The Blockchain Folk Theorem

Bruno Biais
Toulouse School of Economics, CNRS, Université Toulouse Capitole, TSM-R and HEC Paris

Christophe Bisière
Toulouse School of Economics, Université Toulouse Capitole, TSM-R

Matthieu Bouvard
Desautels Faculty of Management, McGill University

Catherine Casamatta
Toulouse School of Economics, Université Toulouse Capitole, TSM-R

Blockchains are distributed ledgers, operated within peer-to-peer networks. We model the proof-of-work blockchain protocol as a stochastic game and analyze the equilibrium strategies of rational, strategic miners. Mining the longest chain is a Markov perfect equilibrium, without forking, in line with Nakamoto (2008). The blockchain protocol, however, is a coordination game, with multiple equilibria. There exist equilibria with forks, leading to orphaned blocks and persistent divergence between chains. We also show how forks can be generated by information delays and software upgrades. Last we identify negative externalities implying that equilibrium investment in computing capacity is excessive. (*JEL* C73, G20, L86)

Received May 31, 2017; editorial decision July 6, 2018 by Editor Itay Goldstein.

Blockchains are decentralized protocols for recording transactions and asset ownership. In contrast with centralized protocols in which one authority is in charge of maintaining a unique common ledger, a blockchain operates within a network, whose participants each possess and update their own version of

Many thanks for helpful comments to our editor I. Goldstein, two anonymous referees, C. Michel, V. Glode, B. Gobillard, C. Harvey, J. Hörner, A. Kirilenko, G. Laroque, T. Mariotti, J. Prat, J. Tirole, S. Villeneuve, participants in the TSE Blockchain working group, the Inquire Conference in Liverpool, the GSE Summer Forum in Barcelona, the Africa Meeting of the Econometric Society in Algiers, the RFS Fintech Workshop and Conference, the Oxford Financial Intermediation Theory Conference, the Swissquote Conference, the Digital Economics Conference in Toulouse, the BoF/CEPR Conference on Money in the Digital Age in Helsinki, the Swiss Finance Institute Research Days in Gerzensee, and seminars at University of Geneva, Sciences Po Paris, IAST Toulouse, INRIA Paris, TSE, University of Tokyo, University of Amsterdam and University of Toronto. 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. Send correspondence to Bruno Biais, HEC Paris, 1 Rue de la Libération, 78350 Jouy-en-Josas, France; telephone: +33 1 39 67 98 06. E-mail: biaisb@hec.fr.

doi:10.1093/rfs/hhy095

the ledger, which is therefore distributed. 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 have been conducted for supply chains, trade finance, and financial transactions, for example by Nasdaq, the Australian Stock Exchange, BHP Billiton, Banque de France, and Ripple.

Our focus is on public blockchains, such as Bitcoin or Ethereum, which are fully decentralized and in which participants are anonymous. Public blockchains stand in contrast to private blockchains, in which a central authority can authorize participants and certify transactions. The main issue we address is whether public blockchains can be expected to generate stable consensus: Is such consensus a necessary byproduct of the blockchain protocol? Or could the blockchain protocol lead to the emergence of different, competing, versions of the ledger?

Each block, in the blockchain, offers an updated version of the ledger, taking into account recent transactions, and chained to a previous version of the ledger—that is, a previous block. In an ideal blockchain, there is a single sequence of blocks, registering the dynamics of the ledger, on which all participants agree. In contrast to this situation, there could be forks in the blockchain. In that case there are competing branches, each registering a potentially different version of the ledger. Such forks could make the ledger less stable, reliable, and useful, as they could create uncertainty about the distribution of property rights. In practice, as discussed in the next section, there have been several forks, some of which have persisted until now. We endeavor to understand the economic forces at the root of forks, and why the blockchain protocol does not seem to always be successful at avoiding them.

To study the dynamics of the blockchain, we analyze the behavior of its key participants: the miners. It is the miners who decide to which previous block a new block is chained and thus give rise to a single chain or trigger forks. We take a game-theoretic approach to analyze the strategies of the miners and the resulting equilibrium blockchain dynamics.

The rules of the game played by the miners in the major public blockchains (such as, e.g., Bitcoin or Ethereum) are set by the protocol known as "proof-of-work," which can be sketched as follows (and is described in more details in the next section): At each point in time, miners try to validate a new block of transactions. This implies solving a purely numerical problem unrelated to the block's content, an activity referred to as "mining." A miner who solves his problem obtains a proof-of-work, which he attaches to his block before disseminating it through the network. The instantaneous probability that a miner solves his proof-of-work problem is increasing in his computing power. In this

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

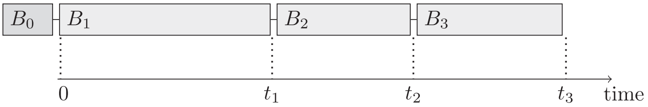

Figure 1
The blockchain
At $t = 0$, there is a genesis block B_0 and a stock of transactions included in a block B_1, chained to B_0. Miners work on a computational problem until a miner solves B_1, at t_1. B_1 is broadcast to all. Nodes check proof-of-work and transactions validity, and express acceptance by chaining the next block to B_1.

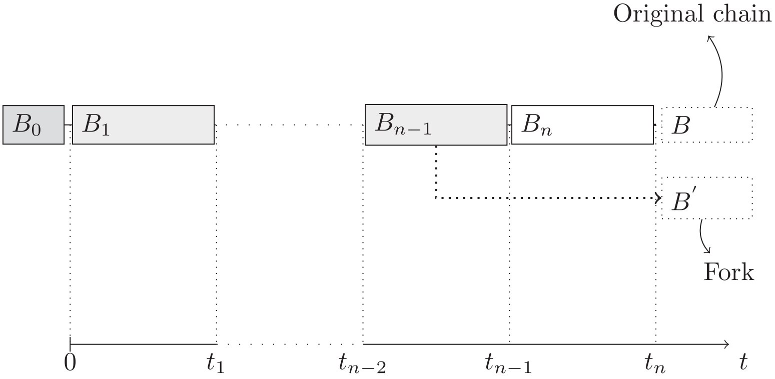

Figure 2
A fork

context, at each point in time, the identity of the miner who proposes the next block is the outcome of a random draw. 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 disseminated through the network, it becomes part of the consensus if miners chain their next block to it.

As argued by Nakamoto (2008), proof-of-work generates a stable consensus, or in other words, a single chain, if miners always take the last solved block as the parent for their next block. This ideal blockchain is illustrated in Figure 1.

Miners, however, may choose to discard certain blocks. Suppose, for example, that the last block solved is B_n, but miner m chains his next block to the parent of B_n, that is, 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 could be called in question.

A fork can also occur when some miners adopt a new version of the mining software that is incompatible with the current version. If miners fail to coordinate on the same software, this triggers a fork.

Does the blockchain protocol rule out the occurrence of forks? In the frictionless case in which information is instantaneously disseminated in the network, and there is no attempt to double-spend (such attempts are described in the next section), it is commonly assumed that a single blockchain will prevail. To examine the validity of that "folk theorem," we design a model that captures the key features of the proof-of-work blockchain protocol: There are M risk-neutral, rational, and strategic miners. Each time a miner solves a block, he obtains a reward in the cryptocurrency associated with the branch to which his block belongs. Miners choose to which previous block to chain their current block. They do so, observing all previously solved blocks, to maximize the expectation of their cumulated rewards at an exogenous liquidation time. We solve for Markov perfect equilibria of this stochastic game.

Our analysis of this game uncovers two important economic forces at play in the blockchain.

- First, miners' actions are strategic complements: Recall their rewards are paid in the cryptocurrency associated with the chain on which they are solving blocks. We assume the value of that cryptocurrency depends on the credibility of the corresponding chain, which is higher if more miners are active on it. 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 (Proposition 1). The same coordination motives, however, sustain equilibria with forks. If at some time (e.g., when a sunspot is observed) a miner anticipates all others to fork and mine a new branch, his best response is to follow them. Indeed, he rationally anticipates that any block solved out of the equilibrium path will not be accepted by other miners and the corresponding reward will be worthless (Proposition 2).
- Second, we identify a countervailing force that we refer to as "vested interest": An important feature of the blockchain is that miners cannot immediately spend or convert the cryptocurrency earned for solving blocks. Consequently, a miner who has accumulated rewards by solving several 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 taking into account the impact of their actions on the blockchain. It is likely, in practice, that large pools of miners, such as AntPool,

BTC.com, or Slush Pool, who each represent more than 10% of the computing power, indeed behave strategically.

Next, we enrich the model and make it more realistic by incorporating further real world characteristics of blockchains, such as information delays, and instances in which miners have to choose between incompatible upgrades of the mining software. We show that like the sunspots in our basic model, these characteristics can trigger forks on the equilibrium path. And we show that the same fundamental interplay of coordination motives and vested interests as in the basic case underlies equilibria with forks in these more realistic extensions of the model.

Finally, we endogenize the computing capacity that each miner installs. Because the difficulty of the mining process is 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 overinvesting (not unlike in Glode, Green, and Lowery 2012 and Biais, Foucault and Moinas 2016). This analysis points to another source of inefficiency in the blockchain design.

Early academic contributions on distributed consensus are in computer science. Since seminal work by Pease, Shostak, and Lamport (1980) and Lamport, Shostak, and Pease (1982), finding protocols that allow network participants to reach an agreement has been a major issue in computer science. In a Byzantine agreement (BA) setting, participants seek to agree on a common output aggregating private inputs, in the presence of "malicious" participants who try to attack, that is, disrupt the agreement. Nakamoto (2008) proposes the proof-of-work blockchain protocol to achieve consensus with high probability, in spite of potential attacks by malicious miners seeking to create a branch faster than the "honest" ones. Miller and LaViola (2014), Pass, Seeman, and Shelat (2017), and Garay, Kiayias, and Leonardos (2015) consider a larger set of attacks.[2] Their results suggest that the protocol is robust as long as honest miners represent the majority of computing power.[3]

While these papers consider ad hoc strategies from exogenously honest or malicious participants, we study optimal strategies from rational and profit-maximizing players in a game. The computer science papers to which our analysis is the closest, by Kroll, Davey, and Felten (2013) and Carlsten et al. (2016), analyze interactions between miners as a game. Kroll, Davey, and Felten (2013) offer interesting economic intuition, but do not develop a formal game-theoretic analysis. Carlsten et al. (2016) focus on specific strategies (regarding the choice of which block to chain to, and which transactions to include in a block) and show that they constitute an equilibrium. Relative to the computer

[2] Eyal and Sirer (2014) analyze a specific strategy called selfish mining, by which some miners maintain a "secret" branch of blocks and then release it to the network.

[3] See Bonneau et al. (2015) for a survey of the literature analyzing Bitcoin.

science literature, our contribution is twofold: To the best of our knowledge, our paper offers the first formal game-theoretic analysis of equilibria in the proof-of-work blockchain protocol. It is also the first to point to the importance of coordination effects in that protocol and show that they lead to multiple equilibria involving forks.

Our paper also relates to an emerging literature in economics and finance on blockchains. Harvey (2016) discusses the pros and cons of blockchains, while Raskin and Yermack (2016) and Yermack (2017) discuss their implications for central banking and corporate governance, respectively. Cong and He (2019) model the effect of blockchains on product market competition: blockchains improve contractibility and favor entry, but increase access to transaction data, which helps sustain collusive equilibria.[4] While these papers focus on the consequences of blockchains deployment, we provide a formal analysis of the blockchain consensus-building mechanism itself under proof-of-work. Chiu and Koeppl (2019) also analyze the proof-of-work protocol, and focus on investors' incentives to double-spend, while we focus on miners' incentives to maintain consensus. The analysis of proof-of-stake protocols in Saleh (2017) is complementary to our analysis of proof-of-work. Several papers (e.g., Evans 2014) note that a 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 analyze how these variations affect incentives.

1. A Primer on Blockchains

In this section we first describe the blockchain protocol and then discuss forks that occurred in blockchains.

1.1 The blockchain protocol

1.1.1 Centralized versus decentralized ledgers. A ledger is a collection of records, regarding ownership, transactions, identity, and so on. 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 centralized 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

[4] Relatedly, Catalini and Gans (2016) argue that blockchains improve verifiability and allow bypassing intermediaries. Khapko and Zoican (2017), who study endogenous settlement time, and Malinova and Park (2017), who study the impact of anonymity in financial markets, are motivated by features and capabilities of blockchains.

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 network participants, each including it in its 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 endeavor to incorporate the incentives of the participants.[5] This justifies our game-theoretic approach that accounts for these incentives to investigate the properties of this protocol.

1.1.2 Proof-of-work. Decentralization of the ledger implies that its validation should not be controlled or manipulated by a single participant or a small number of colluding participants. 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 many IPs."

The solution proposed by Nakamoto (2008) to this so-called Sybil attack is called "proof-of-work." 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. In order to be designated, each participant works to solve a computational problem with adjustable difficulty (referred to as a "moderately-hard puzzle" in computer science; see Dwork and Naor 1993) attached to his block. The term "work" in proof-of-work therefore refers 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 requests using 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). As nodes randomly draw candidate solutions, one of them eventually gets lucky and solves the problem before the others. Thus, proof-of-work is a way to randomize across participants who will propose the next change to the ledger.

A drawback of proof-of-work is that it requires costly computing capacity. An alternative protocol to save on electricity and hardware costs is proof-of-stake (see Saleh 2017). The basic idea is that each participant's probability to be designated to propose a block for validation depends on the amount of cryptocurrency he owns: the larger that amount, the more frequently a participant will be chosen. Ideally, this system ensures that those who have more stake in the network (and are thus more eager to maintain consensus) are more likely to contribute to the validation process. But no major fully

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

decentralized blockchain network uses proof-of-stake yet.[6] For that reason we focus on proof-of-work in our analysis.

A property of moderately-hard puzzles is that the solution is not easy 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) page 8, participants "vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them."

This process gives rise to a chain of consecutively solved blocks, that is, the blockchain, as illustrated in Figure 1.

1.1.3 Miners. The nodes conducting the earlier-mentioned tasks are called miners, as they get rewarded for solving proof-of-work problems with newly-created units of the cryptocurrency (12.5 BTC per block on Bitcoin in 2018, and 3 ETH on Ethereum since October 2017).[7] They also receive transaction fees that the originators of economic transactions can choose to offer for the validation of these transactions. These rewards are included in the block that generates them. 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 May 2018. The figure illustrates that 10 mining pools represented about 86% of the total hash capacity. Pools allow miners to mutualize 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."

1.1.4 Difficulty. As explained in Nakamoto (2008), the time it takes miner m to solve a block problem is exponential with parameter θ_m. Thus, θ_m is the instantaneous probability that m solves the block he is mining. θ_m is determined by the miner's individual computing power, or hash power, h_m, and by the

[6] Protocols inspired by proof-of-stake are currently used by Nxt or BlackCoin. The Ethereum community has been trying for a long time to develop a proof-of-stake algorithm. Their proof-of-work protocol even includes an exponential increase in difficulty (difficulty bomb) to induce miners to switch to the upcoming proof-of-stake protocol. Its development proved difficult, however, and the difficulty bomb has been delayed.

[7] These are the main sources of cryptocurrency creation. Some blockchain protocols can also include additional rewards.

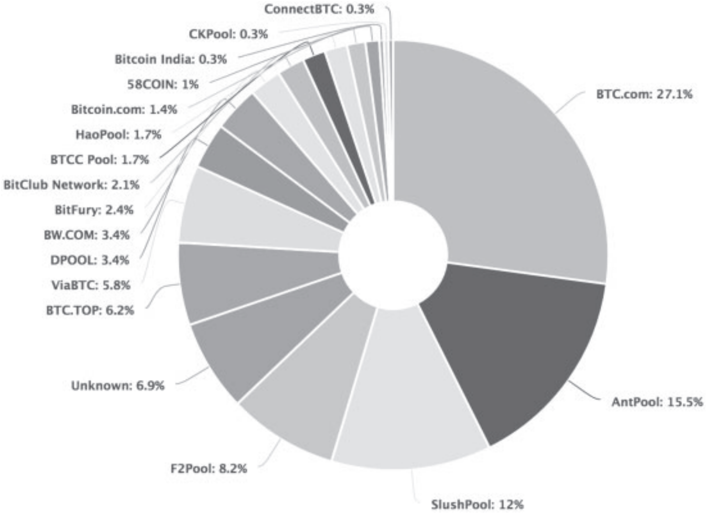

Figure 3
Estimated hashrate distribution of Bitcoin mining pools on May 8, 2018
Source: blockchain.info.

difficulty of the mining task set by the network protocol, D:[8]

$$\theta_m = \frac{h_m}{D}. \tag{1}$$

The difficulty is set by the blockchain protocol so that the expected time between two block solutions is targeted to a constant, X (on Bitcoin $X=10$ minutes, while on Ethereum it is currently 15 seconds). Thus, using the properties of exponential distributions, for given hash powers, the difficulty D is set so that

$$X = \frac{1}{\sum_{i\in\mathcal{M}}\theta_i} = \frac{1}{\sum_{i\in\mathcal{M}}\frac{h_i}{D}} = \frac{D}{\sum_{i\in\mathcal{M}}h_i}. \tag{2}$$

Equation (2) implies that the difficulty D is equal to the average duration between block solutions (X) multiplied by the total computing power:

$$D = X\left(\sum_{i\in\mathcal{M}} h_i\right). \tag{3}$$

If the total computing power increases (e.g., due to the entry of new miners), 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

[8] 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.

2,016 blocks, that is, approximately every two weeks, the difficulty is rescaled to ensure that the average time between blocks remains at 10 minutes.

Together, Equations (1) and (3) imply

$$\theta_m = \frac{1}{X} \frac{h_m}{\sum_{i \in \mathcal{M}} h_i}. \tag{4}$$

Equation (4) implies that the instantaneous probability to find a block for miner m is increasing in his or her own computing capacity, but decreasing in the capacity of the others.

1.2 Forks

Consensus on the decentralized ledger requires that there is only one chain of blocks, observed by all and on which all agree. It is jeopardized if the chain splits into a fork, with two competing branches, each with its own version of the ledger. In the present section we review some reasons why forks can happen in blockchains, and describe forks that actually occurred.

1.2.1 Information 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 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)

1.2.2 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, as soon as the seller delivers the object, the buyer has an incentive 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.[9]

1.2.3 Software upgrades. So far we have 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 to create a different version. As emphasized by Lerner and Tirole (2002), this gives rise to coordination issues on which version is adopted by participants. In a blockchain, a "soft fork" is the introduction of an upgraded version of the software that remains compatible with the previous version: blocks mined with the new version are considered as valid by miners still running the old version (but the opposite is in general not true). In contrast, a "hard fork" is not backward-compatible, as upgraded miners create blocks that are rejected as invalid by non-upgraded miners. As long as some miners do not accept blocks considered as valid by others, a software upgrade can trigger a fork in the blockchain. We describe here some actual unintended or intended forks triggered by software upgrades.

1.2.4 Unintended forks. An important chain split occurred on the Bitcoin blockchain in March 2013, following what developers thought to be an innocuous software upgrade: some time earlier, some miners upgraded to a new version of the software, referred to as 0.8. On March 11, 2013, 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 was in fact 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.

[9] See also Teusch, Jain, and Saxena (2016).

> Pieter Wuille: that is what should be done, but we should have consensus first.

Miners faced a dilemma. Should they follow the LCR 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-miximizing 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 (or a pool such as, e.g., BTC Guild) decides to chain his blocks to the branch that 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.

1.2.5 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 on Ethereum, 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 2018, Ethereum Classic represented about 3.5% of the hash capacity of Ethereum, and the price of ETC was about 3% 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 two hard forks, in the summer and in the fall of 2017. The first fork is linked to the size of blocks that can be mined on the blockchain. The community had long been divided on how to relax the limitation of the network throughput.[10] Several solutions 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 out a compromise solution (SegWit2x). Yet, another way to increase throughput, Bitcoin Cash, was implemented, via a hard fork, on 1 August 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 hard fork was planned for November 2017 to double the size of blocks. 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 Bitcoin community members 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 in 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, 2017, hashrates were similar on the two branches.

The second fork, Bitcoin Gold, which occurred in the fall of 2017, relies on a different proof-of-work algorithm than Bitcoin. It allows miners to mine efficiently using generic graphics processing units (GPU) by preventing the use of specific ASIC (integrated circuits that cannot be used for any other purpose than mining and are produced by a small number of firms). While it was initially unclear whether this fork would attract computing capacity, it is now implemented with a market capitalization around $1.2 billion as of May 2018.

A dozen other hard forks occurred on Bitcoin in December 2017, each proposing a modified version of the mining software or reward policy (as, for instance, Bitcoin God, Super Bitcoin, Bitcoin X, Oil Bitcoin, Bitcoin World, Lightning Bitcoin...). Although it is uncertain how many of these forks will remain, these episodes underscore 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 analyze the economic mechanisms that are at the roots of these forks.

2. Basic Model

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

[10] 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.

2.0.1 Miners and pools. There are $M > 2$ risk-neutral miners, indexed by $m \in \mathcal{M} = \{1, \dots M\}$. Equivalently, each m can be thought of as a mining pool, since, as explained in the previous section, a pool coordinates the strategies of a group of miners.

2.0.2 Mining technology. There is a continuous flow of transactions sent for confirmation by end users.[11] We assume all miners perfectly and instantaneously observe this flow, which they include in the blocks they mine.

As mentioned earlier, the time it takes miner m to solve a block problem is an exponentially distributed random variable with parameter θ_m. We first consider a stationary environment, in which the number of miners, their computer capacity, and the difficulty of the task are constant (so that the θ_m are constant also). Then, in Section 5, we endogenize difficulty and computing capacities.

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.[12] This waiting time is also independent of which block m is mining, and of which 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

$$N_m(t) = \int_{s=0}^{t} dN_m(s). \tag{5}$$

We assume that miners do not update the set of transactions defining the block they mine until a hash problem is solved (transactions that flow in meanwhile are stored in a buffer.) Relaxing that assumption would not alter the economic mechanism we analyze here.

We also 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 or purchased previously to a new miner who also inherits his beliefs and preferences. Thus, exits are compensated by entries and the environment is stationary. The times at which blocks are solved and liquidity shocks occur are all independent.

2.0.3 Blockchain. At time 0, there is an initial state of the ledger, encoded in B_0, and an initial set of transactions to be registered. Starting from B_0, miners start working on the first block, B_1, which contains this initial set of transactions.

[11] We take the flow of transactions to be exogenous, while in practice it can actually be endogenous. For simplicity, we don't model the transactions and model the blockchain process directly at the level of the blocks. See Carlsten et al. (2016) for an analysis of the choice of which transactions to include in a block.

[12] 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.

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 the chain splits into two competing chains, one including B_0 and B_1, the other B_0 and B_2.

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.

We first assume that at any time t, all miners observe the tree of solved blocks $\mathcal{C}^t=\{B^t, E^t, I^t\}$, where $B^t=(B_0,\dots B_n)$ is the set of all blocks that have been solved by time t, $E^t=\{(B_0,B_1),\dots(B_k,B_{k'}),\dots\}$, with $0\leq k<k'\leq n$, is the set of edges chaining these blocks, and $I^t=(m(B_1),\dots m(B_n))$ is the set of identities of miners who solved blocks. We then relax the assumption that $\mathcal{C}^t$ is observed instantaneously to study information delays. 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 $B_k\neq B_{k'}$ and both (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).

2.0.4 Stopping times. 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 get to make a decision. Thus, the sequence of stopping times at which miners make decisions is $\mathcal{T}=\{0,\dots\tau_j,\tau_{j+1},\dots\}$ where the next stopping time after τ_j, τ_{j+1}, is equal to $\tau_{j+1}=\min[\tau^l(\tau_j),\tau^b(\tau_j)]$, $\tau^l(\tau_j)$ 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.

2.0.5 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 $m \in \mathcal{M} = \{1, \dots M\}$ face the same action space.

2.0.6 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.[13] As mentioned in the previous section, Bitcoin imposes a 100-block delay before rewards for mining can be spent. To capture such delays in a simple manner, we assume miner m keeps the units of cryptocurrency he earned until z_m, and then consumes at z_m the rewards he earned throughout the game.

Therefore, a miner's lifetime payoff depends on the number of blocks he solved until z_m, and on the value of the rewards earned from these blocks when z_m occurs. For example, if a miner entered at date 0, and all his blocks solved have the same arbitrary value $G > 0$ at z_m, his payoff is written

$$\mathcal{G}_m = \left[\int_{s=0}^{s=z_m} dN_m(s) \right] G. \tag{6}$$

We now present what determines the value of blocks' rewards. When the cryptocurrency earned as a reward for mining a block is sold, the price it fetches 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—that is, 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 the chain. The market value of the block's reward in the first case (when the block ends up orphaned) is likely to be zero and is bound to be smaller than in the second case. Now turn to 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 can remain positive.

More formally, the payoff for miner m from solving B is an increasing function, $G(.)$, defined on $\{0, \dots, M\}$, of the number of miners active at time z_m in the chain including B.[14] 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, because it was mined before the

[13] For simplicity, we neglect transaction fees offered by final traders, since we do not explicitly model transactions. Carlsten et al. (2016) take the opposite approach by focusing on transaction fees.

[14] A bubble component could be added to the payoff function. In a rational bubble model (see for instance Tirole 1982 or Tirole 1985), the bubble component would be a martingale and would not affect the strategies of the miners since they are risk neutral.

inception of the fork, he earns $G(M-K)+G(K)$.[15] More generally, if a block belongs to I chains, each attracting K_i miners with $\sum_{i=1}^{I} K_i \leq M$, the value of this block is $\sum_{i=1}^{I} G(K_i)$. We set $G(0)=G(1)=0$, that is, when there is only one or no miner on a chain, the associated cryptocurrency has no value. Indeed, any level of consensus requires that at least two miners agree. We also assume $G(M)$ is strictly positive, hence $G(.)$ is strictly increasing in part of its domain. Finally, we assume that G is superadditive: when a fork occurs on a chain and creates two competing chains attracting respectively K_1 and K_2 miners, the total value of a unit of cryptocurrency that belongs to the two competing chains (because it was mined before the inception of the fork) is weakly lower than if the fork had not occurred. Formally, we have $G(K_1)+G(K_2) \leq G(K_1+K_2), \forall (K_1, K_2)$ such that $K_1+K_2 \leq M$. This implies that the value of a block cannot be greater than $G(M)$, whatever the number of chains that block belongs to. Note that in our setting the payoff function $G(.)$ depends on the total number of miners M. This captures the value of consensus: miners' rewards depend on the relative number of miners active on each chain, rather than on the absolute one.

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 UTC on March 11) at which miners started working on two different chains. Chats between miners realizing there was a fork started around 23:30.[16] At 1:30 on March 12, a message posted on Bitcointalk asked miners to stop mining one of 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 realized that miners worked on different branches, this triggered a 25% drop in the value of the virtual currency (from around 48 at 1:00 to around 36 at 3:00).

2.0.7 States. At time $\tau \in \mathcal{T}$, a state ω_τ includes two 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, as in Duggan (2012) or Cole and Kehoe (2000), to enable players to coordinate their actions using a public randomization device,

[15] We also assume that, if z_m occurs just after a miner attempts to fork, the not-yet-realized fork does not reduce the credibility of the current chain. That is, $m(B)$ earns $G(M)$ for any block B on the current chain, as long as no block creating a fork has been mined. An alternative hypothesis is that the attempt to fork reduces mining rewards. Our proofs are robust to that alternative hypothesis.

[16] Source: http://web.archive.org/web/20130421062533/http://bitcoinstats.com/irc/bitcoin-dev/logs/2013/03/11.

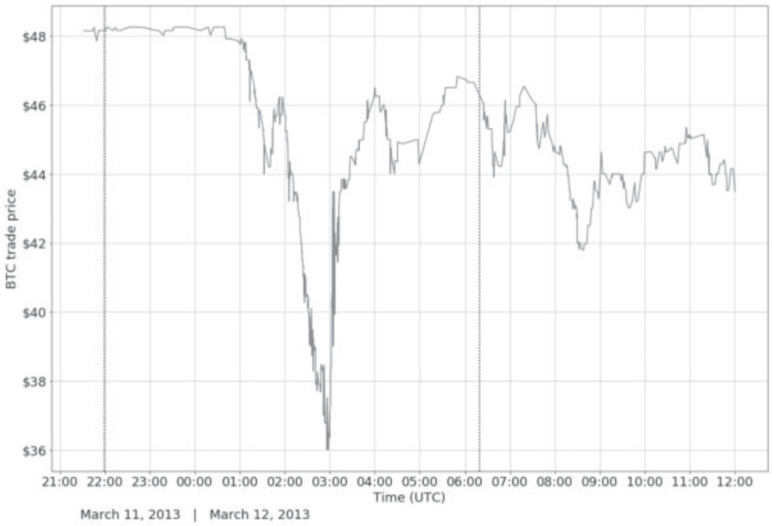

Figure 4
BTC/USD trade prices on Bitstamp exchange, around the March 2013 Bitcoin fork
The graph plots individual transaction prices obtained from a major bitcoin exchange platform, Bitstamp, during the March 2013 fork. The first dotted vertical line represents the time at which the fork started, and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

> we assume that at each time $\tau \in \mathcal{T}$, the realization 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 $\omega_\tau = (\mathcal{C}^\tau, r^\tau)$ and denote by Ω the set of states of the world.

2.0.8 Strategies. Miner m chooses his strategy to maximize 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, for instance, the exact timing of blocks resolution and the previous mining choices. In line with Markov perfection, we consider only 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 $\{N_m\}_{m \in \mathcal{M}}$, yields the transition probabilities from one state of the blockchain to the next.

2.0.9 Equilibrium. The previously described elements define our stochastic game. Our equilibrium concept is Markov perfect equilibrium, that is, subgame

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

perfect equilibrium with strategies restricted to depend only on the current state ω_τ.

3. Equilibrium Analysis of the Basic Model

To analyze equilibrium strategies, it is useful to first note that the maximum lifetime payoff of miner m, entering the market at time 0, is

$$\mathcal{G}_m^{\max} = \left[\int_{s=0}^{s=z_m} dN_m(s) \right] G(M). \tag{7}$$

If the miner enters the market later, to replace an earlier miner hit by a liquidity shock, his maximum payoff is the same as in Equation (7), minus the price he pays for the cryptocurrency bought from the previous miner. This sunk cost does not affect his strategy, and we neglect it hereafter.

$\mathcal{G}_m^{\max}$ is a maximum because (i) the total number of blocks solved by m before z_m is $\int_{s=0}^{z_m} dN_m(t)$, whatever his mining strategy, and (ii) m cannot earn more than $G(M)$ each time he solves a block. At time t, the expectation of $\mathcal{G}_m^{\max}$, conditional on $z_m \geq t$, is

$$\begin{aligned} &\mathbb{E}_t \left[\int_{s=0}^{t} dN_m(s) + \int_{s=t}^{z_m} dN_m(s) | z_m \geq t \right] G(M) \\ &= \left\{ N_m(t) + \mathbb{E} \left[\int_{s=t}^{z_m} dN_m(s) | z_m \geq t \right] \right\} G(M) \\ &= \left\{ N_m(t) + \frac{\theta_m}{\lambda_m} \right\} G(M). \end{aligned} \tag{8}$$

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, $\mathbb{E}[\mathcal{G}_m^{\max}]$.

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, $\mathbb{E}[\mathcal{G}_m^{\max}|\omega_\tau]$, by also abiding to it. Hence there is no profitable one-shot deviation from the strategy that 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 will not follow him, and the block solved out of the equilibrium path will have no value.

In the context of the strategic interaction characterized 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. This arises only when difficulty is constant, however. In Section 5, we study the determination of total computing capacity and discuss its allocations to different branches of the blockchain. In that context, we point out that when one miner devotes larger computing power to one branch, he raises the difficulty for the others. This brings in a competition effect, running in the opposite direction to the coordination effect discussed earlier.

It is 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 behavior. 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 realization of the sunspot variable is such that miners anticipate a fork. As shown in the following, 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 $\varepsilon > 0$ can be arbitrarily small), and let $n(\tau)$ denote the index of the last block solved by time τ. Relying on this notation, we can state our next proposition:

Proposition 2. Consider an arbitrary integer $f > 0$ and $\varepsilon > 0$. 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 the previous block, $B_{n(\tau)}$. 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 characterize the equilibrium strategy profile for any state. The intuition of Proposition 2 is the following: If a miner expects all to fork to $B_{n(\tau)-f}$, but chooses to deviate and not fork to $B_{n(\tau)-f}$, then he expects any block he solves will be orphaned and earn no reward. Rationally anticipating this, he chooses to do like the others and fork to $B_{n(\tau)-f}$.

Miners' behavior in Proposition 2 is reminiscent of actual participants' behavior during the 2013 Bitcoin fork reported in Section 1.2. 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. They fork nevertheless because they anticipate that the others do.

Proposition 2 is stated for any integer f. In practice, Bitcoin Core (the reference implementation of the bitcoin system) includes several hard-coded checkpoints, that is, identifiers of blocks coded into the software so that any fork starting from an earlier block is considered as invalid. In the presence of checkpoints, f cannot be larger than the distance from the last checkpoint (provided that miners do not decide to remove checkpoints from their software code.)

If we measure welfare with respect to miners' payoffs,[18] then the equilibrium in Proposition 2 is Pareto-dominated by the one in Proposition 1. In Proposition 1, every block earns the miner who solved it a reward $G(M)$ when it is sold to a subsequent miner. Any subsequent miner who buys the reward resells it for the same price $G(M)$, which is therefore payoff-neutral for him. In Proposition 2, the same applies for any block that does not belong to the orphaned chain $(B_{n(\tau)-f+1}, B_{n(\tau)})$. However, a miner who solved (or inherited) a block on the orphaned chain obtains a reward worth $G(0)=0$ after the fork, instead of $G(M)$ in the equilibrium of Proposition 1. Hence, the fork makes at least one miner strictly worse off without strictly increasing the payoff of any other miner.

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 Ethereum 2016 and Bitcoin 2017 forks discussed in Section 1.2.

As in Proposition 2, we consider the possibility that, at some time τ^f, the realization of the sunspot suggests forking to a new branch, chained to $B_{n(\tau^f)-f}$. This can give rise to two coexisting chains at time $\tau > \tau^f$, the original chain, including the blocks linked by the sequence of edges

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

[18] $G(.)$ can be thought of as the gain from seigniorage, reflecting the value of the cryptocurrency as a medium of exchange. For simplicity, however, $G(.)$ is exogenous in our model. So we can only define welfare in terms of the miners' gains.

and a new chain, including the blocks linked by

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

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 let $v_m^o(\tau, \tau^f)$ and $v_m^n(\tau, \tau^f)$ denote the vested interests of miner m at time $\tau \geq \tau^f$ on the original and the new chain (respectively) when the fork starts at τ^f. Specifically, suppose miner m keeps mining the original chain after τ^f. At $\tau \geq \tau^f$, his vested interest on the original chain is then equal to $v_m^o(\tau, \tau^f) = N_m(\tau) - N_m(\tau(B_{n(\tau^f)-f}))$ (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 $v_m^n(\tau, \tau^f) = 0$. Alternatively, consider miner m' who mines the new chain from time τ^f on. His vested interest on the original chain at time $\tau \geq \tau_f$ is $v_{m'}^o(\tau, \tau^f) = N_{m'}(\tau^f) - N_{m'}(\tau(B_{n(\tau^f)-f}))$, while his vested interest on the new chain is $v_{m'}^n(\tau, \tau^f) = N_{m'}(\tau) - N_{m'}(\tau^f)$. For miners switching between the original chain and the new one, vested interests are a bit more intricate, but they follow the same logic.

Our next result illustrates the consequences of miners' vested interests. To state that result, rank the miners by their vested interest in the original chain at the start of the fork τ^f as follows:

$$\frac{\Pr(z_m = \tau')}{\Pr(N_m(\tau') - N_m(\tau^f) = 1)} v_m^o(\tau^f, \tau^f)$$
$$\leq \frac{\Pr(z_{m+1} = \tau')}{\Pr(N_{m+1}(\tau') - N_{m+1}(\tau^f) = 1)} v_{m+1}^o(\tau^f, \tau^f), \tag{9}$$

where $\Pr(z_m = \tau')$ is the probability that miner m is hit by a liquidity shock at the next stopping time τ', and $\Pr(N_m(\tau') - N_m(\tau^f) = 1)$ is the probability that he solves his block at τ'. We show in the proof of Proposition 3 that the ratio of these two probabilities simplifies to $\frac{\theta_m}{\lambda_m}$ for miner m, and use this simplification hereafter.

To simplify the exposition, assume that for any $K \in \{0, \dots, M\}, G(K) + G(M-K) = G(M)$. To state our next proposition, we need to define the following condition on the realized distribution of blocks' rewards among miners.

Condition 1. At time τ, there is a single chain and there exists an integer $K^* \geq \frac{M}{2}$ such that for $m > K^*$

$$\frac{\theta_m}{\lambda_m} \big(G(K^*) - G(M-K^*)\big) \leq v_m^o(\tau, \tau)(G(M-K^*) - G(M-K^*-1)) \tag{10}$$

while for $m \leq K^*$

$$\frac{\theta_m}{\lambda_m}(G(K^*) - G(M-K^*)) \geq v_m^o(\tau, \tau)(G(M-K^*+1) - G(M-K^*)). \tag{11}$$

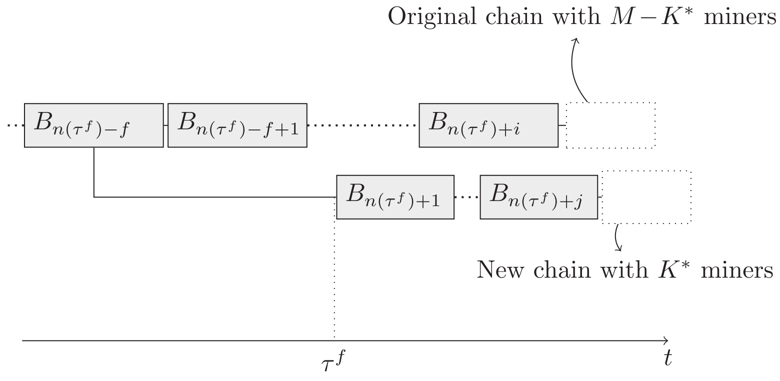

Figure 5
Equilibrium of Proposition 3

To illustrate Condition 1, consider the following example: M=5 and the function $G(.)$ is defined by $G(0)=G(1)=0$; $G(2)=\frac{6}{5}$; $G(3)=\frac{9}{5}$; $G(4)=G(5)=3$, so that all properties of $G(.)$ are satisfied.[19] Assume also that all miners are identical and have the same ratio $\frac{\theta}{\lambda}$. Then Condition 1 holds if there are two miners (ranked 4 and 5) each with a number of blocks solved in the last f blocks larger than $\frac{\theta}{2\lambda}$, and three miners (ranked 1, 2, and 3) each with a number of blocks solved in the last f blocks lower than $\frac{\theta}{\lambda}$. This example also highlights that there can be parameter values for which Condition 1 is never satisfied. This is the case, for instance, if f is smaller than $\frac{\theta}{\lambda}$. The next proposition states that persistent forks can occur in equilibrium, as illustrated in Figure 5.

Proposition 3. Assume that $M \geq 4$. Consider an arbitrary integer $f > 0$ and $\varepsilon > 0$. Define τ^f as the first time τ at which $r^\tau > 1-\varepsilon$, $f < n(\tau)$, and Condition 1 holds. 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.

The intuition for the result is the following. First note that for some miners to fork, we must have that the left-hand side of Equation (11) be positive, which implies that $K^* \geq \frac{M}{2}$. That is, in Proposition 3, persistent forks can occur only if a majority of miners choose to fork and this is expected by all. Next, see that if $M \leq 3$, Conditions (10) and (11) can only be satisfied for the degenerate

[19] A more general example of $G(.)$ that satisfies all our assumptions is: $G(0)=G(1)=0$, $G(M)=G(M-1)=F$, where F is a strictly positive constant, and for all K such that $1 < K < M-1$, $G(K)=F\frac{K}{M}$.

function $G(K)=0$ for all K, which contradicts our assumption that $G(M)>0$. This is why we assume $M\geq 4$ in the statement of Proposition 3.

Now, suppose all miners expect that a majority will fork and this will result in two coexisting chains, and examine whether miner m prefers forking or 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, $\Pr(z_m=\tau')$, that is, if θ_m is large relative to λ_m. This benefit is captured in the left-hand sides of Equations (10) and (11) 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 $v_m^o(\tau,\tau)$ is large. This cost is captured in the right-hand sides of Equations (10) and (11) in Condition 1. The incentive effect of vested interests is self-reinforcing once the fork occurred. After τ^f, miners $m\leq K^{*}$ start accumulating vested interests on the new chain while miners $m>K^{*}$ continue accumulating vest interests on the original chain, entrenching further their respective strategies.

Last, the assumption that for any $K\in\{0,\dots,M\}, G(K)+G(M-K)=G(M)$ ensures that miners only consider the value of rewards from blocks solved after $B_{n(\tau^f)-f}$ when assessing the benefit and cost of following one chain or another.[20] It therefore simplifies the presentation of Condition 1. Were this not true, the value of rewards from these earlier blocks would enter Condition 1 as well, but a persistent fork equilibrium could still be sustained.

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, and his vested interest in the original chain. A miner is more likely to fork when the former is higher and the latter is lower.

Unlike in Proposition 1 and Proposition 2, the equilibrium outcome in Proposition 3 depends on the number of miners. More precisely, the trade-offs 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-dominated by that in Proposition 1. As earlier, this is because no miner can

[20] Note that this assumption does not preclude that for some $K_1+K_2<M, G(K_1)+G(K_2)<G(K_1+K_2)$.

earn more than $G(M)$ per solved block, and blocks that belong to either branch after the fork have a strictly lower value ($G(K^*)$ or $G(M-K^*)$) than $G(M)$.

4. Enriching the Model

So far we have considered a stylized frictionless case, in which, relying on abstract sunspots, we showed that coordination effects could lead to forks and equilibrium multiplicity. We now enrich the analysis by introducing information delays, double-spending attempts, and upgrades. We show that these realistic events can play a similar role as sunspots in triggering forks, also because of coordination effects.

4.1 Information delays

As mentioned previously, Nakamoto (2008) considered the possibility of information delays in the network, generating short-term forks, and argued these forks would be rapidly resolved, as miners would follow the longest chain rule. To examine this point, we extend our model to analyze equilibrium mining in the presence of delays. We show how the interplay between information delays and coordination effects gives rise to multiple equilibria.

To keep things as simple as possible, we assume that a delay in information transmission can happen only once. As long as there has been no delay, 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 $m(B_n)$. 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, along with the information that B_{n+1} has been solved. To simplify proofs, we also assume that $G(M)=G(M-1)+G(1)$ (which implies $M \geq 3$).[21] In this extension of our model we obtain the following result.

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 that 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 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

[21] 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.

conjecture of Nakamoto (2008). The equilibrium described in Proposition 4 therefore illustrates the robustness of the LCR equilibrium of Proposition 1 to information delays. This, however, is not the only equilibrium. Because of coordination effects, other equilibria can be sustained, in which miners don't behave as suggested by Nakamoto (2008). 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. Yet, when an information delay causes a fork, with two equally long branches, miners abandon the chain on which they were active and follow the fork. This equilibrium is in line with Proposition 2: Both show how coordination effects underpin equilibrium multiplicity. Yet, while in Proposition 2 forking was triggered by an abstract sunspot, in Proposition 5 it is triggered by a realistic event, information delays.

In Proposition 5 the fork is only one block long, because the delay can only affect the observation of one block. On Bitcoin, most temporary forks are one block long. We collected data from one full node operated by blockchain.info, which currently reports orphaned blocks since March 2014. We found that out of 935 reported forks between March 2014 and December 2017, 928 were only one block long. Intuitively, the frequency of such forks depends on the relative frequency of block resolution compared to network latency. For example, one-block-long forks are much more frequent on Ethereum (with one block solved every 15 seconds) than on Bitcoin (with one block solved every 10 minutes): on average there have been over 1,000 one-block long forks per day on Ethereum since January 2018 (source: https://etherscan.io/chart/uncles.).

Longer forks could arise if delays affected more blocks. Note that delays are not necessarily due to network latency. In the case of the Bitcoin March 2013 fork, delays occurred because one block was mistakenly rejected by computers running one version of the mining software and it took time for miners to become aware of that problem. As discussed in Section 1.2, miners then found it difficult to coordinate on a single chain. This is consistent with Propositions 4 and 5, which show there is a multiplicity of equilibria, making it difficult for participants to coordinate. Another example occurred during the implementation of a soft fork (BIP66) on Bitcoin in July 2015. Some mining pools representing 40% of the network hashpower were mining blocks without fully validating previous blocks (a strategy referred to as "simplified payment verification mining"). They thereby accepted blocks that they should have considered as invalid, which created a 6-block-long fork on July 4, 2015, and a 3-block-long fork on July 5, 2015. Although these short-lived forks differ from persistent forks sustained by vested interests, our analysis uncovers that in both types of fork, coordination effects play a key role.

4.2 Double-spending

Another important potential issue outlined in Nakamoto (2008) is double-spending. We study later whether it can arise at equilibrium. In the spirit of the modeling of delays earlier, assume that after each block is solved, there is a probability η' that one miner has an opportunity to divert the payment S from a transaction included in the last block. To earn S, the miner needs to create a fork and ensure it becomes the only active chain. Assume the opportunity to double-spend occurs only once and denote $\gamma(m,\tau)$ the probability that at any time τ, m is the miner who solves the next block after τ. As for Proposition 4, we assume that $G(M)=G(M-1)+G(1)$ to simplify proofs (which again implies $M\geq 3$).

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

$$S>\frac{G(M)(2-\gamma(m,\tau))}{\gamma(m,\tau)}. \tag{12}$$

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 endeavors to solve two blocks in a row before the other miners solve any new block on the original chain. If he succeeds, the fork started by the double-spending miner becomes the longest chain. If the other miners follow the longest chain rule, they then chain their blocks to the forking branch. This enables the miner to recover S and spend it again.

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 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.

It is profitable to try to double-spend if Condition (12) holds. The intuition for that condition is the following. When trying to create a fork after spotting S, a miner faces a large risk that his fork does not become the longest chain. The higher $\gamma(m,\tau)$, the lower that risk, and the higher the double-spending S, the larger the compensation for that risk.

In practice, however, Condition (12) is unlikely to hold. A back-of-the-envelope computation suggests that if λ is small and there are 15 identical miners, Condition (12) can be approximated as $S > 30G(M)$. For Bitcoin, $G(M) = 12.5$ bitcoins in 2018, hence for Condition (12) to hold, S must be larger than 375 bitcoins, which is a sizable amount. This condition could be relaxed, however, if instead of deriving an equilibrium where all miners can double-spend, one focuses on an equilibrium in which only the miner with the largest computing power double-spends.

Selfish mining is a related type of attack by which a miner starts a fork in the hope that it will become longer than the original chain. In this attack, the selfish miner delays the broadcasting of the blocks he mines on the forking chain, so that the other miners continue mining the original chain. They therefore bear the risk of losing all the rewards they earned on that chain. As in the case of double-spending, selfish mining can only be profitable if a miner has a benefit other than blocks rewards.[22]

4.3 Upgrades

As discussed in Section 1.2, blockchain participants sometimes introduce 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, $C = 0$, and 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 allow for the possibility that miners derive private benefits from using one technology or the other. Private benefits can reflect a cost advantage. For instance, some argued that the mining pools controlled by Bitmain, an ASIC manufacturer, favored Bitcoin Cash over SegWit because they had access to a patented mining-enhancing device that cannot be used with SegWit.[23] In contrast, Bitcoin Gold is favored by miners who find it costly to rely on ASIC. Private benefits can also 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 these large pools 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.

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

[22] For instance, Eyal and Sirer (2014) consider that miners want to maximize their relative revenue, so that a selfish miner has an interest in reducing the profit of the others.

[23] This is the ASICBOOST technology that can increase the efficiency of the SHA-256 calculation.

another equilibrium in which, on the equilibrium path, all miners follow the LCR and choose technology $C=1$.

If a miner anticipates all the other miners will choose technology C, then it is a best response for this miner to also choose C, whatever his or her private benefits. 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 $C=0$. We now explore how a persistent fork can also be sustained at equilibrium in the presence of private benefits. For simplicity, suppose that there are $M-K^*$ miners with a positive private benefit for the original technology. We assume $b_m(1)=0$, $\forall m$, while $b_m(0)=0$, $\forall m \leq K^*$ and $b_m(0)=b>0$ $\forall m>K^*$. Denote τ^f the time at which the new technology is introduced and assume for simplicity, as in Proposition 3, that $G(K)+G(M-K)=G(M)$ for all $K \in \{0,\ldots,M\}$.

Proposition 8. Assume $M \geq 4$. If

$$b \geq \frac{G(K^*)}{G(M-K^*)}-1, \tag{13}$$

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

Proposition 8 focuses on the case in which the majority of miners (K^* out of M) have no private benefits, while a minority of them strongly prefers not to adopt the upgrade. In this context, if it is expected that the majority will opt for the upgrade, then all the miners who don't have any private benefit anyhow choose to adopt the upgrade, but the minority with strong private benefits sticks to the previous version of the software. This result is reminiscent of the persistent fork equilibrium described in Proposition 3. For similar reasons as in Proposition 3, we must have $M \geq 4$ for the equilibrium to exist. One difference is that here, the fork is initiated by exogenous private benefits, instead of the endogenous distribution of vested interests from past solved blocks in Proposition 3. However, once the fork occurred, vested interests play the same reinforcing role as in Proposition 3 by increasing the incentives of miners to stick to the chain they initially picked.

Proposition 7, in which miners must choose between two versions of the software and coordinate on a unique version, is in line with the situation observed in November 2017, in which all miners eventually coordinated on refusing SegWit2x. Such unanimity contrasts with Proposition 8, in which a minority with strong private benefits rejects the upgrade adopted by the majority. This is in line with the situation observed in July 2016, when a minority of miners rejected the Ethereum hard fork for ideological reasons. This is also in line with the Bitcoin Gold fork, which is mined by a minority of participants, some of which have strong vested interest in that chain due to their initial allocation of Bitcoin Gold.[24]

As mentioned earlier, many hard forks occurred during the year 2017 on Bitcoin. Our analysis highlights that larger, unevenly distributed private benefits can trigger forks. It is likely that the increase in the network activity, and the corresponding increase in Bitcoin value, have generated larger private benefits for some miners, derived from specific technologies or mining rules, which can explain why fork attempts have been more frequent.

5. Computing Capacity and Difficulty

The previous analysis takes difficulty and computing capacity as given and fixed, independently from the choices of the miners. Correspondingly, θ_m is also given and fixed for all m. In the short term this is a realistic assumption. For example, on the Bitcoin blockchain between two resets of the difficulty, that is, approximately within two weeks, difficulty remains constant in all branches. In the longer term, however, this assumption is less realistic, as miners can decide to change the computing capacity they allocate to different blockchains, and difficulties are correspondingly reset. In this section we examine separately (for simplicity) two aspects of these issues: First, holding total capacity constant, we discuss how the allocation of computing capacity to different branches changes difficulty in the different branches. Second, focusing on the equilibrium of Proposition 1 (with a single chain), we analyze the ex ante equilibrium choice of computing capacity. A key point in both aspects of the analysis is that when one miner increases his computing capacity in one branch, he increases the difficulty in that branch, which exerts a negative externality on the other miners operating in that branch.

5.1 Difficulty

Suppose there are two persistently competing branches in the blockchain, x and y, attracting different computing capacity. The key difference between this setting and our previous analysis is that now the different branches are assumed to have different difficulties, reflecting different hash power (which is appropriate in the long term), while previously we assumed the different branches had the same difficulty (which is appropriate in the short term). To discuss the strategic issues arising in this setting, suppose miners $m=1,\dots K$

[24] Bitcoin Gold developers allocated themselves 100,000 units of the new cryptocurrency by pre-mining blocks before opening the chain to the public.

mine branch x while miners $m=K+1,...M$ mine branch y.[25] Consider one miner, for example miner K. If K mines branch x, by Equation (3), his instantaneous probability of solving a block is

$$\theta_K(x)=\frac{1}{X}\frac{h_K}{\sum_{m=1}^{K} h_m}, \tag{14}$$

while his reward from solving a block in that branch is $G(K)$. If, instead, miner K chose to mine in branch y, his instantaneous probability of solving a block would be

$$\theta_K(y)=\frac{1}{X}\frac{h_K}{\sum_{m=K}^{M} h_m}, \tag{15}$$

while his reward from solving a block in that chain would be $G(M-K+1)$.

Suppose that branch x attracts less miners and computing capacity than y. Then miner K faces a trade-off: if he successfully mines in branch x, his reward $G(K)$ is lower than if he succeeds in branch y; on the other hand, his probability to solve a block is higher on x since the difficulty of mining is larger on y.

The first effect, already present in the previous subsections, makes it attractive for miners to mine where they expect the others to mine (“crowding in”). The second effect (“crowding out”) is new to this section and makes it less attractive to mine a branch where many others mine. Characterizing equilibrium in that situation is beyond the scope of the present paper. Yet two different types of externalities can be pointed to.

- On the one hand, when moving from branch x to branch y, miner K exerts a positive externality on miners $m > K$, by increasing the value of the rewards in their branch from $G(M-K)$ to $G(M-K+1)$.
- On the other hand, miner K exerts a negative externality on miners $m > K$, by increasing the computing capacity, and hence the difficulty in branch y.

5.2 Computing capacity

We now endogenize the total computing capacity in the network, that is, the choices of hash rates h_m by all miners at time 0. We investigate the relation between equilibrium hash capacity and the socially optimal one. Our analysis of equilibrium capacity is similar to Dimitri (2017). The contribution of this subsection, relative to Dimitri (2017), is to identify an externality in the computing capacity acquisition game, which drives a wedge between equilibrium and social optimality.

To choose h_m, miner m needs to anticipate how his computing capacity will affect his continuation game payoff. To do so, the miner needs to form

[25] Miners could also split their computing capacity across the two branches, but that would not alter the thrust of the argument.

a conjecture on the equilibrium that will prevail in the mining game. For simplicity, we assume all miners rationally anticipate that the single chain equilibrium described in Proposition 1 will prevail.

The program of miner m is

$$\max_{h_m} \frac{\theta_m}{\lambda_m} G(M) - \frac{c_m h_m}{\lambda_m}, \tag{16}$$

where $c_m h_m$, with $c_m > 0$, is the cost of using h_m per unit of time. Since $c_m h_m$ can be interpreted as the rental cost of the equipment plus the cost of electricity, it is natural to assume that the cost is linear in the capacity. Miner m bears this cost until he is hit by a liquidity shock.[26]

A Nash equilibrium of the computing capacity acquisition game is a vector $\{h_m^*\}_{m=1,...M}$ 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 when all M miners choose to participate.[27]

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

$$h_m^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i} \left(1 - c_m \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right). \tag{17}$$

The total computing capacity installed on the network is

$$\sum_{i \in \mathcal{M}} h_i^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}. \tag{18}$$

Condition $c_m \leq \frac{\sum_{i \in \mathcal{M}} c_i}{M-1}$ holds if costs, c_i, are not too different across miners. It ensures that the solution to Equation (17) is positive for all miners, while condition $\frac{\sum_{i \in \mathcal{M}} c_i}{M-1} \leq G(M)$ ensures that it is always possible to find a value of the difficulty parameter D above one, such that the expected time between two blocks is X. Equation (17) implies that equilibrium individual computing capacity is increasing in the mining reward ($G(M)$), decreasing in the average duration between blocks (X) and in the unit cost c_m. Correspondingly, Equation (18) implies that total network capacity decreases with $\sum_{i \in \mathcal{M}} c_i$.

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

[26] When miner m is hit by a shock, his successor inherits the computing capacity h_m.

[27] There is also an equilibrium such that all miners choose not to participate. Indeed, if a miner anticipates that the others will not install any computing capacity, his best response is not to install any capacity either since $c_m > 0$ and $G(1) = 0$. We focus on the equilibrium in which the network is created.

The corresponding maximization problem is

$$\max_{h} M\left(\frac{1}{\lambda X}\frac{h}{Mh}G(M)-\frac{c}{\lambda}h\right). \tag{19}$$

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 Equation (18) we see that, under condition $\frac{\sum_{i\in\mathcal{M}} c_i}{M-1}\leq G(M)$ of Proposition 9, in equilibrium there is overinvestment in computing capacity: 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.

As in the previous subsection, by devoting computing capacity to the mining task, a miner exerts a negative externality on the others, by increasing the difficulty. This negative externality drives a wedge between equilibrium and social optimality.

6. Conclusion

Miners' incentives are key to the production of a robust consensus in a blockchain. This paper shows that, while miners benefit from coordinating on a single chain, thereby maintaining consensus, coordination motives can also lead to abandoning portions of the blockchain. This can jeopardize the blockchain's key function, that is, producing a stable and immutable history of transactions. In addition, vested interests can lead to the persistence of multiple active chains.

Could these issues be overcome by communication among participants? In practice, communication among participants proved ineffective in the cases of Ethereum in 2016 and SegWit2x in 2017. This is in line with theoretical and experimental findings that communication in games does not necessarily facilitate coordination (see, e.g., Crawford 1998).

Another issue relates to the negative externalities arising in proof-of-work blockchains. First, as shown earlier, when choosing individually optimal computing capacity, miners fail to internalize the negative externality their investment generates for other miners by increasing difficulty. This implies that equilibrium capacity acquisition in proof-of-work mining is excessive. Second, proof-of-work mining generates greenhouse-effect negative externalities, whose order of magnitude is significant. As of May 2018, the electricity consumed for Bitcoin mining was equal to the electricity consumption of over 6 million U.S. households, with an average consumption per transaction of around 850 KWh.[28] Pigovian taxation could curb overinvestment in mining, but

[28] Source: https://digiconomist.net/bitcoin-energy-consumption.

it might also be difficult to put in place, given the international decentralization of mining.

This suggests moving from proof-of-work to proof-of-stake, in which participants do not have to use computing power and energy to propose blocks for consensus. Note, however, that proof-of-stake is exposed to the same coordination problems as proof-of-work, since in both protocols participants must choose which blocks to accept and are rewarded when the others agree with their choice. In addition, proof-of-stake comes with its own problems, in particular the nothing-at-stake effect: a participant can stake his cryptocurrency units on different branches, thus hindering the emergence of consensus.[29]

This points to a major dilemma for distributed ledgers: On the one hand, the anonymity and decentralization of public blockchains expose them to coordination failures and externalities. On the other hand, private blockchains can restore coordination and internalize externalities, but only to the extent that they involve the intervention of a centralized authority, which goes against the fundamental motivation for blockchain.

Appendix

Notation. We summarize here 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.[30]

Lemma A1. Our blockchain game is continuous at infinity.

Proof of Lemma A1. Denote by $\mathcal{G}_m(\sigma_m)$ the lifetime payoff of miner m if he follows strategy σ_m. Consider any 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

$$\mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')]=\Pr(z_m\leq T)\mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')|z_m\leq T]$$
$$+\Pr(z_m>T)\mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')|z_m>T]. \tag{A1}$$

[29] Current proof-of-stake proposals like Casper try to alleviate this problem by imposing coin deposits that can be seized if a participant is observed to simultaneously bet on several competing branches. It is unclear, however, whether these proposals are sufficiently robust to be implementable.

[30] See, for instance, Proposition 5.7.1 in Mailath and Samuelson (2006).

Now, by definition,

$$\mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')|z_m \leq T]=0. \tag{A2}$$

Moreover, recall that $\Pr(z_m > T)=e^{-\lambda_m T}$, so that we have[31]

$$\lim_{T\to\infty} \Pr(z_m > T)=0. \tag{A3}$$

Last, $\mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')|z_m > T]$ is bounded, since $\mathbb{E}[\mathcal{G}_m^{\max}]$ is finite. Hence,

$$\lim_{T\to\infty} \mathbb{E}[\mathcal{G}_m(\sigma_m)-\mathcal{G}_m(\sigma_m')]=0, \tag{A4}$$

which ensures that our game is continuous at infinity. ■

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'.[32] 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 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 $B_{n(\tau)+1}$. 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. ■

[31] Our liquidity shock has the same impact on miner m's long-term payoff as a discount rate equal to λ_m.

[32] The next event can also be that another miner is hit by a liquidity shock. Which block m chose as a parent block is also irrelevant in this case. For brevity, we ignore this case in the remainder of the proofs.

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 exist.

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. ■

Proof of Proposition 3. We start with preliminary steps. Consider τ_f defined in Proposition 3, and define the new chain as in the proof of Proposition 2. Let $v_m^n(\tau,\tau^f) = N_m^n(\tau) - N_m^n(\tau^f)$ be miner m's vested interest 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 a given $K^* \geq \frac{M}{2}$, for $\tau \geq \tau^f$, ω_τ is such that for $m > K^*$

$$v_m^o(\tau,\tau^f)(G(M-K^*)-G(M-K^*-1)) - v_m^n(\tau,\tau^f)(G(K^*+1)-G(K^*))$$
$$\geq \frac{\Pr(N_m(\tau')-N_m(\tau)=1)}{\Pr(z_m=\tau')}(G(K^*)-G(M-K^*)), \quad \text{(A5)}$$

while for $m \leq K^*$

$$v_m^o(\tau,\tau^f)(G(M-K^*+1)-G(M-K^*)) - v_m^n(\tau,\tau^f)(G(K^*)-G(K^*-1))$$
$$\leq \frac{\Pr(N_m(\tau')-N_m(\tau)=1)}{\Pr(z_m=\tau')}(G(K^*)-G(M-K^*)). \quad \text{(A6)}$$

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 $\Delta\omega \equiv \omega^\tau \setminus \omega^{\tau^f}$ (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 original and the new chain are defined with respect to history $\omega^{\tau'} \setminus \Delta\omega$.[33]

As will become explicit later, 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 case 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 of 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

$$\Pr(\tau^f < z_m|\omega_\tau)\left[f+\frac{\theta_m}{\lambda_m}\right]G(M). \tag{A7}$$

Now,

$$\Pr(\tau^f < z_m|\omega_\tau)=\int_{z_m=\tau}^{\infty}\Pr(\tau^f < z_m|\omega_\tau, z_m)\lambda_m e^{-\lambda_m z_m}dz_m. \tag{A8}$$

Observe that

$$\begin{aligned}\Pr(\tau^f < z_m|\omega_\tau, z_m) &< \Pr(\exists\hat{\tau} < z_m, r^{\hat{\tau}} > 1-\varepsilon|\omega_\tau, z_m)\\ &= 1-\Pr(\forall\hat{\tau} < z_m, r^{\hat{\tau}} \leq 1-\varepsilon|\omega_\tau, z_m).\end{aligned} \tag{A9}$$

Moreover,

$$\Pr(\forall\hat{\tau} < z_m, r^{\hat{\tau}} \leq 1-\varepsilon|\omega_\tau, z_m)=\mathbb{E}[(1-\varepsilon)^{\nu(\tau,z_m)}], \tag{A10}$$

where $\nu(\tau,z_m)$ is the number of stopping times between τ and z_m. Hence, $\nu(\tau,z_m)$ follows a Poisson law of parameter $\left(\sum_{m=1}^{M}\theta_m+\lambda_m\right)(z_m-\tau)$. The Laplace transform of $\nu(\tau,z_m)$ is

$$\mathbb{E}\big[e^{\nu(\tau,z_m)}\big]=\exp\left[\left(\sum_{m=1}^{M}\theta_m+\lambda_m\right)(z_m-\tau)\left(e^{-1}-1\right)\right], \tag{A11}$$

which is finite. Next, see that

$$\mathbb{E}\big[e^{\nu(\tau,z_m)}\big]=\mathbb{E}\sum_{k=0}^{+\infty}\frac{\nu(\tau,z_m)^k}{k!}, \tag{A12}$$

which is thus finite. Since $\nu(\tau,z_m)\geq 0$, it follows that the sequence $\left(\frac{\mathbb{E}\left[\nu(\tau,z_m)^k\right]}{k!}\right)_{k\in\mathbb{N}}$ is bounded above.

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

Using a Taylor expansion, we get

$$\mathbb{E}\Big[(1-\varepsilon)^{\nu(\tau,z_m)}\Big]=1-\mathbb{E}[\nu(\tau,z_m)]\varepsilon+\sum_{k\geq 2}\frac{(-1)^k\mathbb{E}\Big[\Pi_{i=0}^{k-1}[\nu(\tau,z_m)-i]\Big]}{k!}\varepsilon^k. \tag{A13}$$

Since $\left(\frac{\mathbb{E}\left[\nu(\tau,z_m)^k\right]}{k!}\right)_{k\in\mathbb{N}}$ is bounded above, all the terms where ε has a degree higher than 2 on the RHS of the previous equation tend to 0 faster than ε. Therefore, for small ε, we have

$$\Pr(\forall\tau<z_m,r^\tau\leq 1-\varepsilon|\omega_\tau,z_m)\approx 1-\mathbb{E}[\nu(\tau,z_m)]\varepsilon. \tag{A14}$$

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', that is, $N_m(\tau')-N_m(\tau)=1$. 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', that is, $z_m=\tau'$. Recall that $N_m^o(\tau)$ is the number of blocks solved by m on the original chain at time τ. Then his payoff under the deviation is

$$v_m^o(\tau,\tau^f)G(M-K^*-1)+v_m^n(\tau,\tau^f)G(K^*+1)+N_m^o(\tau(B_{n(\tau^f)-f}))G(M) \tag{A15}$$

instead of

$$v_m^o(\tau,\tau^f)G(M-K^*)+v_m^n(\tau,\tau^f)G(K^*)+N_m^o(\tau(B_{n(\tau^f)-f}))G(M) \tag{A16}$$

under the equilibrium strategy.[34]

[34] 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 $\tau(B_{n(\tau^f)-f})$.

It follows that there is no profitable deviation if

$$\begin{aligned}\Pr(N_m(\tau')-N_m(\tau)=1)[G(K^*)-G(M-K^*)]\leq&\\ \Pr(z_m=\tau')\quad [v_m^o(\tau,\tau^f)(G(M-K^*)-G(M-K^*-1))&\\ -v_m^n(\tau,\tau^f)(G(K^*+1)-G(K^*))]&\end{aligned}\tag{A17}$$

which is exactly inequality (A5) 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

$$\begin{aligned}\Pr(N_m(\tau')-N_m(\tau)=1)[G(K^*)-G(M-K^*)]\geq&\\ \Pr(z_m=\tau')\quad [v_m^o(\tau,\tau^f)(G(M-K^*+1)-G(M-K^*))&\\ -v_m^n(\tau,\tau^f)(G(K^*)-G(K^*-1))],&\end{aligned}\tag{A18}$$

which is exactly Equation (A6) in Condition 2.

Next, see that at $\tau=\tau^f$, $v_m^n(\tau^f,\tau^f)=0$ for all miners. Inequality (A5) is then written:

$$\begin{aligned}&\frac{\Pr(N_m(\tau')-N_m(\tau^f)=1)}{\Pr(z_m=\tau')}[G(K^*)-G(M-K^*)]\\ &<v_m^o(\tau^f,\tau^f)[G(M-K^*)-G(M-K^*-1)].\end{aligned}\tag{A19}$$

See that $\frac{\Pr(N_m(\tau')-N_m(\tau^f)=1)}{\Pr(z_m=\tau')}=\frac{\theta_m}{\lambda_m}$. To see this, let $x\equiv\sum_{m'\in\mathcal{M}}\lambda_{m'}+\sum_{m'\in\mathcal{M}}\theta_{m'}$ be the cumulative intensity of all liquidity shocks and block solving times. Note that $\Pr(N_m(\tau')-N_m(\tau)=1)=\int_\tau^{+\infty}\theta_m e^{-x(t-\tau)}dt=\frac{\theta_m}{x}$. Similarly, $\Pr(z_m=\tau')=\int_\tau^{+\infty}\lambda_m e^{-x(t-\tau)}dt=\frac{\lambda_m}{x}$.

Hence at $\tau=\tau^f$, (A5) is exactly Inequality (10) in Condition 1. Similarly, Inequality (A6) is then written:

$$\frac{\theta_m}{\lambda_m}[G(K^*)-G(M-K^*)]>v_m^o(\tau^f,\tau^f)[G(M-K^*+1)-G(M-K^*)]\tag{A20}$$

which is exactly Inequality (11) in Condition 1, evaluated at $\tau=\tau_f$.

Furthermore, if miners adhere to the equilibrium strategy, then miners $m\leq K^*$ always mine the new chain so that inequality (11) in Condition 1 at $\tau=\tau_f$ implies that Inequality (A6) in Condition 2 is true at any $\tau>\tau^f$. 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, for $\tau>\tau^f$, Condition 2 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). Following the same logic as in the proof of (b), other deviations can be ruled out. ■

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.[35]

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

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 in what follows when this assumption is used.

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, $p(B_n)$.[36]

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 $\tau(B_n)$.

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 occurrence 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 $B_{n(\tau)+1}$.

- 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

$$N_m^o(\tau(B_n))G(M)+G(M)$$
$$+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt\,|z_m\geq\tau(B_{n(\tau)+1})\right]-\mathcal{L}(\tau(B_{n(\tau)+1})). \quad \text{(A21)}$$

  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.[37] $\mathcal{L}(\tau(B_{n(\tau)+1})$ is the expected loss due to one of m's blocks solved after $\tau(B_{n(\tau)+1})$ becoming orphaned. On the equilibrium path, orphaned blocks occur if and only if a miner observes a block with delay and creates a successful fork.

[36] Since the equilibrium strategies are defined for all states, including those that 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}.

[37] 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.

– 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

$$\begin{aligned}&N_m^o(\tau(p(B_n)))G(M)+2G(M)\\&+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]\\&-\mathcal{L}(\tau(B_{n(\tau)+1})),\end{aligned}\tag{A22}$$

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

It follows that there is no profitable deviation.

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 $N_m^o(z_m)G(M)$ (if there is no fork), or $N_m^o(z_m)(G(M-1)+G(1))=N_m^o(z_m)G(M)$ (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

$$\begin{aligned}&(N_m^o(\tau_{B_n})+1)G(M)+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]\\&-\mathcal{L}(\tau(B_{n(\tau)+1})).\end{aligned}\tag{A23}$$

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[38]

$$\begin{aligned}&\left[N_m^o(\tau(p(B_n))+\mathbb{1}_{\{m=m(B_n)\}}\Pr(B_n=p(B_{n(\tau)+2}))\right]G(M)\\&+\Pr(m=m(B_{n(\tau)+2}))G(M)\\&+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]\\&-\mathcal{L}(\tau(B_{n(\tau)+1})).\end{aligned}\tag{A24}$$

[38] 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.

Indeed, m earns $G(M)$ for all blocks solved on the original chain up to $\tau(p(B_n)$, for B_n if he solved it and it remains on the active chain (if $B_{n(\tau)+2}$ is attached to B_n), and for $B_{n(\tau)+1}$ if it is included in the active chain (if m solves $B_{n(\tau)+2}$).[39]

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 $B_{n(\tau)+1}$, 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

$$\begin{aligned}&[N_m^o(\tau(B_n)) + 1 - \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3}))]G(M)\\ &+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})\right]. \qquad \text{(A25)}\end{aligned}$$

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).

If m deviated by chaining $B_{n(\tau)+1}$ to $p(B_n)$,[40] to earn his reward on $B_{n(\tau)+1}$, m needs to solve $B_{n(\tau)+2}$ so his expected gain is

$$\begin{aligned}&\left[N_m^o(\tau(p(B_n)) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2}))\right] G(M)\\ &+\Pr(m = m(B_{n(\tau)+2}))G(M)\\ &+\mathbb{E}[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt | z_m \geq \tau(B_{n(\tau)+1})]. \qquad \text{(A26)}\end{aligned}$$

As earlier,

$$N_m^o(\tau(B_n)) \geq N_m^o(\tau(p(B_n))) + \mathbb{1}_{\{m=m(B_n)\}} \Pr(B_n = p(B_{n(\tau)+2})). \qquad \text{(A27)}$$

Consequently, there is no profitable deviation if

$$1 - \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})) \geq \Pr(m = m(B_{n(\tau)+2})). \qquad \text{(A28)}$$

That is,

$$1 \geq \Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})), \qquad \text{(A29)}$$

which holds because

$$\begin{aligned}&1 \geq \Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2})) \geq\\ &\Pr(m = m(B_{n(\tau)+2})) + \Pr(m' = m(B_{n(\tau)+2}) = m(B_{n(\tau)+3})). \qquad \text{(A30)}\end{aligned}$$

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

[39] 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.

[40] As earlier, this is the only relevant deviation.

(iii) 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). ■

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

$$\mathbb{1}_{\{m=m(B_{k+1})\}}G(M-1)+\mathbb{1}_{\{m=m(B_n)\}}G(1)+N_m^o(\tau(p(B_n)))G(M), \tag{A31}$$

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

$$\mathbb{1}_{\{m=m(B_{k+1})\}}G(M)+\mathbb{1}_{\{m=m(B_n)\}}G(0)+N_m^o(\tau(p(B_n)))G(M). \tag{A32}$$

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

$$\begin{aligned}&N_m^o(\tau(p(B_n)))G(M)+\mathbb{1}_{\{m=m(B_{k+1})\}}G(M)+\mathbb{1}_{\{m=m(B_n)\}}G(0)+G(0)\\ &+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+1})\right]-\mathcal{L}(\tau(B_{n(\tau)+1})),\end{aligned} \tag{A33}$$

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

$$\begin{aligned}&N_m^o(\tau(p(B_n)))G(M)+\mathbb{1}_{\{m=m(B_{k+1})\}}G(M)+\mathbb{1}_{\{m=m(B_n)\}}G(0)+G(M)\\ &+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+1})\right]-\mathcal{L}(\tau(B_{n(\tau)+1})),\end{aligned} \tag{A34}$$

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 to 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:

$$\begin{aligned}&N_m^o(\tau(p(B_n)))G(M)+G(M)\\&+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]-\mathcal{L}(\tau(B_{n(\tau)+1})).\end{aligned}\tag{A35}$$

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. ■

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 earlier, 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 $N_m^o(\tau_{B_n})G(M)$. If, instead, m followed the equilibrium strategy and chained his block to $p(B_n)$, his payoff is

$$N_m^o(\tau(p(B_n)))[G(M-1)+G(1)]+\mathbb{1}_{\{m=m(B_n)\}}G(M-1).\tag{A36}$$

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 $B_{n(\tau)}$.[41] Since by assumption $G(M-1)+G(1)=G(M)$ and $G(1)=0$, there is no profitable deviation.

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

To analyze 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

$$N_m^o(\tau(B_n))G(M)+2G(M)$$
$$+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})\right]. \qquad \text{(A37)}$$

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

$$N_m^o(\tau_{p(B_n)})G(M)+3G(M)$$
$$+E\left[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})\right]. \qquad \text{(A38)}$$

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

$$N_m^o(\tau_{p(B_n)})G(M)+2G(M)$$
$$+E\left[\int_{\tau(B_{n(\tau)+2})}^{z_m} dN_m(t)G(M)dt|z_m \geq \tau(B_{n(\tau)+2})\right]+S. \qquad \text{(A39)}$$

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.

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.

[41] 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.

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

$$\Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}\left[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})\right]$$
$$+N_m^o(\tau(B_n))G(M)+G(M). \tag{A40}$$

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

$$\Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}\left[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})\right]$$
$$+N_m^o(\tau(p(B_n)))G(M)+2G(M). \tag{A41}$$

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,

$$N_m^o(\tau(p(B_n)))(G(M-1)+G(1))+\mathbb{1}_{\{m=m(B_n)\}}G(M-1)+G(1). \tag{A42}$$

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,

$$N_m^o(\tau_{B_n})G(M)+G(0)+\mathbb{E}\left[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})\right]. \tag{A43}$$

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 blocks solved after $\tau(B_{n(\tau)+2})$.

Since $G(M-1) = G(M)$,[42] gains earned by m on all blocks solved up to $\tau(B_n)$ are the same in the two previous events. Hence m's payoff if he played the equilibrium strategy when he does not solve $B_{n(\tau)+2}$ is:

$$\Pr(z_m > \tau(B_{n(\tau)+2}))\mathbb{E}\left[\int_{\tau_{B_{n(\tau)+2}}}^{z_m} dN_m(t)G(M)dt | z_m > \tau(B_{n(\tau)+2})\right]$$
$$+N_m^o(\tau_{B_n})G(M). \tag{A44}$$

Hence, the net benefit of following the equilibrium strategy rather than deviating when m does not solve $B_{n(\tau)+2}$ is the difference between Equation (A44)

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

and the maximum of Equations (A40) and (A41), that is,

$$-\left\{G(M)+(\mathbb{1}_{\{[m=m(B_{n(\tau)})]\cap[p(B_n(\tau))=p(B_n)]\}}-\mathbb{1}_{\{m=m(B_n)\}})G(M)\right\}. \tag{A45}$$

Overall, m always follows the equilibrium strategy (including when he solved B_n) if and only if

$$\begin{gathered}\Pr[m=m(B_{n(\tau)+2})|m=m(B_{n(\tau)+1})][S-G(M)]\\ >(1-\Pr[m=m(B_{n(\tau)+2})|m=m(B_{n(\tau)+1})])2G(M)\\ \Leftrightarrow S>\frac{G(M)(2-\Pr[m=m(B_{n(\tau)+2})|m=m(B_{n(\tau)+1})])}{\Pr[m=m(B_{n(\tau)+2})|m=m(B_{n(\tau)+1})]}.\end{gathered} \tag{A46}$$

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, which we denote $\gamma(m,\tau)$.

Proof of part (b). 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), and 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

$$G(0)+N_m^o(\tau(B_n))G(M), \tag{A47}$$

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

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

$$G(1)+N_m^o(\tau(p(B_n)))[G(M-1)+G(1)]+\mathbb{1}_{\{m=m(B_n)\}}G(M-1), \tag{A48}$$

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 $B_{n(\tau)}$ becomes orphaned. Therefore, m's expected payoff is

$$\begin{gathered}N_m^o(\tau(B_n))G(M)+G(M)+\mathcal{S}(\tau(B_{n(\tau)+1}))-\mathcal{L}(\tau(B_{n(\tau)+1}))\\ +\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right],\end{gathered} \tag{A49}$$

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's expected payoff is at least equal to

$$\begin{aligned}&N^o_m(\tau_{p(B_n)})G(M)+2G(M)+\mathcal{S}(\tau(B_{n(\tau)+2}))-\mathcal{L}(\tau(B_{n(\tau)+2}))\\&+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+2})\right],\end{aligned}\tag{A50}$$

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), and 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 of Proposition 1, we can restrict attention to the cases where the next event is that m 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 $B_{n(\tau)+1}$.

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

$$\begin{aligned}&(N^o_m(\tau_{B_n})+1)G(M)+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]\\&+\mathcal{S}(\tau(B_{n(\tau)+1}))-\mathcal{L}(\tau(B_{n(\tau)+1})).\end{aligned}\tag{A51}$$

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

$$\begin{aligned}&\big[N^o_m(\tau(p(B_n))+\mathbb{1}_{\{m=m(B_n)\}}\Pr(B_n=p(B_{n(\tau)+2}))\\&+\Pr(m=m(B_{n(\tau)+2}))\big]G(M)\\&+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)G(M)dt|z_m\geq\tau(B_{n(\tau)+1})\right]\\&+\mathcal{S}(\tau(B_{n(\tau)+1}))-\mathcal{L}(\tau(B_{n(\tau)+1})).\end{aligned}\tag{A52}$$

Since $N^o_m(\tau_{B_n})\geq N^o_m(\tau_{p(B_n)})+\mathbb{1}_{\{m=m(B_n)\}}\Pr(B_n=p(B_{n(\tau)+2}))$, 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). ■

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 is 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

$$N_m^0(\tau)(1+b_m(0))G(M)+(1+b_m(0))G(M)$$
$$+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)(1+b_m(0))G(M)dt|z_m \geq \tau(B_{n(\tau)+1})\right] \tag{A53}$$

The first term is m's rewards from blocks 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

$$N_m^0(\tau)(1+b_m(0))G(M)+(1+b_m(C))G(1)$$
$$+\mathbb{E}\left[\int_{\tau(B_{n(\tau)+1})}^{z_m} dN_m(t)(1+b_m(0))G(M)dt|z_m \geq \tau(B_{n(\tau)+1})\right] \tag{A54}$$

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 $C=1$. ■

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 $m > K^*$

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K^*) - (1+b)G(M-K^*)) \leq$$

$$\Pr(z_m = \tau')\left\{\begin{array}{r} (N_m^0(\tau) - N_m^0(\tau^f))(1+b)(G(M-K^*) - G(M-K^*-1)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K^*+1) - G(K^*))) \end{array}\right\}, \qquad \text{(A55)}$$

and for $m \leq K^*$

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K^*) - G(M-K^*)) \geq$$

$$\Pr(z_m = \tau')\left\{\begin{array}{r} (N_m^0(\tau) - N_m^0(\tau^f))(G(M-K^*+1) - G(M-K^*)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K^*) - G(K^*-1))) \end{array}\right\}. \qquad \text{(A56)}$$

Our candidate equilibrium strategy specifies the following:

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

(b) At the hard fork 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 hard fork 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 case 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 hard fork 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', that is, $N_m(\tau') - N_m(\tau) = 1$. 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', that is, $z_m = \tau'$. Then his payoff under the deviation is

$$(N_m^0(\tau) - N_m^0(\tau^f))(1+b)G(M-K^*-1) + (N_m^1(\tau) - N_m^1(\tau^f))G(K^*+1)$$
$$+ N_m^0(\tau^f)(1+b)G(M) \tag{A57}$$

instead of

$$(N_m^0(\tau) - N_m^0(\tau^f))(1+b)G(M-K^*) + (N_m^1(\tau) - N_m^1(\tau^f))G(K^*)$$
$$+ N_m^0(\tau^f)(1+b)G(M) \tag{A58}$$

under the equilibrium strategy.[43] It follows that there is no profitable deviation if

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K^*) - (1+b)G(M-K^*)) \leq$$
$$\Pr(z_m = \tau')\left\{\begin{array}{r}(N_m^0(\tau) - N_m^0(\tau^f))(1+b)(G(M-K^*) - G(M-K^*-1)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K^*+1) - G(K^*)))\end{array}\right\}, \tag{A59}$$

which is exactly Inequality (A55) 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

$$\Pr(N_m(\tau') - N_m(\tau) = 1)(G(K^*) - G(M-K^*)) \geq$$
$$\Pr(z_m = \tau')\left\{\begin{array}{r}(N_m^0(\tau) - N_m^0(\tau^f))(G(M-K^*+1) - G(M-K^*)) \\ -(N_m^1(\tau) - N_m^1(\tau^f))(G(K^*) - G(K^*-1)))\end{array}\right\}, \tag{A60}$$

which is exactly Equation (A56) in Condition 3.

Next, see that at $\tau = \tau^f$, $N_m^C(\tau) = N_m^C(\tau^f)$ for all miners. Inequality (A55) is then written:

$$(1+b)G(M-K^*) \geq G(K^*) \Leftrightarrow b \geq \frac{G(K^*)}{G(M-K^*)} - 1. \tag{A61}$$

Similarly, Inequality (A56) is then written:

$$G(K^*) \geq G(M-K^*) \Leftrightarrow K^* \geq \frac{M}{2}. \tag{A62}$$

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 (A56) 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 (A55) 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. ■

[43] 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. By construction, $D \geq 1$ since $\frac{1}{D}$ is the probability to solve the problem at each trial. If the total computing capacity $\sum_{i \in \mathcal{M}} h_i$ was lower than $1/X$, it would not be feasible to have one block solved every X units of time. To ensure $D \geq 1$ with Equation (3), we impose the technical constraint that

$$\sum_{i \in \mathcal{M}} h_i \geq 1/X. \tag{A63}$$

Substituting Equation (4), each miner's program in Equation (16) is written

$$\max_{h_m} \frac{\frac{h_m}{\sum_{i \in \mathcal{M}} h_i}}{\lambda_m X} G(M) - \frac{c_m h_m}{\lambda_m}, \tag{A64}$$

which yields the following first-order condition:

$$\frac{\left(\sum_{i \in \mathcal{M}} h_i\right) - h_m}{\left(\sum_{i \in \mathcal{M}} h_i\right)^2} \frac{G(M)}{X} = c_m. \tag{A65}$$

See first that miner m's participation constraint is

$$\sum_{i \in \mathcal{M}} h_i \leq \frac{G(M)}{c_m X}. \tag{A66}$$

Evaluated at equilibrium, Equation (A65) yields

$$\left(\sum_{i \in \mathcal{M}} h_i^*\right) - h_m^* = \frac{X}{G(M)} c_m \left(\sum_{i \in \mathcal{M}} h_i^*\right)^2 \tag{A67}$$

$$\left(\sum_{i \in \mathcal{M}} h_i^*\right) - c_m \frac{X}{G(M)} \left(\sum_{i \in \mathcal{M}} h_i^*\right)^2 = h_m^*. \tag{A68}$$

Summing over miners

$$M\left(\sum_{i \in \mathcal{M}} h_i^*\right) - \left(\sum_{i \in \mathcal{M}} c_i\right) \frac{X}{G(M)} \left(\sum_{i \in \mathcal{M}} h_i^*\right)^2 = \left(\sum_{i \in \mathcal{M}} h_{i\in}^*\right) \tag{A69}$$

$$\Leftrightarrow (M-1)\left(\sum_{i \in \mathcal{M}} h_i^*\right) = \left(\sum_{i \in \mathcal{M}} c_i\right) \frac{X}{G(M)} \left(\sum_{i \in \mathcal{M}} h_i^*\right)^2 \tag{A70}$$

$$\Leftrightarrow \frac{(M-1)}{\sum_{i \in \mathcal{M}} c_i} \frac{G(M)}{X} = \sum_{i \in \mathcal{M}} h_m^* \tag{A71}$$

$$\Leftrightarrow \sum_{i \in \mathcal{M}} h_i^* = \frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}, \tag{A72}$$

which is exactly Equation (18) in Proposition 9. Into participation constraint (A66), this yields $(M-1)c_m \leq \sum_{i \in \mathcal{M}} c_i$. Next, substituting Equation (18) into (A68), one gets

$$\begin{aligned} h_m^* &= \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right) - c_m \frac{X}{G(M)} \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right)^2. \\ &= \left(\frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right) \left(1 - c_m \frac{X}{G(M)} \frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right) \\ &= \frac{G(M)}{X} \frac{M-1}{\sum_{i \in \mathcal{M}} c_i} \left(1 - c_m \frac{M-1}{\sum_{i \in \mathcal{M}} c_i}\right), \end{aligned} \tag{A73}$$

which is exactly Equation (17) in Proposition 9. Last, replacing h_m^* into miner m's objective function, one obtains that his equilibrium profit is

$$\frac{G(M)}{\lambda_m X}\left(1-\frac{(M-1)c_m}{\sum_{i\in\mathcal{M}} c_i}\right)^2. \tag{A74}$$

■

References

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

Bonneau, J., A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten 2015. SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. *2015 IEEE Symposium on Security and Privacy*, IEEE, 104–21.

Catalini, C., and J. Gans. 2016. Some simple economics of the blockchain. NBER Working Paper No. 22952.

Carlsten, M., H. Kalodner, S. M. Weinberg, and A. Narayanan. 2016. On the instability of Bitcoin without the block reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 154–67.

Chiu, J., and T. Koeppl. 2019. Blockchain-based settlement for asset trading. *Review of Financial Studies*, 32:1716–53.

Cole, H., and T. Kehoe. 2000. Self-fulfilling debt crises. *Review of Economic Studies* 67(1):91–116.

Cong, L. W., and Z. He. 2019. Blockchain disruption and smart contracts. *Review of Financial Studies*, 32:1754–97.

Crawford, V. 1998. A survey of experiments on communication via cheap talk. *Journal of Economic Theory* 78(2):286–98.

Dimitri, N. 2017. Bitcoin mining as a contest. *Ledger* 2. http://doi.org/10.5195/ledger.2017.96.

Duggan, J. 2012. Noisy stochastic games. *Econometrica* 80(5):2017–45.

Dwork, C., and M. Naor. 1993. Pricing via processing or combatting junk mail. In E. F. Brickell, ed., *Advances in Cryptology, CRYPTO'92, Lecture Notes in Computer Science 740*, 139–47. Berlin: Springer.

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. Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In N. Christin and R. Safavi-Naini, eds., *Financial Cryptography and Data Security: FC 2014, Lecture Notes in Computer Science 8437*, 436–54. Berlin: Springer.

Garay J. A., A. Kiayias, and N. Leonardos. 2015. The Bitcoin backbone protocol: analysis and applications. In E. Oswald and M. Fischlin, eds., *Advances in Cryptology, EUROCRYPT 2015, Lecture Notes in Computer Science 9057*, 281–310. Berlin: Springer.

Glode, V., R. Green, and R. Lowery. 2012. Financial expertise as an arms race. *Journal of Finance* 67(5):1723–59.

Harvey, C. R. 2016. Cryptofinance. Working Paper.

Khapko M., and M. Zoican. 2017. Smart settlement. 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* Vol. 2013.

Lamport L., R. Shostak, and M. Pease. 1982. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4(3):382–401.

Lerner, J., and J. Tirole. 2002. Some simple economics of open source. *Journal of Industrial Economics* 50(2):197–234.

Mailath, G. J., and L. Samuelson. 2006. *Repeated games and reputations: Long run relationships*. Oxford: Oxford University Press.

Malinova, K., and A. Park. 2017. Market design with blockchain technology. Working Paper.

Miller, A., and J. J. LaViola Jr. 2014. Anonymous byzantine consensus from moderately-hard puzzles: A model for Bitcoin. https://socrates1024.s3.amazonaws.com/consensus.pdf.

Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf.

Narayanan, A. 28 July 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.

Pass R., L. Seeman, and A. Shelat. 2017. Analysis of the blockchain protocol in asynchronous networks. In J.-S. Coron and J. Buus Nielsen, eds., *Advances in Cryptology- EUROCRYPT 2017, Lecture Notes in Computer Science 10211*, 643–73. Cham: Springer.

Pease, M., R. Shostak, and L. Lamport. 1980. Reaching agreement in the presence of faults. *Journal of the ACM* 27(2):228–34.

Raskin, M., and D. Yermack. 2016. Digital currencies, decentralized ledgers, and the future of central banking. NBER Working Paper No. 22238.

Saleh, F. 2017. Blockchain without waste: Proof-of-stake. Working Paper.

Teutsch, J., S. Jain, and P. Saxena. 2016. When cryptocurrencies mine their own business. In J. Grossklags and B. Preneel, eds., *Financial Cryptography and Data Security, FC 2016, Lecture Notes in Computer Science 9603*, 499–514. Berlin: Springer.

Tirole, J. 1982. On the possibility of speculation under rational expectations. *Econometrica* 50(5):1163–81.

Tirole, J. 1985. Asset bubbles and overlapping generations. *Econometrica* 53(6):1499–1528.

Yermack, D. 2017. Corporate governance and blockchains. *Review of Finance* 21(1):7–31.