Blockchain Mining with Multiple Selfish Miners

Qianlan Bai, Yuedong Xu, Nianyi Liu, Xin Wang

***Abstract*—This paper studies a fundamental problem regarding the security of blockchain PoW consensus on how the existence of multiple misbehaving miners influences the profitability of selfish mining. Each selfish miner (or attacker interchangeably) maintains a private chain and makes it public opportunistically for acquiring more rewards incommensurate to his Hash power. We first establish a general Markov chain model to characterize the state transition of public and private chains for Basic Selfish Mining (BSM), and derive the stationary *profitable threshold* of Hash power in closed-form. It reduces from 25% for a single attacker to below 21.48% for two symmetric attackers theoretically, and further reduces to around 10% with eight symmetric attackers experimentally. We next explore the profitable threshold when one of the attackers performs strategic mining based on Partially Observable Markov Decision Process (POMDP) that only half of the attributes pertinent to a mining state are observable to him. An online algorithm is presented to compute the nearly optimal policy efficiently despite the large state space and high dimensional belief space. The strategic attacker mines selfishly and more agilely than BSM attacker when his Hash power is relatively high, and mines honestly otherwise, thus leading to a much lower profitable threshold. Last, we formulate a simple model of absolute mining revenue that yields an interesting observation: selfish mining is never profitable at the first difficulty adjustment period, but replying on the reimbursement of stationary selfish mining gains in the future periods. The delay till being profitable of an attacker increases with the decrease of his Hash power, making blockchain miners more cautious on performing selfish mining.**

***Index Terms*—Proof-of-Work, Selfish Mining, Profitability, Markov Chain, Partially Observable MDP.**

I. Introduction

Bitcoin has gained tremendous concerns as the first fully decentralized cryptocurrency since its advent in 2008. All historical transactions between Bitcoin clients are recorded in a global and public data structure known as *blockchain*. The security of Bitcoin-like blockchain is established by a chain of cryptographic Hash puzzles, addressed by a large-scale network of pseudonymous participants called *miners* [1]. Solving a Hash puzzle is deemed as a way to generate Proof-of-Work (PoW) of reaching global consensus. The PoW of Bitcoin demands intensive computations, thus consuming a lot of energy. Each miner competes for this "game", and is rewarded by cryptocurrencies (i.e. bitcoins) if he is the first acknowledged miner to find a valid block. When the population of miners is large, the aggregate Hash power is sufficiently high such that a malicious miner can hardly accumulate enough resource to perform double spending or 51% attack. The PoW consensus of Bitcoin has been widely deployed in public blockchains, serving as the cornerstone of current major cryptocurrencies.

The security of PoW is challenged by the trend of centralization of Hash power. Mining a Bitcoin block is random and it needs more than 10 years on average with a latest-generation ASIC chip [4]. Therefore, blockchain miners operate cooperatively to form pools that have a much larger chance of solving puzzles in a short time. By splitting the mining reward appropriately, they acquire a stable income rate. As a side effect, a small number of mining pools occupy a vast majority of global Hash power, placing blockchain systems at the risk of being overthrown by a gigantic pool or colluding pools. The conventional wisdom believes that PoW is secure as long as no mining pool (or miner interchangeably) controls 51% of total Hash power. However, a miner can choose to mine selfishly instead of conforming to the standard Bitcoin protocol.

Selfish mining refers to a class of block publishing policies in which a miner does not release his newly found block immediately, but forks a private chain unaware to others. At a future epoch, he will release his private blocks strategically to obsolete the current public chain or compete with it for the purpose of obtaining a higher share of valid blocks in the new public chain than his ratio of Hash power. In a word, a selfish miner does not want to destroy the blockchain PoW consensus, but to take advantage of it. The minimum ratio of Hash power that brings more rewards to a selfish miner than this ratio is conventionally called *profitable threshold*. Eyal and Sirer introduced the first selfish mining scheme (namely basic selfish mining, BSM) and pointed out that the profitable threshold of BSM is 25% of total Hash power [2]. Nayak et al. [9] proposed the stubborn mining that improves the revenue of the selfish miner by 13.94% compared to BSM. As the key trick of stubborn scheme, the selfish miner insists on forking if his private chain slightly lags behind the public chain. Sapirshtein et al. modeled the optimal selfish mining as a Markov Decision Process (MDP) that reduces the selfish miner's profitable threshold to 23.21% [7]. Tao et al. [20] introduced the semi-selfish mining attack based on hidden MDP with the control of fork rate. Grunspan and Perez-Marco proved rigorously using martingale theory that selfish mining is an attack on the difficulty adjustment algorithm of blockchain consensus [32]. Recently, a lot of efforts have been devoted to the compound attacks of selfish mining with block withholding attacks [14] [34], bribery attacks [15], eclipse attacks and double spending attacks [22].

Until very recently, the competition of multiple selfish miners came into view. Liu et al. presented the *publish-n* scheme for two selfish mining attackers and simulated their revenues as well as profitable thresholds [3]. Bai et al. modeled the mining race among two BSM miners and one honest miner

Q. Bai and X. Wang are with the School of Computer Science, Fudan University, Shanghai 200438, China.

Y. Xu and N. Liu are with the School of Information Science and Technology, Fudan University, Shanghai 200438, China.

as a Markov process [31]. Zhang et al. [18] simulated the profitable threshold in the presence of multiple selfish miners. Charlie et al. [19] proposed SquirRL, a framework for using deep reinforcement learning (DRL) to analyze selfish mining and block withholding attacks in blockchain systems. Their multi-agent setting is roughly equivalent to multiple selfish miners, and their focus is to train a highly performance DRL model to tackle the strategic mining with large state space and incomplete information. Previous experimental studies, though insightful, lack of theoretical understandings on the principle of competitive selfish mining. It is usually believed that modeling the profitable threshold with multiple selfish miners is hard, and the optimal mining may be intractable due to the above-mentioned challenges.

In this paper, we **theoretically** investigate the profitability of selfish mining with multiple attackers by asking a sequence of key progressive questions: *1) Will selfish mining become more easily profitable with multiple attackers than with a solo attacker? 2) How can we design a nearly optimal mining policy for an attacker despite the complex interactions among miners and the incomplete observations of the system state? 3) How long should an BSM attacker wait from the beginning of selfish mining until being profitable eventually?* Figuring out these questions will provide essential understandings to the security of blockchain PoW consensus.

We formulate a Markov chain model to compute the stationary *relative revenue* of BSM attackers for the first question. The relative revenue of an attacker is the percentage of his *valid* blocks in the public chain. Our model is very general in the sense that it can capture the cases with more than two attackers or allow an attacker to hide multiple blocks privately. In particular, the latter case may cause the complicated chain-reaction release in which the publishing action of one attacker triggers that of the other attacker.

Answering the second question is very challenging if a strategic attacker (Alice), a BSM attacker (Bob) and an honest miner (Henry) coexist in the system. Firstly, the interactions among three chains are more complicated. The state of the mining race is captured by a 10-tuple pertinent to the composition of all chains and the forking status, as opposed to a 3-tuple in the MDP-based optimal policy for a single attacker. The number of actions is larger and the state-action pairs can be 10^2 to 10^3 times larger. Second, Alice as the strategic miner cannot observe the complicated state information. In fact, she is merely aware of 5 attributes at each state related to her private chain and the public chain. We formulate a family of parameterized partially observable Markov decision process (POMDP) models to characterize Alice's strategic mining policy with a continuous belief of the current state. To tackle the large state space and the high-dimensional belief space, we adopt AEMS2, one of the fastest POMDP online algorithms, to compute the nearly optimal mining policy. A binary search method similar to [7] is used to find the maximum relative revenue among the family of POMDP models.

As for the third question, we build a simple model to compute attackers' *absolute revenue*, which is the average number of valid blocks received by each miner per unit of time. Since selfish mining is an attack on the difficulty adjustment algorithm (DAA), it is not profitable instantaneously even if the attacker's Hash power is above the profitable threshold. This model enables us to compute the number of DAA periods that lead to profitable selfish mining eventually.

Our major observations are summarized as below.

- **BSM.** The profitable threshold of Hash power is below 21.48% with two symmetric BSM attackers, as opposed to 25% with a single BSM attacker and 23.21% with a single optimal attacker. More blocks allowed to hold privately or more attackers will dramatically reduce this threshold. When the Hash powers of two attackers are asymmetric, the profitable threshold of one attacker will decrease first and then increase as the other attacker's Hash power increases (i.e. non-monotonic).
- **POMDP.** The POMDP mining policy brings more revenues to the strategic miner than BSM and honest mining, and approaches the performance of MDP mining policy with complete information. When the BSM attacker (Bob) has 34% Hash power, the other attacker's (Alice's) profitable threshold decreases from 29.44% to about 2% if she chooses POMDP rather than BSM. The designed online algorithm can rapidly and effectively compute the near optimal action under the current observable information.
- **Profitable Delay.** A BSM miner receives less absolute revenue than honest mining in the first difficulty adjustment period even if his Hash power is above the profitable threshold, and his gain is achieved in future periods. BSM is profitable after 51 rounds of difficulty adjustment (i.e. 714 days in Bitcoin) if the Hash power of two symmetric selfish miners is 22%. This delay decreases to 5 rounds (i.e. 70 days in Bitcoin) as their Hash power accrues to 33%, which is still quite long.

The remainder of this paper is organized as follows. Section II describes the background of selfish mining. Section III models the relative revenue of basic selfish mining with different attackers. Section IV proposed the POMDP-based mining policy and designed the efficient algorithm. The profitable time of basic selfish mining is modeled in Section V. Section VI validates the revenue model of BSM and the performance of the POMDP-based policy. Section VII introduces the related work, and Section VIII concludes our work.

II. System Model

In this section, we present the block-release procedure of blockchain mining in the presence of two adversarial miners. We further introduce the new features on tie-breaking and chain-reaction release.

A. System Description

Consider a blockchain system with two misbehaving attackers *Alice* and *Bob*, as well as an honest miner, *Henry*[1]. They compete to solve cryptographic puzzles to mine a valid block for the purpose of acquiring bitcoin-like tokens. The

[1] Multiple honest miners can be boiled down to a single miner for the sake of their linear additivity of Hash powers.

proof-of-work (PoW) consensus is adopted and the mining of blocks is stateless: the probability of discovering a block by a miner is proportional to his current Hash power, but inversely proportional to the current aggregate Hash power of the entire blockchain network. The blockchain system dynamically adjusts the difficulty of cryptographic puzzles such that new blocks are generated at a fixed average rate (e.g., one block per 10 minutes on average in Bitcoin). The miners maintain a globally-agreed ordered set of transactions via the adoption and the mining on the longest chain. The relative revenue of a miner is the fraction of blocks mined by him out of all the blocks in the longest chain. The reward of each valid block is normalized as one cryptographic coin.

For simplicity, we make the following assumptions consistent with the literature [2] [7]. The blockchain mining environment has that

- The total Hash power of the blockchain system is normalized as a unit. Then, the Hash power of a miner is represented as a fraction of the total;
- The block discovery time by a miner is exponentially distributed.

The honest miner Henry who finds a valid block will release it immediately. Alice (resp. Bob) may release her blocks strategically by forcing Henry into wasting his computation. When Alice and Bob are both selfish miners, the interaction between two private chains becomes more complicated because none of them know the other's state. In what follows, we capture all the different states that each miner may encounter.

Denote by α_1, α_2 and α_h the Hash powers of Alice, Bob and Henry respectively, i.e., $\alpha_1 + \alpha_2 + \alpha_h = 1$. Denote by γ_1 (resp. γ_2) the proportion that Henry's Hash power mines after Alice's (resp. Bob's) released chain in the tie-breaking between Alice (resp. Bob) and Henry. Denote by θ_1 and θ_2 the probabilities that honest miners choose to mine after Alice's and Bob's chains in the three-party tie-breaking, respectively. When the blockchain system creates a new block, it is mined by pool i with the probability α_i , $\forall i \in \{1, 2, h\}$, owing to the memorylessness of Hash computations.

B. Basic Selfish Mining Mode

Alice maintains a private chain, so does Bob, while Henry operates on the public chain. Alice and Bob are not aware of each other's role, even each other's presence. We suppose that all the miners work on the same public chain at the beginning where the starting point is expressed as "0". The length of the private chain is kept as private information by Alice and Bob, and the length of the public chain is observed by all of them. We consider the selfish mining method proposed by [2], and our analytical approach can be generalized to a variety of other methods.

The *mining procedure* consists of two cases as follows.

- *(Public-chain mining case)* Henry always mines after the public chain. Alice or Bob also mines on the public chain if it is longer than his private chain.
- *(Private-chain mining case)* Alice (resp. Bob) continues to mine on her (resp. his) private chain if she (resp. he) discovers a new block and the private chain is now longer than the public chain.

The *release procedure* is more complicated than the mining procedure. Henry broadcasts his mined block as soon as it is discovered, while Alice and Bob will decide whether to release their mined blocks depending on the length of the public chain.

- *(Forfeit case)* Alice (resp. Bob) abandons her (resp. his) private chain and conforms to mining after the public chain if the latter is longer. Henry also abandons his public chain if Alice or Bob publishes a longer chain.
- *(Risk-avoiding release case)* Alice (resp. Bob) releases her (resp. his) privately mined blocks to the public because of the fear of loss if the new block is mined by the others and the leading advantage of her private chain is no more than two blocks.
- *(Chain reaction case)* When Alice (resp. Bob) releases her (resp. his) blocks to the public chain and updates its length, the release of Bob's (resp. Alice's) private blocks is triggered immediately.

The chain reaction case is the combination of the forfeit and the risk-avoiding cases, whereas the existence of chain reaction complicates evolution of the public chain. Suppose that Alice publishes her private blocks to obsolete the current public chain. After the construction of the new public chain, Bob may release his private chain to forfeit it immediately.

C. Release procedure and tie-breaking Logics

The consensus on the public chain requires that it is the longest. A crucial question is how the public chain evolves when it is of the same length as Alice or Bob. In general, each miner works on his own chain, and the release behavior of Alice and Bob is triggered when Henry mines a new block. We hereby illustrate the evolution of private and public chains where A_k, B_k, and H_k denote that the k^{th} block belongs to Alice, Bob and Henry respectively. The blocks of private chains are in grey and those of public chains are in white.

Risk-avoiding release case. We show the risk-avoiding release of Alice's private chain in Fig. 1. Alice is only one block ahead of Henry after the latter mines a new block for the public chain. Because Alice fears of losing the competition, she publishes her private blocks, obsoleting Henry's public chain, so that both Alice and Henry mine on the new longest chain afterwards.

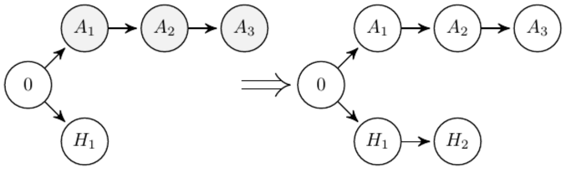

Fig. 1. Alice's risk-avoiding release and Henry's abandonment.

Tie-breaking resolving. If Alice's private chain is only one block ahead of Henry's, Henry may catch up with her. When it happens, Alice publishes her private blocks immediately to compete with Henry. Thus, two public chains of the same length exist in Fig. 2. Since only one public chain prevails, a tie-breaking rule needs to be taken into account. The first case is that the public chains of Alice and Henry have the same length, and Bob's private chain is 0. Hence, we only need to resolve the tie between Alice and Henry. All the miners are possible to mine after block A_1, while Bob and Henry may mine after H_1. There are five possibilities of extending the longest public chain, and the shorter one will be obsoleted. We omit the tie-breaking between Bob and Henry because this can be analyzed in the same way.

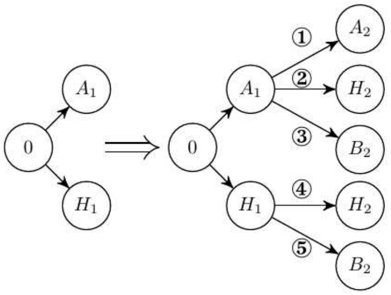

Fig. 2. Tie-breaking case of two public chains.

For the situation that each of Alice and Bob hides one private block, they will publish their private chains instantly after Henry finds a new block. As shown in Fig. 3, there exist three competing public chains. Alice will mine after A_1 and Bob will mine after B_1 for sure; Henry is not aware of which chain is maliciously forked so that he may mine on each public chain. There are also five possible situations. The risk-avoiding release, together with two tie-breaking solutions, constitutes all the dynamics of private and public chains.

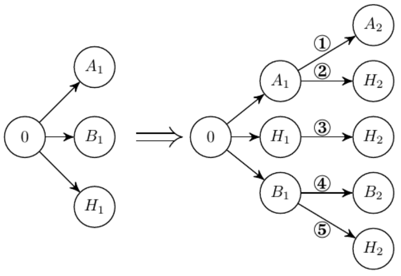

Fig. 3. Tie-breaking case of three public chains.

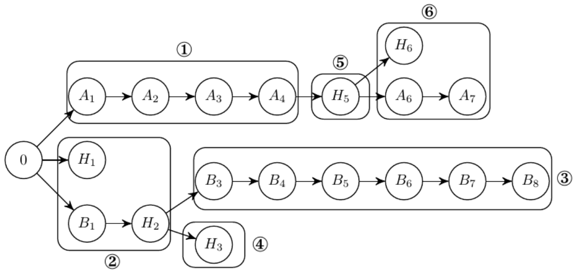

Fig. 4. Chain reaction case.

Chain reaction release. We next introduce the chain reaction release that complicates the evolution of the private and public chains. Note that the chain reaction release consists of a sequence of risk-avoiding releases and tie-breaking resolvings. Fig. 4 illustrates an example of how the chain reaction phenomenon is triggered. At stage 1, Alice's private chain contains four blocks, while the lengths of Bob's private chain and Henry's public chain are 0. After a tie-breaking resolving at stage 2, the longer public chain contains two blocks B_1 and H_2, and the shorter is orphaned. Bob constructs a new private chain starting from B_3 to B_8, while Henry continues to mine one block after H_2 at stage 4. From Alice's perspective, her private chain is merely one block ahead of the public chain. She releases her private blocks in order to avoid the risk of losing the race with Henry. The new public chain now starts from block A_4. Next, stages 5 and 6 constitute a new round of tie-breaking resolving between Alice and Henry, extending the public chain to block A_7. However, the release of A_7 triggers Bob to release all of his private blocks starting from B_3 to B_8. When retrospecting all the mining stages, we observe that the winning branch switches back and forth, making the analysis of selfish mining extremely complicated. To be noted, the chain reaction occurs only when the length of the private chain is greater than three.

III. Stationary Analysis of Basic Selfish Mining

In this section, we present Markov chain models to characterize the block-publishing dynamics with multiple selfish miners. The expected revenues of selfish miners are derived in explicit form.

A. Definition

The theme of our study is placed on the profitability of selfish mining so that the profitable measures should be clarified first. For notational simplicity, we only consider three miners: Alice, Bob and Henry.

Definition 1: (*Relative Revenue*) *Let R_a, R_b and R_h be the expected numbers of valid blocks mined by Alice, Bob and Henry in a mining round, respectively. The relative revenue of a miner, $\hat{R}_i$, is expressed as*

$$\hat{R}_i = \frac{R_i}{R_a + R_b + R_h}, \qquad i \in \{a, b, h\}.$$

It should be emphasized that the *valid* blocks are confirmed blocks in the longest chain. The profitability of selfish mining does not refer to the surplus that the block reward subtracts the cost of cryptographic computation. In fact, it is a contrastive measure to the honest mining that needs an objective index.

Definition 2: (*Profitability*) *The selfish or strategic mining performed by Alice (resp. Bob) is deemed profitable if the relative revenue is higher than the normalized Hash power, i.e. $\hat{R}_a > \alpha_1$ (resp. $\hat{R}_b > \alpha_2$).*

B. Stationary Analysis for Two Attackers

In order to analyze the profitability of selfish mining, we need to compute the fractions of blocks mined by the selfish miners in the public chain. We hereby formulate a discrete-time Markov chain model to characterize the dynamics of the public and private chains. We begin with the assumption that each selfish miner will release his two private blocks immediately after he has mined the second one (i.e. $N = 2$).

The underlying reasons are two-fold. Firstly, the simpler block-release process avoids the complicated representation of Markov states, thus allowing for more tractable mathematical modeling. Secondly, the bursty release of many valid blocks in a very short time usually indicates the existence of selfish mining attacks that can be easily detected. The selfish miners intend to acquire extra mining rewards other than destabilizing the authority of cryptocurrencies.

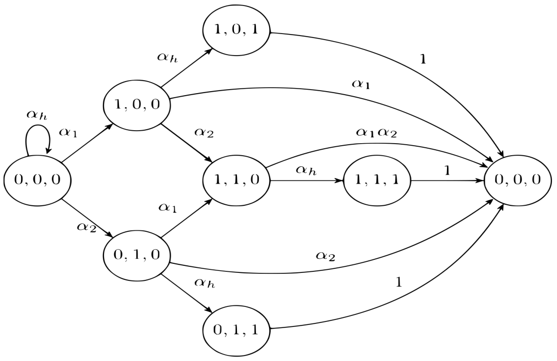

Fig. 5. Markov chain with less than two private blocks.

We define a *state* as a three-tuple consisting of the lengths of Alice's, Bob's and Henry's chains. Fig. 5 illustrates all the states, the state transition indicators and their transition probabilities. For instance, the transition from state $(0, 0, 0)$ to state $(1, 0, 0)$ means that Alice discovers a valid block with probability α_1 and forks the private chain. If the maximum length of any private chain is below 2, Alice and Bob can hide their private chains and continue the selfish mining. All the transitions to state $(0, 0, 0)$ means that the forked chains return to the unanimous public chain and a new round of selfish mining starts. Denote by $\mathbf{P}$ the state transition probability matrix and by $P_{ss'}$ the transition probability from state $s = (i, j, k)$ to $s' = (i', j', k')$. Let π_{ijk} be the stationary distribution of state (i, j, k). According to the *detailed balance equation* [5] [21]

$$\pi_s = \sum_{s'} \pi_{s'} P_{s's}. \tag{1}$$

Then, we can compute π_{000} as

$$\pi_{000} = (1+\alpha_1+\alpha_2+\alpha_1\alpha_h+2\alpha_1\alpha_2+\alpha_2\alpha_h+2\alpha_1\alpha_2\alpha_h)^{-1}, \tag{2}$$

and π_{ijk} at any other state $s = (i, j, k)$ in the same way.

The transitions to state $(0, 0, 0)$ manifest which miner is the final winner in the current round of selfish mining. Therefore, we can compute the expected revenues of Alice, Bob and Henry that are defined as R_a, R_b and R_h respectively. Facilitated by the stationary state distributions, we calculate them as below,

$$\begin{aligned} R_a = \pi_{000} \cdot [&2\alpha_1^2 (1 + \alpha_h) + (\alpha_2 + \alpha_h) \alpha_1\alpha_h\gamma_1 \\ &+ \alpha_1\alpha_2\alpha_h + 4\alpha_1^2\alpha_2 (1 + \alpha_h) + 2\alpha_1\alpha_2\alpha_h^2\theta_1]; \end{aligned} \tag{3}$$

$$\begin{aligned} R_b = \pi_{000} \cdot [&2\alpha_2^2 (1 + \alpha_h) + (\alpha_1 + \alpha_h) \alpha_h\alpha_2\gamma_2 \\ &+ \alpha_1\alpha_2\alpha_h + 4\alpha_2^2\alpha_1 (1 + \alpha_h) + 2\alpha_1\alpha_2\alpha_h^2\theta_2]; \end{aligned} \tag{4}$$

$$\begin{aligned} R_h = \pi_{000} \cdot [&\alpha_1\alpha_h^2 (2 - \gamma_1) + 2\alpha_1\alpha_2\alpha_h^2 (2-\theta_1 - \theta_2) \\ &+ \alpha_h + \alpha_2\alpha_h^2 (2 - \gamma_2) + \alpha_1\alpha_2\alpha_h(2 - \gamma_1 - \gamma_2)]. \end{aligned} \tag{5}$$

The average number of Henry's orphaned blocks in each attack round is calculated as:

$$\begin{aligned} O_h =&\pi_{000}[(\alpha_1+(1-\alpha_1)\gamma_1)\alpha_1\alpha_h + (\alpha_2+(1-\alpha_2)\gamma_2)\alpha_2\alpha_h \\ &+ (\alpha_1 + \alpha_2 + (\theta_1 + \theta_2)\alpha_h)2\alpha_1\alpha_2\alpha_h]. \end{aligned} \tag{6}$$

As a special case that the both selfish miners are homogeneous, i.e. $\alpha_1 = \alpha_2 = \alpha < 0.5$, $\gamma_1 = \gamma_2 = 0.5$ and $\theta_1 = \theta_2 = 1/3$, the expected revenues can be simplified as

$$\pi_{000} = (1 + 4\alpha - 4\alpha^3)^{-1};$$

$$R_a = R_b = \pi_{000} \cdot \tfrac{1}{6}\alpha(25\alpha + 2\alpha^2 + 3 - 32\alpha^3); \tag{7}$$

$$R_h = \pi_{000} \cdot [(1 - 2\alpha)(1 + 3\alpha - \tfrac{7}{3}\alpha^2 - \tfrac{16}{3}\alpha^3)]. \tag{8}$$

We can easily observe that the attackers' (resp. Henry's) expected revenues in Eq. (7) (resp. Eq. (8)) monotonically increase (resp. decrease) with regard to attackers' ratios of Hash power.

C. Scaling to Multiple Attackers

Although the profitable selfish mining demands a high Hash power, it is possible that multiple selfish miners can choose to opt in. The impact of more selfish miners on the profitability is obscure. The honest miner's share of Hash power decreases, and the competition among selfish miners becomes more fierce when the number of selfish miners increases beyond two. Therefore, we consider a general mire scenario with m $(m > 2)$ BSM miners.

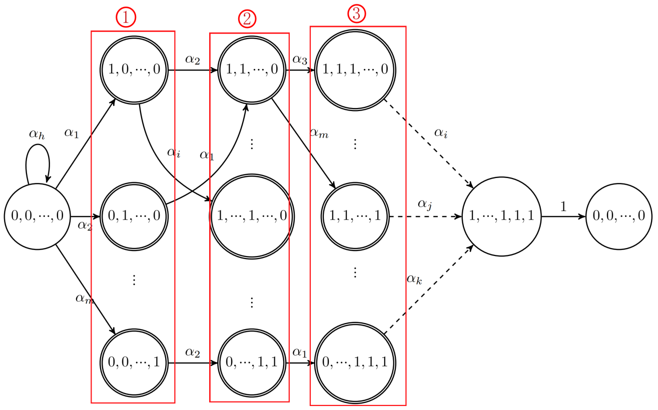

Fig. 6. Markov State Transition with m Attackers.

We model the dynamics of the public and private chains as a Markov process likewise. Fig. 6 illustrates all the states and their transition probabilities. Recall that the assumption of the maximum private chain length also holds. Each state is expressed as a m-tuple, i.e. $L = \{l_1, l_2, \cdots, l_m\}$ with $l_i \in \{0, 1\}$, which consists of the lengths of each attacker's private chain. The state with m zeros, denoted as L_0, indicates the start of the mining competition. The states with a single '1' are grouped together in which one and only one attacker has mined a valid block to build his private chain. Similarly, the states with k elements of '1' indicate that the private chains of k out of m attackers have one valid block. Formally, we denote by $\mathcal{L}$ the set of all states with the cardinality 2^m, and

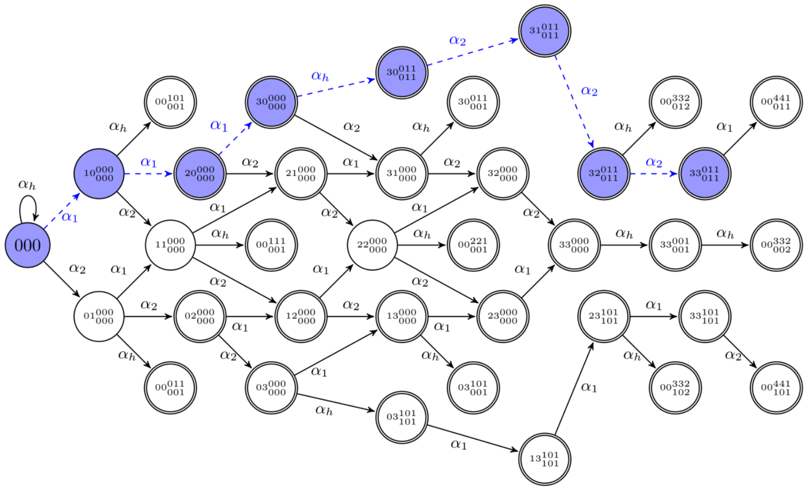

Fig. 7. State machine with $N = 4$.

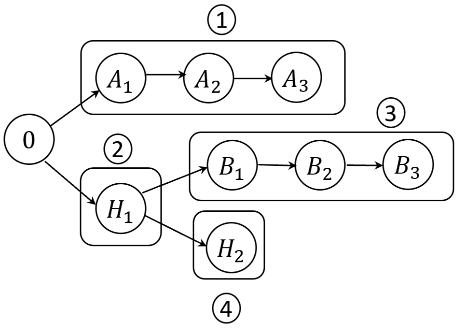

Fig. 8. A path sample with $N = 4$.

by $\mathcal{L}_k \subseteq \mathcal{L}$ the subset in which k selfish miners hold their private blocks.

The state transition probabilities are described as the following. The blockchain system can reside at the initial state with probability α_h, i.e., the block is mined by the honest miner. Denote by e_i the vector whose i^{th} element is 1 and all others are zero. When a valid block is discovered by attacker i, the blockchain system transits to a new state $(L + e_i)$ with probability α_i. At the state $L \in \mathcal{L}_k$, if the i^{th} attacker who has a private block finds a valid block again, the blockchain system returns to the initial state L_0 (in which the state is expressed as a double circle). Otherwise, it jumps to a state in $\mathcal{L}_{k+1}$ with the probability equivalent to the relative Hash power of the selfish miner. The state transition probabilities can be expressed as:

$$\begin{aligned}
&P_{LL'} = \alpha_i \ \ \forall L' = L + e_i \in \mathcal{L}_{k+1} \text{ and } L \in \mathcal{L}_k,\\
&P_{L_0L_0} = \alpha_h,\\
&P_{LL'} = \alpha_i + \alpha_h, P_{L'L_0} = 1, \ \ \forall L \in \mathcal{L}_k, L' \notin \mathcal{L}_{k+1}.
\end{aligned}$$

Using the detailed balance equations, we can compute the stationary distribution of each state. Specifically, the stationary probability at state S_0 is computed explicitly as

$$\pi_{L_0} = \big(1 + \sum_{k=1}^{m} \sum_{L \in \mathcal{L}_k} k! \prod_{j=1}^{m} (\alpha_i \cdot 1_{l_j=1})\big)^{-1}. \tag{9}$$

The stationary probability at state L is given by:

$$\pi_L = k! \cdot \pi_{L_0} \cdot \prod_{j=1}^{m} (\alpha_i \cdot 1_{l_j=1}), \forall L \in \mathcal{L}_k. \tag{10}$$

The revenues of all miners are computed based on the stationary state distribution and the particular transition paths to state L_0. If all miners have the same γ in tie-breaking cases, their revenues can be written as:

$$R_i = \sum_{k=1}^{m} \sum_{L \in \mathcal{L}_k} \begin{cases} \pi_L \cdot [\alpha_h(2\alpha_i + \frac{1 - \sum_{j \in L} \alpha_j \cdot 1_{l_j=1}}{k+1})], & l_i = 1;\\ \alpha_h \cdot \pi_L \cdot \alpha_i, & l_i = 0. \end{cases}$$

$$R_h = (\sum_{k=1}^{m} \sum_{L \in \mathcal{L}_k} \alpha_h \pi_L \alpha_h (\frac{2}{k+1} + \frac{k}{k+1})) + \pi_{L_0} \cdot \alpha_h. \tag{11}$$

D. Scaling to $N \geq 2$

We next model the revenues of all miners when each private chain can hide more than one block. Especially, as the maximum number of private blocks for one attacker is no less than four (i.e. N=4), the "chain reaction" occurs and the resulting finite state machine becomes very much complicated. A state should include not only the lengths of all chains, but also the interleaving of blocks on them. We confine our study to three miners: Alice, Bob and Henry, and investigate the case $N = 4$ without loss of generality. The main difficulty hindering the mathematical analysis is that Alice and Bob have different beliefs in the starting position of the current mining round. Besides, the blocks in the winning chain may belong to Henry and Alice/Bob so that we need to memorize them in order to compute their revenues. In contrast, both Alice and Bob always have the common starting position in the racing without chain reaction (i.e. state $(0, 0, 0)$ in Fig.5).

The state transitions with $N = 4$ are expressed in Fig. 7 where each state consists of eight parameters. The current mining round starts at the leftmost node where no miner has discovered a block. The notation h_1 indicates the distance between the starting position believed by Alice and the real starting position. Similarly, h_2 and h_3 indicate these distances of Bob and Henry. We record h_1, h_2 and h_3 because the blocks between the real and the believed starting positions influence which chain will prevail finally and how the revenues are calculated. The notation l_1 (resp. l_2) represents the number of unreleased blocks at Alice's (resp. Bob's) private chain. μ_1 denotes the number of Henry's block between the real starting position and the Alice-believed starting position. μ_2 and μ_3 are defined for Bob and Henry in the same way. Combined them together, we define an Markov state as $l_1 l_2{}^{h_1 h_2 h_3}_{\mu_1 \mu_2 \mu_3}$ that is also applicable to the situation with $N > 4$.

We hereby present a concrete example of state transition. The related states in Fig. 7 are marked in blue, and their transitions are illustrated in Fig. 8 separately. At stage ①, Alice has mined three blocks stealthily so that the system state jumps from 0 to 30^{000}_{000} through 10^{000}_{000} and 20^{000}_{000}. At stage ②, Henry mines a valid block H_1 and publishes it to the public chain immediately. The system state then jumps from 30^{000}_{000} to 30^{011}_{011}. Bob has mined three blocks after H_1 at stage ③ and the system state moves to 33^{011}_{011}. So far, neither Alice nor Bob will release their private blocks. At stage ④, Henry discovers a new block H_2 that triggers the release action of Alice. After Alice publishes all her blocks to obsolete Henry's chain, Bob finds that the public chain is catching up. As a consequence, Bob publishes all his blocks and wins the competition finally, i.e. the system state returning to the stating position. In this round, Bob receives three block rewards and Henry receives one block reward.

According to the transitive probability, the revenue for each miner can be represented as:

$$\begin{aligned}\pi_{000} = 1/(&1 + \alpha_1 + \alpha_2 + \alpha_1\alpha_3 + \alpha_1^2 + 2\alpha_2\alpha_1 + \alpha_2^2 + \alpha_1^3\\ &+ \alpha_2\alpha_3 + \alpha_1^3 + 3\alpha_2\alpha_1^2 + 2\alpha_1\alpha_2\alpha_3 + 3\alpha_1\alpha_2^2 + \alpha_2^3 + \alpha_1^3\alpha_3\\ &+ 4\alpha_1^3\alpha_2 + 6\alpha_1^2\alpha_2^2 + 4\alpha_1\alpha_2^3 + \alpha_2^3\alpha_3 + \alpha_1^3\alpha_2\alpha_3 + 10\alpha_1^3\alpha_2^2\\ &+ 6\alpha_1^2\alpha_2^2\alpha_3 + 10\alpha_1^2\alpha_2^3 + 4\alpha_1\alpha_2^3\alpha_3 + \alpha_1\alpha_2^3\alpha_3 + \alpha_1^3\alpha_2^2\alpha_3\\ &+ 20\alpha_1^3\alpha_2^3 + \alpha_1^3\alpha_2^3\alpha_3 + \alpha_1^3\alpha_2^2\alpha_3^2 + 20\alpha_1^3\alpha_2^3\alpha_3 + 4\alpha_1^3\alpha_2\alpha_3\\ &+ \alpha_1^4\alpha_2^3\alpha_3 + 20\alpha_1^3\alpha_2^3\alpha_3^2 + \alpha_1^2\alpha_2^3\alpha_3 + \alpha_1^3\alpha_2^3\alpha_3 + \alpha_1^2\alpha_2^3\alpha_3^2\\ &+ \alpha_1^3\alpha_2^4\alpha_3);\end{aligned} \tag{12}$$

$$\begin{aligned}R_1 =& \pi_{000} \cdot [4\alpha_1^4(1+\alpha_h) + 3\alpha_1^3\alpha_h^2 + 16\alpha_1^4\alpha_2 + 4\alpha_1^2\alpha_h\\ &+ 40\alpha_1^4\alpha_2^2(1+2\alpha_2) + \alpha_1\alpha_2\alpha_h(1+\gamma_1+2\theta_1\alpha_h)\\ &+ 10\alpha_1^2\alpha_2\alpha_h + 20\alpha_1^3\alpha_2\alpha_h(3\alpha_2+\alpha_1) + 15\alpha_1^3\alpha_2\alpha_h^2\\ &+ 4\alpha_1^4\alpha_2^2\alpha_h(1+\alpha_h) + 4\alpha_1^4\alpha_2^3\alpha_h^2(\beta_1+20)\\ &+ 5\alpha_1^5\alpha_2^3\alpha_h + 4\alpha_1^4\alpha_2^3\alpha_h(\alpha_2+21) + 3\alpha_1^3\alpha_2^4\alpha_h^2\beta_1\\ &+ \alpha_1\alpha_h^2\gamma_1 + 12\alpha_1^2\alpha_2^2\alpha_h^2\beta_1 + \alpha_1^2\alpha_2^2\alpha_h^3\beta_1(3\alpha_1+2\alpha_2)\\ &+ 6\alpha_1^3\alpha_2^3\alpha_h^2(10\alpha_h\beta_1+1)];\end{aligned} \tag{13}$$

$$\begin{aligned}R_2 =& \pi_{000}[4\alpha_2^4(1+\alpha_h) + 3\alpha_2^3\alpha_h^2 + 16\alpha_1\alpha_2^4 + 4\alpha_2^2\alpha_h\\ &+ 40\alpha_1^2\alpha_2^4(1+2\alpha_1) + \alpha_1\alpha_2\alpha_h(1+\gamma_2+2\theta_2\alpha_h)\\ &+ 10\alpha_1\alpha_2^2\alpha_h + 20\alpha_1\alpha_2^3\alpha_h(3\alpha_1+\alpha_2) + 15\alpha_1\alpha_2^3\alpha_h^2\\ &+ 4\alpha_1^2\alpha_2^4\alpha_h(1+\alpha_h) + 4\alpha_1^3\alpha_2^4\alpha_h^2(\beta_2+20)\\ &+ 5\alpha_1^3\alpha_2^5\alpha_h + 4\alpha_1^3\alpha_2^4\alpha_h(\alpha_1+21) + 3\alpha_1^4\alpha_2^3\alpha_h^2\beta_2\\ &+ \alpha_2\alpha_h^2\gamma_2 + 12\alpha_1^2\alpha_2^2\alpha_h^2\beta_2 + \alpha_1^2\alpha_2^2\alpha_h^3\beta_2(2\alpha_1+3\alpha_2)\\ &+ 6\alpha_1^3\alpha_2^3\alpha_h^2(10\alpha_h\beta_2+1)];\end{aligned} \tag{14}$$

$$\begin{aligned}R_h =& \pi_{000}[\alpha_1\alpha_h^2(2-\gamma_1) + \alpha_2\alpha_h^2(2-\gamma_2) + \alpha_1^2\alpha_2^3\alpha_h^3(2\beta_1{+}\beta_2)\\ &+ 2\alpha_1\alpha_2\alpha_h^2(2-\theta_1-\theta_2) + \alpha_1^2\alpha_2^2\alpha_h^2(6+4\alpha_1\alpha_2)\\ &+ \alpha_1^3\alpha_2^2\alpha_h^3(\beta_1+2\beta_2) + \alpha_1\alpha_2\alpha_h(2-\gamma_1-\gamma_2)\\ &+ \alpha_1^3\alpha_2^3\alpha_h(\alpha_1+\alpha_2) + \alpha_1^3\alpha_2^4\alpha_h^2(2\beta_1+\beta_2) + \alpha_h\\ &+ \alpha_1^4\alpha_2^3\alpha_h^2(\beta_1{+}2\beta_2){+}20\alpha_1^3\alpha_2^3\alpha_h^3{+}2\alpha_1^4\alpha_2^4\alpha_h];\end{aligned} \tag{15}$$

$$\beta_1 = \gamma_1/(\gamma_1+\gamma_2) \quad \beta_2 = \gamma_2/(\gamma_1+\gamma_2). \tag{16}$$

The cases with $N > 4$ can be analyzed in the same way. We validate in our experiments that the relative revenue tends to converge when $N \geq 4$. If N is too large, the repeated chain-reactions will occur, which aggravates the system instability and increases the possibility of detecting selfish mining attacks.

IV. Optimal strategy under multiple attackers

The basic selfish mining policy restricts the choices of withholding and releasing blocks. In this section, we present the *optimal* selfish mining strategy (*POMDP-based mining policy*) for two attackers when one of them chooses the basic selfish mining and the other chooses to be strategic.

A. OPT *for an Upper Bound of Revenue*

The limitations of basic selfish mining are intuitive. An attacker is "conservative" to adopt the public chain when his private chain slightly lags behind it, and is "less wise" to override the public chain when it is catching up. The optimal selfish mining problem with a single attacker was raised in [7] [22] that modeled the mining race as a Markov decision process. Under the optimal policy, the attacker does not adopt the public chain if his private chain is slightly shorter than the public chain; he may intentionally release some blocks to cause the forking if his private chain is ahead of the public chain by several blocks. This strategic selfish mining policy lowers the profitable threshold of Hash power.

The optimal selfish mining (OPT) in the presence of two attackers (Alice as the *strategic attacker* and Bob as the *basic attacker*) is far more challenging than that with a single attacker. First, the state and the state transition are greatly augmented. Alice has to incorporate the status of multiple chains in the state other than merely the lengths of the chains. The racing of three chains also causes complicated state transitions. Second, Alice cannot acquire the information regarding Bob's private chain. In other words, the state is partially observable to Alice. To tackle these difficulties, we begin with the assumption that Alice has the full information of the private chain of Bob. The optimal policy of Alice can be solved based on an MDP model, and the corresponding revenue will be used as the upper bound of revenue when the private chain of Bob is unknown. Meanwhile, the MDP model offers the principle of designing optimal policy with partially observable states.

1) Main components: We formulate an MDP model for the strategic attacker as the four-tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} >$ where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ corresponds to the transition matrix, and $\mathcal{R}$ corresponds to the reward matrix.

State: The state space $\mathcal{S}$ is defined as a 10-tuple in the form $< loc, fork, l_1, l_2, h_1, h_2, h_3, u_1, u_2, u_3 >$. The attribute $loc \in \{1, 2, 3\}$ indicates the branch that Henry is working on. If $loc = 1$ (resp. $loc = 2$), Henry is mining on the public chain that also contains Alice's (resp. Bob's) blocks at the current mining round. If $loc = 3$, the longest public chain contains only Henry's blocks. Note that Alice's and Bob's blocks are mutually exclusive on the public chain at the same mining round because one attacker will not accept the blocks of the other before this attack round ends.

The attribute $fork$ obtains six values, dubbed as $\{ir, r, f_{12}, f_{13}, f_{23}, f_{123}\}$, where r represents that Alice can release blocks to compete with the current public chain when $h_3 > 0$ while ir represents that she can not. If multiple miners are competing on the public chain, $fork$ takes four values $\{f_{12}, f_{13}, f_{23}, f_{123}\}$, indicating that (Alice, Bob), (Alice, Henry), (Bob, Henry) and (Alice, Bob, Henry) are in the competition respectively. The other attributes such l_1, l_2, h_1, h_2, h_3, μ_1, μ_2 and μ_3 have the same meanings as those in the aforementioned Markov chain. Similar to [7], we limit the lengths of private and public chains in a mining round so as to confine the size of state space, i.e. $l_1 \leq N$ and $h_3 \leq h_{3,max}$.

Action: An action is the number of blocks that Alice publishes under a particular state. We define Alice's action space $\mathcal{A}$ as $\mathcal{A} = \{adopt, 0, 1, \cdots, l_1\}$ in which $adopt$ means that Alice gives up her private chain, 0 means that Alice chooses to *wait*, and l_1 is the current number of blocks hold by Alice privately. The action taken by Alice has certain reasonable restrictions: if l_1 reaches N, Alice must release at least one block; if h_3 reaches $h_{3,max}$, Alice either choose "adopt" or to release no less than $(h_3 - h_1)$ blocks to end this mining round.

State Transition: We define the state transition function as $\Pr(s'|s, a \in \mathcal{A})$, the probability that the state s jumps to state s' under the action a. The state transition is triggered by the mining of a new block, and is determined by which miner discovers it. All the transitions are summarized in Table III.

Reward Function: The purpose of "optimal" mining is to acquire a larger share of confirmed blocks on the public chain, or to purse a larger relative revenue in other word. Recall that the relative revenue of a miner is the fraction of his blocks on the public chain for a long period. Obviously, the relative revenue cannot be measured under each state-action pair, and cannot be taken as the corresponding immediate reward. Sapirshtein et al. [7] transform the (long-term) relative revenue into the family of (one-shot) absolute revenues parameterized by the weight $\rho \in [0, 1]$.

This ingenious transformation operates as the following. Define a transformation function $w_\rho : \mathbb{N}^3 \rightarrow \mathbb{R}$ related to Alice's instantaneous reward:

$$w_\rho^i(r_1^i, r_2^i, r_h^i) = (1-\rho) \cdot r_1^i - \rho \cdot (r_2^i + r_h^i), \qquad (17)$$

where r_1^i, r_2^i and r_h^i represent the instantaneous rewards of Alice, Bob and Henry at step i (analogous to time t in classical MDP). We reformulate the original MDP model as $\mathcal{M}_\rho =< \mathcal{S}, \mathcal{A}, \mathcal{P}, w_\rho(r_1, r_2, r_h) >$. The underlying reason of such transformation is that instead of maximizing the relative revenue, we choose to maximize the expected fictitious reward $w_\rho(r_1, r_2, r_h)$. For any admissible policy π, the mean reward denoted by v_ρ^π is characterized as:

$$v_\rho^\pi = \mathbb{E}[\lim_{\xi \to \infty} \frac{1}{\xi} \sum_{i=1}^{\xi} w_\rho(r_1^i(\pi), r_2^i(\pi), r_h^i(\pi))], \qquad (18)$$

where ξ is the total number of state transition steps. The optimal revenue v_ρ^* is given by

$$v_\rho^* = \max_\pi \{v_\rho^\pi\}. \qquad (19)$$

The equivalence between two MDPs $\mathcal{M}$ and $\mathcal{M}_\rho$ is indirect. Sapirshtein et al. [7] presents two propositions to guarantee their equivalence for a single selfish miner.

- If $v_\rho^* = 0$ for some $\rho \in [0, 1]$, then any policy π^* obtaining this value also maximizes the relative revenue, and the relative revenue equals to ρ.
- v_ρ^* is monotonically decreasing in ρ.

The above propositions tell us that by searching an appropriate ρ that yields the mean reward of $\mathcal{M}_\rho$ to be 0, we can obtain the optimal relative revenue of Alice in $\mathcal{M}$. We generalize this idea to the MDP with multiple selfish miners. In addition, the maximum number of steps, ξ, is truncated to avoid excessive computations by tolerating a very gentle loss in the optimal mean reward v_ρ^π.

2) Algorithm: Owing to the monotonicity of v_ρ^* to ρ, a binary search of $\rho \in [0, 1]$ is adequate. For a given ρ, we utilize the *value iteration* method to solve the optimal policy π_ρ^* as [7]. Compared to *policy iteration*, the advantage of *value iteration* is its fast convergence rate especially in large-scale MDPs [12] [13].

B. POMDP-based policy for Optimal Mining

The OPT framework provides important insights on the optimal mining policy in the presence of two attackers, yet its real world deployment is unrealistic. The strategic miner Alice is assumed to know precisely the system state, while in reality Bob's private chain is not observable to Alice, and the blocks on the public chain released by Bob and Henry cannot be differentiated because of their anonymity. In light of the incomplete state information, we reformulate the optimal mining as a Partially Observable Markov Decision Process (POMDP). Before diving into details, we enumerate three key challenges:

- which subset of information is non-observable to Alice;
- how the mining event-driven MDP model is generalized to the POMDP model;
- how the POMDP-based optimal mining policy can be computed efficiently.

1) Main components: The POMDP model is expressed as a six-tuple $\mathcal{M}_{PO} :=< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{Z} >$, where $\mathcal{O}$ is Alice's observation space, $\mathcal{Z}(\cdot)$ is the observation function, and the remaining components inherit the same meanings as their counterparts in MDP.

Observable Information: In the partially observable environment, Alice cannot obtain all the attributes in $\mathcal{S}$. The length of Bob's private chain l_2 cannot be observed absolutely. The attribute loc is non-observable either because Alice is unaware of whether Henry is mining on his own chain or Bob's chain. h_2, μ_2 and μ_3 are non-observable for that the anonymity of mining covers up the owners of the blocks on the public chain. The attribute $fork$ is observable because ir and r are pertinent to Alice's private chain, and the values $f_* \in \{f_{12}, f_{13}, f_{23}, f_{123}\}$ is obtained by counting the number of focks in competition. l_1, h_1 and μ_1 are known for sure; h_3 is actually the length of the longest public chain. In summary, the observation space is represented as $\mathcal{O} :=<$

$fork, l_1, h_1, h_3, \mu_1 > \subset \mathcal{S}$. Given the same observation, Alice is likely to be in many possible states.

State Transition: The state transition function is also denoted as $\Pr(s'|s, a \in A)$. Though taking the similar form, $\mathcal{M}_{PO}$ possesses a different state transition logic from $\mathcal{M}$. In $\mathcal{M}$, an action is triggered by the discovery of a new block, and the state transition follows. In $\mathcal{M}_{PO}$, the pure *event-driven* state transition will restrict Alice from participating in the fork competition. For instance, if Bob hides one block, Alice should publish her private block while there is no observable event to trigger a fork competition. On the contrary, if Bob does not have any private block while Alice believes that the length of Bob's private chain is 1, Alice may publish one or two blocks unnecessarily. Therefore, one can see that a time-slotted plus event-driven POMDP model is appropriate to handle the intricate optimal mining problem.

Due to the memoryless Hash computation [8], the block arrival process is actually a stationary stochastic process. If we slice this stochastic process equally with a slot duration Δt, the number of mined blocks in every slot has the same distribution. By choosing a relatively small Δt, we suppose that at most one block is mined in each slot (the chance of mining two or more blocks is rarer by orders of magnitude). Denote by p the probability of generating a block in one time slot. The probabilities that Alice, Bob and Henry generate it are $\alpha_1 p$, $\alpha_2 p$ and $\alpha_h p$ respectively. Alice can estimate the Hash power α_2 and α_h through mining honestly for a certain period. It is worth highlighting that our POMDP model makes the optimal mining with partially observable states feasible, and is in accordance with realistic blockchain systems. All the transitions are summarized in Table IV.

Observation function: Define $\mathcal{Z} := \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{O})$ as the observation function that specifies the relationship between system states and observations. Here, $z(s, a, o)$ is the probability that observation o will be reached after Alice performs action a and lands in state s:

$$z_{t+1}(s, a, o) = \Pr(o_{t+1} = o | s_{t+1} = s, a_t = a). \tag{20}$$

In $\mathcal{M}_{PO}$, the observation of a state is certain, i.e,

$$\begin{aligned} & s =< loc, fork, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3 >, \\ & o =< fork, l_1, h_1, h_3, \mu_1 >, \\ & z_{t+1}(s, a, o) = 1. \end{aligned} \tag{21}$$

Belief: Our POMDP model is pertinent to a belief b which is a probability distribution over all the possible states. Intuitively, Alice makes a guess on the current state iteratively. The belief on a particular state s at time t is given by:

$$b(s) = \Pr(s_t = s | z_t, a_{t-1}, z_{t-1}, \cdots, a_0, b_0). \tag{22}$$

The updated belief state $b'(s')$ is calculated whenever the action a is taken and the observation o is perceived.

$$b'(s') \equiv \Pr(s'|a, o, b) = \frac{\Pr(s', a, o, b)}{\Pr(a, o, b)} \tag{23}$$

$$= \frac{\Pr(o|s', a) \sum_s \Pr(s'|a, s) b(s)}{\Pr(o|a, b)}, \tag{24}$$

where $\sum_{s \in \mathcal{S}} b(s) = 1$ and $\Pr(o|a, b)$ is a normalization factor given by

$$\Pr(o|a, b) = \sum_{s'} \Pr(o|a, s') \sum_{s} \Pr(s'|a, s) b(s). \tag{25}$$

Reward function: We use $r_1^i(s, a), r_2^i(s, a)$ and $r_h^i(s, a)$ to denote the instantaneous rewards of Alice, Bob and Henry at step i when Alice takes action a at state s. Due to the uncertainty of system state, the expected reward functions $r_1^i(b, a), r_2^i(b, a)$ and $r_h^i(b, a)$ are constructed on the belief of instantaneous rewards, $\forall a \in \mathcal{A}$,

$$r_1^i(b, a) = \sum_{s \in \mathcal{S}} b_i(s) r_1^i(s, a), \tag{26}$$

$$r_2^i(b, a) = \sum_{s \in \mathcal{S}} b_i(s) r_2^i(s, a), \tag{27}$$

$$r_h^i(b, a) = \sum_{s \in \mathcal{S}} b_i(s) r_h^i(s, a). \tag{28}$$

Alice's purpose is to maximize her relative reward that can be transformed into an alternative expression of her absolute reward at each time slot. Similar to the transformation in the MDP model, we replace the relative reward by the absolute reward $w_\rho^i(r_1^i(b_i, a), r_2^i(b_i, a), r_h^i(b_i, a))$ parameterized by ρ using Eq. (17).

We next formulate the optimal mining as a finite-horizon average reward POMDP problem as [7] and [22]. The expected average value function is defined as

$$v_\rho^\pi = \mathbb{E}[\lim_{\xi \to \infty} \frac{1}{\xi} \sum_{i=1}^{\xi} w_\rho(r_1^i(b_i, \pi), r_2^i(b_i, \pi), r_h^i(b_i, \pi))].$$

The optimal policy π^* is a set of decision rules depending on belief-state pairs:

$$\pi^* = \arg\max_{\pi \in \mathcal{A}} \{v_\rho^\pi\}. \tag{29}$$

The parameter ρ that solves $v_\rho^* = 0$ is the relative revenue of Alice under the POMDP model. In practice, the sum of revenues over ξ is truncated by a sufficiently large number ξ_0. Given a precision threshold ϵ, ξ_0 needs to satisfy:

$$|v_\rho^\pi - \mathbb{E}[\frac{1}{\xi_0} \sum_{i=1}^{\xi_0} w_\rho(r_1^i(\pi), r_2^i(\pi), r_h^i(\pi))]| \leq \epsilon. \tag{30}$$

C. Algorithm

A POMDP is essentially an expanded MDP defined on belief space, and classical value iteration methods can be adopted to solve the optimal policy offline. However, the belief space is a high-dimensional continuous space that needs to be segmented into a huge number of belief states. An offline POMDP algorithm will compute the optimal actions at every belief state. Considering a large-scale POMDP like ours, the offline computation is time-consuming because of generating the rewards, updating the beliefs and constructing the optimal policy at each belief. This is in contrast to MDP that only has an exact belief state. For efficiency considerations, we propose using the online POMDP algorithm that explores the future belief states reachable from the current belief state. The policy construction time is often substantially shorter. Furthermore, three important properties can be used to reduce the time of searching for ρ.

***Lemma** 1: Under the same parameter setting, the optimal result of $\mathcal{M}$ is the upper bound of $\mathcal{M}_{PO}$.*

Proof: Among the approximation algorithms of the POMDP problem, the MDP approximation consists in approximating the value function of the POMDP by the value function of its underlying MDP [36]. This value function is an upper bound on the value function of the POMDP [28]. ■

***Lemma** 2: The revenue obtained under the optimal policy based on any ρ is the lower bound of the actual optimal revenue for $\mathcal{M}_{PO}$.*

Proof: In the online algorithm, the policy construction steps and the execution steps are interleaved with one another. Hence, we record the number of blocks that each miner obtains at every time step, i.e, (r_1^i, r_2^i, r_h^i). After enough time steps, we can compute the relative revenue under the current ρ even though it is not optimal. Therefore, the revenue of the current optimal policy obtained under the current ρ is the lower bound of the actual optimal revenue. ■

***Lemma** 3: There exists an optimal stationary deterministic policy for $\mathcal{M}_{PO}$ model.*

Proof: The states, actions and beliefs are countable because the maximum lengths of private and public chains are limited. Since a POMDP problem can be regarded as a belief MDP, the existence of an optimal stationary policy for our POMDP problem is guaranteed by Theorem 7.3.6 of [10]. ■

Our online mining algorithm is described in Algorithm 1. We calculate the optimal revenue based on binary search. The upper bound can be set as the result of the underlying MDP model according to Lemma 1 (lines 2-3). Alice will execute the optimal action based on the current ρ and obtain the relative revenue (lines 16 and 24). Her optimal revenue is no less than the relative revenue according to Lemma 2 (lines 25-29). We do not have to recalculate the optimal action at every step according to Lemma 3. The online algorithm will adopt the corresponding action if the same belief state has met. Otherwise, it will calculate and store the new optimal action (lines 11-15). A block of size 1MB needs 18 seconds to reach three thousand nodes in Bitcoin [27]. Making timely decisions by Alice is very important. In line 5, we use AEMS2 [11] [28], one of the fastest POMDP algorithms, to compute the optimal action (line 14).

Algorithm 1 Algorithm for solving the POMDP.

Input: $\mathcal{M}_{PO}, \mathcal{M}$, a truncation parameter ξ_0, a precision value ε;

Output:ρ;

Static: b_c:The current belief;

a^* : optimal action;

```
 1: $low \leftarrow 0$;
 2: $\rho^* =$QMDP($\mathcal{M}$);
 3: $upper \leftarrow \rho^*$;
 4: while $upper - low > \varepsilon$ do
 5:     $\rho \leftarrow (low + upper)/2$
 6:     RESULT$\leftarrow \{\}$;
 7:     $r_1 \leftarrow 0, r_2 \leftarrow 0, r_h \leftarrow 0$;
 8:     $v_\rho \leftarrow 0$;
 9:     $b_c \leftarrow$ initial belief
10:     while $\xi_0 > 0$ do
11:         if $b_c \in$ RESULT then
12:             $a^* =$RESULT$[b_c]$
13:         else
14:             $a^* \leftarrow AEMS2(b_c, \mathcal{M}_{PO})$
15:         end if
16:         $(r_1^i, r_2^i, r_h^i)\leftarrow$ Execute $a^*$ for $b_c$.
17:         $r_1 + = r_i^i, r_2 + = r_2^i, r_h + = r_h^i$;
18:         $v_\rho + = w_\rho(r_1^i, r_2^i, r_h^i)$;
19:         RESULT$[b_c] = a^*$
20:         Perceive a new observation $z$
21:         $b_c \leftarrow b'(b_c, a^*, z)$        ▷ update algorithm is Eq. (23)
22:         $\xi_0 - = 1$
23:     end while
24:     $R_1' = r_1/(r_1 + r_2 + r_h)$;
25:     if $v_\rho > 0$ then
26:         $low \leftarrow max(\rho, R_1')$;
27:     else
28:         $upper \leftarrow \rho$; $lower \leftarrow \max(low, R_1')$;
29:     end if
30: end while
31: return $\rho$;
```

V. Transient Analysis of Profitability

In this section, we first describe the difficulty adjustment algorithm (DAA) in Bitcoin-like systems, and model the revenue of miners in one difficulty adjustment period. We analytically show that the extra revenue of selfish mining is originated from the DAA.

A. Bitcoin-like Difficulty Adjustment

The essence of Bitcoin mining is to solve a cryptographic puzzle. The header of a block mainly includes the Hash of the previous block, the Merkle root Hash of transactions, the beginning time of calculating header Hash, the nBits used to generate the target difficulty and the *NONCE*. A Bitcoin miner repeatedly enumerates a *NONCE* until the head Hash is below the difficulty target. The smaller a target value is, the more difficult the discovery of a valid *NONCE* will be. For a fixed target difficulty, a larger Hash power means a shorter time of finding a valid *NONCE*.

To maintain a stable block generating interval, Bitcoin and Altcoins introduce difficulty adjustment algorithms (DAAs) to cope with the variable Hash powers in the systems. The Bitcoin DAA is executed after 2016 blocks have been mined. It is actually a feedback control system: if the actual time of mining 2016 blocks is larger than 20160 minutes (10 minutes per block), the target difficulty decreases proportionally, and increases otherwise. To avoid excessive fluctuation, the difficulty of the next period should be within the range $[\frac{1}{4}, 4]$ times of the current difficulty. When a miner performs selfish mining, a lot of blocks are orphaned so that the actual time to mine 2016 blocks becomes longer. In the next difficulty adjustment periods, the target difficulty is lowered down to maintain the fixed block generating rate.

B. Absolute Revenue

Previously we define the revenue of a miner in each mining round and his/her relative revenue. However, the duration of a mining round may not be fixed all the time, and the actual number of valid blocks obtained by a miner in each unit of wall-clock time is overlooked. In Bitcoin system, we denote 10 minutes as the unit time, and denote a DAA period as the expected time units of mining 2016 **valid** blocks.

***Definition** 3:* (*Absolute Revenue*) *The absolute revenue is the average number of blocks obtained in each unit time.*

#1 DAA Period. We treat the first DAA period as the beginning of selfish mining in order to analyze the transient absolute revenues of Alice and Bob. The following normalization rule is made to simplify the analysis by eliminating the randomness of block generating interval. *A block is generated every time unit at the first difficulty adjustment period, i.e.,* $\Delta t_1 = 1$.

With the above assumption, we can easily compute the duration of generating 2016 valid blocks, though slightly sacrificing its rigorousness. Let R_{vld} be the total number of **valid** blocks of all miners in a mining round, and let R_{tot} be the total number of blocks including the **valid and orphan** blocks in the same round. This means that a mining round occupies R_{tot} time units. Denote by T_1 the expected time units to accomplish the first DAA period. One can calculate the time of mining rounds that the total number of valid blocks reaches 2016. Then, T_1 is equivalent to the sum of time units of these mining rounds. There exists

$$\begin{aligned} R_{vld} &= R_1 + R_2 + R_h; \\ R_{tot} &= 1; \\ E[T_1] &= \frac{2016}{R_{vld}} \cdot R_{tot}. \end{aligned} \tag{31}$$

Due to the orphaned blocks, R_{tot} is greater than R_{vld} so that the actual time of the first DAA period is longer than 2016 time units.

Subsequent DAA Periods. After the first DAA period, the blockchain consensus mechanism finds that the time interval of generating a valid block is longer than one time unit. Consequently, the target difficulty decreases to match the current valid (visible or perceptible) Hash power in the system. Given the invariable Hash power of miners, the block generating interval Δt_i becomes smaller for $i \geq 2$. Let T_i be the expected time units of the i^{th} DAA period that has $T_i = 2016$ for $i \geq 2$. It is noted we assume $R_{tot} \leq (4 \cdot R_{vld})$. This is also the goal that DAA is going to achieve.

Absolute Revenue Over Time. Recall that the absolute revenue captures the expected reward of a miner in each time unit. Since our purpose is to investigate the transient profitability of selfish mining, we define $\tilde{R}_i(K)$ as the absolute revenue of the i^{th} miner over K DAA periods. Therefore, we obtain the following expressions for $\forall i \in \{a, b, h\}$

$$\begin{aligned} \tilde{R}_i(K) &= \frac{2016 \cdot K \cdot R_i}{R_{vld}} \cdot \frac{1}{\sum_{k=1}^{K} E[T_k]} \\ &= \frac{2016 \cdot K R_i}{R_{tot} + (K-1) R_{vld}}. \end{aligned} \tag{32}$$

Now we are aware that the selfish mining has a smaller absolute revenue in the first DAA period no matter whether the Hash power of the attacker is above the stationary profitable threshold or not. This claim also holds in different selfish mining policies. As K increases, a selfish miner hopefully can reimburse his/her loss in the first DAA period by his/her extra revenue in the future DAA periods. With our absolute revenue model, we can characterize how much time is needed to make selfish mining profitable eventually.

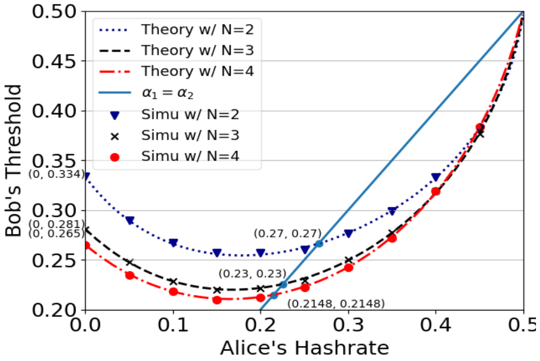

Fig. 9. Bob's threshold under the influence of Alice's Hashrate.

VI. Evaluation

In this section, we develop an event-driven simulator for basic selfish mining and a time-driven simulator for POMDP-based selfish mining. Comprehensive experiments validate the correctness of our models and reveal important properties regarding the profitability of selfish mining.

A. Basic Selfish Mining

***Observation** 1: When there are multiple attackers in Bitcoin-like systems, the attackers' profitable thresholds decrease and the system security is degraded.*

We illustrate Bob's profitable threshold of selfish mining in Fig. 9 as Alice's Hash power increases from 0 to nearly 0.5. To avoid involving too many control variables, the tie-breaking parameters are set to $\gamma_1{=}\gamma_2{=}\frac{1}{2}$ and $\theta_1{=}\theta_2{=}\frac{1}{3}$. One can observe from three curves with different N that Bob's profitable threshold decreases in the beginning and increases afterwards. Especially when $N = 2$ and $\alpha_1 = 0.16$, Bob's profitable threshold is the lowest. Under this situation, Alice's selfish mining may yield less revenue compared with her honest mining. We further draw a $45°$ line to indicate the profitable threshold for both Alice and Bob when their Hash power are symmetric. When N is 2, 3 and 4, the profitable threshold is 26.64%, 22.57% and 21.48%. In contrast, it takes the value of 33.33%, 28.08% and 26.50% respectively, if there has a single attacker. An obvious conclusion is that the existence of multiple attackers makes the selfish mining more easily profitable.

***Observation** 2: The profitable threshold decreases with the increase of N, and remains stable with the attackers of symmetric Hash power as $N \geq 4$; it also decreases when the number of attackers m increases.*

We evaluate the profitable thresholds of BSM with different N and m. Our purposes are twofold: one is analyzing the interplay between this threshold and the environment variables, and the other is justifying the use of $N \leq 4$ in the mathematical modeling. The Hash powers of all the attackers are identical,

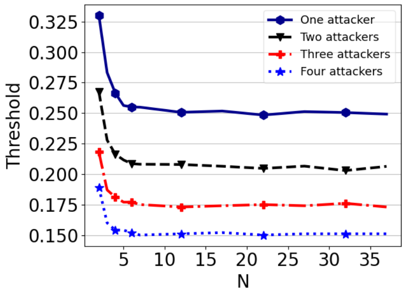

Fig. 10. Profitable threshold vs the maximum number of private blocks (N).

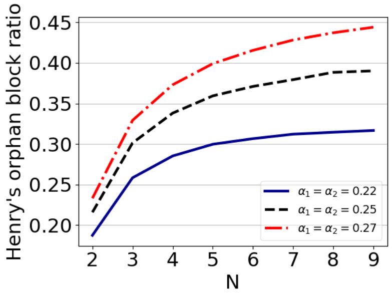

Fig. 11. Henry's orphan block ratio vs the maximum number of private blocks (N).

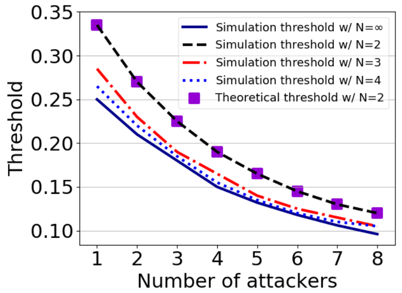

Fig. 12. Profitable threshold vs the number of attackers (m).

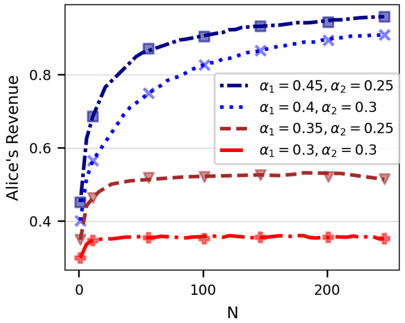

Fig. 13. Alice's Revenue with $\alpha_1 > \max(\alpha_2, \alpha_h)$.

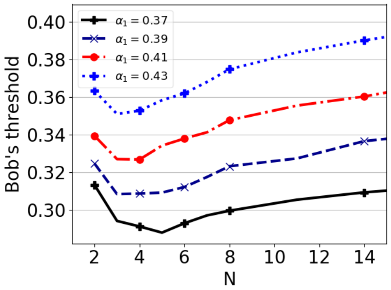

Fig. 14. Simulated threshold for Bob when $\alpha_1 > \max(\alpha_2, \alpha_h)$.

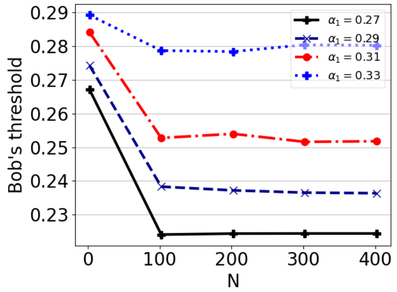

Fig. 15. Simulated threshold for Bob when $\alpha_h > max(\alpha_1, \alpha_2)$.

and the competing chains are indistinguishable upon the tie-breaking rules.

Fig. 10 shows the relationship between N and the profitable threshold. The cases with 1, 2, 3 and 4 symmetric attackers are expressed in solid, dashed, dash-dotted and dotted lines. One can observe that the profitable threshold decreases remarkably for different m as N increases from 2 to 4. The event-driven simulations exhibit a stable profitable threshold when Alice and Bob can hold more than 5 private blocks. For instance, this threshold converges to 25% for a sufficiently large N with a single attacker, which is in line with [2]. With two symmetric attackers ($m = 2$), its value is 20.60% at $N = 30$, slightly different from that at $N = 4$. In general, Alice's or Bob's Hash power is much smaller than Henry's Hash power. The chance that Alice's or Bob's private chain takes a large lead over the public chain in a mining round is very small. Hence, it does not make an evident influence on the profitable threshold when N is already large. Moreover, hiding a long private chain and releasing all the blocks simultaneously will make the selfish mining attack easily detected. Fig. 11 shows Henry's orphan block ratio as N increases from 2 to 9 with $m = 2$. Even though Alice's and Bob's Hash powers are merely (0.22, 0.22), they cause a very high orphan block ratio to Henry, e.g., 18.73% with $N = 2$, 28.53% with $N = 4$ and 31.66% with $N = 9$. Such a high orphan ratio can easily expose the identity of attackers. Therefore, our modeling framework only considers $N \leq 4$ though it is extensible to $N > 4$.

We use mathematical models and event-driven simulations to quantify the impact of m on the profitable threshold in Fig. 10 and 12. One can observe that the increase in the number of attackers reduces the profitable threshold, thus endangering blockchain security. For $N = 4$, the profitable thresholds with $m \in \{2, 4, 8\}$ are $\{0.2148, 0.155, 0.11\}$ respectively. This challenges the cognition that selfish mining is less likely to happen if the Hash power of a mining pool is below 25%. The primary reason that more attackers lead to smaller profitable thresholds lies in that the Hash power of the honest miner declines relatively. Meanwhile, our model coincides with the simulation result at $N = 2$ in Fig. 12, thus validating its correctness.

Since the advent of the seminal work [2], the Bitcoin community tries to constrain mining pools to possess less than 25% of Hash power. However, we prove that 25% is not *enough*: Bitcoin mining is fragile in the presence of multiple selfish miners.

***Observation** 3: If $\alpha_h < \max(\alpha_1, \alpha_2)$, the revenue of the attacker with more Hash power among three miners will increase as N increases. If $\alpha_h > \max(\alpha_1, \alpha_2)$, the revenue of the attacker with more Hash power will increase first and then remain stable as N increases. The common benefit Hash power region of two attackers increases at first and then decreases with the increase of N.*

We explore the revenue of the attackers with more Hash power than the honest miner under different N when $\alpha_h < \max(\alpha_1, \alpha_2)$. In Fig. 13, we show Alice's revenue of Alice at four situations: the Hash powers of Alice, Bob and Henry are (0.45, 0.25, 0.3), (0.4, 0.3, 0.3), (0.35, 0.25, 0.4) and (0.3, 0.3, 0.4) that are labeled as situations 1, 2, 3 and 4. Situations 1 and 2 manifest that the revenue of the attacker with more Hash power than others increases as N increases. The attacker can

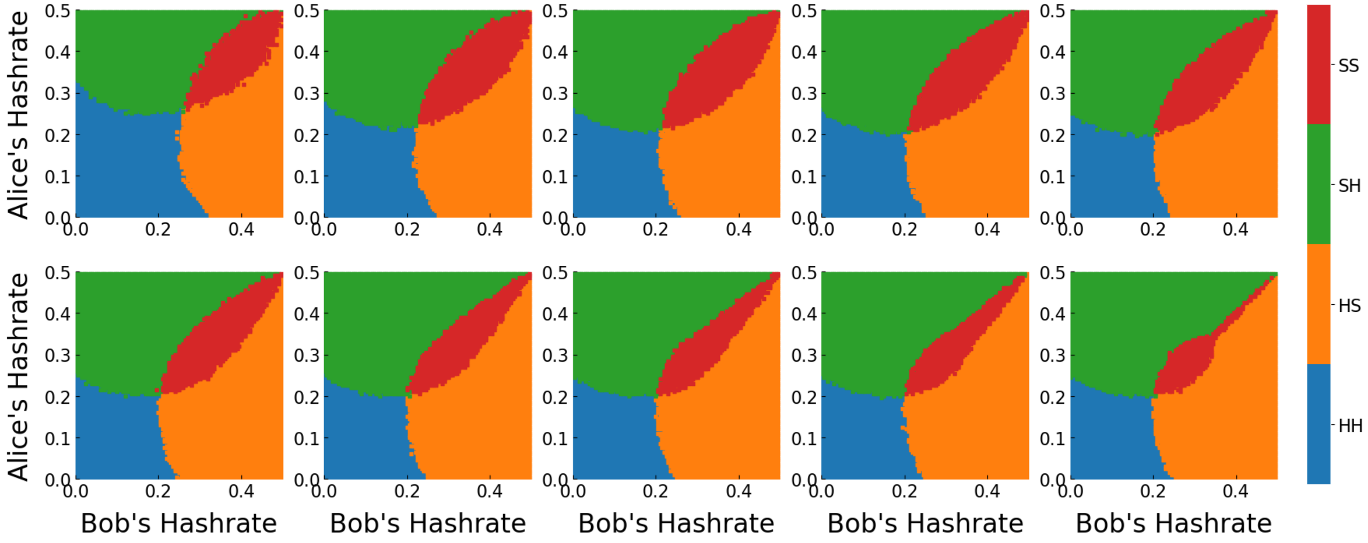

Fig. 16. Profitable regions of both selish mining attackers with $N = 2, 3, 4, 5, 7, 8, 15, 25, 35, \infty$.

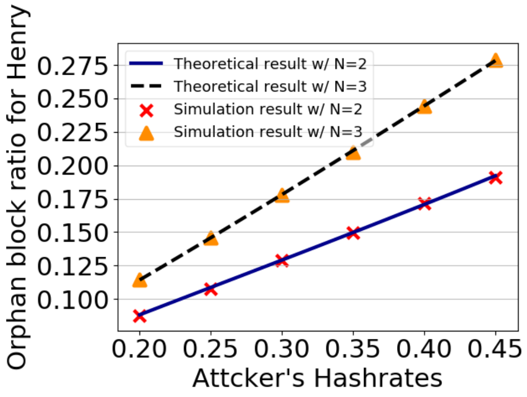

Fig. 17. Estimating Bob's Hash power by observing orphan block ratio.

obtain more than 90% of the revenue, achieving the similar effect as the 51% attack, even though she does not have 51% of Hash power. Situation 3 and 4 show that Alice's revenue converges as N increases if Henry has the largest Hash power. The revenue of Alice converges to 0.524 in situation 3 and 0.357 in situation 4 with $N = 200$. This implies that limiting the attacker's Hash power will prevent the attacker from obtaining too many revenues when N is large.

We then investigate Bob's profitable threshold when Alice has a different Hash power and N is large. Fig. 14 shows Bob's thresholds under different N when Alice has the largest Hash power, i.e. $\alpha_1 > \max\{\alpha_2, \alpha_h\}$. Bob's profitable threshold decreases first and then increases as N increases. That means even if Bob can obtain extra revenue when N is small, he can not obtain extra revenue when N is large enough. Under this circumstance, Alice's revenue can obtain far more than her Hash power proportion, and her attack becomes meaningless because this system can not attract other miners even other selfish mining attackers. Fig. 15 shows Bob's profitable threshold will decrease first and then converge as N increases when Henry has more Hash power than Alice and Bob. This suggests that limiting the attacker's Hash power also encourages other miners including other selfish mining attackers to continue mining even if N is large. According to these analyses, the attacker will not hide too many blocks even he has more Hash power than others.

We now analyze the interplay between Alice's and Bob's profitable thresholds. Fig. 16 shows the profitable regions for each attacker under different N. The blue part indicates that neither attacker can gain additional revenue if they perform the BSM attack, and the red part indicates that both attackers can gain additional revenue through selfish mining. The green (resp. orange) part represents the situation that only Alice (resp. Bob) can obtain extra revenue. The intersection of four regions is actually the profitable threshold with symmetric Hash powers of Alice and Bob that decreases over N and converges gradually. The common profitable region (in red) first expands and then shrinks as N increases. The reason is the following. A large red region basically says that both Alice and Bob are profitable with BSM even if their Hash powers are asymmetric to some extent. When N is very small, Alice and Bob can hide only a couple of blocks so that their ability of wasting Henry's Hash power is restrained. With the increases of N, their selfish mining ability becomes more powerful, and thus could have more chances of obsoleting the public chain even if each of them has a smaller Hash power than Henry. Meanwhile, given the restriction of N, both Alice and Bob may receive a certain amount of extra revenues, resulting in a larger common profitable region. When N is large, the stronger attacker is inclined to dominating the selfish mining. If Alice's Hash power is larger than Bob's, Bob will find it difficult to compete with Alice so that Bob's profitable threshold is getting higher. If Alice's Hash power is the largest, it is similar to 51% attack. Bob's selfish mining is profitable only when Alice's and Bob's Hash powers are sufficiently close, causing the common profitable region to shrink to a line segment.

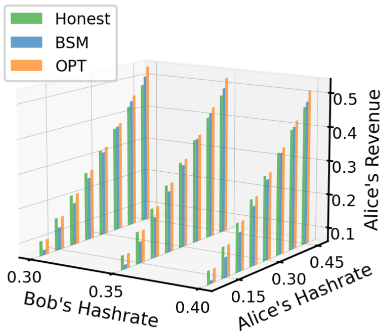

Fig. 18. Optimal revenue for Alice when $N = 2$.

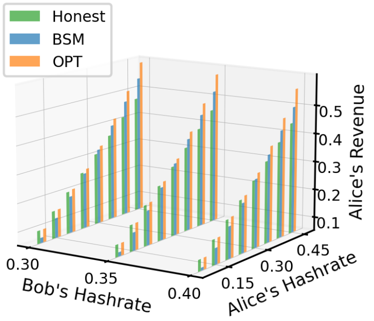

Fig. 19. Optimal revenue for Alice when $N = 3$.

B. MDP and POMDP-based Mining

The roadmap of performing optimal mining is the following. We explore its feasibility by estimating the unknown parameters. Then, we compute the optimal mining policy and the corresponding revenue using MDP, and on this basis we compute the optimal mining policy using POMDP under partially observable states.

Recall that Alice is the optimal miner (OPT) and Bob is the basic selfish miner (BSM). At the first stage, Alice needs to decide whether there exists a selfish miner namely Bob, and if so, what Bob's Hash power is. Note that there has a one-to-one mapping between Henry's orphan block ratio and Bob's Hash power. Fig. 17 shows the orphan ratio of the honest miner as a function of Bob's Hash power with $N = 2$ and $N = 3$. The theoretical and experimental results match well, which indicates the feasibility of calculating Bob's Hash power through the observed orphan block ratio.

We next compute Alice's optimal policy and the corresponding revenue of OPT mining. Let the error parameter $\epsilon = 0.00001$ and the execution number $\xi_0 = 500000$. Fig. 18 illustrates the optimal revenue obtained by Alice when Alice's Hash power is $\alpha_1 \in \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45\}$, Bob's Hash power is $\alpha_2 \in \{0.30, 0.35, 0.40\}$ and $N = 2$. The maximum length of longest public chain is set as $(N+1)$. One can observe that Alice's optimal mining and honest mining yield the same revenue when her mining power is relatively small, e.g. $\alpha_1 = 0.1$, $\alpha_2 = 0.40$ and $\alpha_h = 0.50$. Under these situations, the optimal mining policy is exactly the honest mining, while BSM underperforms the honest mining significantly. On the contrary, when Alice's Hash power is large, e.g., $\alpha_1 = 0.40$, $\alpha_2 = 0.40$ and $\alpha_h = 0.20$, the optimal mining policy results in a higher revenue than the basic selfish mining, and the basic selfish mining is better than the honest mining. Fig. 19 shows a set of similar experiments except for $N = 3$. When $\alpha_1 = 0.30$, $\alpha_2 = 0.30$ and $\alpha_h = 0.4$, Alice's optimal mining policy obviously outperforms both BSM and the basic selfish mining.

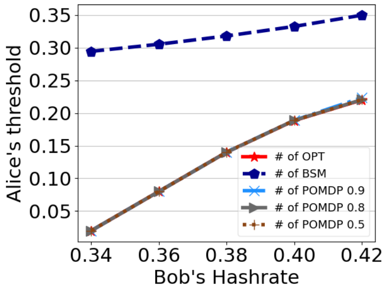

Fig. 20. Profitable threshold for Alice when $N = 2$.

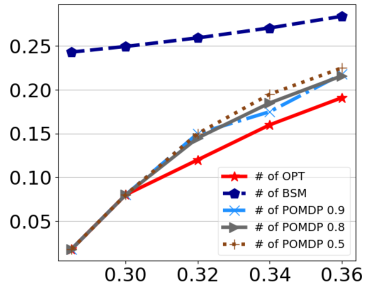

Fig. 21. Profitable threshold for Alice when $N = 3$.

We describe a concrete example of optimal mining policy at a few representative states for simplicity. Table I shows Alice's optimal strategy with α_1=0.2, α_2=0.4 and α_h=0.4. Alice has less Hash power than Bob and Henry. She chooses to adopt the public chain if the length of her chain is shorter than that of Bob or Henry. If the length of the chain (i.e. $l_1 + h_1$) she is mining on is equal to that of the public chain (i.e. h_3), and is able to "fork", she will release l_1 private blocks to seize the chance of winning. If both Alice and Bob hold a private block, she will choose to hide the private block and continue to mine after her own block. By performing the optimal policy, Alice can earn an additional 0.08% revenue, even though Bob has 40% Hash power.

TABLE I
OPTIMAL POLICY FOR ($\alpha_1 = 0.2, \alpha_2 = 0.4, N = 2$).

state	action
$(l_2 + h_2) > (l_1 + h_1)$	*adopt*
$h_3 > (l_1 + h_1)$	*adopt*
$(l_1 + h_1) = (l_2 + h_2) = h3, fork = r$	l_1
$l_1 = l_2 = 1, h_3 = 0$	0
$l_1 = 1, l_2 = 0$	l_1
$(l_1 + h_1) > (l_2 + h_2)$	l_1

***Observation** 4: When Alice uses the POMDP-based policy, and Bob uses the basic selfish mining, Alice has a much lower profitable threshold compared with the basic selfish mining.*

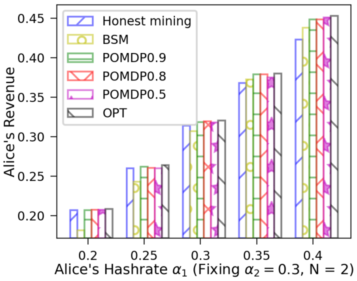

Fig. 22. The revenue for Alice when $\alpha_2 = 0.3, N = 2$.

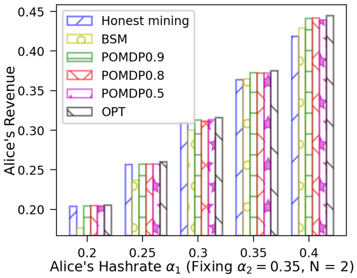

Fig. 23. The revenue for Alice when $\alpha_2 = 0.35, N = 2$.

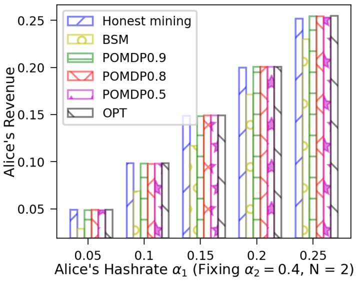

Fig. 24. The revenue for Alice when $\alpha_2 = 0.4, N = 2$.

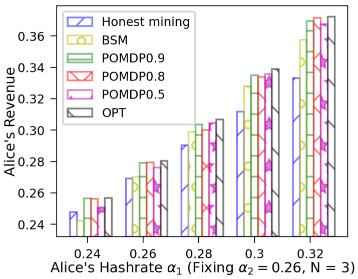

Fig. 25. The revenue for Alice when $\alpha_2 = 0.26, N = 3$.

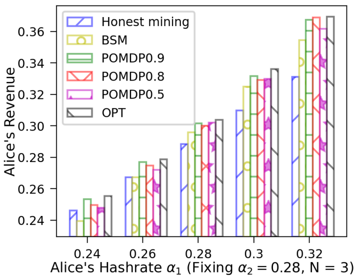

Fig. 26. The revenue for Alice when $\alpha_2 = 0.28, N = 3$.

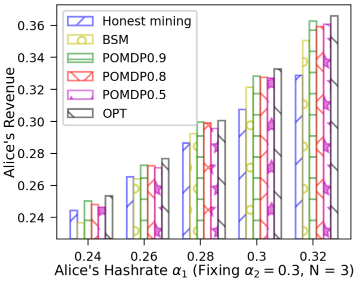

Fig. 27. The revenue for Alice when $\alpha_2 = 0.3, N = 3$.

Her revenue is no less than the honest mining and the basic selfish mining, and is close to the MDP-based policy with complete state information.

We further evaluate Alice's profitable threshold and revenue when she uses the POMDP-based mining policy. Fig. 20 and 21 compare the profitable thresholds of BSM, MDP-based (OPT) and POMDP-based (POMDP) mining strategies when $N = 2, 3$ and α_2 increases from 0.285 to 0.42. An interesting finding is that both OPT and POMDP strategies have much smaller profitable thresholds compared with BSM. For instance, the profitable threshold is 0.02 when $\alpha_2 = 0.34$ and $N = 2$, and is 0.08 when $\alpha_2 = 0.3$ and $N = 3$. The reason is the following. When Bob is playing BSM and Bob's Hash power is much larger than Alice's, Alice will choose to mine honestly. If there is a fork between Bob and Henry, Alice insists on mining on the public chain that contains her own blocks. Alice's policy is equivalent to decreasing γ_2 in the situation of a single attacker. Therefore, Bob will find it difficult to gain more revenues, and Alice as well as Henry benefit from Bob's losses. When Alice's Hash power is below the profitable threshold, the POMDP policy cannot yield the revenue commensurable with her Hash power. In addition, with the increase of Bob's Hash power, Alice's profitable threshold becomes higher. A cross-comparison between Fig. 20 and Fig. 21 shows that a larger N makes Alice hard to compete with Bob if $\alpha_1 \leq \alpha_2$. When Bob's Hash power is 0.36, Alice's profitable threshold is 0.08 when $N = 2$ and is about 21.6% when $N = 3$.

The POMDP policy can improve Alice's revenue under different situations. Fig. 22∼24 plot Alice's revenues using Honest mining, BSM, OPT and POMDP-based mining at $N = 2$; Fig. 25∼27 show those revenues with $N = 3$. In each set of experiments, we fix Bob's Hash power (α_2) and changes Alice's Hash power (α_1) from 0.20 to 0.40. As for the POMDP policy, three cases are considered, in which the probability of generating a block is 0.9, 0.8 or 0.5 at each time slot. Note that this probability does not influence the execution of our POMDP solver. One can easily observe that the POMDP policy has the comparable revenues with the honest and the MDP mining policies when Alice's Hash power is relatively small. While the basic selfish mining seems very "stubborn", causing Alice's revenue much lower than the honest mining in this situation. The PODMP-based policy generates equal or higher revenues than the honest mining and the basic selfish mining, and approaches the performance of the MDP policy.

The designed online algorithm can effectively calculate the optimal revenue of $\mathcal{M}_{OP}$. We compare the number of iterations required to execute the binary search algorithm in [7] and Algorithm 1. The reduction ratios are summarized in Table II. Under different Hash power combinations and different action slots, the efficiency of Algorithm 1 is significantly higher. When $N = 2$, $\alpha_1 = 0.3$, $\alpha_2 = 0.3$ and $p = 0.5$, we can save 46% of the computing time. It takes a long time to simulate half a million times to obtain the revenue for each ρ. The improvement of our search algorithm plays a significant role in solving $\mathcal{M}_{PO}$ rapidly.

C. Selfish Mining on Multiple Difficulty Adjustment Periods

***Observation** 5: A selfish miner obtains a smaller absolute revenue than that of honest mining during the first difficulty adjustment period regardless of his Hash power. However, he might gain profit after a number of periods that is related to the selfish miners' Hash power.*

Fig. 28 shows the *relative revenue* and *absolute revenue* of attackers with same Hashrate 33% and $N = 4$ in each DAA

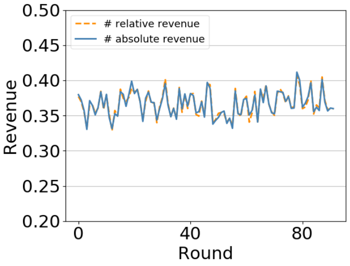

Fig. 28. Relative revenue and absolute revenue when $\alpha_1 = \alpha_2 = 0.33, N = 4$.

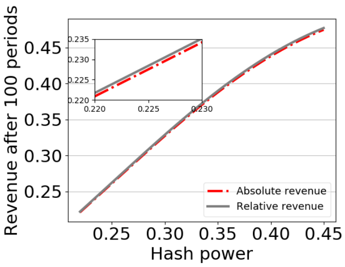

Fig. 29. The theoretical relative revenue and absolute revenue when $\alpha_1 = \alpha_2$ after 100 periods.

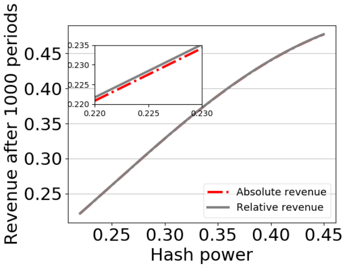

Fig. 30. The theoretical relative revenue and absolute revenue when $\alpha_1 = \alpha_2$ after 1000 periods.

TABLE II
THE TIME REDUCTION RATIO OF ALG. 1 COMPARED TO TRADITIONAL ALGORITHM.

Hash power	p	EFF IMP
$\alpha_1 = 0.3, \alpha_2 = 0.3$ $N = 2$	0.9	53.3%
	0.8	46.6%
	0.5	66.6%
$\alpha_1 = 0.35, \alpha_2 = 0.35$ $N = 2$	0.9	80%
	0.8	60%
	0.5	46.7%
$\alpha_1 = 0.28, \alpha_2 = 0.28$ $N = 3$	0.9	60%
	0.8	46.7%
	0.5	46.7%
$\alpha_1 = 0.3, \alpha_2 = 0.3$ $N = 3$	0.9	73.3%
	0.8	46.7%
	0.5	40%

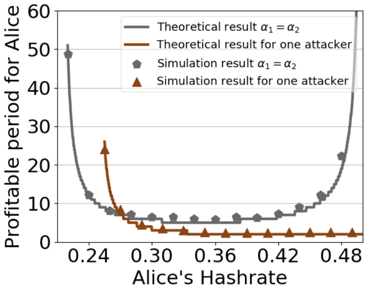

Fig. 31. Profitable time and Hash power.

period. The *relative revenue* and *absolute revenue* are equal within the allowable range of error. Therefore, the *relative revenue* can play the same rule with *absolute revenue* in representing benefit. Fig. 29 and Fig. 30 show the theoretical *relative revenue* and *absolute revenue* after 100 and 1000 DAA periods when $\alpha_1 = \alpha_2$. It can be observed that the *absolute revenue* is always less than the *relative revenue*, but the difference is small. When $\alpha_1 = 0.22$, the *relative revenue* is 0.2217 and the *absolute revenue* is 0.2209. This difference decreases to 0.0001 after 1000 periods. That means the difference between *relative revenue* and *absolute revenue* decreases with the increase of the attack time.

As Eq. (32) shows, when Alice has more Hash powers, she can get illegal revenue earlier. However, if both of the two attackers has large Hashrate, they will benefit late. Fig. 31 shows the simulation results and the theoretical results of profitable time under different Hash powers with symmetric attackers. The horizontal axis represents the attack's Hash power and the ordinate represents the attackers' profitable time, also the curves are theoretical results and the dots are simulation results. The selfish mining is profitable after 51 rounds of difficulty adjustment (i.e. 714 days in Bitcoin) if the Hashrates of selfish miners are both 22% (slightly higher than the profitably threshold). This delay decreases to 5 rounds (i.e. 70 days in Bitcoin) as their Hashrates accrues to 33%, which is still very long. As the attackers gained more computing power, Alice's main competitor became Bob rather than Henry. The benefit time begins to increases for Alice and Bob. When there is only one attacker and she has 25.5% Hashrates, the attacker will obtain extra revenue after 26 rounds (about 364 days).

It shows that when attackers' Hashrate is relatively small, it takes a rather long period to gain profit. When the two attackers all have large Hashrates, it also takes a long period to obtain extra revenue. That means in the real system, it is a little bit hard to perform attack. If the global Hashrate increase, we can also use this formula to calculate when to stop the attack before we can benefit the most.

VII. Related Work

Selfish mining attack is one of the core challenges of blockchain consensus that has been extensively studied in the past several years. We hereby describe the recent advances in selfish mining policies and their analytical or experimental performance.

One attacker. Since the advent of [2], there have been many studies on different forms of selfish mining attacks. Nayak et al. [9] proposed the stubborn mining based on the basic selfish mining, it points out that selfish mining is not always the best for different parameters. The stubborn attack improves about 13.94% revenues compared with the basic selfish mining attack. A more intelligent selfish mining strategy has been proposed in [7] based on the Markov Decision Process and it decreases the profitable threshold to 23.21%. Tao et al. [20] described semi-selfish mining attack on the basis of selfish mining based on hidden Markov decision process, which not only ensures the benefit of the attack, but also reduces the forking rate. Negy et al. [23] introduced *intermittent selfish mining* and showed that the intermittent selfish miner above 37% hash power earns more coins per time unit even when $\gamma = 0$. Negy and Davidson et al. [23] [24] simulated the profitability of selfish mining under several difficulty adjustment algorithms used in popular cryptocurrencies.Selfish mining attack takes on different properties in the Ethereum system because of the uncle block. Grunspan and Ritz introduced the selfish mining attack in Ethereum and found that Ethereum is more vulnerable to selfish mining than Bitcoin [25] [26].

At the same time, selfish mining attack can also be combined with other attacks to achieve greater benefits. Gervais et al. devised optimal strategies for double-spending and selfish mining while taking into account real world constraints such as network propagation, different block sizes, block generation intervals, information propagation mechanism, and the impact of eclipse attacks [22]. Kwon et al. [14] proposed FAW attack which combines the selfish mining attack and withholding attack. The reward for an FAW attacker is always equal to or greater than that for a BWH attacker. Gao et al. extended the work of Kwon et al. proposing the power adjusting withholding attack (PAW) and bribery selfish mining attack (BSM) [15]. They showed PAW could avoid the "miner's dilemma" in BWH attack and BSW increases 10% revenues compared with traditional SM [33]. However, BSW will introduce "venal miner's dilemma". To avoid the "venal miner's dilemma", Yang et al. proposed the IPBSM attack which assumed all attackers take the optimal bribery selfish mining [16].

Multiple attackers. The participation of multiple attackers in the system will greatly change the benefits of selfish mining attacks. Ruan et al. simulated the situation in which there are two attackers in the system and obtained the corresponding threshold will decrease [3]. Francisco et al. proposed semi-selfish mining when there are two attackers and modeled this attack [17]. They obtained the Nash equilibrium under different Hash powers and the threshold for each policy based on the game theory. Also, they model the situation when the number of attackers is more than two. However, they do not get the closed-form result. Azimy et al. designed a Bitcoin network simulator and used it to simulate different configurations of miners to be able to address this problem. Their finding shows that in almost all of the configurations, with the presence of a more powerful selfish miner, selfish mining actually decreases the revenue of the weaker selfish miners and also helps the stronger selfish miner [30]. Sebastian et al. proposed the simulation results that the profitable threshold decreases in proportion to the number of selfish miners [35]. Moreover, there exist Nash equilibrium where all selfish miners mine honestly and simultaneously earn their unfair mining reward. Zhang et al. simulated the situation when there were multiple selfish mining attackers in the system. It shows there are scenarios where it is enough to have 12% mining power to benefit from selfish mining but also that having more than seven selfish miners which benefit simultaneously is highly unlikely [18]. Charlie et al. proposed SquirRL which is a framework for using deep reinforcement learning to analyze attacks on blockchain incentive mechanisms. The revenue of SquirRL is greater than that of the Markov decision attackers when there are multiple selfish mining attackers [19]. Xia et al. explored the impact of multiple miners and propagation delay on selfish mining [29].

VIII. Conclusion

In this paper, we first study how the existence of multiple misbehaving miners influences the profitability of basic selfish mining. By establishing the Markov chain model to describe the action of attackers and honest miners, we can obtain the minimum profitable threshold is symmetric 21.48%, which decreases as the number of symmetric attackers increases. If there are two asymmetric selfish mining attackers in the system, the profitable threshold of one attacker decreases first and then increases with the increase of the other attacker's Hash power. We validate this in both models and experiments. We next investigate the revenues of the attackers when one executes the basic selfish mining and the other implements the strategic mining. A new mining strategy is designed for the miners with incomplete information based on *POMDP*. We can obtain revenue by the new strategy no less than honest mining and basic selfish mining. Considering the difficulty adjustment, we model the transient process and acquire the closed-form solution of the profitable time. It can be discovered that the profitable time is large when the attacker's Hash power is low. Moreover, there is a negative correlation between the profitable time and the attackers' mining power.

References

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

[2] I. Eyal and E. G. Sirer. "Majority is not enough: Bitcoin mining is vulnerable". *In Financial Cryptography and Data Security.* Springer, 2014, pp. 436-454.

[3] Q.H. Liu, N. Ruan, et al. "On the Strategy and Behavior of Bitcoin Mining with N-attackers". *Proc. of the Asia Conference on Computer and Communications Security,* pp. 357-368, 2018.

[4] https://decrypt.co/35373/how-long-does-it-take-to-mine-a-bitcoin, [online].

[5] A. Papoulis, S. U. Pillai. Probability, random variables, and stochastic processes[M]. Tata McGraw-Hill Education, 2002.

[6] R. Pass, E. Shi, "Fruitchains: A fair blockchain". *Proc. of the Asia Conference on Computer Symposium on Principles of Distributed Computing,* pp. 315-324, 2017.

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

[8] S. Jiang and J. Wu, "Bitcoin Mining with Transaction Fees: A Game on the Block Size," *in 2019 IEEE International Conference on Blockchain (Blockchain),* Atlanta, GA, USA, Jul. 2019, pp. 107–115.

[9] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack," *in 2016 IEEE European Symposium on Security and Privacy (EuroS&P),* Saarbrucken, Mar. 2016, pp. 305–320.

[10] Puterman, Martin L. Markov decision processes: discrete stochastic dynamic programming. John Wiley Sons, 2014.

[11] Ross, Stephane, and Brahim Chaib-Draa. "AEMS: An anytime online ´ search algorithm for approximate policy refinement in large POMDPs." *IJCAI.* 2007, pp. 2592-2598.

[12] Karkus, Peter, David Hsu, and Wee Sun Lee. "QMDP-Net: deep learning for planning under partial observability." *Proceedings of the 31st International Conference on Neural Information Processing Systems.* 2017.

[13] P. Ashok, K. Chatterjee, P. Daca, J. Křetínský, and T. Meggendorfer, "Value Iteration for Long-Run Average Reward in Markov Decision Processes," *in Computer Aided Verification,* vol. 10426, Springer International Publishing, 2017, pp. 201–221.

[14] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin," *in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security,* Dallas Texas USA, Oct. 2017, pp. 195–209.

[15] S. Gao, Z. Li, Z. Peng, and B. Xiao, "Power Adjusting and Bribery Racing: Novel Mining Attacks in the Bitcoin System," *in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security,* London United Kingdom, Nov. 2019, pp. 833–850.

[16] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, “IPBSM: An optimal bribery selfish mining in the presence of intelligent and pure attackers,” *Int J Intell Syst,* vol. 35, no. 11, pp. 1735–1748, Nov. 2020.

[17] F. J. Marmolejo-Cossío, E. Brigham, B. Sela, and J. Katz, “Competing (Semi-)Selfish Miners in Bitcoin,” *in Proceedings of the 1st ACM Conference on Advances in Financial Technologies,* Zurich Switzerland, Oct. 2019, pp. 89–109.

[18] S. Zhang, K. Zhang, and B. Kemme, “Analysing the Benefit of Selfish Mining with Multiple Players,” *in 2020 IEEE International Conference on Blockchain (Blockchain),* Rhodes Island, Greece, Nov. 2020, pp. 36–44.

[19] Hou, Charlie, et al. “SquirRL: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning.” *arXiv* 2019.

[20] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, “Semi-selfish mining based on hidden Markov decision process,” *Int J Intell Syst,* vol. 36, no. 7, pp. 3596–3612, Jul. 2021.

[21] Meyn, Sean P., and Richard L. Tweedie. *Markov chains and stochastic stability.* Springer Science & Business Media, 2012.

[22] A. Gervais, et al., “On the Security and Performance of Proof of Work Blockchains,” *in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, Oct. 2016, pp. 3–16.

[23] K. A. Negy, P. R. Rizun, and E. G. Sirer, “Selfish Mining Re-Examined,” *in Financial Cryptography and Data Security*, vol. 12059, J. Bonneau and N. Heninger, Eds. Cham: Springer International Publishing, 2020, pp. 61–78.

[24] Davidson, Michael, and Tyler Diamond. “On the Profitability of Selfish Mining Against Multiple Difficulty Adjustment Algorithms.” *IACR Cryptol. ePrint Arch.* 2020 (2020): 94.

[25] C. Grunspan and R. Perez-Marco, “Selfish Mining in Ethereum,” *in Mathematical Research for Blockchain Economy,* P. Pardalos, I. Kotsireas, Y. Guo, and W. Knottenbelt, Eds. Cham: Springer International Publishing, 2020, pp. 65–90.

[26] F. Ritz and A. Zugenmaier, “The Impact of Uncle Rewards on Selfish Mining in Ethereum,” *in 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),* London, Apr. 2018, pp. 50–57.

[27] https://tradeblock.com/bitcoin/historical.

[28] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, “Online Planning Algorithms for POMDPs”, *jair,* vol. 32, pp. 663–704, Jul. 2008.

[29] Q. Xia et al., “The Impact Analysis of Multiple Miners and Propagation Delay on Selfish Mining”, *in 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, Jul. 2021, pp. 694–703.

[30] H. Azimy and A. Ghorbani, “Competitive Selfish Mining”, *in 2019 17th International Conference on Privacy, Security and Trust (PST),* Fredericton, NB, Canada, Aug. 2019, pp. 1–8.

[31] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, “A Deep Dive Into Blockchain Selfish Mining”, *in ICC 2019 - 2019 IEEE International Conference on Communications (ICC),* Shanghai, China, May 2019, pp. 1–6.

[32] C. Grunspan and R. Pérez-Marco, “On profitability of selfish mining”, *arXiv,* 2019.

[33] I. Eyal, “The Miner’s Dilemma”, *arXiv*, 2014.

[34] X. Dong, F. Wu, A. Faree, D. Guo, Y. Shen, and J. Ma, “Selfholding: A combined attack model using selfish mining with block withholding attack,” *Computers & Security,* vol. 87, p. 101584, Nov. 2019.

[35] T. Leelavimolsilp, L. Tran-Thanh, and S. Stein, ”On the Preliminary Investigation of Selfish Mining Strategy with Multiple Selfish Miners,” *arxiv*, 2018.

[36] Littman et al., ”Learning policies for partially observable environments: scaling up”. *In Proceedings of the 12th International Conference on Machine Learning (ICML-95),* pp. 362–370.

.

Appendix

TABLE III
A description of the transition and reward matrices $\mathcal{P}$ and $\mathcal{R}$ in the decision problem M.

$l_2 + h_2 - h_3 > 2$	$adopt$	$loc! = 2$	α_1	$(1, ir, 1, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, ir, 0, l_2 + 1, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			α_h	$(1, r, 0, l_2, h_3, h_2, h_3 + 1, \mu_3, \mu_2, \mu_3 + 1)$	
		$loc = 2$	α_1	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			α_2	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			α_h	$(1, r, 0, l_2, h_3 - h_2, 0, h_3 - h_2 + 1, h_3 - h_2, 0, h_3 - h_2 + 1)$	
	$h_1 + action < h_3$		α_1	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			α_h	$(loc, r, l_1 - action, l_2, h_1 + action, h_2, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_1 + action = h_3$	$r, action > 0$ $f_{13}, action = 0$	α_1	$(loc, f_{13}, l_1 - action + 1, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, f_{13}, l_1 - action, l_2 + 1, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h \gamma_1$	$(1, r, l_1 - action, l_2, h_3, h_2, h_3 + 1, \mu_1, \mu_2, \mu_1 + 1)$	
			$\alpha_h(1 - \gamma_1)$	$(1, r, l_1 - action, l_2, h_3, h_2, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_3 < h_1 + action < l_2 + h_2 - 1$		α_1	$(1, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_1 + action, \mu_1, \mu_2, \mu_1)$	$(0, 0, 0)$
			α_2	$(1, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_1 + action, \mu_1, \mu_2, \mu_1)$	
			α_h	$(1, r, l_1 - action, l_2, h_1 + action, h_2, h_1 + action + 1, \mu_1, \mu_2, \mu_1 + 1)$	
	$h_1 + action = l_2 + h_2 - 1$		α_1	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			α_2	$(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			α_h	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
	$h_1 + action = l_2 + h_2$		α_1	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h \beta_1$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h \beta_2$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		α_1	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	$(h1 + action - \mu_1, 0, \mu_1)$
			α_2	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	
$l_2 + h_2 - h_3 = 2$	$adopt$	$loc! = 2$	α_1	$(1, ir, 1, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, ir, 0, l_2 + 1, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			α_h	$(2, r, 0, 0, h_3, h_2 + l_2, h_2 + l_2, \mu_3, \mu_2, \mu_2)$	
		$loc = 2$	α_1	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			α_2	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			α_h	$(2, r, 0, 0, h_3 - h_2, l_2, l_2, h_3 - h_2, 0, 0)$	
	$h_1 + action < h_3$		α_1	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			α_h	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
	$h_1 + action = h_3$	$r, action > 0$ $f_{13}, action = 0$	α_1	$(loc, f_{13}, l_1 - action + 1, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, f_{13}, l_1 - action, l_2 + 1, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			α_h	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
	$h_1 + action = l_2 + h_2 - 1$		α_1	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			α_2	$(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			α_h	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
	$h_1 + action = l_2 + h_2$		α_1	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h \beta_2$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h \beta_1$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		α_1	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	$(h1 + action - \mu_1, 0, \mu_1)$
			α_2	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	
$l_2 + h_2 - h_3 = 1$	$adopt$	$loc = 2, loc = 3$	α_1	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			α_2	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			α_h	$(loc, f_{23}, 0, 0, h_3 - h_2, h_3 - h_2 + 1, h_3 - h_2 + 1, 0, 0, 1)$	
	$h_1 + action < h_3$		α_1	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			α_h	$(loc, f_{23}, l_1 - action, 0, h_1 + action, h_3 + 1, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_1 + action = l_2 + h_2$		α_1	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h \beta_2$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h \beta_1$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		α_1	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			α_2	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	

$l_2 + h_2 = h_3$	$adopt$	$f_{23}, f_{123}, loc = 2$	$\alpha_1\gamma_2$	$(1, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_2 - \mu_2, \mu_2)$
			$\alpha_1(1 - \gamma_2)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_3 - \mu_3, \mu_3)$
			α_2	$(2, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 + 1 - \mu_2, \mu_2)$
			$\alpha_h\gamma_2$	$(2, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, \mu_2 + 1)$
			$\alpha_h(1 - \gamma_2)$	$(3, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_3 - \mu_3, \mu_3 + 1)$
		$f_{23}, f_{123}, loc! = 2$	$\alpha_1\gamma_2$	$(1, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_2 - \mu_2, \mu_2)$
			$\alpha_1(1 - \gamma_2)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1 + h_3 - \mu_3, 0, \mu_3)$
			α_2	$(2, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 + 1 - \mu_2, \mu_2)$
			$\alpha_h\gamma_2$	$(2, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, \mu_2 + 1)$
			$\alpha_h(1 - \gamma_2)$	$(3, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(h_3 - \mu_3, 0, \mu_3 + 1)$
		$(r, ir, loc = 2), f_{12}$	α_1	$(3, ir, 1, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, \mu_2)$
			α_2	$(3, ir, 0, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$(3, r, 0, 0, 0, 1, 1, 0, 1, 1)$	
		$(r, ir, loc! = 2), f_{13}$	α_1	$(3, ir, 1, 0, 0, 0, 0, 0, 0, 0)$	$(h_3 - \mu_3, 0, \mu_3)$
			α_2	$(3, ir, 0, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$(3, r, 0, 0, 0, 1, 1, 0, 1, 1)$	
	$action = 0$	f_{12}	α_1	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, r, l_1, l_2, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h\beta_1$	$(2, r, l_1, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h\beta_2$	$(1, r, l_1, l_2, h_1, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
		f_{13}	α_1	$(1oc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2\gamma_1$	$(2, r, l_1, l_2, 0, 1, 1, 0, 0, 0)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_2(1 - \gamma_1)$	$(2, r, l_1, l_2, 0, h_3 + 1, h_3 + 1, \mu_1, \mu_3, \mu_3)$	$(0, 0, 0)$
			$\alpha_h\gamma_1$	$(2, r, l_1, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h(1 - \gamma_1)$	$(2, r, l_1, l_2, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		f_{23}	α_1	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(1, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h\gamma_2$	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h(1 - \gamma_2)$	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	
		f_{123}	α_1	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h\theta_1$	$(2, r, l_1, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h\theta_2$	$(2, r, l_1, 0, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
			$\alpha_h(1 - \theta_1 - \theta_2)$	$(2, r, l_1, 0, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		r, ir	α_1	$(loc, ir, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(loc, ir, l_1, l_2 + 1, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			α_h	$(loc, ir, l_1, l_2, h_1, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_3 + 1)$	
	$action = h_3 - h_1 > 0$	f_{23}	α_1	$(loc, f_{123}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(2, r, l_1 - action, 0, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h\theta_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h\theta_2$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
			$\alpha_h(1 - \theta_1 - \theta_2)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		$r, h_2 = \mu_2$	α_1	$(loc, f_{13}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2\gamma_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_2(1 - \gamma_1)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3, \mu_3)$	$(0, 0, 0)$
			$\alpha_h\gamma_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_h(1 - \gamma_1)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		$r, h_2! = \mu_2$	α_1	$(loc, f_{12}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			α_2	$(2, r, l_1 - action, l_1, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h\beta_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_h\beta_2$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
	$h_1 + action > l_2 + h_2$		α_1	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	$(h1 + action - \mu_1, 0, \mu_1)$
			α_2	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			α_h	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	

TABLE IV
A DESCRIPTION OF THE TRANSITION AND REWARD MATRICES $\mathcal{P}$ AND $\mathcal{R}$ IN THE DECISION PROBLEM $\mathcal{M}_{PO}$.

$l_2 + h_2 - h_3 > 2$	$adopt$	$loc! = 2$	$(1-p)$	$(1, ir, 0, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(1, ir, 1, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, ir, 0, l_2 + 1, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			$\alpha_h p$	$(1, r, 0, l_2, h_3, h_2, h_3 + 1, \mu_3, \mu_2, \mu_3 + 1)$	
		$loc = 2$	$(1-p)$	$(1, ir, 0, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p$	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			α_2p	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			$\alpha_h p$	$(1, r, 0, l_2, h_3 - h_2, 0, h_3 - h_2 + 1, h_3 - h_2, 0, h_3 - h_2 + 1)$	
	$h_1 + action < h_3$		$1 - p$	$(loc, ir, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p$	$(loc, r, l_1 - action, l_2, h_1 + action, h_2, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_1 + action = h_3$	$r, action > 0$ $f_{13}, action = 0$	$(1-p)$	$(loc, f_{13}, l_1 - action, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{13}, l_1 - action + 1, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, f_{13}, l_1 - action, l_2 + 1, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p \gamma_1$	$(1, r, l_1 - action, l_2, h_3, h_2, h_3 + 1, \mu_1, \mu_2, \mu_1 + 1)$	
			$\alpha_h p(1 - \gamma_1)$	$(1, r, l_1 - action, l_2, h_3, h_2, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_3 < h_1 + action < l_2 + h_2 - 1$		$(1-p)$	$(1, ir, l_1 - action, l_2, h_1 + action, h_2, h_1 + action, \mu_1, \mu_2, \mu_1)$	$(0, 0, 0)$
			$\alpha_1 p$	$(1, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_1 + action, \mu_1, \mu_2, \mu_1)$	
			$\alpha_2 p$	$(1, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_1 + action, \mu_1, \mu_2, \mu_1)$	
			$\alpha_h p$	$(1, r, l_1 - action, l_2, h_1 + action, h_2, h_1 + action + 1, \mu_1, \mu_2, \mu_1 + 1)$	
	$h_1 + action = l_2 + h_2 - 1$		$(1-p)$	$(2, ir, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_1 p$	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			$\alpha_2 p$	$(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
	$h_1 + action = l_2 + h_2$		$1 - p$	$(loc, f_{12}, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p \beta_1$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h p \beta_2$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		$(1-p)$	$(3, ir, l_1 - action, 0, 0, 0, 0, 0, 0, 0)$	$(h1 + action - \mu_1, 0, \mu_1)$
			$\alpha_1 p$	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	
$l_2 + h_2 - h_3 = 2$	$adopt$	$loc! = 2$	$(1-p)$	$(1, ir, 0, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(1, ir, 1, l_2, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, ir, 0, l_2 + 1, h_3, h_2, h_3, \mu_3, \mu_2, \mu_3)$	
			$\alpha_h p$	$(2, r, 0, 0, h_3, h_2 + l_2, h_2 + l_2, \mu_3, \mu_2, \mu_2)$	
		$loc = 2$	$(1-p)$	$(1, ir, 0, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p$	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			$\alpha_2 p$	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			$\alpha_h p$	$(2, r, 0, 0, h_3 - h_2, l_2, l_2, h_3 - h_2, 0, 0)$	
	$h_1 + action < h_3$		$(1-p)$	$(loc, ir, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
	$h_1 + action = h_3$	$r, action > 0$ $f_{13}, action = 0$	$(1-p)$	$(loc, f_{13}, l_1 - action, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{13}, l_1 - action + 1, l_2, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, f_{13}, l_1 - action, l_2 + 1, h_3, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
	$h_1 + action = l_2 + h_2 - 1$		$(1-p)$	$(2, ir, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_1 p$	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			$\alpha_2 p$	$(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
	$h_1 + action = l_2 + h_2$		$(1-p)$	$(loc, f_{12}, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p \beta_2$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h p \beta_1$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		$(1-p)$	$(3, ir, l_1 - action, 0, 0, 0, 0, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_1 p$	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	
$l_2 + h_2 - h_3 = 1$	$adopt$	$loc = 2, loc = 3$	$(1-p)$	$(1, ir, 0, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	$(0, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p$	$(1, ir, 1, l_2, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			$\alpha_2 p$	$(1, ir, 0, l_2 + 1, h_3 - h_2, 0, h_3 - h_2, h_3 - h_2, 0, h_3 - h_2)$	
			$\alpha_h p$	$(loc, f_{23}, 0, 0, h_3 - h_2, h_3 - h_2 + 1, h_3 - h_2 + 1, 0, 0, 1)$	
	$h_1 + action < h_3$		$(1-p)$	$(loc, ir, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, ir, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, ir, l_1 - action, l_2 + 1, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p$	$(loc, f_{23}, l_1 - action, 0, h_1 + action, h_3 + 1, h_3 + 1, \mu_1, \mu_2, \mu_3 + 1)$	
	$h_1 + action = l_2 + h_2$		$(1-p)$	$(loc, f_{12}, l_1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{12}, l_1 - action + 1, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p \beta_2$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + l_2 + 1, h_2 + l_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h p \beta_1$	$(2, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
	$h_1 + action > l_2 + h_2$		$(1-p)$	$(3, ir, l_1 - action, 0, 0, 0, 0, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_1 p$	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	

$l_2 + h_2 = h_3$	$adopt$	$f_{23}, f_{123}, loc = 2$	$(1-p)$	$(loc, f_{23}, 0, l_2, 0, \mu_3 - \mu_2, \mu_3 - \mu_2, 0, \mu_3 - \mu_2, \mu_3 - \mu_2)$	$(0, h_3 - \mu_3, \mu_2)$
			$\alpha_1 p \gamma_2$	$(1, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p(1-\gamma_2)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_3 - \mu_3, \mu_3)$
			$\alpha_2 p$	$(2, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 + 1 - \mu_2, \mu_2)$
			$\alpha_h p \gamma_2$	$(2, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, \mu_2 + 1)$
			$\alpha_h p(1-\gamma_2)$	$(3, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_3 - \mu_3, \mu_3 + 1)$
		$f_{23}, f_{123}, loc! = 2$	$(1-p)$	$(loc, f_{23}, 0, l_2, 0, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p \gamma_2$	$(1, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p(1-\gamma_2)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(1 + h_3 - \mu_3, 0, \mu_3)$
			$\alpha_2 p$	$(2, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 + 1 - \mu_2, \mu_2)$
			$\alpha_h p \gamma_2$	$(2, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, 1 + \mu_2)$
			$\alpha_h p(1-\gamma_2)$	$(3, r, 0, 0, 0, 0, 0, 0, 0, 0)$	$(h_3 - \mu_3, 0, \mu_3 + 1)$
		$(r, ir, loc = 2), f_{12}$	$(1-p)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(0, h_2 - \mu_2, \mu_2)$
			$\alpha_1 p$	$(3, ir, 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, 0, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$(3, r, 0, 0, 0, 1, 1, 0, 1, 1)$	
		$(r, ir, loc! = 2), f_{13}$	$(1-p)$	$(3, ir, 0, 0, 0, 0, 0, 0, 0, 0)$	$(h_3 - \mu_3, 0, \mu_3)$
			$\alpha_1 p$	$(3, ir, 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, 0, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$(3, r, 0, 0, 0, 1, 1, 0, 1, 1)$	
	$action = 0$	f_{12}	$(1-p)$	$(loc, fork, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p$	$(1, r, l_1, l_2, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h p \beta_1$	$(2, r, l_1, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h p \beta_2$	$(1, r, l_1, l_2, h_1, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
		f_{13}	$(1-p)$	$(1oc, fork, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(1oc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p \gamma_1$	$(2, r, l_1, l_2, 0, 1, 1, 0, 0, 0)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_2 p(1-\gamma_1)$	$(2, r, l_1, l_2, 0, h_3 + 1, h_3 + 1, \mu_1, \mu_3, \mu_3)$	$(0, 0, 0)$
			$\alpha_h p \gamma_1$	$(2, r, l_1, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h p(1-\gamma_1)$	$(2, r, l_1, l_2, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		f_{23}	$(1-p)$	$(loc, fork, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(1, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	
			$\alpha_h p \gamma_2$	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	
			$\alpha_h p(1-\gamma_2)$	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	
		f_{123}	$(1-p)$	$(loc, fork, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, fork, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p$	$(2, r, l_1, 0, h_1, h_2 + 1, h_2 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h p \theta_1$	$(2, r, l_1, 0, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h p \theta_2$	$(2, r, l_1, 0, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
			$\alpha_h p(1 - \theta_1 - \theta_2)$	$(2, r, l_1, 0, h_1, h_3 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		r, ir	$(1-p)$	$(loc, ir, l_1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, ir, l_1 + 1, l_2, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_2 p$	$(loc, ir, l_1, l_2 + 1, h_1, h_2, h_3, \mu_1, \mu_2, \mu_3)$	
			$\alpha_h p$	$(loc, ir, l_1, l_2, h_1, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_3 + 1)$	
	$action = h_3 - h_1 > 0$	f_{23}	$(1-p)$	$(loc, f_{123}, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{123}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p$	$(2, r, l_1 - action, 0, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h p \theta_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 - \mu_1, 0, \mu_1)$
			$\alpha_h p \theta_2$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2 + 1, \mu_2 + 1)$	$(0, 0, 0)$
			$\alpha_h p(1 - \theta_1 - \theta_2)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		$r, h_2 = \mu_2$	$(1-p)$	$(loc, f_{13}, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{13}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p \gamma_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_2 p(1-\gamma_1)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3, \mu_3)$	$(0, 0, 0)$
			$\alpha_h p \gamma_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_h p(1-\gamma_1)$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
		$r, h_2! = \mu_2$	$(1-p)$	$(loc, f_{12}, l_1 - action, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_1 p$	$(loc, f_{12}, l_1 - action + 1, l_2, h_1 + action, h_2, h_3, \mu_1, \mu_2, \mu_3)$	$(0, 0, 0)$
			$\alpha_2 p$	$(2, r, l_1 - action, l_1, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_2, \mu_2)$	$(0, 0, 0)$
			$\alpha_h p \beta_1$	$(2, r, l_1 - action, l_2, 0, 1, 1, 0, 1, 1)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_h p \beta_2$	$(2, r, l_1 - action, l_2, h_1 + action, h_2 + 1, h_3 + 1, \mu_1, \mu_3 + 1, \mu_3 + 1)$	$(0, 0, 0)$
	$h_1 + action > l_2 + h_2$		$(1-p)$	$(3, ir, l_1 - action, 0, 0, 0, 0, 0, 0, 0)$	$(h_1 + action - \mu_1, 0, \mu_1)$
			$\alpha_1 p$	$(3, ir, l_1 - action + 1, 0, 0, 0, 0, 0, 0, 0)$	
			$\alpha_2 p$	$(3, ir, l_1 - action, 1, 0, 0, 0, 0, 0, 0)$	
			$\alpha_h p$	$((3, r, l_1 - action, 0, 0, 1, 1, 0, 1, 1))$	