1-29.

Awe K F, Malik Y, Zavarsky P, Jaafar F (2020). Validating BGP update using blockchain-based infrastructure. Decentralized Internet of Things: 151-165. Springer.

Babich V, Hilary G (2019). Distributed ledgers and operations: What operations researchers should know about blockchain technology. Manufacturing & Service Operations Management 22(2): 223-240.

Bahack L (2013). Theoretical bitcoin attacks with less than half of the computing power (draft). arXiv preprint arXiv 1312.7013: 1-18.

Bai Q, Zhou X, Wang X, Xu Y, Wang X, Kong Q (2019). A deep dive into blockchain selfish mining. International Conference on Communications: 1-6.

Bano S, Sonnino A, Al-Bassam M, Azouvi S, McCorry P, Meiklejohn S, Danezis G (2019). SoK: Consensus in the age of blockchains. Proceedings of the 1st ACM Conference on Advances in Financial Technologies: 183-198.

Bastiaan M (2015). Preventing the 51%-Attack: A stochastic analysis of two phase Proof-of-Work in bitcoin. The 22nd Twente Student Conference on IT: 1-10.

Bissias G, Levine B N (2020). Bobtail: A Proof-of-Work target that minimizes blockchain mining variance. Network and Distributed Systems Security Symposium: 1-16.

Bissias G, Levine B N, Ozisik A P, Andreescu G (2016). An analysis of attacks on blockchain consensus. arXiv preprint arXiv 1610.07985: 1-19.

Bright L, Taylor P G (1995). Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. Stochastic Models 11(3): 497-525.

Bright, L, Taylor, P G (1997). Equilibrium distributions for level-dependent quasibirth-and-death processes. Matrix-Analytic Methods in Stochastic Models, Marcel Dekker: 359-375.

Cachin C, Vukolic M (2017). Blockchain consensus protocols in the wild. The 31st International Symposium on Distributed Computing: 1-18.

Carlsten M, Kalodner H A, Weinberg S M Narayanan A (2016). On the instability of bitcoin without the block...
reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*: 154-167.

Chang D, Hasan M, Jain P (2019). Spy based analysis of selfish mining attack on multi-stage blockchain. *IACR Cryptology ePrint Archive*: 1-34.

Chang S Y, Park Y, Wuthier S, Chen C W (2019). Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners. *International Conference on Applied Cryptography and Network Security*: 241-258.

Chicarino V, Albuquerque C, Jesus E, Rocha A (2020). On the detection of selfish mining and stalker attacks in blockchain networks. *Annals of Telecommunications* 75(3): 143-152.

Choi S M, Park J Nguyen Q, Cronje A (2018). Fantom: A scalable framework for asynchronous distributed systems. *arXiv preprint arXiv:1810.10360*: 1-36.

Courtot N T, Bahack L (2014). On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718*: 1-15.

Davidson M, Diamond T (2020). On the profitability of selfish mining against multiple difficulty adjustment algorithms. *IACR Cryptology ePrint Archive*: 1-22.

Decker C, Wattenhofer R (2013). Information propagation in the bitcoin network. *IEEE Thirteenth International Conference on Peer-to-Peer Computing*: 1-10.

Dong X, Wu F, Faree A, Guo D, Shen Y, Ma J (2019). Self-Holding: A combined attack model using selfish mining with block withholding attack. *Computers & Security* 87 (101584): 1-11.

Eijkel D, Fehnker A (2019). A distributed blockchain model of selfish mining. *The 1st Workshop on Formal Methods for Blockchains*: 48-59.

Eyali, Gencer A E, Sirer E G, Van Renesse R (2016). Bitcoin-NG: A scalable blockchain protocol. *The 13th USENIX Symposium on Networked Systems Design and Implementation*: 45-59.

Eyali I, Sirer E G (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM* 61(7): 95-102.

Fan C, Ghaemi S, Khazaei H, Musilek P (2020). Performance evaluation of blockchain systems: A systematic survey. *IEEE Access*: 126927-126950.

Fullmer D, Morse A S (2018). Analysis of difficulty control in bitcoin and Proof-of-Work blockchains. *The IEEE Conference on Decision and Control*: 5988-5992.

Gao S, Li Z, Peng Z, Xiao B (2019). Power adjusting and bribery racing: Novel mining attacks in the bitcoin System. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*: 833-850.

Garay J, Kiayias A, Leonardos N (2015). The bitcoin backbone protocol: Analysis and applications. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*: 281-310.

Garay J, Kiayias A, Panagiotakos G (2017). Proofs-of-Work for blockchain protocols. *IACR Cryptology ePrint Archive*: 1-26.

Garay J, Kiayias A, Ostrovsky R M, Panagiotakos G, Zikas V (2020). Resource-restricted cryptography: Revisiting MPC bounds in the Proof-of-Work era. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*: 129-158.

Gervais A, Karame G O, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016). On the security and performance of Proof-of-Work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*: 3-16.

Gervais A, Ritzdorf H, Karame G O, Capkun S (2015). Tampering with the delivery of blocks and transactions in bitcoin. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*: 692-705.

Göbel J, Keeler H P, Krzesinski A E, Taylor P G (2016). Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Performance Evaluation* 104: 23-41.

Grünspan C, Pérez-Marco R (2018). On profitability of selfish mining. *arXiv preprint arXiv:1805.08281*: 1-20.

Grünspan C, Pérez-Marco R (2018). On profitability of stubborn mining. *arXiv preprint arXiv:1808.01041*: 1-16.

Grünspan C, Pérez-Marco R (2018). On profitability of trailing mining. *arXiv preprint arXiv:1811.09322*: 1-19.

Grünspan C, Pérez-Marco R (2019). Bitcoin selfish mining and dyck words. *arXiv preprint arXiv:1902.01513*: 1-5.

Guerrazou R, Wang J (2018). On the unfairness of blockchain. *The 6th International Conference on Networked Systems*: 36-50.

Gupta N (2020). Security and privacy issues of blockchain technology. *Advanced Applications of Blockchain Technology*: 207-226.

Hofmann E, Strewe U M, Bosia N (2017). *Supply Chain Finance and Blockchain Technology: The Case of Reverse Securitisation*. Springer.

Huang H, Kong W, Zhou S, Zheng Z, Guo S (2021). A survey of state-of-the-art on blockchains: Theories, models, and tools. *ACM Computing Surveys (CSUR)* 54(2): 1-42.

Jain P (2019). Revenue generation strategy through selfish mining focusing multiple mining pools. *Bachelor Thesis for the Degree of B Tech in Computer Science & Applied Mathematics, Indraprastha Institute of Information Technology*: 1-24, Delhi.

Javareone M A, Wright C S (2018). Modeling a double-spending detection system for the bitcoin network. *arXiv preprint arXiv:1809.07678*: 1-14.
Natoli C, Yu J, Gramoli V, Esteves-Verissimo P (2019). Deconstructing blockchains: A comprehensive survey on consensus, membership and structure. arXiv preprint arXiv:1908.08316: 1-43.

Nayak K, Kumar S, Miller A, Shi E (2016). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. IEEE European Symposium on Security and Privacy: 305-320.

Neuts M F (1981). Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. Johns Hopkins University Press.

Nguyen G T, Kim K (2018). A survey about consensus algorithms used in blockchain. Journal of Information Processing Systems 14(1): 101-128.

Niu J, Feng C (2019). Selfish mining in ethereum. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS): 1306-1316.

Ritz F, Zugenmaier A (2018). The impact of uncle rewards on selfish mining in ethereum. The 3rd IEEE European Symposium on Security and Privacy Workshops: 50-57.

F. Rosenfeld, M (2014). Analysis of hash-rate-based double spending, arXiv preprint arXiv:1402.2009: 1-13.

Saad M, Njilla L, Kamhoua C, Mohaisen A (2019). Countering selfish mining in blockchains. The 2019 International Conference on Computing Networking and Communications: 1-5.

Saad M, Spaulding J, Njilla L, Kamhoua C, Nyang D, Mohaisen A (2019). Overview of attack surfaces in Blockchain. Blockchain for Distributed System Security: 51-66.

Sapirshtein A, Sompolinsky Y, Zohar A (2016). Optimal selfish mining strategies in bitcoin. The 20th International Conference on Financial Cryptography and Data Security: 515-532.

Solat S, Potop-Butucaru M (2016a). Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. arXiv preprint arXiv:1605.02435: 1-11.

Solat S, Potop-Butucaru M (2016b). ZeroBlock: Preventing selfish mining in bitcoin. HAL Id: hal-01310088: 1-16.

Sompolinsky Y, Lewenberg Y, Zohar A (2016). SPECTRE: A fast and scalable cryptocurrency protocol. IACR Cryptology ePrint Archive (1159): 1-71.

Sompolinsky Y, Zohar A (2015). Secure high-rate transaction processing in bitcoin. The 19th International Conference on Financial Cryptography and Data Security: 507-527.

Stifter N, Schindler P, Judmayer A, Zamyatin A, Kern A, Weippl E (2019). Echoes of the past: Recovering blockchain metrics from merged mining. International Conference on Financial Cryptography and Data Security: 527-549.

Tromp J (2015). Cuckoo Cycle: A memory bound graph-theoretic Proof-of-Work. International Conference on Financial Cryptography and Data Security: 49-62.

Velner Y, Teutsch J, Luu L (2017). Smart contracts make bitcoin mining pools vulnerable. International Conference on Financial Cryptography and Data Security: 298-316.

Wang W, Hoang D T, Hu P, Xiong Z H, Niyato D, Wang P, Wen Y G, Dong I K (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access 7: 22328-22370.

Wang Z, Liu J, Wu Q, Zhang Y, Zhou Z (2019). An analytic evaluation for the impact of uncle blocks by selfish and stubborn mining in an imperfect ethereum network. Computers & Security 87(101581): 1-10.

Werner S M, Pritz P J, Zamyatin A, Knottenbelt W J (2019). Uncle traps: Harvesting rewards in a queue-based ethereum mining pool. Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools: 127-134.

Wright C S (2017). The fallacy of selfish mining in bitcoin: A mathematical critique. SSRN-id3004026: 1-18.

Wright C S, Savanah S (2017). The fallacy of the selfish miner in bitcoin: An economic critique. SSRN-id3151923: 1-6.

Wüst K (2016). Security of blockchain technologies. Master Thesis, Department of Computer Science, ETH Zürich.

Xiao Y, Zhang N, Lou W, Hou Y T (2019). A survey of distributed consensus protocols for blockchain networks. IEEE Communications Surveys & Tutorials 22(2): 1432-1465.

Zhang R (2015). Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin. IACR Cryptology ePrint Archive: 1-19.

Zhang R, Preneel B (2017). Publish or perish: A backward-compatible defense against selfish mining in bitcoin. Cryptographers’ Track at the RSA Conference. Springer, Cham.

Quanlin Li is a full professor at School of Economics and Management Sciences, Beijing University of Technology, Beijing, China. He received his Ph.D. degree at Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China. He has published an English monograph (Constructive Computation in Stochastic Models with Applications: The RG-Factorizations, Springer, 2010), has edited three proceedings of international conferences (2012, 2017, 2019) by Springer, and has published over 60 research papers in a variety of international journals, such as, Advances in Applied Probability, Queueing Systems, Stochastic Models, European Journal of Operational Research, Computer Networks, Performance Evaluation, Discrete Event Dynamic Systems, Computers & Operations Research, Computers & Mathematics with Applications, Annals of Operations Research, and International Journal of Production Economics. Now, his research interests include stochastic models, stochastic processes, the mean-field theory, stochastic process algebra, game theory,
queueing networks, computer networks, resource management in big networks, health care systems, mathematical models of Blockchain, sharing economy and so forth.

Yanxia Chang is a doctoral candidate at School of Economics and Management, Beijing University of Technology, Beijing, China. Now, her research interests include queueing networks, the mean-field theory, service systems, resource management of big networks, and blockchain systems.

Xiaole Wu is a Professor in the Department of Management Science at the School of Management, Fudan University. She received her PhD in management from Olin Business School, Washington University in St. Louis in 2011, and bachelor’s degree in industrial engineering from Tsinghua University in 2006. Her research interests include managing uncertainties and risks in supply chains, and the interfaces between operations and other disciplines. She has published in Management Science, Manufacturing and Service Operations Management, Production and Operations Management, Decision Sciences, Energy Economics, European Journal of Operational Research, etc. She serves as a Senior Editor at Production and Operations Management, and an Associate Editor at Service Science and Modern Supply Chain Research and Applications.

Guoqing Zhang is a professor in industrial engineering, the director of supply chain and logistics optimization research lab at Department of Mechanical, Automotive & Materials Engineering, University of Windsor. He received the Ph.D. degree in management sciences from City University of Hong Kong in 2000. His recent research interests include supply chain management and optimization, logistics, warehouse management, algorithms design and development, operations research, data driven optimization and intelligent decision support systems. He has published over 70 articles on those areas in journals such as Operations Research, European Journal of Operational Research, IIE Transactions, Transport Res B-Meth and E-Log, Omega, International Journal of Production Research, International Journal of Production Economics, and so on. Dr. Zhang has provided consulting to several world leading companies on automotive, energy, airline, and foods industries.