The blockchain folk theorem[*]

Bruno Biais[†] Christophe Bisière[‡] Matthieu Bouvard[§]
Catherine Casamatta[¶]

November 21, 2017

Preliminary

Abstract

Blockchains are distributed ledgers, operated within peer-to-peer networks. If reliable and stable, they could offer a new, cost effective, way to record transactions and asset ownership, but are they? We model the blockchain as a stochastic game and analyse the equilibrium strategies of rational, strategic miners. We show that mining the longest chain is a Markov perfect equilibrium, without forking on the equilibrium path, in line with the seminal vision of Nakamoto (2008). We also clarify, however, that the blockchain game is a coordination game, which opens the scope for multiple equilibria. We show there exist equilibria with forks, leading to orphaned blocks and also possibly to persistent divergence between different chains.

[*]We thank V. Glode, B. Gobillard, I. Goldstein, C. Harvey, J. Hörner, A. Kirilenko, T. Mariotti, J. Tirole, S. Villeneuve, the members of the TSE Blockchain working group, participants in the Inquire Conference in Liverpool, 2017, participants in the GSE Summer Forum in Barcelona, 2017, the Africa Meeting of the Econometric Society in Algiers, 2017, the RFS Fintech workshop, 2017, as well as an anonymous referee for helpful comments. Financial support from the FBF-IDEI Chair on Investment banking and financial markets value chain is gratefully acknowledged. This research also benefited from the support of the Europlace Institute of Finance, and the Jean-Jacques Laffont Digital chair.

[†]Toulouse School of Economics, CNRS (TSM-Research)

[‡]Toulouse School of Economics, Université Toulouse Capitole (TSM-Research)

[§]Desautels Faculty of Management, McGill University

[¶]Toulouse School of Economics, Université Toulouse Capitole (TSM-Research)

1 Introduction

Blockchains are decentralised protocols for recording transactions and asset ownership. In contrast with centralised protocols in which one authority is in charge of recording or certifying transactions in a unique centralised ledger, blockchain operates within a network, whose participants each possess and update their own version of the ledger. The blockchain design was the main innovation underlying the digital currency network Bitcoin (Nakamoto, 2008), but its potential benefits in terms of cost-efficiency, speed and security, for a variety of assets and contracts, have attracted interest from a broad range of institutions and businesses.[1] Blockchain experiments, and in some cases, deployments, have been conducted by the Australian Stock Exchange, the Nasdaq, BHP Billiton and major banks around the globe. As blockchains are being embedded into major transaction platforms, we propose to investigate the stability of the protocol: how efficient is a blockchain at building a stable consensus among participants about the history of past transactions? This question is particularly relevant in public blockchains where participants are anonymous and no formal authority may coordinate their behaviour in last resort.[2] Our game-theoretic approach pins down the tradeoffs faced by the key players in a blockchain's decentralised certification process, the "miners."

We study the protocol known as "proof-of-work" that miners follow to build consensus.[3] This protocol, which we describe in more details in the next section, can be sketched as follows. At each point in time, miners try to validate a new block of transactions. This implies verifying that the transactions in the block do not contradict each other or past transactions, but also solving a purely numerical problem unrelated to the block's content, i.e., "mining." The first miner to solve this problem obtains a proof-of-work, which he attaches to the new block before diffusing it to the network. Hence, the identity of the miner who validates a given block is the outcome of a random draw in which each miner's probability of winning is his share of

[1] The blockchain is cost effective in that the administrative costs of running it are limited compared to those incurred within older technologies and institutions, such as notaries, banks or depositories.

[2] Bitcoin and Ethereum are best-known examples of public blockchains. There also exist private blockchains, which use the same technology, but whose participants are selected and which can have specific coordination devices. Our paper focuses on public blockchains.

[3] Major public blockchains such as Bitcoin or Ethereum run on proof-of-work protocols.

the total computing power in the network. This ensures that miners take turns to validate blocks, and therefore no single miner has control over the whole verification process. When a new block is diffused to the network, it becomes part of the consensus if miners express acceptance by chaining to it their next block. Each block includes a reward for the miner who solved it. This process illustrated in Figure 1.

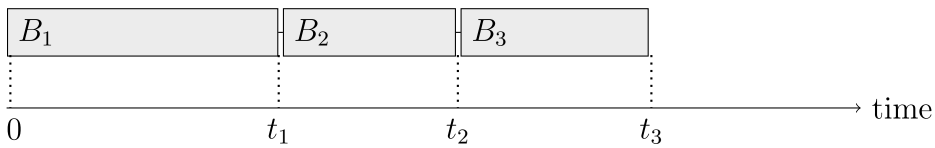

Figure 1: The Blockchain

At $t = 0$, there is an initial block B_0 and a stock of transactions included in a block B_1, chained to B_0. Miners work on a cryptographic problem until a miner solves B_1 at t_1. B_1 is broadcast to all. Nodes check proof-of-work and transactions validity, and express acceptance by chaining the next block to B_1.

Proof-of work generates a stable consensus, or in other words, a single chain, if miners always accept a newly solved block as the parent for their next block. Miners, however, may choose to discard certain blocks. Suppose, e.g., that the last block solved is B_n, but miner m chains his next block to the parent of B_n, i.e., B_{n-1}. This starts a fork, as illustrated in Figure 2. If some miners follow m, while others continue to attach their blocks to the original chain, there are competing versions of the ledger. This reduces the credibility and reliability of the blockchain, especially if the fork is persistent. Even if, eventually, all miners agree to attach their blocks to the same chain, the occurrence of the fork is not innocuous. The blocks in the chain eventually abandoned are orphaned. They have been mined in vain, and the corresponding computing power and energy have been wasted. Moreover, the transactions recorded in the orphaned blocks may be called in question.

In the present paper, we first consider an environment in which miners' payoffs are solely composed of rewards for solving blocks, and information is instantaneously disseminated in the network. In this frictionless world, it is commonly argued, in particular by the blockchain community, that blockchains should give rise to a single and stable consensus, and thus offer a reliable way to record transactions and ownership. We examine the validity of that "folk theorem" in a stochastic game that captures the key features of

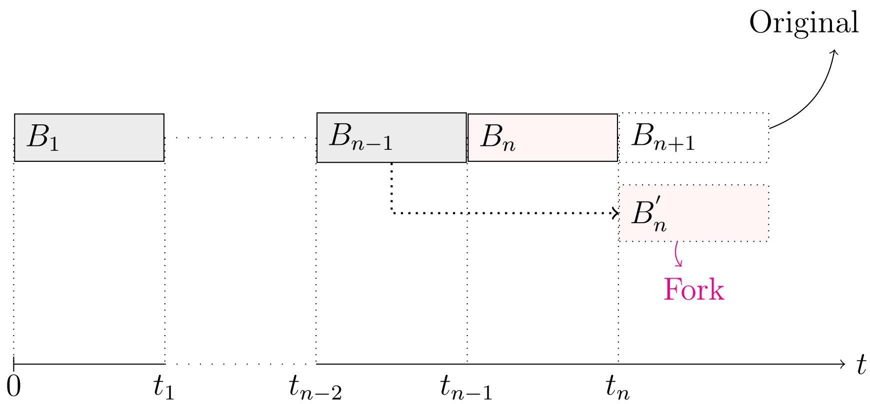

Figure 2: A fork

a blockchain and allows a formal analysis of the economic forces at play in that environment. In particular, we are interested in mechanisms pushing towards stability or instability of the distributed ledger. Our analysis uncovers two important economic forces at play in the blockchain.

First, miners' actions are strategic complements. Indeed, their rewards are paid in the cryptocurrency associated with the chain on which they are solving blocks. The value of that cryptocurrency depends on the credibility of the corresponding chain, which is higher if more miners are active on it. This generates complementarities: If more miners are expected to mine a given chain going forward, the expected profit from mining blocks on that chain increases. Hence, miners benefit from coordinating on a single chain, which they can achieve by playing the longest chain rule (hereafter LCR), as suggested by Nakamoto (2008). This sustains a Pareto-optimal equilibrium in which every new block becomes part of the consensus. However, the same coordination motives sustain equilibria with forks (Proposition 2). Intuitively, if at some time (a sunspot) a miner anticipates all other miners to fork and create a new chain, his best response is to follow them. Indeed, he rationally anticipates that any block solved out of the equilibrium path, for instance on the original chain, will not be accepted by other miners and the corresponding reward will be worthless.

Second, we identify a counterveiling force which we refer to as "vested interest." Because miners cannot immediately spend or convert the cryptocurrency earned while mining, a miner who has accumulated rewards by solving blocks on a given chain has a vested interest in this chain remaining active. In particular, the value of these rewards would drop if he moved to a different chain. Vested interests may counteract coordination motives for a group of miners, inducing them to keep mining a minority chain, and sustaining persistent forks in equilibrium (Proposition 3). Unlike temporary forks that only rely on coordination motives and would arise with atomistic miners, equilibria with persistent forks depend on miners internalising how their actions affect the value of their rewards.

Overall, this analysis suggests that the blockchain design, by generating complementarities and vested interest, is subject to instability. Next, we investigate how frictions typically associated with dissensus and forks relate to these economic forces. For instance, communication delays may generate transient forks as some miners do not immediately realise that a new block has been solved. Some miners may also derive extrinsic benefits from creating a fork: it can allow them to void previous transactions and recover the corresponding cryptocurrencies ("double-spending"), or to push technical solutions that give them a competitive edge ("upgrades"). We incorporate these frictions in our model and show that while they may act as triggers (instead of sunspots), the same fundamental interplay of coordination motives and vested interests as in the frictionless case underlies equilibria with forks.

Finally, we endogenise the computing capacity that each miner installs. Because the difficulty of the mining process is typically adjusted upwards when the total computing capacity in the network increases, a miner's investment in computing power exerts a negative externality on all other miners. This gives rise to an arms race in which each miner ends up over-investing (not unlike the over-investment in financial expertise noted by Glode, Green and Lowery (2012)). This analysis points to another source of inefficiency in the blockchain's decentralised design.

Literature: Most existing literature on blockchains is in computer science, with the notable exceptions of Harvey (2016), who discusses the pros and cons of blockchains and Yermack (2017) who discusses their implications for corporate governance.

Computer science papers offer insightful analyses of potential strategic problems, but usually do not rely on the same type of formalism as in economics. Bonneau et al. (2016) analyse how mining pools (i.e., groups of miners) controlling a large fraction of the computing power could attack the chain. Eyal and Sirer (2014) show how colluding miners can obtain a larger revenue than their fair shares. Teusch, Jain and Saxena (2016) study how a strategic miner can fork and attack the blockchain to double spend. The paper to which our analysis is the closest is Kroll, Davey and Felten (2013). They note that the interaction between miners should be analysed as a game. They argue that the LCR is a Nash equilibrium. While their analysis offers interesting economic intuition, it does not offer a formal analysis and proof of equilibrium. Another difference between our analysis and theirs is our analysis of forks on the equilibrium path.

Several papers (e.g., Evans, 2014) note that an additional problem with the Bitcoin mining incentive scheme is that miners are paid with bitcoins, which have a volatile value. In our analysis, the only source of variation in the value of rewards to a given block is the extent to which the chain including that block is actively mined. We analyse how these variations affect incentives. Schrijvers et al. (2016) study a different type of incentive problems than that we consider. They study the behaviour of miners in a pool, assuming that the pool organiser does not observe when miners solve blocks nor the computing power they dedicate to that task. They analyse how to incentivise miners to reveal that they have solved a block as soon as they have done so.

The remainder of the paper is organised as follows. The next section offers an introduction to blockchain environments. Section 3 presents the model. Section 4 develops our equilibrium analysis and contains our main results. Extensions of the model are provided in Sections 5 and 6, and Section 7 concludes. All proofs are in the Appendix.

2 A primer on blockchains

In this section we first describe the blockchain protocol and then discuss some problems that can arise in blockchains.

2.1 The blockchain protocol

Centralised vs decentralised ledgers: A ledger is a collection of records, regarding ownership, transactions, identity, etc. For ledgers to facilitate interactions among economic agents, it is essential that the agents reach a consensus about the state of the ledger and its authenticity. Historically, a central authority, e.g., the state and its delegates, ensured this consensus by managing and certifying the ledger. Such centralised ledgers, however, cannot operate satisfactorily if the central authority behaves opportunistically, e.g., by excluding some participants or transactions, or by distorting the ledger.

The distributed ledger technology (DLT) can overcome that obstacle. Within a distributed ledger, there is a network of participants, and each participant maintains its own ledger. When an economic transaction occurs, the trading parties send this information to the network, so that it can be validated by the network participants, each including it into his or her own ledger. Ledgers should eventually be the same for all participants, giving rise to consensus on a single, distributed, ledger.

Blockchain is the distributed ledger protocol invented by Nakamoto (2008) when he created Bitcoin. The elegance and novelty of his solution relies in particular on its endeavour to incorporate the incentives of the participants.[4] This justifies our game theoretic approach that accounts for these incentives to investigate the properties of this protocol.

Proof-of-work: Decentralisation of the ledger implies its validation should not be controlled or manipulated by a single participant or a small number of colluding participants. All participants should be equally able to contribute to the validation process. One way to associate all participants to the validation of transactions would be to rely on majority voting. However, as noted by Nakamoto (2008), page 3:

> "If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate IPs."

The solution proposed by Nakamoto (2008) to this so-called "Sybil attack" is called "proof-of-work."[5] For convenience, participants do not validate each transaction individually but group them in blocks. One participant is designated to send his block to the network to update the ledger. To obtain this right, each participant works to solve a difficult cryptographic problem, attached to his block. The term "work" in "proof-of-work" therefore refers

[4]Section 6 of Nakamoto (2008) seminal paper is entitled "Incentive."

[5]Alternative protocols include proof-of-stake, currently used by NXT, BlackCoin and Peercoin. However, and in spite of serious efforts, there is no satisfying alternative to proof-of-work so far.

to using computers and electricity to perform this task. The problem to solve has nothing to do with the economic transactions included in a block. Rather, it allows each participant to use computing power to perform independent trials (similar to draws under replacement) until one finds a solution to an arbitrary numerical problem (a hash value lower than a given threshold.) The larger the computing power the larger the number of draws per unit of time, the shorter the expected time it takes to find the solution.

A property of cryptographic problems is that the solution is hard to find, but easy to verify. Hence, when they receive a block for validation, participants easily check whether the sender actually found the solution. If participants accept this block, they take it as the parent of the new block they start mining. Thus, as written by Nakamoto (2008), participants

> "vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid block by refusing to work on them."(page 8)

Proof-of-work is essentially a way to randomise across participants who will propose the next change to the ledger. Otherwise stated, proof-of-work is a way to randomise who will get to vote on the next state of the ledger.

This process gives rise to a chain of consecutively solved blocks, i.e., the blockchain, as illustrated in Figure 1 and summarised by Nakamoto (2008) on page 3:

> "The steps to run the network are as follows:
> 1) New transactions are broadcast to all nodes.
> 2) Each node collects new transactions into a block.
> 3) Each node works on finding a difficult proof-of-work for its block.
> 4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
> 5) Nodes accept the block only if all transactions in it are valid and not already spent.
> 6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash."[6]

[6] This is a "hash pointer", i.e., a value that identifies the block (note from the authors).

Miners: The nodes conducting the above mentioned tasks are called "miners", as they "mine" to solve proof-of-work problems and get rewarded for this in units of the cryptocurrency whose ownership is recorded in the blockchain. This includes the creation of new units of the cryptocurrency allocated to the miner (12.5 BTC per block on Bitcoin in 2017, and 3 ETH on Ethereum since Oct. 2017) plus transaction fees which the originators of economic transactions can choose to offer for the validation of these transactions. Bitcoin imposes a 100-block delay before rewards earned through mining can be spent.

In practice, miners gather in large pools. Figure 3 presents the distribution of computing power of the pools operating on Bitcoin in November 2017. The figure illustrates that 13 mining pools represented about 95 % of the total hash capacity. Pools allow miners to mutualise block discovery risk. They also coordinate individual mining strategies. For example, on `https://www.bitcoinmining.com/bitcoin-mining-pools/`, one can read:

> "If you participate in a Bitcoin mining pool then you will want to ensure that they are engaging in behavior that is in agreement with your philosophy towards Bitcoin. [...] Therefore, it is your duty to make sure that any Bitcoin mining power you direct to a mining pool does not attempt to enforce network consensus rules you disagree with."

Difficulty: The time it takes a miner to solve a block depends on the difficulty of the cryptographic problem and the miner's computing power. The difficulty is set by the blockchain protocol to keep the average duration between two blocks close to a target (10 minutes on Bitcoin and between 9 and 17 seconds on Ethereum). If the total computing power increases (e.g., due to the entry of new miners and new pools), the protocol ensures that the difficulty is scaled up so that average duration between two blocks remains equal to the desired level. Thus, on Bitcoin every 2,016 blocks, i.e., approximately every 2 weeks, the difficulty is rescaled to ensure that the average time between blocks remains at 10 minutes.

2.2 Forks

Consensus on the decentralised ledger requires that there is only one chain of blocks, observed by all and on which all agree. It is jeopardised if the chain

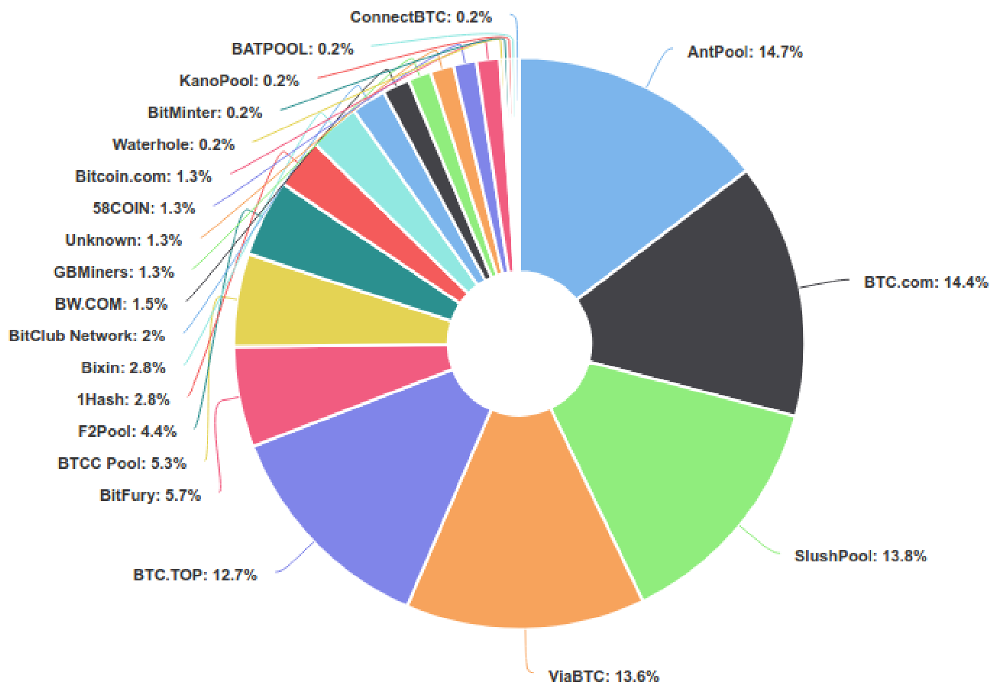

Figure 3:
Hashrate distribution of Bitcoin mining pools on November 15, 2017.
Source: blockchain.info. AntPool servers are located in China. The other three main pools have servers in China, Japan and the US.

splits into a fork, with two competing branches, each with its own version of the ledger. In the present subsection we review some reasons why forks can happen in blockchains, and describe forks which actually occurred.

Communication delays: In practice, the information that a block has been solved is not transmitted instantaneously and simultaneously to all network participants. For example, miners in Siberia might learn before miners in Iceland that a block has been solved in China. Thus, it will routinely happen that a block has just been solved but some participants are not yet aware of that. If these participants solve their own block in the meantime, this starts a *fork* with two competing blocks attached to the same parent. Nakamoto (2008) identified that problem and suggested it would be solved if miners always chained their blocks to the longest chain (following the Longest

Chain Rule, hereafter LCR):

> "Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one."Nakamoto (2008), page 3.

Double spending: The blockchain protocol was also designed to prevent "double spending." Suppose a miner buys an object from a seller, paying for it with bitcoins. The corresponding transaction is recorded in a block B. When the latter is validated, the seller delivers the object and is entitled to use the bitcoins. This could give an incentive for the buyer to start a fork: he could try to solve a block that does not contain his transaction, and that is chained to the parent of B, in the hope of attracting miners to his chain. If he succeeded in doing so, no block would be chained to B (i.e. B would be "orphaned"), which would void the transfer of his bitcoins to the seller. Nakamoto (2008) argues that double spending is unlikely to be successful because it would require solving blocks faster than the rest of the network.

Software upgrades: So far we used the term "fork" to refer to chain splits. There is another possible use of the term: In the context of open source software, forking means copying the source code of a computer program and modifying it. In a blockchain, a "soft fork" is the introduction of an upgraded version of the software which remains compatible with the previous version: blocks mined with the new version are considered as valid by miners still running the old version. In that case, a soft fork does not trigger a fork in the blockchain, even if not all miners upgrade to the new version. In contrast, a "hard fork" is not backward-compatible, as upgraded miners might create blocks that will be rejected as invalid by non-upgraded miners. Therefore, if not all miners upgrade, a hard fork can be a way to trigger a fork in the blockchain. We describe below some actual unintended or intended forks triggered by software upgrades.

Unintended forks: An important chain split occurred on the Bitcoin blockchain in March 2013, following what developers thought to be an innocuous soft fork: On March 11, 2013, some miners upgraded to a new version of the software, referred to as 0.8. There turned out to be a bug so that the miners operating with the 0.7 version rejected as invalid one block solved by the 0.8 miners and consequently the subsequent ones (Thus the 0.8 upgrade turned out to be an unintended hard fork, which miners identified with a delay). From that point on, the 0.8 miners worked on a chain stemming from that block, while the 0.7 worked on a competing chain, stemming from its parent. When they discovered the split, participants all wanted to revert to a single chain, but they had to decide which branch to keep and which to orphan. Achieving coordination turned out to be difficult, as illustrated by the following discussion between Bitcoin software developers and miners (reported by Narayanan (2015)):

> "Gavin Andresen: the 0.8 fork is longer, yes? so majority hashpower is 0.8 ... first rule of bitcoin: majority hashpower wins
>
> Luke Dashjr: if we go with 0.8 we are hard forking
>
> BTC Guild: I can single handedly put 0.7 back to the majority hashpower. I just need confirmation that that's what should be done.
>
> Pieter Wuille: that is what should be done, but we should have consensus first."

Miners faced a dilemma. Should they follow the longest chain rule and mine the 0.8 chain which had attracted the majority of the computing power? Or should they revert to the 0.7 version? Miners wanted to abide to the consensus, but they first needed to coordinate on what the consensus should be.

Eventually, BTC Guild, which was one of the largest pools at the time, chose to downgrade to the 0.7 version. This resulted in the 0.7 chain becoming the longest, and all miners coordinating back to it. It took 8 hours before participants could solve the problem. Consequently more than 24 blocks, solved on the 0.8 chain, became orphaned, and their miners (including BTC Guild) lost the corresponding rewards. Commenting on this situation, Narayanan (2015) wrote:

> "One way to look at this is that BTC Guild sacrificed revenues for the good of the network. But these actions can also

> be justified from a revenue-maximising perspective. If the BTC Guild operator believed that the 0.7 branch would win anyway (perhaps the developers would be able to convince another large pool operator), then moving first is relatively best, since delaying would only take BTC Guild further down the doomed branch."

This discussion underscores that miners' coordination, or the lack thereof, plays an important role in the emergence and resolution of forks. It also underscores the importance of beliefs in this context: An individual miner (such as, e.g., BTC Guild) decides to chain his blocks to the branch which he believes the others will choose. Thus, beliefs about the actions of others influence one's action. This generates a form of beauty contest, in which coordination effects are critical.

Intended forks: Ethereum underwent a hard fork in the summer of 2016. Following the hack of TheDAO, a large venture capital fund operating through smart contracts, members of the Ethereum community suggested to roll back the blockchain in order to cancel the transactions that diverted the fund's money. They hoped all participants would coordinate on that hard fork, leading to a single active chain. Other members, however, refused to alter the history of the ledger and rejected the hard fork. Consequently, Ethereum split in two incompatible branches, Ethereum and Ethereum Classic. These two branches still exist, each corresponding to a different ledger and history of trades, and a different cryptocurrency. As of may 2017, Ethereum Classic represented about 10% of the hash capacity of Ethereum, and the price of ETC was about 10% of the ETH price. This episode illustrates the difficulty to achieve coordination on a single chain, the uncertainty about miners' actions and the resulting occurrence of persistent competition between alternative chains.

Bitcoin also underwent a hard fork, in the summer of 2017. The community had long been divided on how to relax the limitation of the network throughput.[7] Two main solutions, Segregated Witness (SegWit) and Bitcoin Unlimited (BU), were supported by different participants, with the threat of some to fork in order to impose their preferred solution. In the New York Agreement signed on may 2017, most mining pool operators agreed to roll

[7] The Bitcoin protocol sets the maximum size of a block of transactions to one megabyte. This limit slows down the speed of transactions validation and hinders the development of the network itself.

out a compromise solution (SegWit2x). Yet another way to increase throughput, Bitcoin Cash, was eventually implemented, via a hard fork, on August 1st, 2017. Bitcoin then split in two branches, with two different cryptocurrencies, Bitcoin and Bitcoin Cash. On the former branch, following the New York Agreement, a hark fork was planed for November 2017. There was a lot of uncertainty, and discussion among miners, about who would adopt SegWit2x, and whether there would be a new chain split. Many bloggers, developers and mining pool operators announced that a chain split was very likely. At odds with those forecasts, the SegWit2x hard fork was abandoned. One could have thought this meant participants would coordinate on Bitcoin. Quite to the contrary, a large fraction of miners reacted by shifting from Bitcoin to Bitcoin Cash. While, early November, 12-hour average hashrates were about 10 Exahashes (10^{18} hashes) per second on Bitcoin versus less than 2 on Bitcoin Cash, on November 12, hashrates were similar on the two branches. Again, this episode underscores that it is difficult for miners to coordinate on a single chain, that chain splits are not uncommon, and that the outcome is hard to predict, even for major participants.

In the next sections, we will show how these features of blockchain in the real life also arise, in equilibrium, due to standard economic forces.

3 Model

In line with the above description of the blockchain technology, we consider the following model.

Miners and pools: There are $M \geq 2$ risk-neutral miners, indexed by $m \in \mathcal{M} = \{1, ...M\}$. While, in our model, we refer to each m as a miner, m can also represent a mining pool, which coordinates the efforts of its miners as regards which blocks they mine.

Mining technology: There is a continuous flow of transactions sent for confirmation by end-users.[8] For the moment, for simplicity, we assume all miners perfectly and instantaneously observe this flow, which they include in the blocks they mine.

[8]For simplicity we take the flow of transactions to be exogenous, while in practice it can actually be endogenous. In fact, we don't model the transactions and model the blockchain process directly at the level of the blocks.

As explained in Nakamoto (2008), the time it takes miner m to solve a block problem is exponential with parameter θ_m. For a given computational power, the greater the difficulty, the lower the intensity θ_m. A key property of the exponential distribution is that it is memoryless: at each point in time, the distribution of the waiting time until the miner finds a solution is independent from how long the miner has been working on the problem.[9] This waiting time is also independent of which block m is mining, and also from the blocks the other miners are mining. We denote by N_m the Poisson process jumping each time miner m solves a block. Thus, the number of blocks solved by miner m between time 0 and time t, is

$$N_m(t) = \int_{s=0}^{t} dN_m(s).$$

For simplicity we assume (in line with what happens in practice) that miners do not update the set of transactions defining the block they mine until they have solved the hash problem (transactions that flow in meanwhile are stored in a buffer.) Relaxing that assumption would not alter the economic mechanism we analyse below.

We assume that at time z_m, exponentially distributed, with parameter λ_m, miner m is hit by a liquidity shock. At time z_m the miner must leave the game and sell the cryptocurrencies he earned previously to a new miner who also inherits his beliefs and preferences.[10] Thus, exits are compensated by entries and the environment is stationary.

Blockchain: At time 0, there is an initial state of the ledger, encoded in B_0, and a set of transactions. Starting from B_0, miners start working on the first block, B_1, which contains the initial set of transactions. Once B_1 is solved, miners must choose to which parent block to chain the next block (B_2) they mine. If miners choose B_1 as a parent block, they continue the first chain. Alternatively, miners can choose to disregard B_1 and attach B_2 to B_0. In that case, miners start a fork and there are two competing chains, one including B_0 and B_1, the other B_0 and B_2.

[9] Another key property of the exponential distribution is that the minimum of two exponentials, with parameters θ and θ', is also exponential, with parameter $\theta + \theta'$. Thus, when interpreting the M players in our game as M pools, we interpret the intensity of pool m, θ_m, as the sum of the intensities of all the miners active in that pool.

[10] We explain below the process through which miner m accumulates cryptocurrencies.

As the game unfolds, a tree of blocks develops. At each vertex B_k, the tree includes a label, identifying the miner who solved the corresponding block, $m(B_k)$. The indices of the blocks give the order in which they have been solved. That is, if $k < n$, then block B_k was solved before block B_n.

Thus, at any time t, one can observe a tree of solved blocks $\mathcal{C}^t = \{B^t, E^t, I^t\}$, where $B^t = (B_0, ...B_n)$ is the set of all blocks that have been solved by time t, $E^t = \{(B_0, B_1), ...(B_k, B_{k'}), ...\}$, with $0 \leq k < k' \leq n$, is the set of edges chaining these blocks, and $I^t = (m(B_1), ...m(B_n))$ is the set of identities of miners who solved blocks. Within a tree, a chain is a sequence of connected blocks in which each block is connected to at most one subsequent block. Thus, each fork starts a new chain. More formally, we define a fork as follows:

Definition 1 *Fork: There is a fork at time t if and only if there exists $(B_i, B_k, B_{k'})$ included in B^t such that (B_i, B_k) and $(B_i, B_{k'})$ belong to E^t.*

It is also useful to define the original chain for a given tree $\mathcal{C}^t$, as follows:

Definition 2 *Original Chain: Suppose E^t contains (B_i, B_k) and $(B_i, B_{k'})$. A chain that includes (B_i, B_k) preexists a chain that includes $(B_i, B_{k'})$ if and only if $k < k'$. We call the original chain the chain that preexists all other chains in $\mathcal{C}^t$.*

Note that the original chain is well defined since the "preexist" relation provides a complete ranking of all chains (as all chains have at least one common block, B_0).

Stopping times: We assume miners make decisions at different points in time, corresponding to a sequence of stopping times. Whenever a block is solved or a miner is hit by a liquidity shock, all miners make a decision. Miners can also make a decision, after a time interval of length Δ, if no block is solved and no liquidity shock has occurred during that interval. Δ can be arbitrarily small to approximate a continuous time environment.[11] Thus, the sequence of stopping times at which miners make decisions is $\mathcal{T} = \{0, ...\tau_j, \tau_{j+1}, ...\}$ where the next stopping time after τ_j, τ_{j+1}, is equal to $\tau_{j+1} = \min[\tau_j + \Delta, \tau^l(\tau_j), \tau^b(\tau_j)]$, $\tau^l(\tau_j)$ being the first time a liquidity shock occurs after τ_j and $\tau^b(\tau_j)$ the first time a block is solved after τ_j.

[11] This discretisation enables us to avoid technical issues regarding the definition of strategies in continuous time games.

Action space: At any time $\tau \in \mathcal{T}$, miners observe the set B^τ of all the blocks that have been solved previously. A miner's action is the choice of which block in B^τ to attach his current block to. All miners $m \in \mathcal{M} = \{1, ...M\}$ face the same action space.

Payoffs: When miner m solves a block in a given chain, he receives a reward, included in the block he mined, and expressed in the cryptocurrency corresponding to that chain.[12] We assume that miner m consumes the rewards he earned throughout the game at time z_m. That is, we assume that, until time z_m, the miner keeps the units of cryptocurrency he earned.[13]

At time z_m, the payoff from each solved block depends on the credibility of the chain that contains the block. Consider two polar cases: In the first case, a block solved by a miner becomes orphaned, i.e., no further blocks are attached to it, so that no miner expresses acceptance of that block and the transfer of cryptocurrency it encodes. In the second case there is a single chain to which all blocks belong, reflecting consensus on the blocks in that chain. The value of a reward in the first case, is likely to be zero, and is bound to be smaller than in the second case. Next, consider an intermediate case, in which the block is included in a chain competing with another one. As long as a significant fraction of the miners are working on each of the chains, the value of rewards included in the blocks of the two chains, while uncertain, can remain positive.

More formally, the payoff for miner m from solving B is an increasing function, $G(.)$, of the number of miners active at time z_m in the chain including B. For example, suppose there are two active chains at time z_m. If there are K miners active in the chain including B, and $M - K$ in the other, the payoffs from solving blocks are the following: The miner who solved block B, which we denote by $m(B)$, earns $G(K)$ for block B. A miner who solved a block in the other chain earns $G(M - K)$ for that block. If a miner solved a block that belongs to both chains, he earns $G(M - K) + G(K)$.[14] We set

[12] For simplicity, we neglect transaction fees offered by final traders, since we do not model explicitly transactions.

[13] On Bitcoin, miners have to wait for at least 100 blocks before using their rewards. In most digital currencies, a so called "k-blocks rule" prevents miners from using rewards before sufficiently many blocks have been chained to their rewarding block. Our model takes the simplified view that the vesting period lasts until z_m.

[14] We also assume that, if z_m occurs just after a fork starts, the not yet realised fork does not reduce the credibility of the current chain. That is, $m(B_n)$ earns $G(M)$ for B_n

$G(0) = G(1) = 0$ since, when there is only one or no miner on a chain, the associated cryptocurrency has no value. Finally, we assume that when several chains compete, the total value of a unit of cryptocurrency that belongs to the competing chains is weakly lower than if it belonged to a single chain that was the consensus of all miners: $G(M-K) + G(K) \leq G(M), \forall K$.

Our assumption that the value of the virtual currency is reduced by forks is illustrated by Figure 4, which plots the decline in bitcoin value during the March 2013 fork. The first vertical line indicates the time (around 22:00) at which miners started working on two different chains. Chats between miners realising there was a fork, started around 23:30.[15] At 1:30 am, a message posted on Bitcointalk asked miners to stop mining one the two branches of the chain (the 0.8 branch). The second vertical line (approximately at 6:20) indicates the time at which the 0.7 branch caught up the 0.8 branch. By 7:30, miners had stopped mining the 0.8 branch, which became orphaned, so that the fork was no longer active. The figure illustrates that, when the market realised that miners worked on different branches this triggered a 25% drop in the value of the virtual currency (from around \$48 at 1:00 am to around \$36 at 3:00).

States: At time $\tau \in \mathcal{T}$, a state ω_τ includes three elements:

- First, ω_τ includes the tree of solved blocks $\mathcal{C}^\tau = \{B^\tau, E^\tau, I^\tau\}$. The entire set of previously solved blocks, B^τ, is relevant for the miners, since they can chain a new block to any of these previously solved blocks. For each miner, the set of blocks he solved, measurable with respect to I^τ, determines his payoff, and therefore influences his actions.

- Second, ω_τ includes the number of miners active on branches stemming from each of the previously solved blocks:[16] $A^\tau = (A^\tau(B_1), ..A^\tau(B_k), ...A^\tau(B_n))$, where $A^\tau(B_k)$ is the number of miners mining at time τ a block directly

in that case. Alternative hypotheses could be that the attempt to fork reduces mining rewards. Our proofs are robust to the assumption that the reward is reduced to some arbitrary $g < G(M)$ or to $G(K)$ if K miners are active on the chain.

[15] Source: http://web.archive.org/web/20130421062600/http://bitcoinstats.com:80/irc/bitcoin-dev/logs/2013/03/12.

[16] In practice, miners cannot directly observe the current distribution of the computing power across the different branches of the chain, but estimate it based on the observed frequency of block resolutions. In our analysis, equilibrium strategies only depend on A^τ via miners' payoffs at z_m.

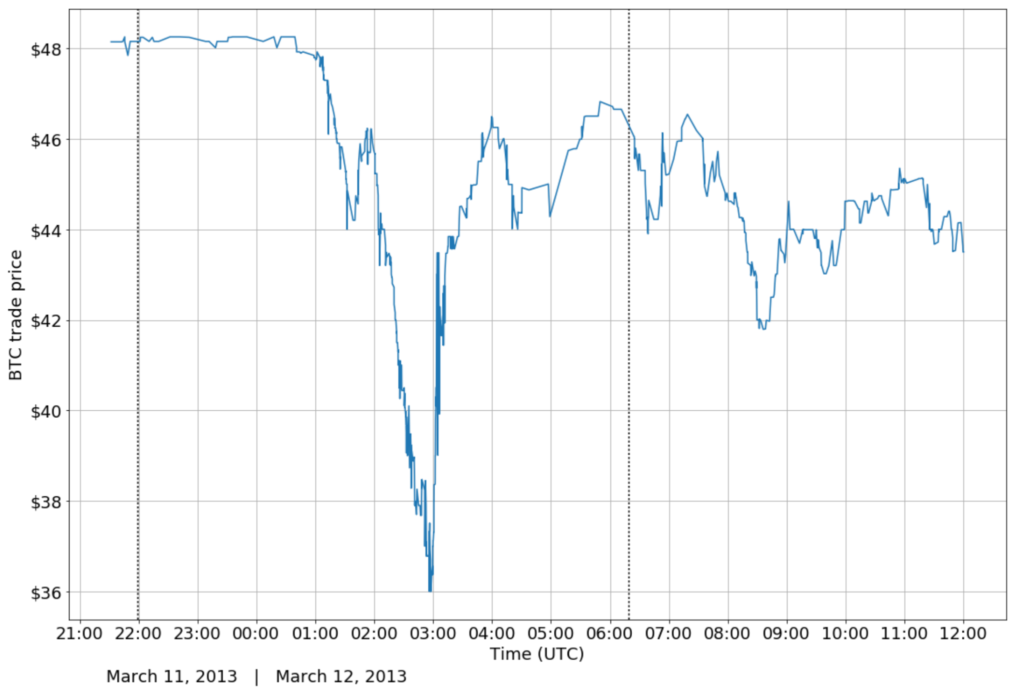

Figure 4:
BTC/USD trade prices on Bitstamp exchange, around the March 2013 Bitcoin fork.
The graph plots individual transaction prices obtained from a major bitcoin exchange platform, Bitstamp, during the March 11-12, 2013 fork. The first dotted vertical line represents the time at which the fork started, and the second dotted vertical line represents the time at which the original chain caught up the fork. Data source: Kaiko

chained to B_k, determines the value of each miner's reward if he's hit by a liquidity shock.

- Finally, as in Duggan (2012) or Cole and Kehoe (2000), to enable players to coordinate their actions using a public randomisation device, we assume that at each time $\tau \in \mathcal{T}$, the realisation of a sunspot random variable r^τ is observed by all, and we include it in the state. r^τ is uniformly distributed on $[0, 1]$ and i.i.d. over time.

Thus, we define $\omega_\tau = (\ \mathcal{C}^\tau,\ A^\tau,\ r^\tau)$ and denote by Ω the set of states of the world.

Strategies: Miner m chooses his strategy to maximise his expected payoff at time z_m. At each time $\tau \in \mathcal{T}$, miners observe the whole history of the game, that is, the state ω_τ, as well as, e.g., the exact timing of blocks resolution and the previous mining choices. In line with Markov perfection, we only consider strategies that are measurable with respect to ω_τ.[17] A pure strategy for miner m is a function σ_m^τ mapping each possible state of the blockchain $\omega_\tau \in \Omega$, into an element of the action space B^τ. We denote the strategy of miner m throughout the entire history of the game by σ_m and the profile of strategies for the M miners by $\sigma = \{\sigma_m\}_{m\in\mathcal{M}}$. σ, combined with the random variables $\{z\}_{m\in\mathcal{M}}$ and $\{N_m\}_{m\in\mathcal{M}}$, yield the transition probabilities from one state of the blockchain to the next.

Equilibrium: The above elements define our stochastic game. Our equilibrium concept is Markov Perfect Equilibrium, i.e., Subgame Perfect Equilibrium with strategies restricted to depend only on the current state ω_τ.

4 Equilibrium analysis

To analyse equilibrium strategies, it is useful to first note that an upper bound on the lifetime payoff miner m can earn is

$$\mathcal{G}_m^{\max} = \left[\int_{s=0}^{s=z_m} dN_m(s)\right] G(M),$$

minus the price he paid for the cryptocurrency if he was not there at time 0. This sunk cost does not affect his strategies and we neglect it hereafter. $\mathcal{G}_m^{\max}$ is an upper bound because i) the total number of blocks solved by m before z_m is $\int_{s=z_m^-}^{z_m} dN_m(t)$, whatever his mining strategy, and ii) m cannot earn more than $G(M)$ each time he solves a block. At time t, the expectation

[17]Indeed, the timing of previous block resolutions, as well as previous mining choices, are payoff irrelevant.

of $\mathcal{G}_m^{\max}$, conditional on $z_m \geq t$, is

$$
\begin{aligned}
& E_t\left[\int_{s=0}^{t} dN_m(s) + \int_{s=t}^{z_m} dN_m(s)|z_m \geq t\right] G(M) \\
= \; & \left\{N_m(t) + E\left[\int_{s=t}^{z_m} dN_m(s)|z_m \geq t\right]\right\} G(M) = \left\{N_m(t) + \frac{\theta_m}{\lambda_m}\right\} G(M).
\end{aligned}
$$

Does there exist a natural strategy enabling miners to achieve this maximum expected payoff? The definition of $\mathcal{G}_m^{\max}$ implies that, to obtain the maximum expected payoff, all miners should be on the same chain, when any of them is hit by the liquidity shock. This is the case if all miners stick to the original chain at any time $\tau \in \mathcal{T}$. If they do so the longest chain rule (LCR) trivially holds. Our first proposition states that there exists an equilibrium in which miners follow this strategy.

Proposition 1 *There exists a Markov Perfect Equilibrium such that on the equilibrium path there is a single chain and all miners follow the LCR, thus obtaining their maximum expected payoff,* $E[\mathcal{G}_m^{\max}]$.

The intuition for Proposition 1 is the following. When all miners up to τ attach their blocks to the original chain, thus following the LCR, there is a single chain at τ. If the others abide to this strategy, then m can obtain his maximum possible expected payoff, $E[\mathcal{G}_m^{\max}|\omega_\tau]$, by also abiding to it. Hence there is no profitable one shot deviation from the strategy which consists in extending the original (and thereby longest) chain. Precisely, each miner rationally anticipates that if he deviates and solves a block, the other miners would not follow him, and the block solved out of the equilibrium path would have no value.

In the context of the strategic interaction characterised in Proposition 1, miners are not really competing to solve their block before the others. That another miner solves his block before m does not, in itself, reduce m's gains. The only thing that matters for miners to obtain the maximum payoff they get in Proposition 1 is that they coordinate well and all work on the same chain.

It is also noteworthy that the result in Proposition 1 does not depend on the number of miners M. The economic mechanism involved in Proposition 1 does not hinge on strategic behaviour. It is purely driven by coordination effects, which would also be at play in a competitive environment.

Proposition 1 emphasises that attaching blocks to the original chain is a simple way for miners to coordinate their actions, and results in a single chain with no fork. There might, however, be other ways for miners to coordinate in our stochastic game. In particular they could rely on the sunspot variable r^τ. We now exhibit an equilibrium in which conditioning actions on r^τ leads to equilibria with forks.

Intuitively, suppose miners follow the original chain until the realisation of the sunspot variable is such that miners anticipate a fork. As shown below, because of coordination effects, this anticipation is self fulfilling.

More precisely, let τ^f be the first time at which the sunspot variable is above $1-\varepsilon$ (where ε can be arbitrarily small), and let $n(\tau)$ denote the index of the last block solved by time τ. We now state our next proposition:

Proposition 2 *Consider an arbitrary integer f. There exists a Markov Perfect Equilibrium such that the following occurs on the equilibrium path: As long as $r^\tau \leq 1-\varepsilon$, or $f \geq n(\tau)$, there is a single chain and all miners chain their current block to $B_{n(\tau)}$. At the first time τ such that $r^\tau > 1-\varepsilon$ and $f < n(\tau)$, each miner chains his current block to $B_{n(\tau)-f}$. Afterwards, miners chain their current block to the last solved block on the chain including the edge $(B_{n(\tau)-f}, B_{n(\tau)+1})$.*

In the statement of the proposition we focus on what happens on the equilibrium path. In the proof in the appendix, we characterise the equilibrium strategy profile for any state. The intuition of Proposition 2 is the following: If I expect all to fork to $B_{n(\tau)-f}$, and if I choose to deviate and *not* fork, any block I solve will not be followed by the other miners, and I will earn no reward for this block. Rationally anticipating this, the rewards I obtain on the new chain become more valuable, therefore I choose to do like the others and fork.

Miners' behaviour in Proposition 2 is reminiscent of actual participants' behavior during the 2013 Bitcoin fork reported in Subsection 2.2. Just like BTC Guild lost the rewards from blocks mined on the 0.8 branch, miners in Proposition 2 lose rewards from blocks $B_{n(\tau)-f+1}$ to $B_{n(\tau)}$, since the fork stemming from $B_{n(\tau)-f}$ becomes the only active chain.[18] They fork nevertheless because they anticipate that the others do. Consequently, these miners earn less than $\mathcal{G}_m^{\max}$, while the other miners do not earn more than $\mathcal{G}_m^{\max}$.

[18] This might also eliminate some of the underlying transactions included in blocks $B_{n(\tau^f)-f+1}$ to $B_{n(\tau^f)}$.

Thus the forking equilibrium in Proposition 2 is Pareto dominated by the single chain equilibrium in Proposition 1.

Observe that, like Proposition 1, Proposition 2 does not depend on the number of miners M. Both propositions hinge on coordination effects, which also arise in a competitive environment.

While in the previous proposition, in spite of forking, there was eventually a single chain, we now show that forking can lead to the persistent coexistence of different branches, as in the examples of the Ethereum 2016 and Bitcoin 2017 forks.

As in Proposition 2, we consider the possibility that, at any time τ^f, the realisation of the sunspot can suggest that some miners fork to a new chain. This can, for instance, give rise to two coexisting chains at time $\tau > \tau^f$, the original chain, including the blocks linked by the sequence of edges

$$(B_0, B_1), ...(B_{n(\tau^f)-f}, B_{n(\tau^f)-f+1}), ...$$

and a new chain, including the blocks linked by

$$(B_0, B_1), ...(B_{n(\tau^f)-f}, B_{k+1}), ...$$

with $k \geq n(\tau^f)$.

The number of blocks solved by m after $B_{n(\tau^f)-f}$ on any of these two chains defines the vested interest of m on that chain. We denote the vested interests of miner m at time τ on the original and the new chain by $v_m^o(\tau)$ and $v_m^n(\tau)$ respectively. For example, suppose miner m keeps mining the original chain. The vested interest of that miner on the original chain at time τ is equal to $v_m^o(\tau) = N_m(\tau) - N_m(\tau(B_{n(\tau^f)-f}))$ (where $\tau(B_{n(\tau^f)-f})$ is the stopping time at which $B_{n(\tau^f)-f}$ is solved), while his vested interest on the new chain is $v_m^n(\tau) = 0$. Alternatively, consider miner m' who mines the new chain from time τ^f on. The vested interest of that miner on the original chain at time τ is $v_{m'}^o(, \tau) = N_{m'}(\tau^f) - N_{m'}(\tau(B_{n(\tau^f)-f}))$, while his vested interest on the new chain is $v_{m'}^n(\tau) = N_{m'}(\tau) - N_{m'}(\tau^f)$. For miners switching between the original chain and the new one, vested interests are a bit more intricate, but follow the same logic.

In our model miners hold their rewards until z_m. Our next result illustrates the consequences of these vested interests. To state that result, rank the miners by their vested interest in the original chain at time τ^f as follows

$$\frac{\Pr(z_m = \tau')}{\Pr(N_m(\tau') - N_m(\tau^f) = 1)} v_m^o(\tau^f) \leq \frac{\Pr(z_{m+1} = \tau')}{\Pr(N_{m+1}(\tau') - N_{m+1}(\tau^f) = 1)} v_{m+1}^o(\tau^f),$$

where $\Pr(z_m = \tau')$ is the probability that at the next stopping time τ', miner m is hit by a liquidity shock, and $\Pr(N_m(\tau') - N_m(\tau^f) = 1)$ is the probability that he solves his block at τ'.

Consider the following condition.

Condition 1 *For any M and any $K < M$, $G(K)+G(M-K) = G(M)$, and ω_τ is such that there is a single chain and there exists $K \in \{Int(\frac{M}{2})+2, ...M\}$ (where Int denotes the integer part) such that*

$$G(M-K) \leq \frac{G(M-K-1)+G(M-K+1)}{2} \tag{1}$$

and for $m > K$

$$\frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K)-G(M-K)) < v_m^o(\tau)(G(M-K)-G(M-K-1)) \tag{2}$$

while for $m \leq K$

$$\frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K)-G(M-K)) > v_m^o(\tau)(G(M-K+1)-G(M-K)) \tag{3}$$

The assumption that for any M and any $K < M$, $G(K) + G(M-K) = G(M)$, simplifies the presentation of Condition 1. However, Proposition 3 below also holds in the more general case where $G(K)+G(M-K) \leq G(M)$.[19]

Consider an arbitrary integer f. Let τ^f be the first time at which $r^\tau > 1-\varepsilon$, $f < n(\tau)$ and Condition 1 holds.

Proposition 3 *For ε sufficiently small, there exists a Markov Perfect Equilibrium in which, on the equilibrium path, the following occurs: As long as $\tau < \tau^f$ there is a single chain and all miners chain their current block to $B_{n(\tau)}$. At τ^f, all miners $m \leq K$ (defined in Condition 1) chain their current block to $B_{n(\tau^f)-f}$ and follow that chain afterwards, while the other miners chain their current block to $B_{n(\tau^f)}$ and follow that chain afterwards.*

[19] In addition to notational changes, it would require imposing an (arbitrarily large) upper bound on miners' vested interests.

The intuition for this result is the following. First note that for some miners to fork, we must have that the left-hand-side of (3) be non negative, which implies that $K \geq \frac{M}{2} + 1$. That is, in Proposition 3, persistent forks can occur only if a majority of miners choose to fork and this is expected by all.

Now, suppose all miners expect that a majority will fork and this will result in two coexisting chains and consider the choice of miner m between forking and remaining on the original chain. For m, the benefit from forking is that the blocks he will mine on the new chain will be worth $G(K)$, which is larger than the value of blocks mined on the original chain, $G(M-K)$. This benefit is large if the probability that m solves a block in any given period, $\Pr(N_m(\tau') - N_m(\tau) = 1)$, is large relative to the probability that m leaves the game because of a liquidity shock, $\Pr(z_m = \tau')$. Note that the ratio of these probabilities increases with the ratio of the mining intensity θ_m to the liquidity shock intensity, λ_m. This benefit is captured in the left-hand-side of equations (2) and (3) in Condition 1.

On the other hand, the cost of mining the new chain is that it reduces the value of the blocks already mined on the original chain. For instance, if miner $m > K$ deviates from the equilibrium strategy and mines the new chain, he reduces the value of all the blocks he solved on the original chain from $G(M-K)$ to $G(M-K-1)$. This cost is large if m has large vested interests in the original chain, that is, if $v_m^o(\tau)$ is large. This cost is captured in the right-hand-side of equations (2) and (3) in Condition 1.

Overall, Proposition 3 shows that the endogenous sorting between miners who prefer to stick to the original chain and those who fork is driven by two forces: the number of blocks that a miner expects to solve in the future, $\frac{\theta_m}{\lambda_m}$, and his vested interest in the original chain, $v^o(m,\tau)$. A miner is more likely to fork when the former is higher, and the latter is lower.

Last, inequality (1) ensures that the set of miners who choose to stick to the original chain has no intersection with the set of miners who prefer to fork. Figure 5 represents the competing chains sustained at the equilibrium of Proposition 3.

Unlike Proposition 1 and Proposition 2, the conditions in Proposition 3 depend on the number of miners. More precisely, the tradeoffs faced by the miners involve the effect of their mining strategy on the value of their rewards. If miners were competitive and their choice had no impact on the value of their rewards, this strategic effect would not arise.

Finally note that the equilibrium outcome in Proposition 3 is Pareto

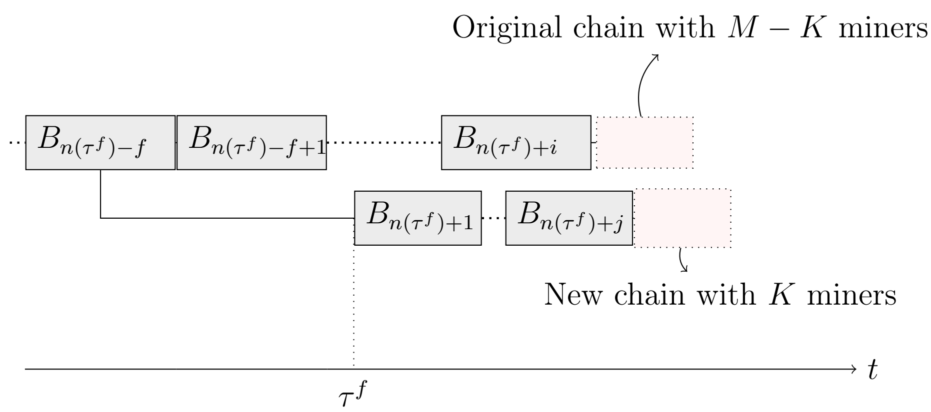

Figure 5: Equilibrium of Proposition 3

dominated by that in Proposition 1. Again, forking reduces the total gains of the miners, and yet it can arise in equilibrium.

5 Frictions

So far we have considered the frictionless case, in which, relying on abstract sunspots, we have shown that coordination effects can lead to forks and multiplicity of equilibria. We now introduce frictions by allowing for information delays, double spending attempts, and hard fork upgrades, and show that they can play a similar role as sunspots in triggering forks. This underlines that the fundamental problem is not the frictions per se, but the coordination effects that necessarily arise when seeking consensus in a decentralised protocol.

5.1 Information transmission delays

One way to introduce frictions is to consider delays in the dissemination of information through the network. Such delays could induce short term forks. As mentioned above, Nakamoto (2008) considered that possibility and conjectured that miners would follow the LCR and that this would resolve short term forks. We explore below how delays can give rise to forks and multiplicity of equilibria.

To model information transmission, we introduce the following modification of our framework: To keep things as simple as possible, we assume that a delay in information transmission can happen only once. Thus, as long as all miners have observed when all previous blocks were solved, each time a new block B_n is solved there is a probability η that one (and only one) of the miners does not observe that event. In that case each of the $M-1$ miners has an equal chance of not observing the block solved by the other miner. (when this happens all miners don't have the same stopping times.) As soon as the next block (B_{n+1}) is solved, the miner who did not observe that B_n was solved learns that information.

Proposition 4 *When miners can observe solved blocks with a delay, there exists a Markov Perfect Equilibrium such that on the equilibrium path miners always mine the chain which they perceive to be the longest. If there are two chains of the same length, each miner keeps mining the chain he was mining just before.*

At the equilibrium presented in Proposition 4, because of information transmission delays, two chains of the same length can appear. At this point, there is a fork. In that case, miners continue mining the chain on which they were active before the fork. When one chain becomes strictly longer, miners apply the LCR and the shortest branch of the fork becomes orphaned. This is in line with the conjecture of Nakamoto (2008). The equilibrium described in Proposition 4 therefore illustrates the robustness of the LCR equilibrium of Proposition 1 in the presence of observation delays. Yet, reflecting coordination issues, other equilibria can be sustained in which miners deviate from the Nakamoto (2008) rule. This is illustrated in the following proposition.

Proposition 5 *When miners can observe solved blocks with a delay, there exists a Markov Perfect Equilibrium such that on the equilibrium path miners always mine the chain that they perceive as the longest. If there are two chains of the same length, miners always chain to the forking branch, and the original chain becomes orphaned.*

In Proposition 5, miners follow the LCR on the equilibrium path, but, when the information delay causes a fork, they abandon the chain on which they were active and follow the fork. While similar in spirit to the sunspot analysed in Section 4, Proposition 5 offers a concrete example of an event that can trigger a fork. In the proposition, the fork is only one-block-long because the delay can only affect the observation of one block. By extension, longer forks could be sustained if delays affected more blocks. Note that delays are not necessarily due to network latency. In the case of the Bitcoin March 2013 Fork, a delay in the observation of several blocks occurred, because one block was mistakenly rejected by computers using one version of the mining software.

5.2 Double spending

Another important potential issue outlined in Nakamoto (2008) is double spending. We study below whether it can arise at equilibrium. In the spirit of the modelling of delays above, assume that after each block is solved, there is a probability η' that one miner can divert the payment S from a transaction included in the last block solved on the original chain. To earn S, the miner needs to create a fork from the parent of that block, that becomes the only active chain. Assume that this opportunity to double spend occurs only once and denote $\Pr(m = m(B))$ the probability that at any time τ, m solves the next block.

Proposition 6 *Assume each miner can receive an opportunity to double spend and that for any miner*

$$S > \frac{G(M)(2 - \Pr(m = m(B)))}{\Pr(m = m(B))}. \tag{4}$$

There exists a Markov Perfect Equilibrium such that on the equilibrium path miners always mine the longest chain, except the miner who has the opportunity to double spend. The latter tries to create a fork longer than the original chain. If he succeeds, all miners chain to his fork and the original chain is orphaned.

When a miner spots a double spending opportunity, he endeavours to solve two blocks in a row before the other miners solve any new block on the original chain. If he succeeds, a fork occurs which enables the miner to recover S and therefore double spend it. The equilibrium described in Proposition 6 relies on a similar coordination effect as that of Proposition 4: miners have an interest in following the longest chain when they anticipate that all other miners (except possibly a miner who spotted a double-spending opportunity) do the same. This in turn can induce a miner to create a fork that will be followed by all miners if it becomes the longest. But in contrast with the case of delays, the fork does not start accidentally, it is intentionally triggered by the miner who tries to double spend. This miner finds it profitable to do so if (4) holds. Condition (4) is explained as follows. When trying to create a fork after spotting S, a miner has a high probability that his fork does not become the longest chain. Hence, he bears the risk of mining blocks on the fork in vain, and obtaining no reward. The higher the double spending S, the more it compensates for these losses. Similarly, the higher $\Pr(m = m(B))$, the higher the probability that the fork succeeds. In Proposition 6, (4) holds for all miners, so that all miners can start a fork. Alternatively, one can construct an equilibrium such that only the miners with the largest computing power can double spend. The condition under which double spending can occur would then be less demanding than (4).

The theoretical possibility of double spending is only relevant in practice if condition (4) is likely to hold. A back-of-the-enveloppe computation suggests that if λ is small and there are 15 identical miners, (4) can be approximated as $S > 30G(M)$. For Bitcoin, $G(M) = 12.5$ bitcoins in 2017, hence for (4) to hold, S must be larger than 375 bitcoins, which is a sizeable amount.

5.3 Upgrades

Blockchain participants may want to change the rules governing their mining process by introducing upgraded versions of the mining software. To study the impact of these upgrades, we assume it is common knowledge that just after the n^{th} block on the original chain has been solved, a new technology is introduced. Then, miners must choose between staying with the existing technology, $C = 0$, or adopting the new technology $C = 1$. From this point on, miners choose between $C = 0$ and $C = 1$ for each block they mine. To capture the notion of hard fork, we assume that miners can only chain their block to a block solved with the same technology. We also assume that each miner m derives a private benefit from using technology C. Private benefits can reflect a cost advantage of using one technology over another. For instance, it is argued that the mining pools controlled by Bitmain, an ASIC manufacturer, favoured the Bitcoin Cash solution because they have access to a patented mining-enhancing device[20] that cannot be used with the SegWit solution adopted on Bitcoin in august 2017. Alternatively, private benefits can reflect ideological preferences: the attempt at increasing the size of blocks on Bitcoin with the SegWit2X hard fork was supported by a group of large mining pools. It was eventually defeated in november 2017 by the Bitcoin core developers who opposed the principle of a hard fork (and the idea of letting the SegWit2X proponents impose their solution).

To model private benefits, we assume that when solving a block with technology C, miner m obtains a reward $(1 + b_m(C))G(K)$ where K is the number of miners active on the chain containing this block and $b_m(C) > -1$ for all m. Without loss of generality, we normalise $b_m(1)$ to 0.

Proposition 7 *There exists a Markov Perfect Equilibrium in which, on the equilibrium path, all miners follow the LCR and choose technology $C = 0$, and another equilibrium in which, on the equilibrium path, all miners follow the LCR and choose technology $C = 1$.*

If miners anticipate that all other miners will choose one technology, say, $C = 0$, then it is a best response to choose $C = 0$ as well, whatever the level of private benefit associated with each technology. This is because each miner anticipates that the others will follow the longest chain rule with the same equilibrium technology, so there is no gain in mining a block with a different technology. The equilibria described in Proposition 7 therefore hinge on the same coordination effects as in Propositions 1 and 2.

When miners coordinate on the same technology and on following the LCR, the level and distribution of private benefits does not affect which equilibrium will prevail. In particular, it can be that $C = 1$ is chosen at equilibrium even if all miners have a preference for $C = 0$ (i.e., even if $b_m(0) > 0\ \forall m$). We explore below how a persistent fork can also be sustained at equilibrium in the presence of private benefits. To do so, assume miners $m \in \{1, 2, ...K\}$ have $b_m(0) = 0$, and miners $m \in \{K+1, ...M\}$ have $b_m(0) = b > 0$. Denote τ^f the time at which the new technology is introduced.

Proposition 8 *If $b \geq \frac{G(K)}{G(M-K)} - 1$ and $K \geq \frac{M}{2}$, there exists a Markov Perfect Equilibrium in which, on the equilibrium path, for $\tau < \tau^f$ all miners follow the LCR and for $\tau \geq \tau^f$, miners $m \leq K$ choose $C = 1$ and follow the LCR*

[20] This is the ASICBOOST technology that can increase the efficiency of the SHA-256 hash function.

on the chain of blocks mined with $C = 1$, and miners $m > K$ choose $C = 0$ and follow the LCR on the chain of blocks mined with $C = 0$.

When some miners have sufficiently large private benefits, the introduction of a new technology can give rise to a disagreement on the technology choice, which triggers a persistent fork on the blockchain. For a fork to happen, it must be that a majority of miners have no private benefit (i.e., $K \geq \frac{M}{2}$): indeed, if a miner has no private benefit attached to a technology, he chooses to mine blocks on the branch that yields the highest reward per block solved, that is, the branch with the majority of miners. If $K \geq \frac{M}{2}$, a majority of miners have no private benefit associated to $C = 0$, and can coordinate on choosing $C = 1$ (even though they do not have any private benefit associated to $C = 1$ either). When K miners choose $C = 1$, it is costly for other miners not to adopt the same technology. They are nevertheless willing to stick to $C = 0$ if their level of private benefit b is sufficiently large to compensate the loss in reward when mining blocks on a chain with only $M - K \leq K$ active miners.

This result is reminiscent of the persistent fork equilibrium described in Proposition 3 in which a minority of miners agree to split from the majority because they have vested interests attached to one branch. One difference with Proposition 3 is that here private benefits are exogenous, while vested interests arose endogenously from the distribution of past solved blocks. Another difference is that here the fork can happen with probability one, simply because some miners have different interests.

The introduction of upgrades in the blockchain bears similarities with patterns observed in open source programs, in which coordination effects and conflicting interests can lead to the splintering of programs into several variants (see Lerner and Tirole (2002)).

6 Computing capacity

We now endogenise computing power in the network to investigate the relation between equilibrium hash capacity and the socially optimal one. The determination of equilibrium capacity is similar to that in Dimitri (2017). Our contribution relative to Dimitri (2017) is to identify an externality in the computing power acquisition game, which drives a wedge between equilibrium and social optimality.

For each miner m, the instantaneous probability of solving a block, θ_m, is determined by his individual computing power, or hash power, and by the difficulty of the mining task set by the network protocol, D:[21]

$$\theta_m = \frac{h_m}{D}. \tag{5}$$

The difficulty is set so that the expected time between two blocks is equal to a constant, X (on Bitcoin $X = 10$ minutes), that is

$$X = \frac{1}{\sum_{i \in \mathcal{M}} \theta_i}. \tag{6}$$

Substituting (5) into (6), we get

$$D = X \sum_{i \in \mathcal{M}} h_i. \tag{7}$$

By construction, the difficulty D must be larger than or equal to 1. Moreover, if the total computing power $\sum_{i \in \mathcal{M}} h_i$ was lower than $1/X$ it would not be feasible to have one block solved every X units of time. Consequently we impose the technical constraint that

$$\sum_{i \in \mathcal{M}} h_i \geq 1/X. \tag{8}$$

Therefore

$$\theta_m = \frac{1}{X} \frac{h_m}{\sum_{i \in \mathcal{M}} h_i}. \tag{9}$$

We now analyse the individually optimal choice of h_m by miner m. To make this choice the miner needs to anticipate how his computing power will affect his continuation game payoff. To do so, the miner needs to form a conjecture on the equilibrium that will prevail in the mining game. For simplicity, we assume that all miners rationally anticipate that the single chain equilibrium described in Proposition 1 will prevail.

The program of miner m is

$$\max_{h_m} \frac{\theta_m}{\lambda_m} G(M) - \frac{c_m(h_m)}{\lambda_m},$$

[21] Indeed, when miners try to solve the hash problem, at each trial they have a probability $\frac{1}{D}$ to solve the problem. The hash power h_m is the number of trials per unit of time.

where $c_m(h_m)$ is the cost of using h_m per unit of time and therefore can be thought of as the rental cost of the equipment plus the cost of electricity. Miner m bears this cost until he is hit by a liquidity shock.[22]

Substituting (9) this is

$$\max_{h_m} \frac{\frac{h_m}{\sum_{i\in\mathcal{M}} h_i}}{\lambda_m X} G(M) - \frac{c_m(h_m)}{\lambda_m}.$$

It is reasonable to assume that the cost function is linear, that is $c_m(h_m) = c_m h_m$, which yields the following first order condition:

$$\frac{\left(\sum_{i\in\mathcal{M}} h_i\right) - h_m}{\left(\sum_{i\in\mathcal{M}} h_i\right)^2} \frac{G(M)}{X} = c_m. \tag{10}$$

A Nash equilibrium of the computing power acquisition game is a vector $\{h_m^*\}_{m=1,\ldots M}$ such that h_m^* is the optimal choice of miner m when he anticipates the others will choose h_{-m}^*. The following proposition presents the equilibrium computing capacity of the network.

Proposition 9 *Assume that* $c_m \leq \frac{\sum_{i\in\mathcal{M}} c_i}{M-1} \leq G(M), \forall m$. *When miners anticipate that no fork will occur on the network, their equilibrium computing capacity is defined by*

$$h_m^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i} \left(1 - c_m \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right). \tag{11}$$

The total computing capacity installed on the network is

$$\sum_{i\in\mathcal{M}} h_i^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}. \tag{12}$$

Condition $c_m \leq \frac{\sum_{i\in\mathcal{M}} c_i}{M-1}$ ensures that the solution to (11) is positive for all miners and $\frac{\sum_{i\in\mathcal{M}} c_i}{M-1} \leq G(M)$ ensures that $\sum_{i\in\mathcal{M}} h_i^* \geq \frac{1}{X}$ as requested by (8). Equilibrium individual computing power is increasing in the mining reward ($G(M)$), decreasing in the average duration between blocks (X) and in the unit cost c_m. Consequently, total network capacity decreases with $\sum_{i\in\mathcal{M}} c_i$.

[22] For simplicity, when miner m is hit by a shock, his successor inherits the computing capacity h_m.

We now compare the equilibrium network capacity to what miners would choose if they could coordinate their choice and maximise their joint profit. To do so in the simplest way, we consider the special case in which all miners have the same cost c and the same λ.

The corresponding maximisation problem is

$$\max_h M(\frac{1}{\lambda X}\frac{h}{\sum_m h}G(M) - \frac{c}{\lambda}h).$$

This is decreasing in h. So miners coordinate on the smallest possible value of h, corresponding to $D = 1$. We then have $\sum_{i\in\mathcal{M}} h_i = \frac{1}{X}$. Comparing this to (12), and recalling that $\frac{\sum_{i\in\mathcal{M}} c_i}{M-1} \leq G(M)$, there is overinvestment in computing power: if miners could cooperatively decide on their individual computing capacity, each would invest $\frac{1}{MX}$. But if all miners invest $\frac{1}{MX}$, then any miner m has an incentive to increase his own h in order to increase his probability to solve blocks, leading to an arms race in computing capacity. This uncovers an additional cost of decentralised ledgers compared to centralised ones.

7 Conclusion

Our analysis suggests that miners' incentives are key to the production of a robust consensus in a blockchain. While miners benefit from coordinating on a single chain, thereby maintaining consensus, coordination motives may also lead them to abandon portions of the blockchain. This jeopardises the blockchain's key function, i.e., producing a stable and immutable history of transactions. In addition, vested interests, by counteracting coordination motives, may lead to the persistence of multiple active chains.

Our approach could be taken in different directions going forward. First, the current design could be enriched by providing miners with additional tools to avoid coordination failures. In particular, communication among miners could play a role in this context, and provide a rationale for information channels such as IRC networks and forums, or flags attached to blocks to signify support for an upgrade. Second, one could study how modifying the current design would affect stability. A natural candidate is miners' payment scheme which could, for instance, reward some orphaned blocks, as is already the case on Ethereum. Another component of miners' profits are the rewards attached to transactions to provide incentives for faster validation.

On Bitcoin, these payments are bound to become more significant relative to rewards for solving blocks and could affect the nature of the interaction between miners. Finally, alternative protocols, such as proof-of-stake, have been put forward, if not fully implemented, and their properties could be compared to the design we study here. Overall, we hope that the current model can be a first step towards a better understanding of decentralised transaction systems.

Appendix

Notation

We summarize below notation we use throughout the proofs:

- $\tau(B_n)$ is the stopping time at which block B_n is solved,
- $n(\tau)$ is the index of the last block solved by time τ,
- $N_m(\tau)$ is the total number of blocks solved by miner m by time τ,
- $N_m^{\mathcal{C}}(\tau)$ is the number of blocks solved by miner m on chain $\mathcal{C}$ by time τ. In particular, $N_m^o(\tau)$ is the number of blocks solved by m on the original chain,
- $p(B_n)$ is the index of the block to which B_n is chained (his parent).

The following Lemma implies that a candidate strategy profile $\{\sigma_m^*\}_{m\in\mathcal{M}}$ forms a Markov Perfect Equilibrium (MPE) if and only if no miner has a profitable one-shot deviation after any possible history of the game ω_τ.

Lemma 10 *Our blockchain game is continuous at infinity.*

Proof of Lemma 10

Denote by $J(\sigma_m)$ the expected payoff of miner m if he follows strategy σ_m. Consider an alternative strategy, σ_m', that prescribes the same actions as σ_m until time T and differs afterwards. The difference between the two expected payoffs can be written as

$$\begin{aligned} J(\sigma_m) - J(\sigma_m') &= \Pr(z_m \leq T)\mathbb{E}[J(\sigma_m) - J(\sigma_m')|z_m \leq T] \\ &\quad + \Pr(z_m > T)\mathbb{E}[J(\sigma_m) - J(\sigma_m')|z_m > T]. \end{aligned}$$

Now, by definition,

$$E[J(\sigma_m) - J(\sigma_m')|z_m \leq T] = 0.$$

Moreover

$$\lim_{T\to\infty} \Pr(z_m > T) = 0,$$

and $J(\sigma_m) - J(\sigma'_m)$ is bounded, since $\mathcal{G}_m^{\max}$ is finite. Hence,

$$\lim_{T \to \infty} J(\sigma_m) - J(\sigma'_m) = 0,$$

which ensures that our game is continuous at infinity.

QED

Proof of Proposition 1

The candidate equilibrium strategy specifies that miners always chain their block to the last block on the original chain. We let B_n denote that block, and check that no miner has a profitable one-shot deviation.

Consider the strategy of miner m at time τ after history ω_τ. We break the analysis into three cases, the probabilities of which are independent of the miners' actions (they reflect the distributions of independent Poisson processes with exogenous intensities).

i) Suppose the next event is z_m. The equilibrium strategy prescribes that all miners mine the original chain. Therefore if m follows the equilibrium strategy and chains his block to the last block on the original chain, B_n, he earns $G(M)$ for each block he solved on the original chain and $G(0) = 0$ for any other block.

   Suppose m deviates and does not chain his block to B_n. By definition, he cannot earn more than $G(M)$ for each block he solved on the original chain. In addition, he cannot earn more than $G(1) = 0$ for each block he solved on forks since all other miners mine the original chain. Hence deviating is not strictly profitable in this case.

ii) Suppose the next event is that a block is solved by another miner than m at time τ'.[23] Then the state of the blockchain at τ', $\omega_{\tau'}$, does not depend on m's action at τ. By definition, $\omega_{\tau'}$ captures all the payoff-relevant information, hence m's action at τ does not affect his payoff.

iii) Suppose the next event is that m solves block $B_{n(\tau)+1}$ at time τ'. Since all other miners play the equilibrium strategy going forward and m himself

[23] The next event can also be that nothing happens, or that another miner is hit by a liquidity shock. Which block m chose as a parent block is also irrelevant in those cases. For brevity, we ignore these cases in the remainder of the proofs.

reverts to mining the original chain after τ' (one-shot deviation), which block m chose as a parent block at τ does not affect the payoff m expects from previously mined blocks or from future blocks. Consequently, m's payoff in any one shot deviation differs from his equilibrium payoff only in the reward he obtains for B_n. This reward is $G(M)$ if m played the equilibrium strategy and chained $B_{n(\tau)+1}$ to B_n, and $G(0)$ if he chained $B_{n(\tau)+1}$ to any other block as in that case, no miner will ever chain a block to $B_{n(\tau)+1}$. Hence deviating is strictly dominated in this case.

Overall, there is no state ω_τ in which a one-shot deviation gives m a strictly higher expected payoff than the candidate equilibrium strategy, which therefore forms a MPE.

QED

Proof of Proposition 2

Let τ^f be the first time the sunspot variable is above $1-\varepsilon$ and f is strictly lower than the number of blocks $n(\tau)$. We call "new chain" the chain created by the fork. Formally, for every $\tau > \tau^f$, the new chain, if it exists, is the chain containing $(B_{n(\tau)-f}, B_k)$ that preexists all other chains containing $(B_{n(\tau)-f}, B_k)$, where $k \equiv \min\{\hat{k} > n(\tau^f), (B_{n(\tau)-f}, B_{\hat{k}}) \in \omega_\tau\}$.

Our candidate equilibrium strategy specifies the following:

a) *Before the fork:* If $\tau < \tau^f$, miners chain their block to the last block on the original chain.

b) *At the fork inception and after the fork:* If $\tau \geq \tau^f$, miners chain their block to the last block on the new chain, or to $B_{n(\tau)-f}$ if the new chain does not exists.

We now show that miner m does not have a profitable one-shot deviation from this strategy at time τ.

a) *Before the fork.* Since m's actions do not affect the occurrence of the sunspot, for $\tau < \tau^f$ the proof operates along the same lines as the proof of Proposition 1.

b) *At the fork inception and after the fork.* Suppose $\tau \geq \tau^f$. As in the proof of Proposition 1, we can restrict attention to the case where m solves the next block, $B_{n(\tau)+1}$, at time τ'. In that case, m's equilibrium payoff differs from his payoff in a one-shot deviation only in the reward for block $B_{n(\tau)+1}$. Since m expects all miners, including himself, to attach their block to the last block on the new chain after τ', m's reward for block $B_{n(\tau)+1}$ is $G(M)$ if he played the equilibrium strategy and chained his block to the last block on the new chain (or to $B_{n(\tau)-f}$), and $G(0) = 0$ if he chained $B_{n(\tau)+1}$ to any other block.

QED

Proof of Proposition 3

Preliminary steps

We define the new chain as in the proof for Proposition 2, and let $v^n_m(\tau) = N^n_m(\tau) - N^n_m(\tau^f)$ be miner m's vested interests on that chain, that is, the number of blocks he solved on the new chain after τ^f.

To define our equilibrium strategies, we use the following condition, which we will derive explicitly in the proof:

Condition 2 *For $\tau \geq \tau^f$, ω_τ is such that for $m > K$*

$$v^o_m(\tau)(G(M-K) - G(M-K-1)) - v^n_m(\tau)(G(K+1) - G(K)) \geq \frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K) - G(M-K)), \quad (13)$$

while for $m \leq K$

$$v^o_m(\tau)(G(M-K+1) - G(M-K)) - v^n_m(\tau)(G(K) - G(K-1)) \leq \frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K) - G(M-K)). \quad (14)$$

Our candidate equilibrium strategy profile specifies the following:

a) *Before the fork:* If $\tau < \tau^f$, miners chain their block to the last block on the original chain.

b) *At the fork inception and after the fork:* If $\tau \geq \tau^f$ and Condition 2 holds, miners $m \leq K$ chain their block to $B_{n(\tau^f)-f}$ if the new chain does not exist, and to the last block on the new chain otherwise, while miners $m > K$ chain their block to the last block on the original chain.

c) *After the fork off-path:* Suppose $\tau > \tau^f$ and Condition 2 does not hold. Let $\Delta\omega \equiv \omega^\tau \backslash \omega^{\tau^f}$ (i.e., $\Delta\omega$ contains the history of the game between τ^f and τ). Then for every $\tau' \geq \tau$, all miners play the strategy prescribed after history $\omega^{\tau'} \setminus \Delta\omega$ that is defined in b). In playing strategies defined in b), miners consider that the original and the new chain are defined with respect to history $\omega^{\tau'} \setminus \Delta\omega$. [24]

As will become explicit below, the specification of the equilibrium strategy in states described in c) is useful to rule out certain types of deviations.

To show that a miner does not have a profitable one shot deviation, we consider each of the cases above in turn.

Proof of part a): Before the fork.

Miner m's one-shot deviation from equilibrium at time $\tau < \tau^f$ has two effects on his expected payoff. First, it can affect the distribution of vested interests on the original chain at future times τ such that $r^\tau > 1-\varepsilon$. Second, as in the proof for Proposition 2, it can impact the value of the block m chooses to mine. As in the previous proofs, these effects are affected by m's action at τ only if he solves the next block, $B_{n(\tau)+1}$.

Consider the first effect. The occurrence of a fork may reduce the payoff participants receive from the blocks they will mine after τ^f, as well as some of the blocks they have mined before τ^f, namely, the f blocks between the last block solved before the sunspot, $B_{n(\tau^f)}$ and the first block that does not belong to the new chain, $B_{n(\tau^f)-f+1}$. For each of these blocks, as well as for the blocks solved after τ^f, the maximal loss for miner m is $G(M)$. In addition m's deviation has an impact on the materialisation of this loss only if the sunspot occurs before m's liquidity shock when m plays the equilibrium strategy. Hence, an upper bound on this loss, or equivalently, on the gain from reducing the likelihood of a fork via a deviation is

$$\Pr(\tau^f < z_m|\omega_\tau)[f + \frac{\theta_m}{\lambda_m}]G(M).$$

[24]In words, miners play as if the blocks solved between τ^f and τ do not exist.

Now,

$$\Pr(\tau^f < z_m|\omega_\tau) = \int_{z_m=\tau}^{\infty} \Pr(\tau^f < z_m|\omega_\tau, z_m)\lambda_m e^{-\lambda_m z_m} dz_m.$$

Observe that

$$\Pr(\tau^f < z_m|\omega_\tau, z_m) < \Pr(\exists \tau < z_m, r^\tau > 1-\varepsilon|\omega_\tau, z_m) = 1-\Pr(\forall \tau < z_m, r^\tau \leq 1-\varepsilon|\omega_\tau, z_m).$$

Moreover,

$$\Pr(\forall \tau < z_m, r^\tau \leq 1-\varepsilon|\omega_\tau, z_m) = \mathbb{E}[(1-\varepsilon)^{\nu(\tau,z_m)}|\omega_\tau, z_m],$$

where $\nu(\tau, z_m)$ is the number of stopping times between τ and z_m. Now, for small ε, a Taylor expansion yields

$$(1-\varepsilon)^{\nu(\tau,z_m)} \approx 1 - \nu(\tau, z_m)\varepsilon.$$

Hence, for small ε,

$$\Pr(\forall \tau < z_m, r^\tau \leq 1-\varepsilon|\omega_\tau, z_m) \approx 1 - \mathbb{E}[\nu(\tau, z_m)]\varepsilon.$$

Hence, if ε is close enough to 0, $\Pr(\tau^f < z_m|\omega_\tau, z_m)$, and therefore the gain from reducing the likelihood of a fork via a deviation, can be arbitrarily small.

Next consider the second effect. If miner m solves $B_{n(\tau)+1}$ but this block is not on the original chain, no further block will be chained to it, since all miners henceforth play the equilibrium strategy. Hence the expected payoff for this block is $G(0) = 0$. If instead m was following the equilibrium strategy when he solved $B_{n(\tau)+1}$, the expected payoff from this block is strictly positive.

Overall, the first effect, which reflects the maximum gain from a one-shot deviation can be set arbitrarily close to 0, while the second effect, which reflects the cost of a one shot deviation, is bounded away from 0. Hence, there is no profitable one-shot deviation.

Proof of part b): At or after the fork:

i) Consider first a deviation by a miner $m > K$.

   Any deviation other than chaining to the last block on the new chain is ruled out by similar arguments as in Proposition 1. Hence we just check that m prefers to chain his block to the last block on the original chain, rather than to the last block on the new chain. As in the proof of Proposition 1, a one-shot deviation affects m's payoff only if the next stopping time τ' corresponds to two possible events: either m is hit by a liquidity shock or m solves a block.

- Suppose miner m solves a block at τ', i.e., $N_m(\tau') - N_m(\tau) = 1$. If Condition 2 is still true at τ', since every miner, including m, reverts to the equilibrium strategy from τ' on, the only impact of the deviation is that m earns $G(K)$ for block $B_{n(\tau')}$ instead of $G(M-K)$ under the equilibrium strategy. If Condition 2 is not true at τ', then from c), the impact of the deviation is that m earns 0 for block $B_{n(\tau')}$ instead of $G(M-K)$ under the equilibrium strategy, and loses all rewards for blocks solved between τ^f and τ'.
- Suppose miner m is hit by a liquidity shock at τ', i.e., $z_m = \tau'$. Then his payoff under the deviation is

$$v_m^o(\tau)G(M-K-1) + v_m^n(\tau)G(K+1) + N_m^o(\tau(B_{n(\tau^f)-f}))G(M)$$

instead of

$$v_m^o(\tau)G(M-K) + v_m^n(\tau)G(K) + N_m^o(\tau(B_{n(\tau^f)-f}))G(M)$$

under the equilibrium strategy.[25]

It follows that there is no profitable deviation if

$$\Pr(N_m(\tau') - N_m(\tau) = 1)[G(K) - G(M-K)] \leq$$
$$\Pr(z_m = \tau')[v_m^o(\tau)(G(M-K) - G(M-K-1)) - v_m^n(\tau)(G(K+1) - G(K))],$$

which is exactly inequality (13) in Condition 2.

ii) Consider next a deviation by a miner $m \leq K$. A symmetric reasoning yields that there is no profitable deviation if

$$\Pr(N_m(\tau') - N_m(\tau) = 1)[G(K) - G(M-K)] \geq$$
$$\Pr(z_m = \tau')[v_m^o(\tau)(G(M-K+1) - G(M-K)) - v_m^n(\tau)(G(K) - G(K-1))],$$

which is exactly (14) in Condition 2.

[25]Note that we used the assumption that $\forall K$, $G(M) = G(M-K) + G(K)$ to write down miner m's payoff from blocks solved before $\tau(B_{n(\tau^f)-f})$.

Next, see that at $\tau = \tau^f$, $v_m^n(\tau^f) = 0$ for all miners. Inequality (13) is then written:

$$\frac{\Pr(N_m(\tau') - N_m(\tau^f) = 1)}{\Pr(z_m = \tau')}[G(K)-G(M-K)] < v_m^o(\tau^f)[G(M-K)-G(M-K-1)],$$

which is exactly inequality (2) in Condition 1. Similarly, inequality (14) is then written:

$$\frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}[G(K)-G(M-K)] > v_m^o(\tau)[G(M-K+1)-G(M-K)]$$

which is exactly inequality (3) in Condition 1.

Furthermore, if miners adhere to the equilibrium strategy, then miners $m \leq K$ always mine the new chain so that inequality (3) in Condition 1 implies that inequality (14) in Condition 2 is true at any $\tau \geq \tau^f$. Symmetrically, given that miners $m > K$ stick to the original chain, Condition 2 is always verified after τ^f. Hence, given that Condition 1 holds at τ^f, then for $\tau > \tau^f$, Condition 2 holds on the equilibrium path.

Last, see that inequality (1) in Condition 1 guarantees that (2) and (3) cannot be satisfied jointly for the same miner m.

Proof of part c): After the fork off-path

Suppose ω_τ is as described in c). Then given that all other players play the equilibrium, m's payoff from adhering to the equilibrium strategy is as in b) above. Following the same logic as in the proof of b), other deviations can be ruled out.

QED

Proof of Proposition 4

Our candidate equilibrium strategy specifies the following:

a) If a miner solved a block outside the original chain thereby creating a one-block-long fork as long as the original chain, that miner chains his next block to the block he just solved.

b) Otherwise, each miner chains his current block to the last block solved on the original chain, except if there is a fork starting with two blocks consecutively solved by the same miner, longer than the original chain. In that case, each miner chains his block to the longest chain, which miners consider to be the original chain from that point on.[26]

We further assume that $G(M-1)+G(1)=G(M)$. Indeed, there can be a transient fork created by one miner who did not observe in time the actual state of the original chain. If another miner is hit by a liquidity shock precisely when the fork is being formed, the blocks previously solved by that other miner, which with certainty will not become orphaned, are worth at the time of the fork $G(M-1)+G(1)$. Given equilibrium strategies, the same blocks will be worth $G(M)$ just after the fork is resolved. Our assumption means that these blocks have the same value at and after the fork. This assumption also mirrors the assumption that $G(1)=G(0)$: if only one miner is not on the same chain as the others, the reward for solving blocks is not affected. We specify below when this assumption is used.[27]

Proof of part a)

Let B_n be the last block solved on the original chain. Suppose that at time τ, miner m has just created a one-block-long fork as long as the original chain by solving $B_{n(\tau)}$ that is chained to B_n's parent, $p(B_n)$.[28]

As earlier, the relevant choice for m is between chaining his next block to $B_{n(\tau)}$ (the equilibrium strategy) and chaining it to B_n (the only relevant deviation). As in the proof of Proposition 1, a one-shot deviation affects m's payoff only if the next stopping time corresponds to two possible events: either m is hit by a liquidity shock or m solves a block.

i) Suppose the next event is z_m. If m deviated and chained his block to B_n his payoff is

$$G(0)+N_m^o(\tau(B_n))G(M).$$

Indeed, all miners, including m, chain to B_n. Hence, m earns $G(0)$ for solving block $B_{n(\tau)}$ and $G(M)$ for every block he solved on the original chain up to $\tau(B_n)$.

[26] This is to define equilibrium strategies if a second fork occurs off the equilibrium path.

[27] This assumption simplifies the proofs but is not necessary to establish the results. If we instead assume that $G(M-1)+G(1)<G(M)$, the proposition would still hold provided that the probability of a liquidity shock is sufficiently small.

[28] Since the equilibrium strategies are defined for all states, including those which are not on the equilibrium path, we cannot exclude that out of equilibrium, some blocks are solved outside the original chain before or after B_n is solved: $p(B_n)$ is not necessarily B_{n-1}, and $B_{n(\tau)}$ is not necessarily B_{n+1}.

If, instead, m followed the equilibrium strategy and chained his block to $B_{n(\tau)}$ his payoff is

$$G(1) + N_m^o(\tau_{p(B_n)})[G(M-1) + G(1)] + \mathbb{1}_{\{m=m(B_n)\}} G(M-1).$$

Since m is the only miner chaining to $B_{n(\tau)}$, he earns $G(1)$ for block $B_{n(\tau)}$. In addition, m earns $G(M-1)+G(1)$ for each block he solved on the original chain up to $p(B_n)$, reflecting the occurence of a fork where m chains to $B_{n(\tau)}$ and the $M-1$ other miners chain to B_n. Finally, m earns $G(M-1)$ for B_n if he solved it.

Since by assumption $G(M-1) + G(1) = G(M)$ and $G(1) = 0$, the deviation is not strictly profitable in that case

ii) Suppose the next event is that m solves block $B_{n(\tau)+1}$.

- If m deviated and chained his block to B_n, the original chain becomes the only active chain and $B_{n(\tau)}$ is orphaned. m's expected gain is

$$N_m^o(\tau(B_n))G(M) + G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

  Indeed, m earns $G(M)$ for each block he solved on the original chain up to B_n and for $B_{n(\tau)+1}$. The conditional expectation is his expected reward for the blocks solved after $\tau(B_{n(\tau)+1})$ if none becomes orphaned.[29] $\mathcal{L}(\tau(B_{n(\tau)+1})$ is the expected loss due to one m's blocks solved after $\tau(B_{n(\tau)+1})$ becoming orphaned. On the equilibrium path, orphaned blocks occur iff a miner observes a block with delay and creates a successful fork.

- If instead m played the equilibrium strategy and chained his block to $B_{n(\tau)}$, the chain including $B_{n(\tau)}$ and $B_{n(\tau)+1}$ becomes the longest, hence the only active one. m's expected gain is

$$N_m^o(\tau(p(B_n)))G(M) + 2G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})),$$

[29] As before, if z_m occurs when a fork starts, these previously solved blocks are worth $G(M-1) + G(1)$ which is equal to $G(M)$ by assumption in that case.

where the second term capture rewards for $B_{n(\tau)}$ and $B_{n(\tau)+1}$.

It follows that a deviation is strictly dominated in this case.

Proof of part b)

As earlier, B_n is the last block solved on the original chain.

1) Suppose that at time τ, there is no fork of two consecutive blocks solved by the same miner and longer than the original chain.

For any miner m (who has not started a fork), the two relevant choices are to follow the equilibrium strategy and chain his block to B_n, or to create a fork by chaining his block to $p(B_n)$ and try solving two blocks in a row (other deviations are ruled out by the same reasoning as in Proposition 1).

As in the proof of Proposition 1, a one-shot deviation affects m's payoff only if the next stopping time corresponds to two possible events: either m is hit by a liquidity shock or m solves a block. If the next event is z_m, m's payoff is $N^o_m(z_m)G(M)$ (if there is no fork), or $N^o_m(z_m)(G(M-1)+G(1)) = N^o_m(z_m)G(M)$ (if a fork was started by another miner) whether he follows the equilibrium strategy or deviates. If the next event is that m solves block $B_{n(\tau)+1}$, there are two possible continuations: Either another miner does not observe that m solved $B_{n(\tau)+1}$ or all miners observe that m solved $B_{n(\tau)+1}$. The probabilities of these two events are independent of m's action, we consider them in turn.

i) If all miners observe that m solved $B_{n(\tau)+1}$, m's expected gain if he followed the equilibrium strategy and chained $B_{n(\tau)+1}$ to B_n is

$$(N^o_m(\tau_{B_n})+1)G(M) + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] \\ - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

The first term is the reward for blocks solved up to $\tau(B_n)$ plus the reward for mining $B_{n(\tau)+1}$ when the latter remains on the original chain. The conditional expectation is m's expected reward for solving blocks after $\tau(B_{n(\tau)+1})$. The last term, $\mathcal{L}(\tau(B_{n(\tau)+1}))$ is the expected loss due to one of m's blocks solved after $\tau(B_{n(\tau)+1})$ becoming orphaned.

m's expected gain if he deviated and chained $B_{n(\tau)+1}$ to $p(B_n)$ is[30]

$$\left[N_m^o(\tau(p(B_n)) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})) + \Pr(m = m(B_{n(\tau)+2}))\right] G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

Indeed, m earns $G(M)$ on all blocks solved on the original chain up to $\tau(p(B_n)$, on B_n if he solved it and it remains on the active chain (if $B_{n(\tau)+2}$ is attached to B_n), and on $B_{n(\tau)+1}$ if it is included in the active chain (if m solves $B_{n(\tau)+2}$).[31]

The second term is the continuation payoff for all blocks solved after $B_{n(\tau)+1}$ if they are not orphaned afterwards. The third term is the expected loss due to one of m's blocks solved after $\tau(B_{n(\tau)+1})$ becoming orphaned. Neither term depends on which block m chains $B_{n(\tau)+1}$ to.

Since $N_m^o(\tau(B_n)) \geq N_m^o(\tau(p(B_n)) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+1}))$, if all miners observe that m solved B_{k+1}, m's expected payoff is larger if he followed the equilibrium strategy than if he deviated.

ii) If one miner (m') did not observe that m solved $B_{n(\tau)+1}$, m's expected gain if he followed the equilibrium strategy is

$$[N_m^o(\tau(B_n)) + 1 - \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3}))]G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})].$$

The first term is m's expected reward for solving blocks up to $B_{n(\tau)+1}$, reflecting the risk that $B_{n(\tau)+1}$ become orphaned if m' solves $B_{n(\tau)+2}$ and $B_{n(\tau)+3}$. The second term is m's continuation payoff, reflecting that m will be mining on the single active chain (be it the original one or a fork that becomes the consensus).

[30] Clearly, this is the only relevant deviation since m cannot obtain more if he chained $B_{n(\tau)+1}$ to $B_{n(\tau)}$ if $B_{n(\tau)}$ started a fork: $B_{n(\tau)+1}$ will never be on the active chain given the equilibrium strategies, even if m solves $B_{n(\tau)+2}$. A fortiori, m cannot obtain more if he decides to chain $B_{n(\tau)+1}$ to any block B_i with $i < n(\tau)$ outside the original chain.

[31] A fork can happen if one miner does not observe $B_{n(\tau)+1}$, but even in that case, $B_{n(\tau)}$, as well as all previously solved blocks, will be on the active chain and yield $G(M)$ or $G(M+1) + G(1) = G(M)$ depending on when z_m occurs.

If m deviated by chaining $B_{n(\tau)+1}$ to $p(B_n)$,[32] to earn his reward on $B_{n(\tau)+1}$, m needs to solve $B_{n(\tau)+2}$ so his expected gain is

$$\left[N_m^o(\tau(p(B_n)) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})) + \Pr(m = m(B_{n(\tau)+2}))\right] G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})].$$

As above

$$N_m^o(\tau(B_n)) \geq N_m^o(\tau(p(B_n))) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})).$$

Consequently, there is no profitable deviation if

$$1 - \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})) \geq \Pr(m = m(B_{n(\tau)+2})).$$

That is

$$1 \geq \Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})),$$

which holds because

$$1 \geq \Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2})) \geq$$
$$\Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})).$$

This completes the first part of the proof of the optimality of the strategy stated in b).

2) Suppose there is a fork starting with two blocks consecutively solved by the same miner and longer than the original chain. The equilibrium strategy then prescribes that miners chain their block to the longest chain.

If one miner observed a block with delay, we are in the same situation as in Proposition 1, and there is no profitable deviation from mining the longest chain. Off the equilibrium path, however, that fork could have occurred for other reasons, and a new fork could still occur because of a delay in the future. In that case there is no profitable deviation (in particular, trying to create a fork by solving two blocks in a row is dominated by the equilibrium strategy), as shown in the first part of b).

QED

[32] As above, this is the only relevant deviation.

Proof of Proposition 5

Our candidate equilibrium strategy specifies the following:

a) If a miner solved a block outside the original chain thereby creating a one-block-long fork as long as the original chain, all miners chain their next block to the fork, which miners consider to be the original chain from that point on.

b) Otherwise, each miner chains his current block to the last block solved on the original chain.

Proof of part a)

Let B_n be the last block solved on the original chain, suppose B_{k+1}, with $k \geq n$, is chained to $p(B_n)$. As above, the relevant choice for m at time τ is between chaining his next block to B_{k+1} (the equilibrium strategy) and chaining it to B_n (the only relevant deviation). As in the proof of Proposition 1, a one-shot deviation affects m's payoff only if the next stopping time corresponds to two possible events: either m is hit by a liquidity shock or m solves a block.

i) Supppose the next event is z_m. If m deviated and chained his block to B_n his payoff is

$$\mathbb{1}_{\{m=m(B_{k+1})\}}G(M-1) + \mathbb{1}_{\{m=m(B_n)\}}G(1) + N_m^o(\tau(p(B_n)))G(M),$$

where $G(M-1)$ is his reward if he solved block B_{k+1}, and $G(1)$ his reward if he solved B_n. If, instead, m followed the equilibrium strategy and chained his block to B_{k+1} his payoff is

$$\mathbb{1}_{\{m=m(B_{k+1})\}}G(M) + \mathbb{1}_{\{m=m(B_n)\}}G(0) + N_m^o(\tau(p(B_n)))G(M).$$

Since by assumption $G(M-1) \leq G(M)$ and $G(1) = G(0)$, the deviation is not strictly profitable.

ii) Suppose the next event is that m solves block $B_{n(\tau)+1}$.

If m chained his block to B_n, $B_{n(\tau)+1}$ becomes orphaned since all miners, including m mine the fork after $\tau(B_{n(\tau)+1})$. m's expected gain is

$$\begin{aligned}&N_m^o(\tau(p(B_n)))G(M) + \mathbb{1}_{\{m=m(B_{k+1})\}}G(M) + \mathbb{1}_{\{m=m(B_n)\}}G(0) + G(0)\\ &+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})),\end{aligned}$$

since blocks B_n and $B_{n(\tau)+1}$ are orphaned and earn $G(0)$. As before, $\mathcal{L}(\tau(B_{n(\tau)+1}))$ is the expected loss due to one of the blocks solved by m after $\tau(B_{n(\tau)+1})$ becoming orphaned.

If instead m had chained his block to B_{k+1}, m's expected gain is

$$N_m^o(\tau(p(B_n)))G(M) + \mathbb{1}_{\{m=m(B_{k+1})\}}G(M) + \mathbb{1}_{\{m=m(B_n)\}}G(0) + G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})),$$

since now m earns $G(M)$ for solving B_{k+2}.

It follows that the deviation is strictly dominated.

Proof of part b)

As before, B_n is the last block solved on the original chain. Assume there is no one-block-long fork of the same length as the original chain at time τ. The only relevant deviation for miner m is to try and start a fork by chaining his current block to $p(B_n)$. If z_m occurs, or if another miner solves the next block, m's payoff is not affected by which block he currently mines.

Consider the case where the next event is that m solves block $B_{n(\tau)+1}$. If m deviated and chained $B_{n(\tau)+1}$ to $p(B_n)$, his payoff is:

$$N_m^o(\tau(p(B_n)))G(M) + G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+1})] - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

Indeed, all miners chain their future blocks to the chain that contains $B_{n(\tau)+1}$, therefore m earns $G(M)$ for $B_{n(\tau)+1}$. If m played the equilibrium strategy and chained $B_{n(\tau)+1}$ to B_n, he obtains the same payoff, since he earns $G(M)$ for $B_{n(\tau)+1}$ as well. Therefore there is no profitable deviation.

QED

Proof of Proposition 6

Our candidate equilibrium strategy specifies the following

a) If a miner has the opportunity to double spend, he mines a block chained to the parent of the last block solved on the original chain.

b) If a miner solves a block that creates a one-block-long fork as long as the original chain, that miner chains his next block to the block he just solved, except if he spots an opportunity to double-spend, in which case he plays according to a).

c) Otherwise, each miner chains his current block to the last block solved on the original chain, except if there is a fork starting with two blocks consecutively solved by the same miner, longer than the original chain. In that case, each miner chains his block to the longest chain, which miners consider to be the original chain from that point on.

The general structure of the proof is similar to that of Proposition 4. As in this previous proof, we assume that $G(M-1)+G(1) = G(M)$ to simplify the exposition. We also clarify that the miner who earns the reward S is the one who completes a double-spending fork before being hit by his liquidity shock z_m. In particular, a miner who initiates a double-spending fork but is hit by a liquidity shock before the fork is resolved does not earn S. By contrast, a miner who successfully completes a double-spending fork initiated by the miner he replaced does earn S.

Proof of part a)

Let B_n be the last block solved on the original chain. Consider the strategy of miner m who spots the opportunity to double spend at time τ.

Following the same reasoning as above, the relevant choice for m is between chaining his next block to $p(B_n)$ (the equilibrium strategy), chaining it to B_n, or chaining it to $B_{n(\tau)}$ if $B_{n(\tau)}$ is chained to $p(B_n)$ and $m = m(B_{n(\tau)})$. (This can happen off path if m started a fork from $p(B_n)$ and spots the double-spending opportunity right after. By assumption, S can only be earned if m creates a new fork from $p(B_n)$.) As in the proof for Proposition 1, we can restrict attention to the cases where the next event is that m either is hit by a liquidity shock, or solves a block.

Suppose first that the next event is z_m. If m deviated and chained his block to B_n his payoff is $N^o_m(\tau_{B_n})G(M)$. If, instead, m followed the equilibrium strategy and chained his block to $p(B_n)$ his payoff is

$$N^o_m(\tau(p(B_n)))[G(M-1) + G(1)] + \mathbb{1}_{\{m=m(B_n)\}}G(M-1).$$

Last, suppose $m = m(B_{n(\tau)})$ and $B_{n(\tau)}$ is chained to $p(B_n)$. If m deviated and chained his block to $B_{n(\tau)}$, his payoff is the same as when chaining his block to

$p(B_n)$, plus $G(1)$ for solving $B_{n(\tau)}$.[33] Since by assumption $G(M-1)+G(1) = G(M)$ and $G(1)=0$, the deviations are not strictly profitable.

Alternatively, suppose next event is that m solves $B_{n(\tau)+1}$.

To analyse this case, we condition m's payoffs on the event that $m = m(B_{n(\tau)+2})$, that is m solves $B_{n(\tau)+2}$ before being hit by a liquidity shock. This event's probability is independent from m's strategy, and if m follows the equilibrium strategy, then m earns S if and only if this event is true.

i) Suppose m solves $B_{n(\tau)+2}$.

If m deviated and chained his block to B_n, his payoff is

$$N_m^o(\tau(B_n))G(M)+2G(M)+\mathbb{E}[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})].$$

m earns $G(M)$ for all the blocks solved on the original chain up to $\tau(B_n)$. m earns $G(M)$ for solving $B_{n(\tau)+1}$ and $B_{n(\tau)+2}$ which belong to the original chain, and for all the future blocks solved after $\tau(B_{n(\tau)+2})$, since m knows that no other double spending opportunity will be spotted.

If $m = m(B_{n(\tau)})$ and $B_{n(\tau)}$ is chained to $p(B_n)$, if m deviated and chained $B_{n(\tau)+1}$ to $B_{n(\tau)}$, his payoff is

$$N_m^o(\tau_{p(B_n)})G(M)+3G(M)+E[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})].$$

m's fork has succeeded and he earns $G(M)$ on all blocks solved up to $p(B_n)$ on the original chain, plus on $B_{n(\tau)}$, $B_{n(\tau)+1}$ and $B_{n(\tau)+2}$.

If m played the equilibrium strategy and chained $B_{n(\tau)+1}$ to $p(B_n)$, his payoff is

$$N_m^o(\tau_{p(B_n)})G(M)+2G(M)+E[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})]+S.$$

m earns $G(M)$ for all the blocks he solved before the fork (up to $p(B_n)$), $G(M)$ for $B_{n(\tau)+1}$ and for $B_{n(\tau)+2}$, and for all the future blocks solved after $\tau_{B_{n(\tau)+2}}$, since on the equilibrium path all miners mine on the chain including $B_{n(\tau)+2}$. In addition, m earns S from double-spending.

[33] In that case a fork has started so the reward for solving B_n is $G(M-1)$ which is lower than $G(M)$. This makes the deviation even less profitable.

Hence, the net benefit of following the equilibrium strategy rather than deviating is $S - \max\{\mathbb{1}_{\{m=m(B_n)\}}; \mathbb{1}_{\{[m=m(B_{n(\tau)})]\bigcap[p(B_{n(\tau)})=p(B_n)]\}}\}G(M)$.

ii) Suppose that either z_m occurs before $\tau(B_{n(\tau)+2})$ or z_m occurs after $\tau(B_{n(\tau)+2})$ but m does not solve $B_{n(\tau)+2}$. To write m's payoff, we will distinguish the two events when needed.

If m deviated and chained $B_{n(\tau+1)}$ to B_n, his payoff is

$$\begin{aligned}&N^o_m(\tau(B_n))G(M) + G(M)\\&+ \Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})].\end{aligned} \tag{15}$$

m earns $G(M)$ for all the blocks solved up to $\tau(B_n)$, for $B_{n(\tau)+1}$ (since it is on the original chain), and for blocks solved after $\tau(B_{n(\tau)+2})$ if $z_m > \tau(B_{n(\tau)+2})$.

If $m = m(B_{n(\tau)})$ and $B_{n(\tau)}$ is chained to $p(B_n)$, if m deviated and chained $B_{n(\tau)+1}$ to $B_{n(\tau)}$, his payoff is

$$\begin{aligned}&N^o_m(\tau(p(B_n)))G(M) + 2G(M)\\&+ \Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})].\end{aligned} \tag{16}$$

m's fork has succeeded and he earns $G(M)$ on all blocks solved up to $p(B_n)$ on the original chain, plus on $B_{n(\tau)}$ and $B_{n(\tau)+1}$.

If m played the equilibrium strategy and chained $B_{n(\tau)+1}$ to $p(B_n)$, his payoff is

- if z_m occurs first,

$$N^o_m(\tau(p(B_n)))(G(M-1) + G(1)) + \mathbb{1}_{\{m=m(B_n)\}}G(M-1) + G(1).$$

  In that case, m has created a one-block-long fork as long as the original chain when he is hit by his liquidity shock. Therefore, he earns $G(M-1) + G(1)$ for all the blocks he solved on the original chain up to $\tau(p(B_n))$. He also earns $G(M-1)$ for B_n if he solved it and $G(1)$ for $B_{n(\tau)+1}$.

- if $B_{n(\tau)+2}$ is solved by another miner before z_m,

$$N^o_m(\tau_{B_n})G(M)+G(0)+\mathbb{E}[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt|z_m > \tau(B_{n(\tau)+2})].$$

  m's fork fails, therefore he earns $G(M)$ for all the blocks he solved on the original chain up to $\tau(B_n)$ and for the blocs solved after $\tau(B_{n(\tau)+2})$.

Since $G(M-1) = G(M)$,[34] gains earned by m on all blocks solved up to $\tau(B_n)$ are the same in the two events above. Hence m's payoff if he played the equilibrium strategy when he does not solve $B_{n(\tau)+2}$ is:

$$N^o_m(\tau_{B_n})G(M)+\Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt|z_m > \tau(B_{n(\tau)+2})]. \tag{17}$$

Hence, the net benefit of following the equilibrium strategy rather than deviating when m does not solve B_{k+2} is the difference between (17) and the max of (15) and (16), that is,

$$-\left\{G(M) + (\mathbb{1}_{\{[m=m(B_{n(\tau)})]\cap[p(B_n(\tau))=p(B_n)]\}} - \mathbb{1}_{\{m=m(B_n)\}})G(M)\right\}.$$

Overall, m always follows the equilibrium strategy (including when he solved B_n) iff

$$\begin{aligned}
&\Pr[m = m(B_{n(\tau)+2})|m = m(B_{n(\tau)+1})][S - G(M)] \\
&> (1 - \Pr[m = m(B_{n(\tau)+2})|m = m(B_{n(\tau)+1})])2G(M) \\
\Leftrightarrow \quad & S > \frac{G(M)(2 - \Pr[m = m(B_{n(\tau)+2})|m = m(B_{n(\tau)+1})])}{\Pr[m = m(B_{n(\tau)+2})|m = m(B_{n(\tau)+1})]}.
\end{aligned}$$

Remark that $\Pr(m = m(B_{n(\tau)+2})|m = m(B_{n(\tau)+1}))$ is just the probability that at any time m solves the next block.

Proof of part b)

[34] Again, allowing for $G(M-1) < G(M)$ only makes the condition under which the equilibrium exists more intricate.

As earlier, B_n is the last block solved on the original chain. Suppose that at time τ, miner m has just created a one-block-long fork as long as the original chain by solving $B_{n(\tau)}$ that is chained to B_n's parent, $p(B_n)$. The relevant choice for m at τ is between chaining his next block to $B_{n(\tau)}$ (the equilibrium strategy) and chaining it to B_n (the only relevant deviation). The reasoning is analogous to the proof of Proposition 4 part a), hence we only sketch it here.

i) Suppose that the next event is z_m.

If m deviated and chained his block to B_n, the original chain remains the only active chain and m's payoff is

$$G(0) + N_m^o(\tau(B_n))G(M),$$

where the first term is the reward for $B_{n(\tau)}$.

If, instead, m followed the equilibrium strategy and chained his block to B_{k+1} his payoff is

$$G(1) + N_m^o(\tau(p(B_n)))[G(M-1) + G(1)] + \mathbb{1}_{\{m=m(B_n)\}}G(M-1),$$

where the first term is the reward for $B_{n(\tau)}$. Since $G(M-1) + G(1) = G(M)$ and $G(1) = 0$, this deviation is not strictly profitable.

ii) Suppose the next event is that m solves $B_{n(\tau)+1}$.

If m deviated and chained his block to B_n, the original chain remains the only active chain and B_{k+1} becomes orphaned. Therefore, m's expected payoff is

$$N_m^o(\tau(B_n))G(M) + G(M) + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+1})]$$
$$+ \mathcal{S}(\tau(B_{n(\tau)+1})) - \mathcal{L}(\tau(B_{n(\tau)+1})),$$

where the second term is the reward for $B_{n(\tau)+1}$. As earlier, $\mathcal{L}(\tau(B_{n(\tau)+1}))$, is the expected loss due to one of m's blocks solved after $\tau(B_{n(\tau)+1})$ becoming orphaned. $\mathcal{S}(\tau(B_{n(\tau)+1}))$ is the expected benefit from m spotting a double-spending opportunity after $\tau(B_{n(\tau)+1})$. Note that both $\mathcal{L}(\tau(B_{n(\tau)+1}))$ and $\mathcal{S}(\tau(B_{n(\tau)+1}))$ are conditional on m's information at τ. For instance, if m already had a double-spending opportunity, then $\mathcal{L}(\tau(B_{n(\tau)+1})) = \mathcal{S}(\tau(B_{n(\tau)+1})) = 0$.

If instead m played the equilibrium strategy and chained $B_{n(\tau)+1}$ to $B_{n(\tau)}$, the chain including $B_{n(\tau)}$ and $B_{n(\tau)+1}$ becomes the longest one, and all miners hereafter chain their blocks to it. Thus m expected payoff is at least equal to

$$N^o_m(\tau_{p(B_n)})G(M) + 2G(M) + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+2})]$$
$$+ \mathcal{S}(\tau(B_{n(\tau)+2})) - \mathcal{L}(\tau(B_{n(\tau)+2})),$$

where the second term is the reward for $B_{n(\tau)+1}$ and $B_{n(\tau)+2}$. This payoff is higher by S if m has the double-spending opportunity (the only case on the equilibrium path).

Since $\mathbb{1}_{\{m=m(B_n)\}} \leq 1$, m prefers to follow the equilibrium strategy.

Proof of part c)

The reasoning is analogous to the proof of Proposition 4 part b), we only sketch it here. B_n is the last block on the original chain

1. First consider the case in which there is no fork of two consecutive blocks solved by the same miner and longer than the original chain. For any miner m who does not have the double-spending opportunity, the only two relevant choices are to chain his block to B_n (the equilibrium strategy) and to create a fork and try solving two blocks in a row (the only relevant deviation). As in the proof for Proposition 1, we can restrict attention to the cases where the next event is that m either is hit by a liquidity shock, or solves a block.

   - Suppose the next event is z_m. If m followed the equilibrium strategy, his payoff is $N^o_m(z_m)G(M)$ (if there is no fork), or $N^o_m(z_m)(G(M-1) + G(1)) = N^o_m(z_m)G(M)$ (if a fork has started). If m deviated, his payoff is at most equal to $N^o_m(z_m)G(M)$.
   - Suppose the next event is that m solves block $B_{n(\tau)+1}$.
     If m followed the equilibrium strategy and chained $B_{n(\tau)+1}$ to B_n, his expected payoff is

$$(N^o_m(\tau_{B_n}) + 1)G(M) + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})]$$
$$+ \mathcal{S}(\tau(B_{n(\tau)+1})) - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

If m deviated and chained $B_{n(\tau)+1}$ to $p(B_n)$, his expected payoff is

$$\left[N_m^o(\tau(p(B_n)) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})) + \Pr(m = m(B_{n(\tau)+2}))\right] G(M)$$
$$+ \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})] + \mathcal{S}(\tau(B_{n(\tau)+1})) - \mathcal{L}(\tau(B_{n(\tau)+1})).$$

Since

$$N_m^o(\tau_{B_n}) \geq N_m^o(\tau_{p(B_n)}) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})),$$

m's expected payoff is larger if he followed the equilibrium strategy than if he deviated.

2. Consider the case in which there is a fork starting with two blocks consecutively solved by the same miner and longer than the original chain. If that fork occurred because one miner exploited a double-spending opportunity, we are in the same situation as in Proposition 1, and there is no profitable deviation from mining the longest chain.

   If that fork occurred for other reasons (off the equilibrium path), a new fork could still occur because of a double-spending opportunity in the future. In that case there is no profitable deviation (in particular, trying to create a fork by solving two blocks in a row is dominated by the equilibrium strategy), as shown in the first part of c).

QED

Proof of Proposition 7

Let τ^f be the time at which the n^{th} block is solved on the original chain. Hence, $B_{n(\tau^f)}$ is the n^{th} block on the original chain. We say that a chain "conforms" to technology C if every block on that chain solved after τ^f was mined with technology C. We call "C-chain" the chain that contains $B_{n(\tau^f)}$, conforms to C, and preexists all other chains containing $B_{n(\tau^f)}$ and conforming to C. $N_m^C(\tau)$ is the number of blocks solved by m up to τ on the C-chain.

Our candidate equilibrium strategy specifies the following:

a) For every $\tau < \tau^f$, miners chain their block to the last block on the original chain,

b) For every $\tau \geq \tau^f$, miners choose $C = 0$ and chain their block to the last block on the chain that contains $B_{n(\tau^f)}$, conforms to $C = 0$, and preexists all other chains containing $B_{n(\tau^f)}$ and conforming to $C = 0$. If such a chain does not exist, miners choose $C = 0$ and chain their block to $B_{n(\tau^f)}$.

We consider each case in turn.

a) Suppose $\tau < \tau^f$. Then using the same reasoning as in Proposition 1, a deviation is not profitable.

b) Suppose $\tau \geq \tau^f$. As in the proof of Proposition 1, it is sufficient to compare payoffs when m solves the next block, $B_{n(\tau)+1}$.

If miner m played the equilibrium strategy, that is, chose $C = 0$ and chained his block to the last block on the 0-chain, or to $B_{n(\tau^f)}$, his payoff is

$$N_m^0(\tau)(1 + b_m(0))G(M) + (1 + b_m(0))G(M) \\ + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)(1 + b_m(0))G(M)dt | z_m \geq \tau(B_{n(\tau)+1})]$$

The first term is m's rewards from blocks he solved on the 0-chain up to τ. The second term is the reward from solving $B_{n(\tau)+1}$, and the last term is the expected value of solving future blocks on the 0-chain.

If instead, miner m deviated and chained his block to another block than the last one on the 0-chain, using any technology C, his payoff is

$$N_m^0(\tau)(1 + b_m(0))G(M) + (1 + b_m(C))G(1) \\ + \mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)(1 + b_m(0))G(M)dt | z_m \geq \tau(B_{n(\tau)+1})]$$

Hence, the only difference between m's payoff if he deviates and his equilibrium payoff is the reward from solving block $B_{n(\tau)+1}$. This reward is $(1 + b_m(C))G(1) = 0$ if he deviates and $(1 + b_m(0))G(M) > 0$ if he plays the equilibrium strategy. It follows that a deviation is not profitable.

A symmetric argument sustains the equilibrium in which all miners choose $C = 1$.

QED

Proof of Proposition 8

We use the same notation as in the proof of Proposition 7 for the C-chain.

To define our equilibrium strategies, we need to introduce the following condition, which we will derive explicitly in the proof:

Condition 3 *For $\tau \geq \tau^f$, ω_τ is such that for $m > K$*

$$\begin{aligned} \Pr(N_m(\tau') - N_m(\tau) = 1)(G(K) - (1+b)G(M-K)) \leq \\ \Pr(z_m = \tau') \left\{ \begin{aligned} (N_m^0(\tau) - N_m^0(\tau^f))(1+b)(G(M-K) - G(M-K-1)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K+1) - G(K))) \end{aligned} \right\}, \end{aligned} \tag{18}$$

and for $m \leq K$

$$\begin{aligned} \Pr(N_m(\tau') - N_m(\tau) = 1)(G(K) - G(M-K)) \geq \\ \Pr(z_m = \tau') \left\{ \begin{aligned} (N_m^0(\tau) - N_m^0(\tau^f))(G(M-K+1) - G(M-K)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K) - G(K-1))) \end{aligned} \right\}. \end{aligned} \tag{19}$$

Our candidate equilibrium strategy specifies the following

a) *Before the hardfork:* If $\tau < \tau^f$, miners chain their block to the last block on the original chain.

b) *At the hardfork or after:* If $\tau = \tau^f$, or if $\tau > \tau^f$, and Condition 3 holds, miners $m \leq K$ choose $C = 1$ and chain their block to $B_{n(\tau^f)}$ if the 1-chain does not exist, and chain their block to the last block solved on the 1-chain otherwise, while miners $m > K$ choose $C = 0$ and chain their block to $B_{n(\tau^f)}$ if the 0-chain does not exist, and chain their block to the last block on the 0-chain otherwise.

c) *After the hardfork off-path:* Suppose $\tau > \tau^f$ and Condition 3 does not hold. Let $\Delta\omega \equiv \omega^\tau \setminus \omega^{\tau^f}$ (i.e., $\Delta\omega$ contains the history of the game between τ^f and τ). Then for every $\tau' \geq \tau$, all miners play the strategy prescribed after history $\omega^{\tau'} \setminus \Delta\omega$ that is defined in b). In playing strategies defined in b), miners consider that the 0-chain and the 1-chain are defined with respect to history $\omega^{\tau'} \setminus \Delta\omega$.

We need to prove that a miner does not have a profitable one shot deviation from σ^*. We hereafter consider each of the cases above in turn.

Proof of part a): Before the fork

Following the reasoning of the proof of Proposition 1, there is no profitable deviation. In particular, it is not profitable for any miner $m \leq K$ to choose $C = 1$ before τ^f since any block solved with $C = 1$ will, by definition, not belong to the 1-chain and yield a reward of 0. Also, note that unlike in the proof of Proposition 3, miners' actions before τ^f cannot after the condition under which the hardfork occurs.

Proof of part b): at or after the fork

i) Consider first a deviation by a miner $m > K$.

   Any deviation other than chaining to the last block on the 1-chain is ruled out by similar arguments as in Proposition 1. Hence check that m prefers to mine blocks on the 0-chain, rather than on the 1-chain. As earlier, this one-shot deviation affects m's payoff only if the next stopping time τ', corresponds to two possible events: either m solves his block, or z_m occurs.

   - Suppose miner m solves a block at τ', i.e., $N_m(\tau') - N_m(\tau) = 1$. If Condition 3 is still true at τ', since every miner, including m, reverts to the equilibrium strategy from τ' on, the only impact of the deviation is that m earns $G(K)$ for block $B_{n(\tau')}$ instead of $(1 + b)G(M - K)$ under the equilibrium strategy. If Condition 3 is not true at τ', given c), the impact of the deviation is that m earns 0 for block $B_{n(\tau')}$ instead of $(1+b)G(M-K)$ under the equilibrium strategy and loses all rewards for blocks solved between τ^f and τ'.
   - Suppose miner m is hit by a liquidity shock at τ', i.e., $z_m = \tau'$. Then his payoff under the deviation is

     $$(N^0_m(\tau) - N^0_m(\tau^f))(1 + b)G(M - K - 1) + (N^1_m(\tau) - N^1_m(\tau^f))G(K + 1) + N^0_m(\tau^f)(1 + b)G(M)$$

     instead of

     $$(N^0_m(\tau) - N^0_m(\tau^f))(1 + b)G(M - K) + (N^1_m(\tau) - N^1_m(\tau^f))G(K) + N^0_m(\tau^f)(1 + b)G(M)$$

under the equilibrium strategy.[35] It follows that there is no profitable deviation if

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K) - (1+b)G(M-K)) \leq$$
$$\Pr(z_m = \tau') \begin{Bmatrix} (N^0_m(\tau) - N^0_m(\tau^f))(1+b)(G(M-K) - G(M-K-1)) \\ -(N^1_m(\tau) - N^1_m(\tau^f))(G(K+1) - G(K))) \end{Bmatrix},$$

which is exactly inequality (18) in Condition 3.

ii) Consider next a deviation by a miner $m \leq K$. A symmetric reasoning yields that there is no profitable deviation if

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K) - G(M-K)) \geq$$
$$\Pr(z_m = \tau') \begin{Bmatrix} (N^0_m(\tau) - N^0_m(\tau^f))(G(M-K+1) - G(M-K)) \\ -(N^1_m(\tau) - N^1_m(\tau^f))(G(K) - G(K-1))) \end{Bmatrix},$$

which is exactly (19) in Condition 3.

Next, see that at $\tau = \tau^f$, $N^C_m(\tau) = N^C_m(\tau^f)$ for all miners. Inequality (18) is then written:

$$(1+b)G(M-K) \geq G(K) \Leftrightarrow b \geq \frac{G(K)}{G(M-K)} - 1.$$

Similarly, inequality (19) is then written:

$$G(K) \geq G(M-K) \Leftrightarrow K \geq \frac{M}{2}.$$

Furthermore, if miners adhere to the equilibrium strategy, then miners $m \leq K$ always mine the 1-chain so that if $K \geq \frac{M}{2}$, inequality (19) in Condition 3 is true at any $\tau \geq \tau^f$. Given that miners $m > K$ stick to the 0-chain, if $b \geq \frac{G(K)}{G(M-K)} - 1$, inequality (18) is always verified after τ^f. Hence, for $\tau \geq \tau^f$, Condition 3 holds on the equilibrium path.

Proof of part c): After the fork off-path

Suppose ω_τ is as described in c). Then given that all other players play the equilibrium, m's payoff from adhering to the equilibrium strategy is as in b) above. Following the same logic as in the proof of b), other deviations can be ruled out.

QED

[35] Note that we used the assumption that $\forall K$, $G(M) = G(M-K) + G(K)$ to write down miner m's payoff from blocks solved before τ^f.

Proof of Proposition 9

See first that miner m's participation constraint is

$$\sum_{i\in\mathcal{M}} h_i \leq \frac{G(M)}{c_m X}. \tag{20}$$

Evaluated at equilibrium, (10) yields

$$\left(\sum_{i\in\mathcal{M}} h_i^*\right) - h_m^* = \frac{X}{G(M)} c_m \left(\sum_{i\in\mathcal{M}} h_i^*\right)^2$$

$$\left(\sum_{i\in\mathcal{M}} h_i^*\right) - c_m \frac{X}{G(M)} \left(\sum_{i\in\mathcal{M}} h_i^*\right)^2 = h_m^*. \tag{21}$$

Summing over miners

$$M\left(\sum_{i\in\mathcal{M}} h_i^*\right) - \left(\sum_{i\in\mathcal{M}} c_i\right) \frac{X}{G(M)} \left(\sum_{i\in\mathcal{M}} h_i^*\right)^2 = \left(\sum_{i\in\mathcal{M}} h_{i\in}^*\right)$$

$$(M-1)\left(\sum_{i\in\mathcal{M}} h_i^*\right) = \left(\sum_{i\in\mathcal{M}} c_i\right) \frac{X}{G(M)} \left(\sum_{i\in\mathcal{M}} h_i^*\right)^2$$

$$\frac{(M-1)}{\sum_{i\in\mathcal{M}} c_i} \frac{G(M)}{X} = \sum_{i\in\mathcal{M}} h_m^*$$

$$\sum_{i\in\mathcal{M}} h_i^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i},$$

which is exactly (12) in Proposition 9. Into participation constraint (20), this yields $(M-1)c_m \leq \sum_{i\in\mathcal{M}} c_i$. Next, substituting (12) into (21)

$$h_m^* = \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right) - c_m \frac{X}{G(M)} \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right)^2.$$

$$h_m^* = \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right) \left(1 - c_m \frac{X}{G(M)} \frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right)$$

$$h_m^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i\in\mathcal{M}} c_i} \left(1 - c_m \frac{M-1}{\sum_{i\in\mathcal{M}} c_i}\right),$$

which is exactly (11) in Proposition 9. Last, replacing h_m^* into miner m's objective function, one obtains that his equilibrium profit is

$$\frac{G(M)}{\lambda_m X}\left(1-\frac{(M-1)c_m}{\sum_{i\in\mathcal{M}} c_i}\right)^2.$$

References

Biais, B., T. Foucault and S. Moinas, 2016, "Equilibrium Fast-Trading," *Journal of Financial Economics*, 116(2), 292–313.

Bonneau, J., E. W. Felten, S. Goldfeder, J. A. Kroll, and A. Narayanan, 2016, "Why buy when you can rent? Bribery attacks on Bitcoin consensus," Princeton University working paper.

Cass, D. and K. Shell, 1983, "Do sunspots matter?," *Journal of Political Economy*, 91(2), 193–227.

Cole, H. L. and T. Kehoe, 2000, "Self-Fulfilling Debt Crises", *The Review of Economic Studies*, 67(1), 91-116.

Dimitri, N, 2017, "Bitcoin Mining as a Contest", *Ledger*, 2, doi:https://doi.org/10.5195/ledger.2017.96.

Duggan, J., 2012, "Noisy Stochastic Games", *Econometrica*, 80(5), 2017–2045.

Evans, D.S., 2014, "Economic aspects of Bitcoin and other decentralized public-ledger currency platforms," Coase-Sandor Working Paper Series in Law and Economics, University of Chicago Law School.

Eyal, I., and E. G. Sirer, 2016, "Majority is not enough: Bitcoin mining is vulnerable," in International Conference on Financial Cryptography and Data Security, Springer, 436–454.

Glode, V., R. Green, and R. Lowery, 2012, "Financial Expertise as an Arms Race," *Journal of Finance*, 67, 1723–1759.

Harvey, C. R., 2016, "Cryptofinance," working paper.

Kroll, J. A., I. C Davey, and E. W. Felten, 2013, "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries," in Proceedings of WEIS, 2013.

Lerner, J, and J. Tirole, 2002, "Some simple economics of open source," *The Journal of Industrial Economics*, 2, 197-234.

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

Narayanan, A., july 28, 2015, "Analyzing the 2013 Bitcoin fork: centralized decision-making saved the day," *http://freedom-to-tinker.com/2015/07/28/analyzing-the-2013-bitcoin-fork-centralized-decision-making-saved-the-day.*

Pease, M., R. Shostak, and L. Lamport, 1980, "Reaching Agreement in the Presence of Faults," *J. ACM*, 27(2), 228-234.

Schrijvers, O., J. Bonneau, D. Boneh, and T. Roughgarden, 2016, "Incentive compatibility of Bitcoin mining pool reward functions," *Financial Cryptography and Data Security*.

Teutsch, J., S. Jain, and P. Saxena, 2016, "When cryptocurrencies mine their own business," *Financial Cryptography and Data Security (FC 2016).*

Yermack, D., 2017, "Corporate governance and blockchains," *Review of Finance*, forthcoming.