WORKING PAPERS

N° TSE- 817

May 2017

"The blockchain folk theorem"

Bruno Biais, Christophe Bisière, Matthieu Bouvard
and Catherine Casamatta

INSTITUT D'ECONOMIE INDUSTRIELLE

Toulouse School of Economics

The blockchain folk theorem*

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

May 22, 2017

Preliminary

Abstract

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

*We thank B. Gobillard, C. Harvey, J. Hoerner, A. Kirilenko, T. Mariotti, S. Villeneuve, the members of the TSE Blockchain working group, participants in the Inquire Conference in Liverpool, 2017, as well as an anonymous referee for helpful comments. Financial support from the FBF-IDEI Chair on Investment banking and financial markets value chain is gratefully acknowledged. This research also benefited from the support of the Europlace Institute of Finance.

[†]Toulouse School of Economics, CNRS (CRM-IAE)
[‡]Toulouse School of Economics, Université Toulouse Capitole (CRM-IAE)
[§]Desautels Faculty of Management, McGill University
[¶]Toulouse School of Economics, Université Toulouse Capitole (CRM-IAE)

1 Introduction

Blockchains are decentralised protocols for recording transactions and asset ownership. The blockchain design was the main innovation underlying the digital currency network Bitcoin (Nakamoto, 2008), but its potential benefits in terms of cost-efficiency, speed and security, for a variety of assets and contracts, have attracted interest from a broad range of institutions and businesses.[1] Blockchain experiments, and in some cases limited deployments, have been conducted by the Australian Stock Exchange, the Nasdaq, BHP Billiton and major banks around the globe. As blockchains are being embedded into major transaction platforms, we propose to investigate the stability of the protocol: how efficient is a blockchain at building a stable consensus among participants about the history of past transactions? This question is particularly relevant when blockchains are public, that is when participants are anonymous and there is no formal authority to coordinate their behaviour in last resort.[2] We take a game-theoretic approach that captures the key features of a blockchain design and allows to pin down the tradeoffs faced by the key players (the "miners") in the blockchain's decentralised certification process.

Nakamoto (2008) (Section 5) gives the following description of the blockchain's functioning.

> "The steps to run the network are as follows:
>
> 1. New transactions are broadcast to all nodes.
> 2. Each node collects new transactions into a block.
> 3. Each node works on finding a difficult proof-of-work for its block.
> 4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
> 5. Nodes accept the block only if all transactions in it are valid and not already spent.

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

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

6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash."

The nodes conducting the above mentioned tasks are called "miners", as they "mine" to solve proof-of-work problems,[3] and get rewarded for this in bitcoins. When mining, a miner sets a computer capacity that performs trials to find a hash value lower than a given threshold. Each trial is independent: past failures do not affect the probability of success of a future trial. Once a trial is successful, the winning miner sends the block with the solution to other participants. If participants accept this block as the new consensus, they take it as the parent of the new block they start mining. In that case (unless the consensus is altered), the miner who solved the block gets a reward.[4] This process is illustrated in Figure 1.

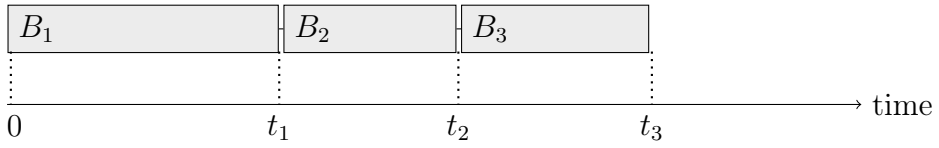

Figure 1: The Blockchain

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

Ideally, there is only one chain, to which all miners attach their blocks. One of the major questions about blockchain is whether such an outcome will arise. The alternative outcome is one in which miners do not all attach their block to the same chain. Suppose, e.g., that the last block solved is B_n, but miner m chains his next block to the parent block of B_n, i.e., B_{n-1}. This starts a fork, as illustrated in Figure 2. If some miners follow m,

[3]The problem to be solved by the miners is a purely numerical problem, completely unrelated to the economic nature of the transactions in the block. Once found, the solution to this problem is easy to verify.

[4]This includes rewards given by the blockchain system plus transaction fees which the originators of trade can choose to offer for the validation of their transactions.

while others continue to attach their blocks to the original chain, there are competing versions of the ledger. This reduces the credibility and reliability of the blockchain, especially if the fork is persistent. Even if, eventually, all miners agree to attach their blocks to the same chain, the occurrence of the fork is not innocuous. The blocks in the chain eventually abandoned are orphaned. They have been mined in vain, and the corresponding computing power and energy have been wasted. Moreover, the transactions recorded in the orphaned blocks may be called in question.

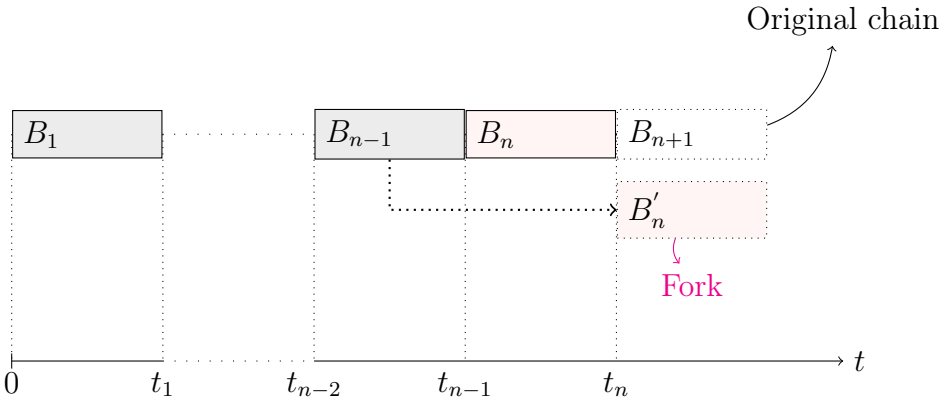

Figure 2: A fork

Blockchain networks did experience major forks in the past. One of the most significant was the fork occurred on Bitcoin in March 2013: Due to a bug in a software upgrade, two competing branches started. It took more than 8 hours for miners to identify the fork and abandon one of the branches. Another example is the July 2016 fork on Ethereum, the major smart contract network. Following the hack of TheDAO, a large venture capital fund operating through smart contracts, members of the Ethereum community suggested to roll back the blockchain in order to cancel the transactions that diverted the fund's money. Other members defended the principle that the history of the ledger should not be altered in any way, for the sake of the network's credibility. Ethereum eventually split in two branches that still exist today, giving rise to two different cryptocurrencies. The possibility of major forks is still lurking. The Bitcoin community is currently divided on which technical solution to adopt to address the limitation of the network

transaction throughput.[5] Two main solutions, Segregated Witness (SegWit) and Bitcoin Unlimited (BU), are supported by different Bitcoin community members, with the threat of some to fork in an attempt to impose their preferred solution. As of May 2017, it is not clear which solution will be adopted, nor whether it will lead to a fork.

How do forks happen? The above coordination issues, which can arise following a technological change or an unpredictable event (like the hacking of TheDAO) have been overlooked and it is a contribution of the present paper to undescore and analyse them. Coming from a different angle, an often mentioned potential cause of forks is "double-spending." Suppose miner m from the example above buys an object from some party Y and the transfer of m's bitcoins to Y is recorded in B_n. This could give an incentive for miner m to mine from B_{n-1}, trying to attract miners to his chain, to orphan B_n and void the transfer of his bitcoins to Y. m would then be able to spend his bitcoins again, i.e., would "double spend."

Non-instantaneous dissemination of information through the network is another potential reason why forks, i.e., competing versions of the ledger, could arise. Nakamoto (2008) identified that problem and suggested it would be solved if miners always chained their blocks to the longest chain:

> "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."

In the present paper, we abstract from these two problems, assuming miners do not attempt to double spend and also that information is instantaneously disseminated in the network. In this frictionless world, it is commonly argued, in particular by the blockchain community, that blockchains should give rise to a single and stable consensus, and thus offer a reliable way

[5]Precisely, the protocol sets the maximum size of a block of transactions to one megabyte, which slows down the speed of transactions validation and hinders the development of the network itself.

to record transactions and ownership. We examine the validity of that "folk theorem" and analyse how consensus can emerge from miners' interactions.

To do so, we rely on a theoretical model, capturing the key features of the blockchain technology. We model the blockchain as a stochastic game, we analyse miners' best responses and beliefs, and we characterise the properties of the corresponding equilibria. Explicitly writing down the blockchain as a game, and explicitly writing down the action space, states, beliefs and strategies of the miners, is necessary to pin down precisely the economic forces at play in that environment, the tradeoffs faced by the miners, and the mechanisms pushing towards stability or instability of the distributed ledger.

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

First, because the value of the rewards for mining a block in a given blockchain depends on the credibility of that chain and correspondingly on the number of miners active on that chain, the blockchain game is a coordination game: If I anticipate all the others to mine a given chain, this increases my incentives to mine that chain. As often in coordination games, there can be multiple equilibria and instability. We show that there exists a Markov perfect equilibrium involving a single chain and in which the longest chain rule (hereafter LCR) suggested by Nakamoto (2008) holds (Proposition 1). In that equilibrium, miners do not want to deviate because they rationally anticipate that if they did, the other miners would not follow them, so that the blocks they would solve on forks out of the equilibrium path would carry no reward. On the other hand, we also show that the same coordination effects can give rise to Markov perfect equilibria involving forks (Proposition 2). In such equilibria, a sunspot variable realises, suggesting miners to fork. No miner wants to deviate from forking for the same reason as above: each miner rationally anticipates that any block solved out of the equilibrium path will not be accepted by the others, and will have no value. The possibility of these forks creates uncertainty about the allocation of property rights and undermines the stability and reliability of the distributed ledger.

Second, we identify another force which we refer to as "vested interest:" When a miner solves blocks on a chain, he is rewarded with units of the cryptocurrency associated with that chain. As long as the miner has not sold this cryptocurrency, he has a vested interest in that chain becoming the consensus. Now, miners cannot immediately sell the cryptocurrency they receive as reward for the blocks they hold. They must keep them until

sufficiently many blocks have been attached to that chain (this is the so-called "k-blocks rule"). This can lead miners working on different chains to continue to do so, in order to beat the competing chain. This can contribute to the emergence of persistent forks (Proposition 3).

While the persistent forks result hinges on the strategic behaviour of miners, who anticipate their strategy will affect the value of their rewards, the emergence of forks, making the previously longest chain orphan, relies only on coordination effects, and would also arise in a competitive environment.

In the last section of the current paper, we discuss how integrating frictions in our model, such as attempts to double-spend or non-instantaneous dissemination of information, could provide further insights into the blockchain's stability. We also suggest to endogenise the computing capacity that each miner installs on the network. In the Bitcoin protocol, total computing capacity determines the difficulty to solve blocks. Because each miner does not take into account the impact of his computing capacity on the difficulty of the cryptographic problem faced by other miners, we conjecture that an arms race can occur, leading to over-investment in computing power (not unlike the over-investment in financial expertise noted by Glode, Green and Lowery (2012)). This provides a roadmap for our future research.

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

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

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

The remainder of the paper is organised as follows. The next section presents the model. Section 3 develops our equilibrium analysis and contains our main results. Future extensions of the model are provided in Section 4, and Section 5 concludes. All proofs are in the Appendix.

2 Model setup

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

Miners and pools: There are $M \geq 2$ risk-neutral miners, indexed by $m \in \mathcal{M} = \{1, ...M\}$. While, in our model, we refer to each m as a miner, in practice miners work in pools, which coordinate the efforts of their miners, in particular as regards which blocks they mine. 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."

Thus, we can think of M as the number of pools. Figure 3 presents the distribution of computing power of the pools operating on Bitcoin in April 2017: 14 mining pools represented about 93% of the total hash capacity.

Thus, a reasonable order of magnitude for M is around 15. Because the number of pools is finite, it is appropriate to take a game theoretic approach, in which each of the M players behaves strategically. In the discussion below, we will highlight which results rely on this strategic behaviour and which would also obtain in a competitive environment.

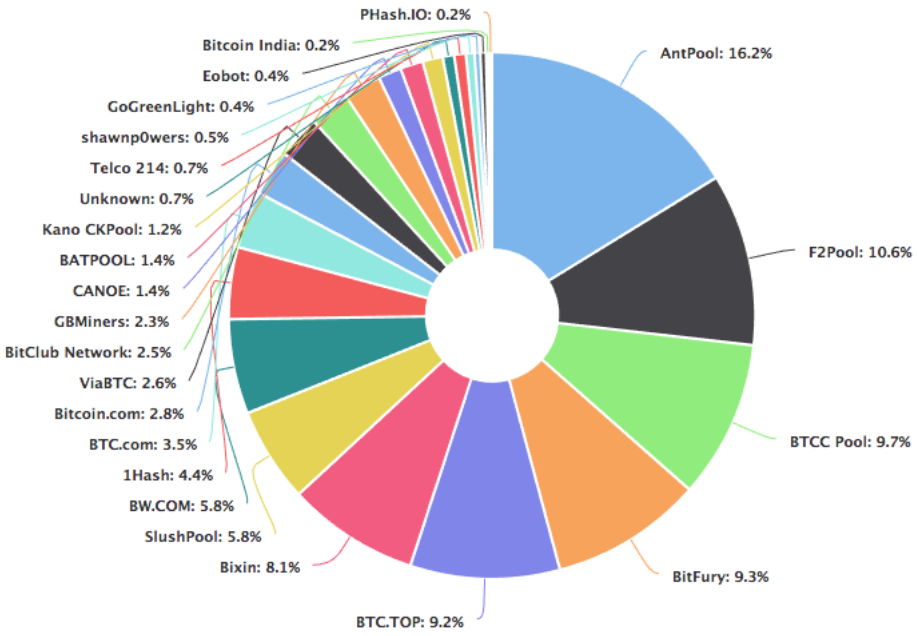

Figure 3: Hashrate distribution of Bitcoin mining pools on April 20, 2017. Source: blockchain.info. AntPool servers are located in China. The other three main pools have servers in China, Japan and the US.

Mining technology: There is a continuous flow of transactions sent for confirmation by end-users.[6] For the moment, for simplicity, we assume all miners perfectly and instantaneously observe this flow, which they include in the blocks they mine. The time it takes a miner to solve his block depends on the difficulty of the cryptographic problem and the miner's computing

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

power. The difficulty is set by the blockchain protocol to keep the average duration between two blocks close to a target (10 minutes on Bitcoin and between 10 and 20 seconds on Ethereum). Correspondingly, as long as the total computing power in the network does not change, the difficulty of the cryptographic problem does not change. If the total computing power increases (e.g., due to the entry of new miners and new pools), the difficulty is scaled up so that average duration between two blocks remains equal to the desired level. Thus, on Bitcoin every 2,016 blocks, i.e., approximately every 2 weeks, the difficulty is rescaled to ensure that the average time between blocks remains at 10 minutes. In the present paper we consider a stationary environment, in which the number of miners and the difficulty of the task are constant.

As explained in Nakamoto (2008), the time it takes miner m to solve a block problem is exponential with parameter θ_m. For a given computational power, the greater the difficulty, the lower the intensity θ_m. A key property of the exponential distribution is that it is memoryless: at each point in time, the distribution of the waiting time until the miner finds a solution is independent from how long the miner has been working on the problem.[7] An important feature of the blockchain is that this waiting time is also independent of which block m is mining, and also from the blocks the other miners are mining. These properties have important strategic consequences. For example, suppose m has been mining block B, and another miner solves a block (possibly B or possibly another one). At this point, the duration until the next time at which m solves a block is independent of whether m continues to mine B or any other block. We denote by N_m the Poisson process jumping each time miner m solves a block. Thus, the number of blocks solved by miner m between time 0 and time t, is

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

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

[7]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.

economic mechanism we analyse below.

We assume that at time z_m, exponentially distributed, with parameter λ_m, miner m is hit by a liquidity shock. At time z_m the miner must leave the game and sell the cryptocurrencies he earned previously to a new miner who also inherits his beliefs and preferences.[8] Thus, exits are compensated by entries and the environment is stationary.

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

As the game unfolds, a tree of blocks develops. In the above example, once B_2 is solved, the tree has three vertices: B_0, B_1 and B_2. If miners continue the first chain, by attaching B_2 to B_1 the two edges (or branches) of the tree are (B_0, B_1) and (B_1, B_2). In contrast, if miners start a fork, the two edges are (B_0, B_1) and (B_0, B_2). At each vertex B_k, the tree also 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.

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

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

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

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

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

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

Stopping times: We assume miners make decisions at different points in time, corresponding to a sequence of stopping times. Whenever a block is solved or a miner is hit by a liquidity shock, all miners make a decision. Miners can also make a decision, after a time interval of length Δ, if no block is solved and no liquidity shock has occurred during that interval. Δ can be arbitrarily small to approximate a continuous time environment.[9] Thus, the sequence of stopping times at which miners make decisions is $\mathcal{T} = \{0, ...\tau_j, \tau_{j+1}, ...\}$ where the next stopping time after τ_j, τ_{j+1}, is equal to $\tau_{j+1} = \min[\tau_j + \Delta, \tau^l(\tau_j), \tau^b(\tau_j)]$, $\tau^l(\tau_j)$ being the first time a liquidity shock occurs after τ_j and $\tau^b(\tau_j)$ the first time a block is solved after τ_j.

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

Payoffs: When miner m solves a block in a given chain, he receives a reward, included in the block he mined, and expressed in the cryptocurrency corresponding to that chain.[10] We assume that miner m consumes the rewards he earned throughout the game at time z_m. That is, we assume that, until time z_m, the miner keeps the units of cryptocurrency he earned. In

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

[10] For example, when a miner solves a hash problem on Bitcoin or Ethereum, he is rewarded in bitcoins (BTC) or ethers (ETH). On Bitcoin, miners receive in 2017 12.5 BTC for each block, on Ethereum they receive 5 ETH per block. For simplicity, we neglect further fees offered by final traders to reward the certification of their transactions, since we do not model explicitly transactions.

practice, miners do not sell their reward immediately after they have earned it. In particular, the so called "k-blocks rule" implies that the cryptocurrency obtained by m for solving a block will be accepted by others only after sufficiently many blocks have been chained to that block.

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

More formally, we assume that the payoff for miner m from solving B is an increasing function, $G(.)$, of the number of miners active at time z_m in the chain including B. For example, suppose there are two active chains at time z_m. If there are K miners active in the chain including B, and $M-K$ in the other, the payoffs from solving blocks are the following: The miner who solved block B, which we denote by $m(B)$, earns $G(K)$ for block B. A miner who solved a block in the other chain earns $G(M-K)$ for that block. If a miner solved a block that belongs to both chains, he earns $G(M-K)+G(K)$.[11] We assume that $G(0)=G(1)=0$ since, when there is only one or no miner on a chain, the associated cryptocurrency has no value. Finally, we assume that when several chains compete, the total value of a

[11] One must also specify what happens if z_m occurs just after a fork starts, after a block B_n has just been solved. The probability of this event is very small, and in practice it is not a very relevant consideration, but, for completeness, we need to specify the value of the reward earned by $m(B_n)$ when K miners chain the block they currently mine to B_n, while $M-K$ chain their block to B_{n-1}. Suppose there was a single chain up to and including B_n. Three alternative hypotheses are possible. First, one could posit that the not yet realised fork does not reduce the credibility of the current chain. In that case, $m(B_n)$ earns $G(M)$ for B_n. Second, one could posit that, irrespective of how many miners fork, the attempt to fork reduces the overall credibility of the chain, reducing the reward for B_n to some arbitrary $g < G(M)$. Third, one could posit that the reward for B_n is worth $G(K)$. We will highlight in the proofs the extent to which these alternatives affect our construction.

unit of cryptocurrency that belongs to the competing chains is weakly lower than if it belonged to a single chain that was the consensus of all miners. To ensure this we assume that $G(M-K)+G(K) \leq G(M), \forall K$.

Our assumption that the value of the virtual currency is reduced by forks is illustrated by Figure 4, which plots the decline in bitcoin value during the March 2013 fork. The first vertical line indicates the time (around 22:00) at which miners started working on two different chains. Chats between miners realising there was a fork, started around 23:30.[12] At 1:30 am, a message posted on Bitcointalk asked miners to stop mining one the two branches of the chain (the 0.8 branch). The second vertical line (approximately at 6:20) indicates the time at which the 0.7 branch caught up the 0.8 branch. By 7:30, miners had stopped mining the 0.8 branch, which became orphaned, so that the fork was no longer active. The figure illustrates that, when the market realised that miners worked on different branches this triggered a 25% drop in the value of the virtual currency (from around 48 at 1:00 am to around 36 at 3:00).

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

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

- Second, ω_τ includes the number of miners active on branches stemming from each of the previously solved blocks:[13] $A^\tau = (A^\tau(B_1), ..A^\tau(B_k), ...A^\tau(B_n))$, where $A^\tau(B_k)$ is the number of miners mining at time τ a block directly chained to B_k, determines the value of each miner's reward if he's hit by a liquidity shock.

- Finally, as in Duggan (2012) or Cole and Kehoe (2000), to enable players to coordinate their actions using a public randomisation device, we

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

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

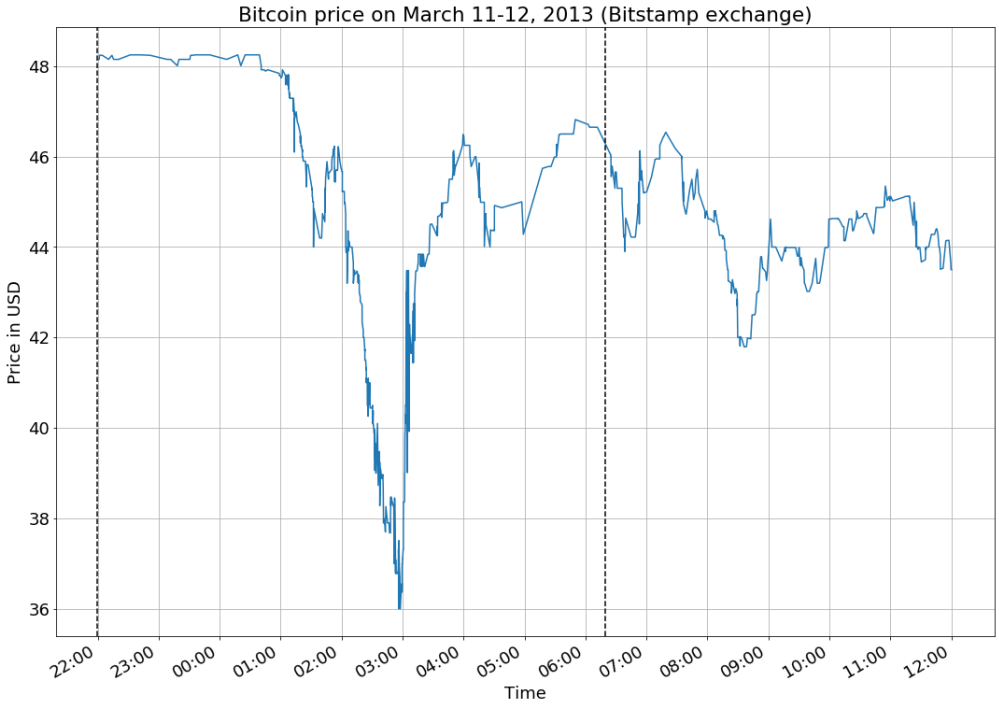

Figure 4: Bitcoin price during the March 2013 fork.
The graph plots individual transaction prices obtained from a major bitcoin exchange platform, Bitstamp, during the March 11-12, 2013 fork. The first dotted vertical line represents the time at which the fork started, and the second dotted vertical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

> assume that at each time $\tau \in \mathcal{T}$, the realisation of a sunspot random variable r^τ is observed by all, and we include it in the state. r^τ is uniformly distributed on $[0, 1]$ and i.i.d. over time.

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

Strategies: Miner m chooses his strategy to maximise his expected payoff at time z_m. At each time $\tau \in \mathcal{T}$, miners observe the whole history of the game, that is, the state ω_τ, as well as, e.g., the exact timing of blocks resolution and the previous mining choices. In the spirit of Markov perfection,

we only consider strategies that are measurable with respect to ω_τ.[14] A pure strategy for miner m is a function σ_m^τ mapping each possible state of the blockchain $\omega_\tau \in \Omega$, into an element of the action space B^τ. We denote the strategy of miner m throughout the entire history of the game by σ_m and the profile of strategies for the M miners by $\sigma = \{\sigma_m\}_{m\in\mathcal{M}}$. σ, combined with the random variables $\{z\}_{m\in\mathcal{M}}$ and $\{N_m\}_{m\in\mathcal{M}}$, yield the transition probabilities from one state of the blockchain to the next.

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

3 Equilibrium analysis

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

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

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

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

Does there exist a natural strategy enabling miners to achieve this maximum expected payoff? The definition of $\mathcal{G}_m^{\max}$ implies that, to obtain the maximum expected payoff, all miners should be on the same chain, when any

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

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 in which, on the equilibrium path there is a single chain and all miners follow the LCR, thus obtaining their maximum expected payoff,* $E[\mathcal{G}_m^{\max}]$.

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

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

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

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

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

More precisely, set a threshold ε, which can be arbitrarily small, consider the first time, τ^f, at which the sunspot variable is above $1-\varepsilon$ and denote by $B_{n(\tau^f)}$ the last block in the chain at that time ($n(\tau)$ denotes the index of the last block solved by τ). In the sunspot equilibrium of our next proposition, at τ^f all participants fork and mine a new block whose parent is $B_{n(\tau^f)-f}$. This fork becomes the only active chain. Since it does not include blocks $B_{n(\tau^f)-f+1}$ to $B_{n(\tau^f)}$, miners do not earn any reward for these blocks.[15] We now state our next proposition:

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

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

The March 12, 2013 Bitcoin fork illustrates the strength of coordination issues in shaping miners' strategies. On March 11 some miners upgraded to a new version of the software, referred to as 0.8. There turned out to be a bug so that the miners operating in the 0.7 version rejected as invalid one block solved by the 0.8 miners (and consequently the subsequent ones). 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. After a while participants became aware that a fork had occurred and had to decide on which branch to coordinate. Arvind Narayanan (2015) reports the following discussion, among miners and developers, from the log of the #bitcoin-dev IRC channel:

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

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

As illustrated by the above quoted discussions, miners faced a dilemma. Should they follow the longest chain rule and continue mining the 0.8 chain which had attracted the majority of the computing power? Or should they fork from it, reverting to a different version of the blockchain? The above discussion shows that the overarching concern of the miners was that they wanted to follow the consensus. BTC Guild, which was one of the largest pools at the time, eventually 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. Consequently more than 24 blocks, solved on the 0.8 chain, became orphaned, and their miners (including BTC Guild) lost the corresponding rewards. Commenting on this situation, Narayanan (2015) wrote:

> "One way to look at this is that BTC Guild sacrificed revenues for the good of the network. But these actions can also be justified from a revenue-maximising perspective. If the BTC Guild operator believed that the 0.7 branch would win anyway (perhaps the developers would be able to convince another large pool operator), then moving first is relatively best, since delaying would only take BTC Guild further down the doomed branch."

This illustrates the behaviour of miners in Proposition 2: if one miner expects all the others to fork, then he is better off following them. Similarly to the 0.7 chain in the 2013 Bitcoin fork, in Proposition 2, the fork stemming from $B_{n(\tau)-f}$ becomes the only active chain. Since it does not include blocks $B_{n(\tau)-f+1}$ to $B_{n(\tau)}$, the miners who solved these blocks lose their rewards. Consequently, these miners earn less than $\mathcal{G}_m^{\max}$, while the other miners do not earn more than $\mathcal{G}_m^{\max}$. Thus the forking equilibrium in Proposition 2 is Pareto dominated by the single chain equilibrium in Proposition 1.

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

While in the previous proposition, in spite of forking, there was eventually a single chain, we now show that forking can lead to the persistent coexistence of different branches. The Ethereum network offers an example of a persistent fork. In response to the TheDAO hacking, on July 20, 2016 80% of the nodes moved to a new, forked chain, that kept the name Ethereum. It was believed that the remaining 20% would follow. Instead, the initial blockchain continued to be mined and took the name Ethereum Classic, which gave rise to a new currency, denoted ETC. Today, two networks coexist: As of may 2017, Ethereum Classic represented about 10% of the hash capacity of Ethereum, and the price of ETC was about 10% of the ETH price.

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

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

and a new chain, including the blocks linked by

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

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

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

In our model miners hold their rewards until z_m and therefore have vested interests. In practice, miners cannot sell their rewards immediately after solving blocks, due to the k-blocks rule. Our model takes a simplified view of this situation by assuming that the vesting period lasts until z_m. Our next result illustrates the consequences of vested interests. To state that result, rank the miners by their vested interest in the original chain at time τ^f as follows

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

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

Consider the following condition.

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

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

and for $m > K$

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

while for $m \leq K$

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

The assumption that for any M and any $K < M$, $G(K) + G(M-K) = G(M)$, simplifies the presentation of Condition 1. However, Proposition 3 below also holds in the more general case where $G(K) + G(M-K) \leq G(M)$.[16]

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

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

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

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

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

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

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

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

of Proposition 3.

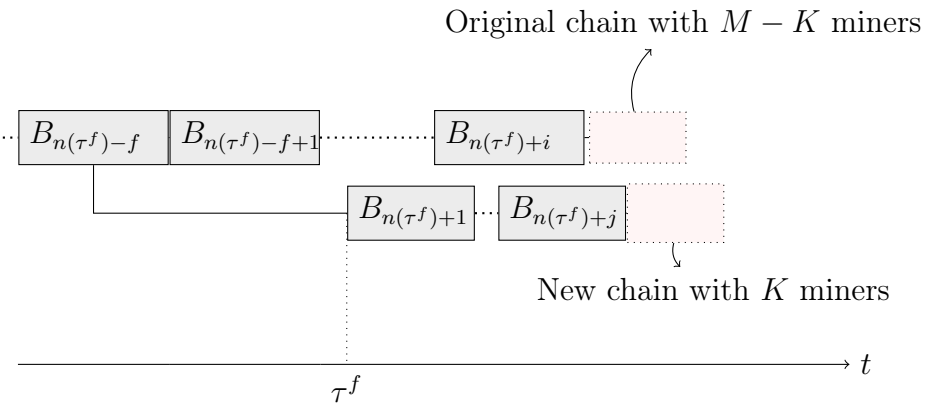

Figure 5: Equilibrium of Proposition 3

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

Finally note that the equilibrium outcome in Proposition 3 is Pareto dominated by that in Proposition 1. Again, forking reduces the total gains of the miners, and yet it can arise in equilibrium.

4 Extensions (work-in-progress)

So far we have considered the case in which i) there are no frictions, and ii) computing capacity is given and constant. In future research, we will work on relaxing some or all of these assumptions and examine to what extent the economic mechanisms we already identified are still at play and what new effects arise. We will also endeavour to distill the implications of our theoretical analysis.

4.1 Frictions

To introduce frictions in our modeling framework, one possible approach would be to consider double spending. This was an important potential issue outlined in Nakamoto (2008), and hopefully could be tackled in the context of our modeling framework. Suppose miner m transferred S bitcoins to another player and this transaction has just been validated in block B_n. Could it be optimal, on the equilibrium path, for miner m to try to start a fork, from B_{n-1}, to recover his S bitcoins and spend them again? Denote by B'_n the initial block on that fork, which m would try to mine. Suppose all other miners always follow the LCR. Then, if m mines B'_n and B'_{n+1} before the others solve any block, the fork stemming from B'_n becomes the longest chain, and hence the consensus. It would be interesting to study if that can occur on the equilibrium path. We conjecture that S could play a role similar to that of vested interests in Proposition 3 above. The equilibrium reaction of the other miners to m's attempt would depend on their gains when the fork succeeds and when it fails, which one would need to specify to analyse that problem.

Another way to introduce frictions would be to consider delays in the dissemination of information through the network. Such delays could induce short term forks. As mentioned above, Nakamoto (2008) considered that possibility and conjectured that miners would follow the LCR and that this would resolve short term forks. It would be interesting, in an extension of our model, to study if this is an equilibrium and if it is the only one. In general, our conjecture is that, with frictions also, coordination effects would arise, which could induce multiplicity of equilibria.

4.2 Computing capacity

In the preliminary analysis above, we took the computing capacity of each miner, and correspondingly the intensity of his block solving process (θ_m), as given. It could be interesting to endogenise it. One could assume that, initially, each miner m spends cost $c(\theta_m)$ to obtain the computing capacity giving intensity θ_m. Each miner's optimality condition would equalise the marginal gain and the marginal cost of additional capacity. In the aggregate, the sum of all miners' investments would determine the total computing capacity. Because of the Bitcoin protocol, that computing capacity would, in

turn, determine the difficulty of the proof-of-work.[17] In this context, the individual decisions of the miners exert negative externalities on their competitors: When m increases his computing power, this raises the difficulty for all the other miners. Thus, we conjecture that an arms race occurs (not unlike in the case of financial technologies and high frequency trading, as noted by Glode, Green and Lowery (2012) and by Biais, Foucault and Moinas (2016)). This could imply over-investment, in the sense that equilibrium aggregate investment would be larger than its socially optimal counterpart. Such over-investment would be particularly undesirable, given that the proof-of-work mechanism is already criticised for allocating computing power and energy to solving useless cryptographic problems.

5 Conclusion

Our analysis suggests that mining in a blockchain is a coordination game. Coordination games usually have multiple equilibria, some of which are Pareto dominated. Our first results illustrate that this can be the case in the blockchain, and raise an important point in the policy debate on blockchains: when record keeping is decentralised, efficient decentralisation requires coordination, while coordination problems can lead to inefficient equilibria. It would be interesting to study if and how inefficient equilibria could be avoided. Maybe cheap talk could play a role in this context. This might provide a rationale for communication channels among miners and developers, such as IRC channels and forums. Another communication device used in practice by miners is flags attached to blocks to convey messages to other miners, such as, e.g., support for an upgrade, which might then lead to or help avoid a fork. It would also be interesting to identify the main drivers of blockchain instability. For example, one could analyse if concentration of computing power can be dangerous. One could also study if other reward schemes than that currently used in blockchains could generate better outcomes. For example, while Bitcoin does not reward orphaned blocks, Ethereum does, to some extent. Should one expect the latter reward scheme to generate better outcomes than the former?

[17] Intuitively, if difficulty was kept constant, an increase in computing capacity would lead to an increase in the frequency of blocks validations. Instead of one block every 10 minutes, there could be, for example, one block every 8 minutes. The increase in difficulty would bring the average duration between two blocks back to 10 minutes.

We also conjecture that, with endogenous computing power there can be negative externalities and excess investment. Could such inefficiencies be corrected by appropriate regulation or taxation? More generally, it would be useful to better understand the social costs and benefits of having more computing power in the network and to examine if policy intervention is called for.

Appendix

Throughout the proofs we will use the following lemma:

Lemma 1 *Our blockchain game is continuous at infinity.*

Proof of Lemma 1: Denote by $J(\sigma_m)$ the expected payoff of miner m if he follows strategy σ_m. Consider an alternative strategy, σ'_m, that prescribes the same actions as σ_m until time T and differs afterwards. The difference between the two expected payoffs can be written as

$$J(\sigma_m)-J(\sigma'_m) = \Pr(z_m \leq T)E[J(\sigma_m)-J(\sigma'_m)|z_m \leq T]+\Pr(z_m > T)E[J(\sigma_m)-J(\sigma'_m)|z_m > T].$$

Now, by definition,

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

Moreover

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

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

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

which ensures that our game is continuous at infinity.

QED

Proof of Proposition 1: By Lemma 1, a strategy profile forms a subgame perfect equilibrium if and only if there is no profitable one shot deviation from that strategy at any stage in the game.

Our candidate equilibrium, $\{\sigma_m^*\}_{m\in\mathcal{M}}$, is that, for any ω_τ, miners chain their block to the most recent block in the original chain.

To prove that this is a Markov perfect equilibrium we now show that, in any state ω_t, any miner prefers to follow the equilibrium, and chain his block to the most recent one in the original chain, rather than engaging in a one shot deviation, chaining his block to another block, $\tilde{B}$, at time τ, and then reverting to the equilibrium strategy.

To do so, consider three cases, whose probabilities are independent of the miners' actions (since they reflect the distributions of independent Poisson processes whose intensities are exogenous):

The first case is when the next event is z_m. The equilibrium strategy prescribes that all miners mine the original chain at τ. Therefore if m follows the equilibrium strategy, he earns $G(M)$ for each previously solved block on that chain and 0 on any block potentially solved on another chain. If instead m deviates, his payoff from his previously solved blocks cannot be larger:

- If he continues a previous fork, he does not increase the payoffs of his previously mined blocks on that fork, since he is the only one mining it (and $G(1) = 0$), and he cannot increase his payoff from his other solved blocks.

- If he starts a new fork, under the first two assumptions in Footnote 11, which block m was mining is irrelevant, while under the third assumption m earns less if he deviates than if he follows the equilibrium.

The second case is when the next event is that a block is solved by another miner than m. Then, again, which block m chose as a parent block is irrelevant. Observe first that the choice of parent block by m at τ does not affect which chain is the original one after τ. Therefore it does not affect future actions and m's expected payoff from future blocks. It does not affect either the payoff m expects from previously mined blocks, since that payoff depends only on what happened before τ and on the (equilibrium) actions that will be chosen in the future.

The third case is when the next event is that m solves $B_{n(\tau)+1}$ (where $n(\tau)$ is the index of the last block solved by time τ). If m had deviated by chaining $B_{n(\tau)+1}$ to $\tilde{B}$, since all other miners play the equilibrium strategy going forward, and m himself reverts to equilibrium after solving $B_{n(\tau)+1}$ (one shot deviation), m anticipates that no miner will chain to $B_{n(\tau)+1}$ (since the $B_{n(\tau)+1}$ is not in the original chain). Hence, as above, 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 $B_{n(\tau)+1}$. He anticipates that reward to be $G(0) = 0$ for any one shot deviation.

Overall, there is no state ω_τ at which a one shot deviation gives m a strictly higher expected payoff than σ_m^*. Consequently, $\{\sigma_m^*\}_{m\in\mathcal{M}}$ is a Markov perfect equilibrium.

QED

Proof of Proposition 2: Denote by $n(\tau)$ the index of the last block solved by time τ, by $B_{n(\tau)}$ the corresponding block and by τ^f the first time at which the sunspot variable is above $1-\varepsilon$ and $f < n(\tau)$.

Our candidate equilibrium strategy profile, σ^*, specifies the following:

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

b) *At the fork inception:* If $\tau = \tau^f$, or $\tau > \tau^f$ and ω_t does not include an edge $(B_{n(\tau^f)-f}, B_{k+1})$, with $k \geq n(\tau)$, miners chain their block to $B_{n(\tau^f)-f}$.

c) *After the fork:* If $\tau > \tau^f$ and ω_τ includes an edge $(B_{n(\tau^f)-f}, B_{k+1})$, with $k \geq n(\tau^f)$, miners chain their block to the most recently solved block in the chain including $(B_{n(\tau^f)-f}, B_{k^*+1})$, (with $k^* = \min\{k \geq n(\tau^f)$ s.t. there exists an edge $(B_{n(\tau^f)-f}, B_{k+1})\}$), whose index is the index of its parent plus one or, if such a block does not exist, to B_{k^*+1}.

Note that if all miners follow σ^*, their behaviour on the equilibrium path is as described in Proposition 2. To prove that this is a Markov perfect equilibrium, we need to prove that a miner does not have a profitable one shot deviation from σ^*. We hereafter consider the three cases a), b) and c) in turn.

a) *Before the fork*: Bearing in mind that miner's actions don't affect the occurrence of the sunspot, at all times before τ^f the proof of a) operates along the same line as the proof of Proposition 1.

b) *At the fork inception:* Compare m's expected gain if he follows the equilibrium strategy (chaining his block to $B_{n(\tau^f)-f}$) to his expected gain from deviating once by chaining his block to $\tilde{B} \neq B_{n(\tau^f)-f}$ and then reverting to the equilibrium strategy. As earlier, the only relevant case is when the next event is that m solves $B_{n(\tau)+1}$. If he had chained $B_{n(\tau)+1}$ to $\tilde{B}$, then he expects that at later stages no miner (including himself) will chain any block to $B_{n(\tau)+1}$, since he anticipates the equilibrium strategy to be followed. Consequently, his reward for mining $B_{n(\tau)+1}$ attached to $\tilde{B}$ is 0 (and therefore less than his gain if he had followed the equilibrium). Moreover, as before, his expected payoff from previously solved blocks as well as from future blocks, is unaffected by which block he has just mined.

c) *After the fork:* The proof follows the same arguments as in cases a) and b).

QED

Proof of Proposition 3:

Preliminary steps

As mentioned in the text, 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 $(B_{n(\tau^f)-f}, B_{k^*+1})$ that preexists all other chains containing $(B_{n(\tau^f)-f}, B_{k^*+1})$, where $k^* \equiv \min\{k \geq n(\tau^f), (B_{n(\tau^f)-f}, B_{k+1}) \in \omega\tau\}$. We let $v^n(m,\tau)$ denote miners' vested interest in that chain, that is, the number of blocks solved by m on the new chain after τ^f.

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

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

$$v^o(m,\tau)(G(M-K) - G(M-K-1)) - v^n(m,\tau)(G(K+1) - G(K)) \geq \frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K) - G(M-K)), \quad (4)$$

while for $m \leq K$

$$v^o(m,\tau)(G(M-K+1) - G(M-K)) - v^n(m,\tau)(G(K) - G(K-1)) \leq \frac{\Pr(N_m(\tau') - N_m(\tau) = 1)}{\Pr(z_m = \tau')}(G(K) - G(M-K)). \quad (5)$$

We turn now to our candidate equilibrium strategy profile, σ^*, which 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:* If $\tau = \tau^f$, or if $\tau > \tau^f$, Condition 2 holds and the new chain does not exist, miners $m \leq K$ chain their block to $B_{n(\tau^f)-f}$, while miners $m > K$ chain their block to the last block on the original chain.

c) *After the fork:* If $\tau > \tau^f$, Condition 2 holds and the new chain exists, then miners $m \leq K$ chain their block to the last block on the new chain, while miners $m > K$ chain their block to the last block on the original chain.

d) *After the fork off-path:* Suppose there are two consecutive times τ and τ' such that Condition 2 holds at τ but not at τ'. Note that this can only happen after a deviation from the strategy prescribed in b) and c) where $B_{\tau'}$ is either chained to the last block on the new chain (or $B_{n(\tau^f)-f}$) by a miner $m > K$ or chained to the last block on the original chain by a miner $m \leq K$. The equilibrium strategy then prescribes that miners exclude $B_{\tau'}$. More precisely, suppose $B_{\tau'}$ was chained to the original chain by $m \leq K$. Then at and after τ', miners $m \leq K$ chain their block to the last block on the new chain. At τ' miners $m > K$ chain their block to the last block solved by time τ on the original chain, $B^o_{n(\tau)}$, and do so until a new block B^o_{new} is solved and chained to $B^o_{n(\tau)}$. After B^o_{new} is solved, miners $m > K$ chain their block to the last block on the chain that contains $(B^o_{n(\tau)}, B^o_{new})$ that preexists all chains that contain $(B^o_{n(\tau)}, B^o_{new})$. This chain then replaces the original chain in Condition 2. Symmetrically, if $B_{\tau'}$ was chained to the new chain by $m > K$, then at and after τ', miners $m > K$ chain their block to the last block on the original chain, while miners $m \leq K$ exclude $B_{\tau'}$ by mining from the last block solved by time τ on the new chain, and then mining from the chain that originates from that fork. This chain then replaces the new chain in Condition 2.

As will become explicit below, the specification of the equilibrium strategy in states described in d) is useful to rule out certain types of deviations. Note that if the state is as described in d) and all miners play the equilibrium strategy, then the expected payoff of a miner, say $m \leq K$, at τ' is

$$v^o(m,\tau)G(M-K) + v^n(m,\tau)G(K) + \frac{\theta_m}{\lambda_m}G(K)$$

which is exactly his expected payoff at τ under the equilibrium strategy described in b) and c). This also holds for miners $m > K$.

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

a) *Before the fork*:

If miner m goes for a one shot deviation from equilibrium at time $\tau < \tau^f$ it has two effects on his expected payoff. First, m's deviation can affect the distribution of vested interests on the original chain at future times τ such that $r^\tau > 1 - \varepsilon$. Second, as in the proof for Proposition 2, it can impact the value of the block m chooses to mine. These two effects materialise only if the next event is that m solves his block.

Consider the first effect. The occurrence of a fork reduces the payoff that participants receive from the block 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 on the original chain after the fork, $B_{n(\tau^f)-f+1}$ (or in other words, the miners' vested interests in the original chain). For each of these blocks, as well as for the blocks solved after τ^f, the maximal loss for miner m is $G(M)$. In addition m's deviation has an impact on the materialisation of this loss only if the sunspot occurs before m's liquidity shock when m plays the equilibrium strategy. Hence, an upper bound on this loss, or equivalently, on the gain from reducing the likelihood of a fork via a deviation is

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

Now,

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

Observe that

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

Moreover,

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

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

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

Hence, for small ε,

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

Hence, one can set ε so that $\Pr(\tau^f < z_m|\omega_\tau, z_m)$, and correspondingly the gain from reducing the likelihood of a fork via a deviation, is arbitrarily close to 0.

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 will follow σ^*. Hence the expected payoff for this block is 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.

b) c) *At or after the fork:*

1) 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 have to 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 previous proofs, this one shot deviation affects m's payoff only if the next stopping time τ', corresponding to two possible events: either m solves his block, or z_m occurs.

(i) Suppose miner m solves a block at τ', i.e., $N_m(\tau') - N_m(\tau) = 1$. If Condition 2 is still true at τ', since every miner, including m, reverts to the equilibrium strategy from τ' on, the only impact of the deviation is that m earns $G(K)$ for block $B_{n(\tau')}$ instead of $G(M-K)$ under the equilibrium strategy. If Condition 2 is not true at τ', the only impact of the deviation is that m earns 0 for block $B_{n(\tau')}$ instead of $G(M-K)$ under the equilibrium strategy. Indeed both under the equilibrium strategy and the deviation, his expected payoff at τ' is his expected payoff at τ plus the reward he receives for block $B_{n(\tau')}$, which is 0 under the deviation when Condition 2 does not hold since from d) no miner will ever chain a block to $B_{n(\tau')}$.

(ii) Suppose miner m is hit by a liquidity shock at τ', i.e., $z_m = \tau'$. Then his payoff under the deviation is

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

instead of

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

under the equilibrium strategy.[18] It follows that there is no profitable deviation if

$$\begin{aligned}&\Pr(N_m(\tau')-N_m(\tau)=1)(G(K)-G(M-K))\leq\\&\Pr(z_m=\tau')(v^o(m,\tau)(G(M-K)-G(M-K-1))-v^n(m,\tau)(G(K+1)-G(K))),\end{aligned}$$

which is exactly inequality (4) in Condition 2.

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

$$\begin{aligned}&\Pr(N_m(\tau')-N_m(\tau)=1)(G(K)-G(M-K))\geq\\&\Pr(z_m=\tau')(v^o(m,\tau)(G(M-K+1)-G(M-K))-v^n(m,\tau)(G(K)-G(K-1))),\end{aligned}$$

which is exactly (5) in Condition 2.

Next, see that at $\tau=\tau^f$, $v^n(m,\tau^f)=0$ for all miners. Inequality (4) is then written:

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

which is exactly inequality (2) in Condition 1. Similarly, inequality (5) is then written:

$$\frac{\Pr(N_m(\tau')-N_m(\tau)=1)}{\Pr(z_m=\tau')}(G(K)-G(M-K))>v^o(m,\tau)(G(M-K+1)-G(M-K))$$

which is exactly inequality (3) in Condition 1.

Furthermore, if miners adhere to the equilibrium strategy, then miners $m\leq K$ always mine the new chain so that inequality (3) in Condition 1 implies that inequality (5) in Condition 2 is true at any $\tau\geq\tau^f$. Symmetrically, given that miners $m>K$ stick to the original chain, Condition 2 is always verified after τ^f. Hence, given that Condition 1 holds at τ^f, then for $\tau>\tau^f$, Condition 2 holds on the equilibrium path.

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

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

d) After the fork off-path

Suppose $\omega_{\tau'}$ is as described in d) and consider a deviation by miner m before block $B_{n(\tau')+1}$ is solved. If m chains his block to $B_{n(\tau')}$ and the next event is not that m solves his block, then his deviation is irrelevant. If the next event is that he solves his block, $B_{n(\tau')+1}$, then the only impact of the deviation is that m earns 0 for this block, since all miners play the equilibrium strategy going forward so that no miner will ever chain his block to $B_{n(\tau')+1}$.

Suppose that miner m deviates and chains his block to a block that was solved before $B_{n(\tau')}$. Then, as above, the only relevant deviations are for a miner $m \leq K$ to mine a block chained to the last block on the original chain solved by time τ or for $m > K$ to mine a block chained to the last block on the new chain solved by time τ. By construction, the payoffs from these deviations and from adhering to the equilibrium strategy are identical to the payoffs derived in b) and c). Furthermore, given that no miner chains his block to $B_{n(\tau')}$ which therefore yields 0 to the miner who solved it, a deviation is not profitable if Condition 2 holds when block $B_{n(\tau')}$ is substracted from miners' vested interests $v^o(m, \tau')$ and $v^n(m, \tau')$, that is, if $v^o(m, \tau)$ and $v^n(m, \tau)$ are such that Condition 2 holds, which is true.

Finally, consider a deviation by miner m after block $B_{n(\tau')+1}$ is solved, then the state is such that Condition 2 holds, hence, from b) and c), there is no profitable one shot deviation. QED

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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