Blockchain Mining Games with Pay Forward*

Elias Koutsoupias† Philip Lazos‡ Paolo Serafino§ Foluso Ogunlana¶

May 17, 2019

Abstract

We study the strategic implications that arise from adding one extra option to the miners participating in the bitcoin protocol. We propose that when adding a block, miners also have the ability to pay forward an amount to be collected by the first miner who successfully extends their branch, giving them the power to influence the incentives for mining. We formulate a stochastic game for the study of such incentives and show that with this added option, smaller miners can guarantee that the best response of even substantially more powerful miners is to follow the expected behavior intended by the protocol designer.

1 Introduction

Bitcoin, currently the most-widely used cryptocurrency, was introduced in a 2009 white paper [12] by the mysterious Satoshi Nakamoto as a form of decentralized, distributed, peer-to-peer digital currency. The backbone of the bitcoin protocol is the *blockchain*, a distributed ledger that (ideally) takes the form of a chain where bitcoin *transactions* are stored into *blocks*. Blocks are created by special nodes of the bitcoin network called miners that: (*i*) collect and validate the set of transactions to be included in a block and (*ii*) solve a crypto-puzzle (the so-called *proof of work*) that cryptographically links the newly created (*mined*) block to the tail of the existing blockchain. The main purpose of the blockchain is to solve the problem of *distributed consensus*, where the consensus to be obtained is on the history (and relative order) of the bitcoin transactions.

Since mining new blocks is a computationally intensive (and expensive) task, miners need a reward scheme to keep mining blocks. Bitcoin has two main reward schemes for miners: (*i*) *transaction fees*, that are left by bitcoin users on a voluntary basis (and hence vary in frequency and size) and *coinbase* (or block) rewards, which are constituted by a fixed set amount (currently 12.5Ƀ) of newly minted bitcoins. Coinbase rewards are the only way by which bitcoins can ever be created, and the protocol specifies that (to avoid inflation) coinbase rewards be decreasing over time, until the limit of 21 million bitcoins are created: at that point the only incentive scheme left for miners will be transaction fees. Currently, most of the rewards for mining a block are coinbase.

In a distributed setting populated by selfish miners, however, the blockchain will hardly be a chain at all, but rather a *tree*. For instance, network delays may lead miners to add newly mined blocks to different blocks that they believed to be the tail of the chain. For this

*This work was generously supported by the ERC Advanced Grant 321171 (ALGAME) and by the ERC Advanced Grant 788893 (AMDROMA).

†Department of Computer Science, University of Oxford. Email: elias@cs.ox.ac.uk

‡Department of Computer Science, Sapienza University of Rome. Email: plazos@gmail.com

§Gran Sasso Research Institute. Email: paolo.serafino@gssi.it

¶Consensys. Email: foogunlana@yahoo.com

reason, newly mined blocks are not part of the consensus until sufficient time has passed since their creation (i.e., d levels of other blocks are added to the blockchain – currently $d = 100$), and block rewards become available to miners only then. Worse still, even though the bitcoin protocol prescribes that miners should mine from the last known block in the chain (the so-called FRONTIER strategy), self-interested miners may try to create a *fork* by intentionally adding a sequence of blocks (a *branch*) constituting a parallel history of the transactions, in an attempt to reap more block rewards or to double-spend bitcoins (once per each branch of the fork).

It should be clear from now that the bitcoin protocol is rife with game-theoretic issues. Nakamoto [12] analyzed in a simple model, providing rough estimates showing that if a large majority of miners follow FRONTIER, then their chain will be the longest and contain the agreed upon history.

Since Nakamoto's paper, there has been significant work about the strategies of miners, mainly under the assumption that the reward per block is fixed (see for example [5, 16, 8]). In particular, these *mining games* have been systematically evaluated through the lens of game-theory by Kiayias et al. [8]. They considered mining games in which miners can influence the blockchain in two ways: by strategically choosing to mine at different branches in an attempt to overtake the longest branch and by withholding their mined blocks and releasing them at the right time, wasting everyone else's computational power. In [16, 8], it was shown by both formal proofs and simulations that the Nakamoto protocol is stable when no miner has computation power more than 0.33 of the total. Furthermore, it was also shown that when the miners do not withhold blocks, the protocol is stable if no miner has computation power more than 0.42.

This raises the issue of *fairness* and *compliance* in the bitcoin protocol. By deviating from the protocol and mining strategically, large miners could affect the network in two ways. By increasing their own rewards disproportionately to their computational power they limit the mining rewards claimed by smaller miners, which is unfair. Also, their constant forking in an attempt to add more blocks upsets the protocol and can cause delays, as well as negate transactions that have been already processed.

In this work, we propose a slight modification to the protocol whereby miners can entice other miners to mine at their block by adding a *pay-forward* amount. This pay-forward amount is collected by the miner of the next block, thus providing incentives to other miners to try to extend the tree from this particular branch. When we extend the available strategies for the miners by adding the ability to pay-forward, we reclaim some stability by ensuring that the attacking (computationally powerful) miner is incentivized to follow the protocol, even if he still reaps more than his fair share of rewards.

The main technical contribution is the study of mining games in which the strategies are extended by pay-forward. We show that with this extension, the stability of the protocol increases significantly: the computational power needed by a dishonest miner to disrupt the blockchain is substantially higher. This shows an interesting trade-off for small miners: on one hand they lose the pay-forward amount but on the other hand, they provide the right incentives to large miners to play FRONTIER and thus secure that their block will end up in the longest chain.

1.1 Our Results

We consider two types of *stochastic games*, whose states are rooted trees. The game is played in discrete time-steps. At the beginning of each time-step, every miner chooses a block and tries to mine from it. Each player i has probability p_i to mine a new block, proportional to his computational power. In the end, the new block may be added to the blockchain or kept hidden and released later. After many rounds of playing, the utility of each player is the fraction of bitcoins he owns (from block and pay-forward rewards) over the total value that has been mined in the longest chain.

Let p be the computational power of the strongest player, called Miner 1. If p is large enough, his best response, given that everyone else is playing FRONTIER, may be a different strategy. We find thresholds on p and w, the pay forward amount of every other player, to guarantee that mining at the end of the longest chain is a best response for Miner 1. As in [8], we consider two variants. In the immediate release case, any mined block has to be added to the blockchain immediately for other players to use. In this case, without pay forward rewards, it was proven in [8] that FRONTIER is a best response for Miner 1 for $p < h$, where $0.361 \leq h \leq 0.455$. Experimentally, they also showed that there is always a deviating strategy for $h \approx 0.42$. In the strategic release case, the newly mined blocks are public knowledge, but can only be used by other players when their creators decide to *release them*. As before, it was shown in [8] that there is a threshold $\hat{h} \geq 0.308$ (experimentally shown to be approximately 0.33 [8, 16]) such that $p < \hat{h}$ leads to FRONTIER as a best response for Miner 1.

We improve these thresholds to $h = 0.5$ and $\hat{h} \geq 0.344$ by showing that there exists some w (as a function of p) that guarantees Miner 1's best response is to play FRONTIER when all remaining miners play FRONTIER and pay forward w. In addition, we experimentally show that for $h \leq 0.44$ that outcome is also a pure Nash equilibrium for the immediate release case, while for the strategic release case we find that $\hat{h} \approx 0.38$. We also devise a linear program to calculate the minimum value of w for any given p.

1.2 Related Work

Bitcoin, originally introduced by Nakamoto in [12] and followed by several cryptocurrencies, such as Litecoin, Ethereum and Ripple, initiated the research on blockchains. Nakamoto's original paper analyses double spending attacks while Rosenfeld provides a more detailed analysis [15]. Bonneau et al. [2] and Tschorsch and Scheuermann [20] provided extensive surveys of the research and challenges in cryptocurrencies.

Kroll et al. [10] was one the first papers to consider the economics of Bitcoin mining, assuming that participants behave according to their incentives. Eyal and Sirer [7] showed that the security of bitcoin is not guaranteed by a majority of honest miners as was previously assumed. They gave a specific mining strategy and argued that a pool of miners, with at least a 1/3 of the total processing power, can get extra profit regardless of the block propagation characteristics of the network. With sufficiently favorable block propagation, this threshold of processing power falls to 0. Extending this, Sapirshtein et al. [16] provided systematic analysis of the space of selfish mining strategies based on computational results.

A similar approach was taken by Kiayias et al. [8] who provided a framework for studying the strategic considerations made by Bitcoin miners. They formulated two abstract stochastic games and proved rigorous bounds on the threshold of computational power below which the honest strategy is a Nash Equilibrium. Carlsten et al. [3] extended the previous approaches by investigating mining games when the reward for miners varies and comes mainly from transaction fees. They observed that the random block arrival times lead to high variance in rewards and they considered strategies that lead to instability. They showed via simulation that an equilibrium exists, but it has the counter intuitive and undesirable effect of a growing backlog of unprocessed transactions.

Several other studies examine possible attacks on the protocol and suggest adaptations to ensure its security. Rosenfeld [14] and Courtois and Bahack [4] discuss pool mining attacks. Eyal [5] introduces an attack in which mining pools infiltrate one another resulting in a pool game. Lewenberg et al. [11] also provide a game theoretic analysis of pool mining. Babaioff et al. [1] consider Sybil attacks on the network and propose a reward scheme to prevent miners from hiding transactions in competition with other miners.

There is also considerable work on the performance and scalability of blockchain inspired networks. Sompolinsky and Zohar's [18] Greedy-Heaviest-Observed-Sub-Tree rule is an alternative consensus mechanism. Eyal [6] proposes Bitcoin-NG which increases the throughput of Bitcoin. Poon and Dryja's Lightning Network [13] scales via off-chain transactions and hashed commitments. Sompolinsky and Zohar's PHANTOM [19] achieves near unlimited transaction throughput. SPECTRE [17], by Sompolinsky et al, fully orders the transactions in blocks using recursive elections and can used in combination with PHANTOM. Kiayias et al. developed Ouroboros[9], a proof of stake network with provable security, utilising a secure multi-party coin flipping protocol.

2 Model and Notation

The model contains subtleties in the abstractions that at first glance might be overlooked by readers who are too familiar with the bitcoin protocol. We do our best to point out any of those instances, but be especially vigilant when reading about the way payoffs are calculated.

The bitcoin mining game with pay forward is an abstraction of the actual protocol, simplified in a way that can only accentuate its game theoretic issues (i.e. negative results hold a fortiori for the actual protocol) while being open to rigorous analysis. The parameters of the game are:

1. the number n of miners (or players).

2. the probabilities $p_1, \ldots, p_n$, representing the *hash power* of each miner, which is the probability that they solve the crypto-puzzle. They are proportional to the computational power of each miner and such that: $p_i \geq 0$ for each i and $\sum_{i=1}^{n} p_i = 1$.

3. the depth d after which mined blocks are 'paid', i.e. their coinbase becomes usable by the respective miner. Without loss of generality we mostly consider $d = \infty$ which gives the attacker extra power.

We assume that each block has a *fixed reward*, which by appropriate scaling is equal to 1. This is reasonable, as the large majority of rewards come from the fixed coinbase rewards. The fluctuating transaction fees (while not insignificant) are still comparatively small.

During the execution of the protocol, miners add their blocks to the blockchain one at a time[1]. Their goal is to maximize the fraction of their blocks on the longest branch of the blockchain, as the longest branch represents the consensus and all other branches are pruned and the computational power spent for their creation is wasted. This does not mean that all the miners try to extend the longest branch at every time step, because some times they might benefit by either adding a block to a shorter branch or withholding it for later.

Definition 1. A public state is a rooted tree. Every node is labelled by one of the players and the amount of money he pays-forward. The nodes represent mined blocks and the label indicates the player who mined the block. The pay-forward amount is collected by the next miner in the longest branch. Every level of the tree has at most one node labelled i because there is no reason for a player to mine twice the same level.

[1] Due to the decentralized nature of bitcoin, it is possible for more than one miners to mine a block simultaneously. This event does occur by chance, but it is very rare and it is not something a miner can plan for. For this reason, we assume that exactly one block is mined at every step, but most of our analysis could be easily extended to the case in which each miner succeeds with probability p_i independently.

A private state of player i is similar to the public state except it may contain more nodes called private nodes and labelled by i. The public tree is a subtree of the private tree and has the same root.

The incomplete information case (where the p_i's or private states are unknown) is significantly more complicated. In this work we only consider the full information case where even the private states are common knowledge (but only become part of the public state when their respective miners add them). We consider two variants:

Immediate-release model Every mined block is added to the public tree immediately.

Strategic-release model Mined blocks can be withheld: every miner is aware of their existence, but they can only mine from them once they are added to the public tree.

The second model has no counterpart in practice, but it serves as an intermediate step between the full and incomplete information models and allows us to study issues around strategic release of blocks. If a strategy is not dominant in this model it cannot be dominant in the incomplete information setting.

Definition 2 (Strategy)**.** The strategy of miner i can be fully characterized by three functions $\mu_i, r_i, \mathrm{PF}_i$:

- the mining function μ_i selects a block of the public state to mine from
- the release function r_i which is the rooted tree he releases to the public. It is a subtree of his private state that contains the public state.
- the pay-forward function PF_i which is the amount of money to be left for the next miner.

Both functions depend on the public and private states of every player.

The original suggested strategy in [12] is FRONTIER.

Definition 3 (FRONTIER)**.** A miner follows strategy FRONTIER if he releases mined blocks immediately, always mines at the deepest node in the blockchain and pays forward 0 Ƀ.

In the following, we refer to FRONTIER(w) as the strategy that mines and releases like FRONTIER but always pays forward w.

The game is played in *phases*. At each phase exactly one miner will mine a block. Then he will choose if he wants to release it, which may trigger a cascade of releases from other players. Eventually, once no one else wants to release anything the public knowledge is updated and the next phase begins. Even if no one releases a block, miners know when the phase ends since everything is public knowledge.

To incentivise miners to mine new blocks, payments are necessary. When a block is added, it is unclear if it is going to remain on the longest branch permanently. To remedy this, blocks are paid for only after their branch is increased by d blocks, after which time it is safe to assume it will stay in the longest branch. Blocks are paid through coinbase, transaction fees and pay-forward. Currently, coinbase rewards (which can be claimed at $d = 100$) far outweigh transaction fees. However, by design Bitcoin will eventually do away with coinbase rewards to limit inflation, at which point only transaction fees will remain, leading to potential instability [3]. We only consider coinbase rewards in this work. For game theoretic analysis we rigorously define payments:

Definition 4 (Payments)**.** For some nodes of the tree, the miners who discovered them will get a payment (coinbase rewards are normalized to 1). The payments comply with the following rules:

- the blocks that receive a payment must form a path from the root. This immediately adds the restriction that at every level of the tree exactly one node receives payment.
- among the blocks of a single level that satisfy the above condition, the first one which succeeds in having a descendant d generations later receives payment.

Since only one block per level is paid for and they form the longest branch, the utility of a miner in the long run is defined as the fraction of pay-forward and coinbase rewards he obtained minus the pay-forward he paid in that branch, over it's length.

When a block is paid for, strategic miners will ignore any branch that starts at an earlier node. This limits the shape of the public state to a long path (the *trunk*) with ignored *stale* branches dangling from it. Only at the end of the trunk we may find multiple active branches, competing to reach length d and render the others stale. We will also consider the case where $d = \infty$. Here there is the possibility that two competing branches will go on forever, but since one of them will have less computational power, a case of gamblers ruin makes this impossible.

3 Immediate Release

In this section we show that if FRONTIER(w) (for some value of w) is followed by all miners but one, the remaining miner's best response is FRONTIER, provided his hash power is $p < 0.5$. Since only one miner might deviate, we can view this situation as Miner 1 being the potentially deviant large miner with hash power $p < 0.5$ and Miner 2 all small miners combined into one, which we call the honest miners, since they follow the same strategy.

In essence, every honest player mines at the deepest node and always pays forward w. This amount propagates across the small miners forming Miner 2 but Miner 1 can claim it for himself, leading to an interesting stochastic game where Miner 1 is incentivise to mine the longest chain, since the rewards don't accumulate.

The blockchain itself is a rooted tree, but after pruning abandoned branches we are left with one long, undisputed, main path followed by two branches of length a and b, mined by Miner 1 and Miner 2 respectively. Therefore, the possible states of the game have the form (a, b, c), where $a \leq b+1$ since Miner 2 always mines the longest branch (except briefly, right when Miner 1's branch overtakes the longest) and c is either 0 or 1, to indicate whether the *last block before the fork* contained a pay forward reward.

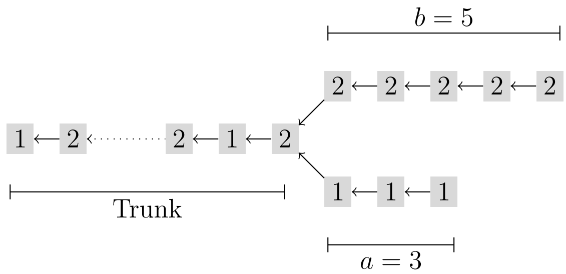

Figure 1: A typical state tree. The trunk represents the blocks whose rewards have already been collected. The current state $(3, 5, 1)$ of the game is represented by the blocks mined by Miner 1 and Miner 2 and $c = 1$ because Miner 2 controls the block before the fork.

The set of states (a, b, c) can be partitioned as follows:

1. **Winning states**: the set W of states where Miner 2 capitulates and starts mining from the tip of Miner 1's branch. This happens exactly when $a = b + 1$, therefore $W = \{(b + 1, b, c) \mid b \geq 0 \text{ and } c \in \{0, 1\}\}$. After Miner 1 overtakes, the new state of the game is $(0, 0, 0)$ since they both mine at the same point.

2. **Capitulation states**: the set C of states where Miner 1 capitulates, abandons his branch and mines from a block of Miner 2's branch, thus moving the game to state $(0, s, 1)$ for some s s.t. $0 \leq s < b$. We say that Miner 1 capitulates *at* (a, b, c) and *to* $(0, s, 1)$. Clearly, after capitulating there would be a pay forward reward available.

3. **Mining states**: the set M of states where Miner 1 and Miner 2 mine their respective branches.

Miner 1 can capitulate to any state $(0, s, 1)$ and will always choose the one that maximizes his payoff. Since he is rational, when capitulating from state (a, b, c) he would only go to states $(0, s, 1)$ with $s < b$, otherwise he would be undercutting his own tentative branch. The strategy FRONTIER has $M = \{(0, 0, 0), (0, 0, 1)\}$, $C = \{(0, 1, 0), (0, 1, 1)\}$ and always capitulates to $(0, 0, 1)$.

Let $g_k(a, b, c)$ denote the optimal expected gain of Miner 1 starting from state (a, b, c) when the longest chain is extended by k levels. Knowing Miner 1 will never pay-forward, we can recursively define:

$$g_k(a, b, c) = \begin{cases} g_{k-1}(0, 0, 0) + a + w \cdot c & \text{if } a = b + 1 \\ \max \begin{cases} \max_{s=0,\ldots,b-1} g_k(0, s, 1) \\ p g_k(a + 1, b, c) + (1 - p) g_{k-1}(a, b + 1, c) \end{cases} & \end{cases} . \tag{1}$$

The first line represents Miner 2 capitulating to Miner 1's chain. The second line containing the max has two cases. In the first case, Miner 1 capitulates to the optimal point within Miner 2's chain. In the second, they both mine and extend their chains with probability p and $(1 - p)$ respectively. Notice that the first term is $pg_k(a+1, b, c)$ and not $pg_{k-1}(a+1, b, c)$. Since $a < b+1$, if Miner 1 adds a block the deepest level of the blockchain does not change, thus k remains the same.

Clearly, Miner 1 eventually will add some block to the chain and the game will restart at state $(0, 0, 0)$. Therefore, we expect that the initial state (a, b, c) has no effect asymptotically. We define the expected gain per level $g^\star$ as:

$$g^\star = \lim_{k \to \infty} \frac{g_k(a, b, c)}{k}. \tag{2}$$

We can decompose $g^\star$ into two separate quantities:

$$g^\star = q_M + q_{\mathrm{PF}} \cdot w, \tag{3}$$

where q_M is the expected fraction of blocks he mined and q_{PF} the expected fraction of blocks mined and containing pay forward rewards.

Lemma 1. *If Miner 1 follows* FRONTIER*, we have $q_M = p$ and $q_{\mathrm{PF}} = p(1 - p)$.*

Proof. If Miner 1 is honest then the probability that he mined any arbitrary block is p. That block also contains pay forward rewards if the one that immediately preceded it was mined by Miner 2, which occurs with probability $p(1 - p)$. □

We are ready to state the main theorem of this section.

Theorem 1. *For every $p < 0.5$, there exists $w \geq 0$ large enough so that if every miner but one follows* Frontier(w)*, the best response of the remaining miner with hash power p is* Frontier.

Proof. We will show that q_{PF} attained by Frontier is at least as large as that of any optimal strategy, and in particular it is larger for $0 < p < 0.5$.

Lemma 2. *If* Frontier *is not the optimal strategy and $0 < p < 0.5$, we have that $p(1-p) > q_{\mathrm{PF}}$. For $p = 0$ or $p = 0.5$ we have equality.*

Proof. Assuming Frontier is not an optimal strategy, we consider different cases, depending on the actions of the optimal strategy starting from state $(0, 1, c)$.

Clearly, we have that $g_k(0, 1, 1) \geq g_k(0, 1, 0)$ for all k. To see this, consider what happens if we apply exactly the same sequence of actions starting from $(0, 1, 0)$ and from $(0, 1, 1)$. As the only difference between the two states is the initial pay forward amount, if we start from $(0, 1, 1)$ the reward will be the higher (in expectation) by some fraction of w. Therefore, if the optimal strategy capitulates at $(0, 1, 1)$ it must also capitulate at $(0, 1, 0)$. This is exemplified in Figure 2, where blocks with a red outline represent those mined by the optimal strategy, and red arrows show how the action of the optimal strategy in one case implies the same action in the other.

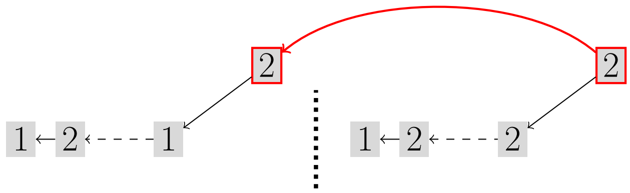

Figure 2: Mining like Frontier

But in this case, the optimal strategy capitulates at exactly the same states as Frontier, as shown in Figure 2. Since the blockchain starts empty, at $(0, 0, 0)$ this strategy would behave exactly the same as Frontier and would therefore not be optimal.

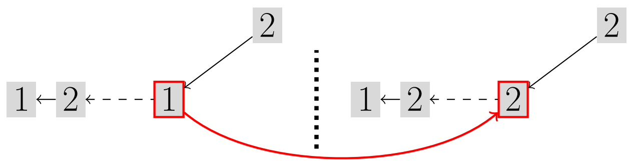

Figure 3: Case 1

In Case 1 (shown in Figure 3), we assume that the optimal strategy mines at $(0, 1, 0)$. As before, since $(0, 1, 1)$ is a more beneficial state it would mine from there as well. We can therefore assume that the optimal strategy always capitulates to states $(0, s, 1)$ with $s \geq 1$. Blocks are permanently added to the blockchain only after a capitulation. Given that, in order to compute q_{PF} for Miner 1, we construct the Markov chain shown in Figure 4, with states $(0, 0, 0)$ and $(0, s, 1)$ indicating that Miner 2 or Miner 1 capitulated respectively. Each transition is labelled with a pair where the first element is the transition probability, whereas the second element is the minimum number of blocks added to the chain when the transition occurs.

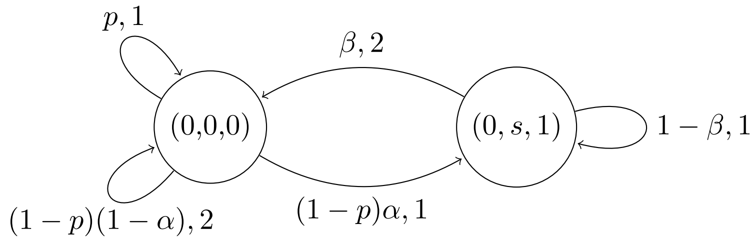

Figure 4: Case 1 Markov chain

Starting from $(0,0,0)$, with probability p Miner 1 adds a block and the game restarts. With probability $1-p$ we move to state $(0,1,0)$ and the optimal strategy has probability α of losing the race and capitulating to some state $(0,s,1)$. From there, the optimal strategy has probability β of winning and returning to $(0,0,0)$.

As q_{PF} is an asymptotic quantity, we are interested in the fraction of pay forward rewards per block mined at the stationary distribution. Let π be the stationary probability of state $(0,s,1)$. Then:

$$\pi = \frac{\alpha - \alpha \cdot p}{\alpha - \alpha \cdot p + \beta} \tag{4}$$

In the actual game, multiple blocks are potentially mined in each edge transition, and only the first block might claim a pay forward reward. On every transition at least one block is permanently added, *except from* $(0,s,1)$ *to* $(0,0,0)$ *and* $(0,0,0)$ *to itself through transition* $(1-p)(1-\alpha)$. In these cases at least two blocks are added, since Miner 1 must mine at least two blocks to be ahead of Miner 2. From $(0,s,1)$ to $(0,0,0)$ is the only transition where Miner 1 wins a block containing pay forward. Dividing the blocks with pay-forward over the total amount mined at the stationary distribution:

$$q_{\mathrm{PF}} \leq \frac{\pi\beta}{(1-\pi)(2(1-\alpha)(1-p) + \alpha(1-p) + p) + \pi(2\beta + 1 - \beta)} \tag{5}$$

We now need to set the parameters α and β optimally to maximize the upper bound on q_{PF}. We take the derivative of the upper bound of q_{PF} with respect to α and β to get:

$$\frac{dq_{\mathrm{PF}}}{d\alpha} = \frac{\beta^2(2-p)(1-p)}{(\alpha(p-1) + \beta(p-2))^2} \geq 0 \tag{6}$$

and

$$\frac{dq_{\mathrm{PF}}}{d\beta} = \frac{\alpha^2(1-p)^2}{(\alpha(p-1) + \beta(p-2))^2} \geq 0. \tag{7}$$

From [8, Lemma 1], we know that the probability that Miner 1 reaches a winning state starting from (a,b) is at most $(p/(1-p))^{(b-a+1)}$. Therefore, the optimal strategy must have

$$\alpha \leq 1 \text{ and } \beta \leq \left(\frac{p}{1-p}\right)^{s+1} \leq \left(\frac{p}{1-p}\right)^2 \tag{8}$$

Since q_{PF} is an increasing function of α, β, the upper bound is obtained by setting them to their highest possible value, leading to:

$$q_{\mathrm{PF}} \leq \frac{(1-p)p^2}{1-(1-p)p(3-2p)}. \tag{9}$$

Compared to the honest outcome $p(1-p)$, we have:

$$p(1-p) - \frac{(1-p)p^2}{1-(1-p)p(3-2p)} = \frac{(1-p)p(1-2p)}{1-(1-p)p(3-2p)} \geq 0, \tag{10}$$

with equality obtained only for $p = 0$ and $p = 0.5$. We also plot this result at the end of the proof.

In the last case, the optimal strategy capitulates at $(0, 1, 0)$ but mines at $(0, 1, 1)$.

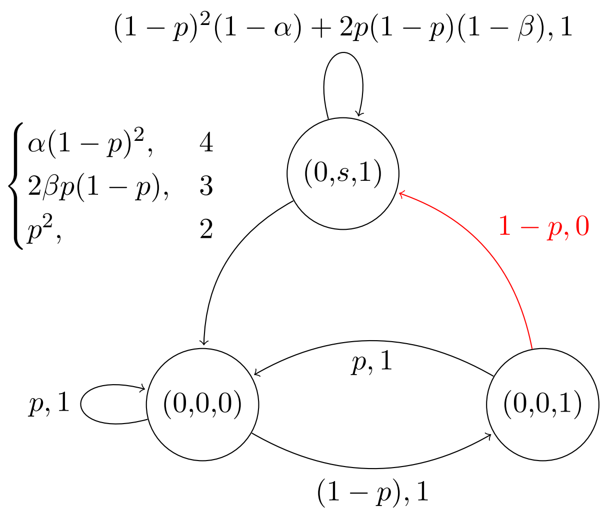

Figure 5: Case 2 Markov chain

The Markov chain in this case (Figure 5) is trickier. As before each state represents a capitulation: $(0, 0, 0)$ for Miner 2 and $(0, 0, 1)$, $(0, s, 1)$ with $s \geq 1$ for Miner 1. However, the transition indicated with the red arrow is *not* a capitulation: it is merely a move to a state that Miner 1 could also capitulate *to*. To more accurately model the transitions of state $(0, s, 1)$, we consider multiple moves ahead. Specifically, three outcomes could happen before the optimal strategy considers capitulating. With probability p^2, Miner 1 could add two blocks and move to $(0, 0, 0)$. Miner 2 could also add two blocks and move to $(0, s+2, 1)$, with probability $(1-p)^2$. Then, Miner 1 wins with probability α. Finally, Miner 1 and Miner 2 could both add one block leading to $(1, s+1, 1)$ with probability $2p(1-p)$. From there, Miner 1 wins with probability β.

Carefully adjusting for the more complicated transitions, the same analysis as before holds. For both cases, we plot the difference $p(1-p) - q_{\text{PF}}$ in Figure 6.

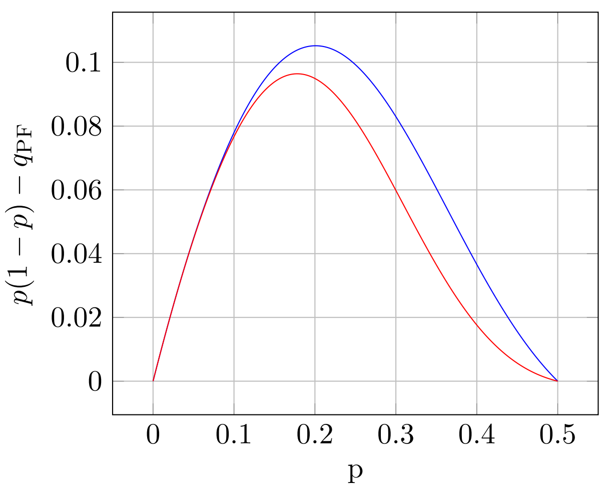

Figure 6: Advantage of FRONTIER in collecting q_{PF} for Case 1 (blue) and Case 2 (red)

$\square$

Therefore, by (3) and Lemma 1 we have that

$$g^\star = q_M + q_{\mathrm{PF}} \cdot w \leq p + (1-p) \cdot w \Rightarrow w \geq \frac{q_M - p}{(1-p) - q_{PF}}$$

and since $(1-p) > q_{PF}$ for $0 < p < 0.5$ there must be some value of w large enough to make Frontier the best response. □

Unfortunately, this result is not enough to calculate w. Giving crude upper bounds on q_M or q_{PF} independently is not too hard, we can use [8, Lemma 1] from example. However, because the trade-off between the two is hard to establish and the actual blockchain does not use $d = \infty$, it is more useful to develop an algorithm that finds the minimum w required for a specific d, to any degree of accuracy.

3.1 Calculating the optimal w for finite d

In reality, the values of w needed are not too large. We could attempt to find the minimum w necessary by computing $g_k(0,0,0)/k$ for large values of k and comparing with $p + p(1-p)w$, but a naive implementation would take $O(k^3)$ time. To simplify this search we can define a potential as in [8]

$$\phi(a,b,c) = \lim_{k\to\infty} g_k(a,b,c) - k \cdot g^\star, \tag{11}$$

that captures the advantage of Miner 1 at different states. This quantity is bounded, because irrespective of the initial state (a,b,c) the game repeats: in particular, when Miner 1 gets ahead of Miner 2 they both restart at state $(0,0,0)$. Therefore, the gain per k asymptotically converges to $g^\star$ and ϕ is well defined. Following recurrence (1), we have:

$$\phi(a,b,c) = \begin{cases} \phi(0,0,0) + a + c \cdot w - g^\star & \text{if } a = b+1 \\ \max \begin{cases} \max_{s=0,\ldots,b-1} \phi(0,s,1) \\ p \cdot \phi(a+1,b,c) \\ \quad +(1-p) \cdot (\phi(a,b+1,c) - g^\star) \end{cases} & \end{cases}, \tag{12}$$

where we set $\phi(0,0,0) = 0$. Finding a ϕ that satisfies these constraints is even harder. However, if we truncate the game at depth d, it becomes more feasible through linear programming. Notice that (12) (which holds for $d = \infty$) does not explicitly define the value of $\phi(a,b,c)$: the recursion is unbounded. By truncating the game at d we limit the available states to $a, b \leq d$. Specifically, Miner 1 *has* to capitulate when $b = d$.

The exact value of d has a small effect on the game. In particular, Miner 1 has more options as d increases, since he does not have to capitulate early. Therefore, the minimum w needed for compliance is a slightly increasing function of d. The effect of this is quite muted however; the probability of having two branches with length greater than d, without Miner 1 winning or capitulating, decreases exponentially in d. In this section we select $d = 8$, as it is large enough to show the trend, but not too large to render the LP computationally infeasible.

Relaxing the two maxes by inequalities, we get the following LP:

$$\begin{array}{ll} \text{minimize} & g + \frac{1}{D}\sum_{a=0}^{d}\sum_{b=0}^{d}\sum_{c=0}^{1}\phi(a,b,c) \\ \text{subject to} & \\ \quad \phi(b+1,b,c) \geq \phi(0,0,0) + b + 1 + c\cdot w - g & \\ \qquad \text{for } b < d, c \leq 1 & \\ \quad \phi(a,b,c) \geq \phi(0,s,1) & \\ \qquad \text{for } a \leq b < d, s < b, c \leq 1 & \\ \quad \phi(a,b,c) \geq p\cdot\phi(a+1,b,c) + (1-p)(\phi(a,b+1,c) - g) & \\ \qquad \text{for } a \leq b < d, c \leq 1 & \end{array},$$

where $\phi(a,b,c) \geq 0$ and $D \gg d^2$ is a normalizing factor to keep the sum of states insignificant compared to g. The constraints are straightforward enough, but the objective is a minimization, which might appear odd at first. The second term of the sum is only there to ensure that feasible solutions only contain potential functions ϕ that tightly satisfy (12). Since many states could be unreachable (for example, if w is high enough so that Miner 1 plays FRONTIER, the only mining states would be $(0,0,0)$ and $(0,0,1)$), their exact value does not affect g and we need this 'extra' term to keep them in check, just to bias the LP towards certain feasible solutions. We minimize g to use the following lemma:

Lemma 3. *For any g, ϕ pair satisfying (12) we have:*

$$g_k(a,b,c) \leq \phi(a,b,c) + k\cdot g.$$

Proof. We use induction on k. Clearly, for $k=0$ we have $g_0(a,b,c) = 0 \leq \phi(a,b,c)$. For fixed k we do strong induction on b and backwards induction on a. Starting with $b=0$, we begin the induction on $a = b+1$:

$$\begin{aligned} g_k(1,0,c) &= g_{k-1}(0,0,0) + 1 + c\cdot w \\ &\leq \phi(0,0,0) + (k-1)g + 1 + c\cdot w \\ &= \phi(0,0,0) + 1 + c\cdot w - g + k\cdot g \\ &= \phi(1,0,c) + k\cdot g \end{aligned}$$

and

$$\begin{aligned} g_k(0,0,c) &= p\cdot g_k(1,0,c) + (1-p)g_{k-1}(0,1,c) \\ &\leq p(\phi(1,0,c) + k\cdot g) + (1-p)(\phi(0,1,c) + (k-1)g) \\ &= p\cdot\phi(1,0,c) + (1-p)(\phi(0,1,c) - g) + k\cdot g \\ &= \phi(0,0,c) + k\cdot g. \end{aligned}$$

For $b > 0$ the proof works the same, starting from $g_k(b+1,b,c)$ as the base case for a. For $a < b+1$:

$$\begin{aligned} g_k(a,b,c) &= \max\begin{cases} \max_{s=0,\ldots,b-1} g_k(0,s,1) \\ pg_k(a+1,b,c) + (1-p)g_{k-1}(a,b+1,c) \end{cases} \\ &\leq \max\begin{cases} \max_{s=0,\ldots,b-1} \phi(0,s,1) + kg \\ p\cdot(\phi(a+1,b,c) + kg) \\ \quad +(1-p)(\phi(a,b+1,c) + (k-1)g) \end{cases} \end{aligned}$$

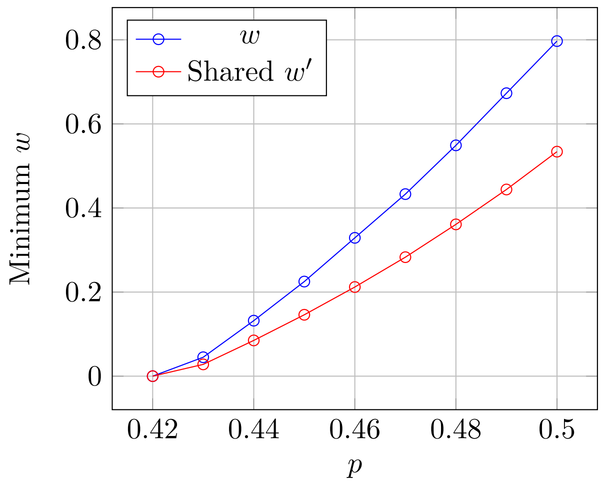

Figure 7: Minimum w for the immediate release case.

$$= k \cdot g + \max \begin{cases} \max_{s=0,\ldots,b-1} \phi(0, s, 1) \\ p \cdot \phi(a+1, b, c) \\ \quad +(1-p)(\phi(a, b+1, c) - g) \end{cases}$$
$$= \phi(a, b, c) + kg,$$

where strong induction was used for the capitulation case. □

Using this lemma, we can find the minimum value of w that leads to Frontier being a best response for different values of p by using binary search on w, checking that the ϕ produced from the LP satisfies (12) and that $g = p + p(1-p)w$. By the definition of $g^\star$:

$$g^\star = \lim_{k\to\infty} \frac{g_k(a,b,c)}{k} \leq \lim_{k\to\infty} \frac{\phi(a,b,c) + k \cdot g}{k} = p + p(1-p)w,$$

For $d = 8$, we obtain the graph in Figure 7.

It is not a mistake that $w < 1$ for $p = 0.5$. The reason is that d is finite. For $d = \infty$, the strategic miner with $p \geq 0.5$ never has a reason to capitulate, no matter w: the probability that his branch eventually overtakes the honest one is always quite high. However, the probability that he will win the race *within d steps* is much smaller, leading to this result.

If not all honest miners are required to pay forward exactly w, they can share the costs. In particular, one strategy which also makes Miner 1 comply with the protocol is for the first honest miner after Miner 1 to pay forward w' and every other honest miner to pay forward $2w'$, until Miner 1 adds a block and claims the reward for himself, where the process restarts. As we can see from the Figure 7, although the total amount paid forward is larger (as $2w' \geq w$), the maximum contribution of any honest miner is $w' \leq w$. We can verify compliance is indeed a best response for Miner 1 by using a slightly modified LP.

3.2 Pure Nash Equilibria for small miners

So far, we have only considered the best responses of Miner 1, given a society of honest, identical miners and identified the minimum value w needed to ensure his compliance. The natural question is to consider if this also a best response from the other miners. Clearly, when treated as a group this is not their best response. If they do not pay forward any amount, Miner 1 would have an average gain of $g' \geq p$, by mining strategically. By paying forward enough, they need to *increase* his gain to $g^\star \geq g'$, as he can always employ strategic mining while they also pay forward. Therefore, it is more costly for Miner 2 to pay forward w and ensure compliance than to pay forward nothing and allow strategic mining.

We will argue however that under certain conditions paying forward can be a pure Nash equilibrium, if Miner 2 consists of many *small* miners that are treated individually. A miner is considered *small* if his chance of mining a block is negligible. Of course at every step some small miner will mine a block with probability $1-p$. To see why paying forward is an equilibrium, consider the options of a small miner at the *exact* moment when he mines a block: he can either pay forward and guarantee it is included in the chain or keep everything for himself, but risk getting undercut by Miner 1. Being a small miner, his payoff from mining another block in the future is insignificant compared to the current reward at stake. Formally, the following strategy, called StrictFrontier(w), is part of a PNE:

Definition 5 (StrictFrontier)**.** Every miner (except Miner 1) pays forward w and mines at the end of the longest chain that contains blocks with pay forward values either w or 0

Miner 1's strategy is to mine at the frontier without paying forward, as before. Note, that paying forward some value other than w or 0 *cannot* be a best response, since that block will be ignored by the honest miners, who have the majority of the computational power[2]

Theorem 2. *For $d = 8$ and any $p \leq 0.44$, there exists w such that Miner 1 playing* Frontier *and every other small miner playing* StrictFrontier(w) *is a pure Nash equilibrium.*

Proof sketch. From the perspective of the small miner, he has two opportunities *not* to pay forward: after another small miner or after Miner 1. We can modify the LP from Section 3.1 to contain both these deviations by the small miner. Examining the tightness of the constraints, we can identify the mining states of Miner 1. Specifically, for $p \leq 0.44$, we observe that in both cases, after the small miner adds his block without paying forward, Miner 1 immediately forks the chain behind the small miner and starts mining in parallel. In particular, states $(0,1,c),(1,1,c),(1,2,c)$ and $(2,2,c)$ become mining states and $(0,3,c),(1,3,c),(2,3,c)$ capitulation states, for $c \in \{0,1\}$. Essentially he mines while one step behind and capitulates if the honest miners add 2 more blocks. Overall, he overtakes the honest branch with probability at least $p^2 + (1-p)p^3$.

Therefore, the expected payoff of the small miner is at most $(1+w)(1-p^2-(1-p)p^3)$, compared to at least $1-w$ which would have been his payoff had he paid forward w (since in this case he would always keep his mined block). Plugging in the p, w pairs found in Figure 7 we verify that StrictFrontier(w) is indeed a best response. □

It would be very interesting to see if there exists a strategy profile that is a PNE where Miner 1 plays Frontier for $0.45 < p < 0.5$. The analysis seems quite challenging and would likely require different techniques to analyze the emerging stochastic game. Ideally, every small miner comprising Miner 1 would use the same w, but it could be the case that better bounds could be obtained if the pay forward amount use depends on the pay forward received from the previous block.

4 Strategic Release

Similarly to the immediate release case, we identify the maximum hash power p such that if all miners but one follow Frontier(w), the remaining miner's best response is Frontier if his

[2] The reason StrictFrontier accepts pay forward of 0 is just to incentivize Miner 1 to cooperate. Without it, every block would have the same pay forward, negating any effects this may have and reducing to the classic bitcoin protocol.

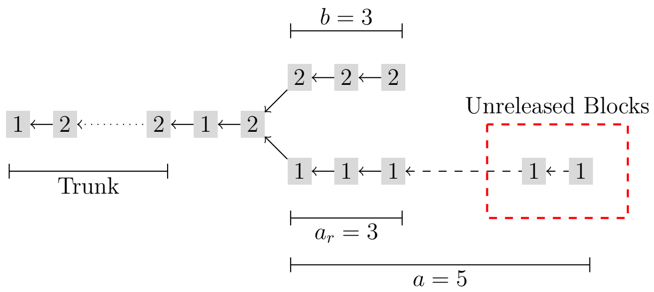

Figure 8: This tree represents state $(3, 5, 3, 1)$. Miner 1 has already mined 5 blocks ahead of the fork, but has only released 3 of them. Miner 2 knows this because the game is complete information. However, he cannot mine there until Miner 1 releases his blocks. Miner 1 can strategically release blocks right when Miner 2's branch is about to become the main one, thus wasting Miner 1's computational power.

hash power is less than p. As before, it is enough to consider a two player game where Miner 1 is the deviant miner with hash power $p < 0.5$ and Miner 2 represents all small miners, as they follow the same strategy.

The blockchain retains its structure: it still contains a long trunk of blocks followed by two branches, corresponding to the released blocks of Miner 1 and Miner 2. Since Miner 1 could also have more unreleased blocks in his branch the state is now a quadruple (a_r, a, b, c) where a is the number of blocks in Miner 1's branch, of which a_r have been released, and b is the number of blocks in Miner 2's branch. Also, contrary to the immediate release case we can have $a > b + 1$.

Since Miner 2 follows the FRONTIER(w) strategy, he will capitulate if $a_r \geq b+1$ and continue mining his branch for $a_r < b + 1$. Therefore, without loss of generality we can assume that at state (a_r, a, b, c) if $a < b + 1$ then $a_r = a$ and if $a \geq b + 1$ then $a_r = b$, otherwise Miner 2 would immediately capitulate and the game would continue at state $(0, a - a_r, 0, 0)$. Therefore, we can encode the states of the mining game with strategic release by the triplet (a, b, c), where $a_r = \min(a, b)$.

As before, we need to recursively define the gain of Miner 1 after the longest branch is extended by k levels. When $a < b + 1$, the actions of Miner 1 are exactly the same as in the immediate release case: he can either capitulate to $(0, s, 1)$ or mine. However, for $a \geq b + 1$ he can release one more block to force Miner 2 to capitulate and continue at state $(a - b - 1, 0, 0)$.

$$\hat{g}_k(a,b,c) = \max \begin{cases} k + b + c \cdot w & \text{if } a \geq k + b \\ \max \begin{cases} \max_{s=0,\dots,b-1} \hat{g}_k(0,s,1) \\ p\hat{g}_k(a+1,b,c) \\ \quad +(1-p)\hat{g}_{k-1}(a,b+1,c) \\ \hat{g}_{k-1}(a-b-1,0,0) \\ \quad +b+1+c \cdot w \end{cases} \end{cases} \tag{13}$$

where the last term applies for $a \geq b + 1$. Equivalently, we can define $g_k(a, 0, c) = -\infty$ for all k and $a < 0$. The first term is necessary, as without it the recurrence is ill-defined and Miner 1 can keep mining forever. Once he reached the horizon k he can safely release all his blocks. We define the expected gain per level $\hat{g}^\star$ and potential $\hat{\phi}$ as we did in (2) for the immediate release. Note that the case where $a = k + b$ does not appear, since ϕ is the asymptotic advantage as

$k \to \infty$ and a, b are bounded.

$$\hat{\phi}(a,b,c) = \max \begin{cases} \max_{s=0,\dots,b-1} \hat{\phi}(0,s,1) \\ p\hat{\phi}(a+1,b,c) + (1-p)(\hat{\phi}(a,b+1,c) - \hat{g}^\star) \\ \hat{\phi}(a-b-1,0,0) + b + 1 + c \cdot w - \hat{g}^\star \end{cases} . \tag{14}$$

Releasing blocks only causes Miner 2 to capitulate if $a \geq b+1$, so we set $\phi(a,0,c) = -\infty$ for $a < 0$. Before moving on, we need a useful inequality of the *immediate release* potential ϕ, to use for bounding the advantage of some states relative to others.

Lemma 4. *For nonnegative integers a, b and $\ell \in \{0,1\}$ we have that:*

$$\phi(a+\ell, b+\ell, 1) \leq \phi(a,b,0) + (\ell + w) \cdot \left(\frac{p}{1-p}\right)^{b-a+1} . \tag{15}$$

Proof. We first need to establish that for $c \in \{0,1\}$

$$g_k(a+\ell, b+\ell, c) \leq g_k(a,b,c) + \ell \cdot \left(\frac{p}{1-p}\right)^{b-a+1} \tag{16}$$

and

$$g_k(a+\ell, b+\ell, 1) \leq g_k(a,b,0) + (\ell + w) \cdot \left(\frac{p}{1-p}\right)^{b-a+1} . \tag{17}$$

Suppose that from state (a,b) Miner 1 follows the same strategy as state $(a+\ell, b+\ell)$. This is possible, as winning or capitulating depends only on the difference $b-a$. Let $\bar{g_k}(a,b,c)$ be the gain from playing this suboptimal strategy and $\bar{r}(a,b)$ the probability of winning from state (a,b). Clearly, we have:

$$g_k(a,b,c) \geq \bar{g}_k(a,b,c) = g_k(a+\ell, b+\ell, c) - \ell \cdot \bar{r}(a,b). \tag{18}$$

Given that $\bar{r}(a,b) \leq (p/(1-p))^{b-a+1}$ from [8, Lemma 1] we get the first inequality.

For the second, let $r(a,b)$ be the probability of winning from (a,b) with the optimal strategy. As before:

$$g_k(a,b,1) = g_k(a,b,0) + w \cdot r(a,b) \leq g_k(a,b,0) + w \cdot \left(\frac{p}{1-p}\right)^{b-a+1} , \tag{19}$$

which we substitute in (16).

To complete the proof, we take limits in (16) and use the definition of ϕ. □

In [8] it was shown that for $p \leq 0.361$ Frontier is a best response, therefore (for $w = 0$) there exists a ϕ such that for $g^\star = p$ (12) is satisfied, meaning that honest mining maximizes the fraction of blocks added by Miner 1. By Lemma 2, increasing w can only strengthen this result, therefore for $p \leq 0.361$ and any $w \leq 1$ there exists ϕ that satisfies (12) for $g^\star = p + p(1-p)w$. Using this, we extend the potential to states (a,b,c) with $a > b+1$:

$$\bar{\phi}(a,b,c) = \begin{cases} \phi(a,b,c) & \text{if } a \leq b+1 \\ a\lambda + b\mu + \kappa + c \cdot w & \text{otherwise} \end{cases} , \tag{20}$$

where $\lambda = \frac{(p-1)^2(1-pw)}{1-2p}$, $\mu = \frac{p\left(p-(p-1)^2 w\right)}{2p-1}$ and $\kappa = \frac{(p-1)p(pw-1)}{2p-1}$. The constants have been selected so that $\bar{\phi}(a,b,c) = p\bar{\phi}(a+1,b,c) + (1-p)(\bar{\phi}(a,b+1,c) - \hat{g}^\star)$ and $\bar{\phi}(b+1,b,c) = b+1+c\cdot w - \hat{g}^\star = \phi(b+1,b,c)$, for $\hat{g}^\star = p + p(1-p)w$ which is the honest gain. Doing this, we can use known results about ϕ for smaller states while having a tight enough closed form of $\hat{\phi}$ on states which are unlikely to be reached.

Following the notation of [8] we define $\bar{\phi}_M$ for when Miner 1 continues to mine, $\bar{\phi}_R$ when Miner 1 releases some blocks and $\bar{\phi}_C$ when he capitulates.

$$\begin{aligned}\bar{\phi}_M(a,b,c) &= p\cdot\bar{\phi}(a+1,b,c)+(1-p)(\bar{\phi}(a,b+1,c)-\hat{g}^\star)\\ \bar{\phi}_R(a,b,c) &= \bar{\phi}(a-b-1,0,0)+b+1+c\cdot w-\hat{g}^\star\\ \bar{\phi}_C(a,b,c) &= \max_{s=0,\dots,b-1}\bar{\phi}(0,s,1).\end{aligned}$$

Theorem 3. *For every $p < 0.344$, there exists $w \geq 0$ large enough so that if every miner but one follows* Frontier(w)*, the best response of the remaining miner with hash power p is* Frontier*.*

Proof. We need to show that $\bar{\phi}$ is a valid potential and satisfies (14) for $\hat{g}^\star = p + p(1-p)w$, or equivalently:

$$\bar{\phi}(a,b,c) = \max\{\bar{\phi}_M(a,b,c), \bar{\phi}_R(a,b,c), \bar{\phi}_C(a,b,c)\}$$

Lemma 5. *The potential $\bar{\phi}$ and $\hat{g}^\star = p + p(1-p)w$ satisfy recurrence (14) for $p \leq 0.344$ (root of the polynomial $-p^4 + 3p^3 - 7p^2 + 5p - 1$) and some $w < 1$.*

Proof.

Claim 1. For states (a,b,c) with $a < b+1$:

$$\bar{\phi}(a,b,c) = \phi(a,b,c) = \max\{\bar{\phi}_M(a,b,c), \bar{\phi}_R(a,b,c), \bar{\phi}_C(a,b,c)\}.$$

In this case $\bar{\phi}_R(a,b,c) = -\infty$ and therefore $\bar{\phi}(a,b,c) = \phi(a,b,c)$ which satisfies (12) and (14) by definition, as releasing is only possible for $a \geq b+1$.

Claim 2. For states (a,b,c) with $a > b+1$:

$$\bar{\phi}(a,b,c) = \bar{\phi}_M(a,b,c) = \max\{\bar{\phi}_M(a,b,c), \bar{\phi}_R(a,b,c), \bar{\phi}_C(a,b,c)\}.$$

By definition, $\bar{\phi}(a,b,c) = \bar{\phi}_M(a,b,c)$. We have:

$$\bar{\phi}(a,b,c) - \bar{\phi}_R(a,b,c) = \frac{b(2-4p)+(1-p)(2-3p)(1-pw)}{1-2p} > 0.$$

Since $p \leq 0.344$ we know that $\phi(a,b,c)$ corresponds to the potential of the honest mining strategy, in the immediate release case. Therefore:

$$\begin{aligned}\bar{\phi}_C(a,b,c) &= \max_{s=0,\dots,b-1}\bar{\phi}(0,s,1)\\ &= \max_{s=0,\dots,b-1}\phi(0,s,1) = \phi(0,0,1) \leq w\cdot\frac{p}{1-p}\end{aligned}$$

as $(0,0,1)$ and $(0,0,0)$ are the only mining states and by using Lemma 4 for the last inequality. Also:

$$\begin{aligned}\bar{\phi}_R(a,b,c) &= \bar{\phi}(a-b-1,0,0)+b+1+c\cdot w-\hat{g}^\star\\ &\geq 1-p(1-p)w\\ &\geq w\cdot\frac{p}{1-p}\\ &\geq \bar{\phi}_C(a,b,c),\end{aligned}$$

for $p < 0.344$.

Claim 3. For states $(b+1, b, c)$:

$$\bar{\phi}(b+1, b, c) = \bar{\phi}_R(b+1, b, c) \\ = \max\{\bar{\phi}_M(b+1, b, c), \bar{\phi}_M(b+1, b, c), \bar{\phi}_C(b+1, b, c)\}.$$

Exactly as before, we have $\bar{\phi}_R(b+1, b, c) = b+1+c\cdot w - \hat{g}^\star \geq \bar{\phi}_C(b+1, b, c)$. For $\bar{\phi}_M(b+1, b, c)$:

$$\begin{aligned}\bar{\phi}_M(b+1, b, c) &= p\bar{\phi}(b+2, b, c) + (1-p)\left(\bar{\phi}(b+1, b+1, c) - g\right) \\ &\leq p((b+2)\lambda + b\mu + \kappa + c\cdot w) \\ &\quad + (1-p)\left((b+1+c\cdot w)\frac{p}{1-p} - g\right),\end{aligned}$$

by the definition of $\bar{\phi}$ and Lemma 4. Then:

$$\begin{aligned}&\bar{\phi}_R(a, b, c) - \bar{\phi}_M(a, b, c) \geq \\ &\quad b(1-2p)^2 + (p-1)\left(p^3 w - p^2(w+1) + p(2c\cdot w + 5) - c\cdot w\right) \\ &\quad + 1.\end{aligned}$$

Since $b \geq 0$ we only need:

$$(p-1)\left(p^3 w - p^2(w+1) + p(2c\cdot w + 5) - c\cdot w\right) + 1 \geq 0.$$

Setting $w = 1$, it holds for $p \leq 0.344$ for $c = 0$ and $p \leq 0.442$ for $c = 1$. For smaller values of p it is not necessary to set w to it's highest value. □

Now, we can use the equivalent of Lemma 3(whose proof has very minor differences) for the strategic release case to get that:

$$\hat{g}^\star = \lim_{k\to\infty} \frac{\hat{g}_k(a, b, c)}{k} \leq \lim_{k\to\infty} \frac{\bar{\phi}(a, b, c) + k\cdot(p + p(1-p)w)}{k} \\ = p + p(1-p)w.$$

This shows that his gain is at most what he would get by playing Frontier, hence a best response. □

Through a procedure similar to Section 3.1, adjusting the LP for this case, we obtain the following graph for $d = 8$ for the minimum value of w required.

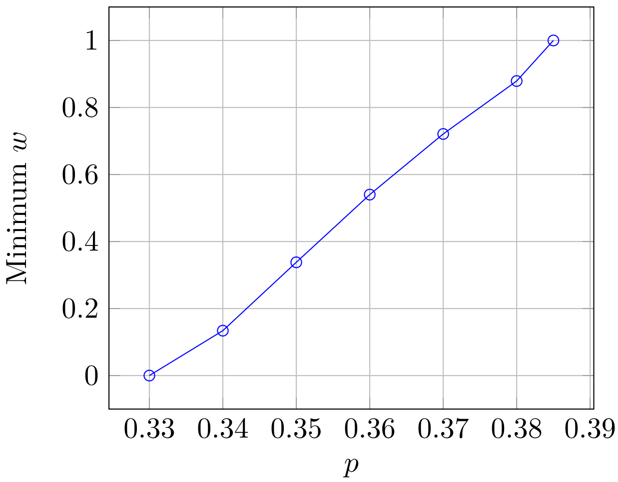

Figure 9: Minimum values of w for the strategic release case.

Contrary to Figure 7, for $p = 0.385$ we have $w \approx 1$. As $d \to \infty$, we reach $w = 1$ for $p \approx 0.38$.

References

[1] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 56–73, New York, NY, USA, 2012. ACM.

[2] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121, May 2015.

[3] Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167. ACM, 2016.

[4] Nicolas T. Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *CoRR*, abs/1402.1718, 2014.

[5] I. Eyal. The miner's dilemma. In *2015 IEEE Symposium on Security and Privacy*, pages 89–103, May 2015.

[6] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *NSDI*, pages 45–59, 2016.

[7] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM*, 61(7):95–102, June 2018.

[8] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 365–382, New York, NY, USA, 2016. ACM.

[9] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 357–388, Cham, 2017. Springer International Publishing.

[10] Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013, page 11, 2013.

[11] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S. Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 919–927, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[12] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf.

[13] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. *draft version 0.5*, 9:14, 2016.

[14] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[15] Meni Rosenfeld. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009*, 2014.

[16] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.

[17] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. SPECTRE: A fast and scalable cryptocurrency protocol. *IACR Cryptology ePrint Archive*, 2016:1159, 2016.

[18] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

[19] Yonatan Sompolinsky and Aviv Zohar. PHANTOM: A scalable blockdag protocol. *IACR Cryptology ePrint Archive*, 2018:104, 2018.

[20] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys Tutorials*, 18(3):2084–2123, thirdquarter 2016.