Agent-based Modelling of Bitcoin Consensus without Block Rewards

Benjamin Kraner
Blockchain & Distributed Ledger Technologies Group
University of Zurich
Zürich, Switzerland
benjamin.kraner@uzh.ch

Sheng-Nan Li
Blockchain & Distributed Ledger Technologies Group
UZH Blockchain Center
University of Zurich
Zürich, Switzerland
shengnan.li@uzh.ch

Andreia Sofia Teixeira
LASIGE and Faculty of Sciences
University of Lisbon
Lisboa, Portugal
asteixeira@ciencias.ulisboa.pt

Claudio J. Tessone
Blockchain & Distributed Ledger Technologies Group
UZH Blockchain Center
University of Zurich
Zürich, Switzerland
claudio.tessone@uzh.ch

2022 IEEE International Conference on Blockchain (Blockchain) | 978-1-6654-6104-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/BLOCKCHAIN55522.2022.00015

***Abstract*—Trust is key to the efficient functioning of any fiat or crypto-currency and so is for the consensus algorithm behind the functioning of blockchain systems. By an arbitrary design choice, Bitcoin and most Proof-of-Work (PoW) blockchains have a limited supply. Once block rewards vanish, only transaction fees will remain as an incentive for miners to partake in the verification process. In this paper, we analyse the impact that miners bargaining over block composition has on consensus in the absence of block rewards: in this situation, competing blocks at the same height may be more attractive to peers by including less transactions (i.e. sharing the mempool). The mining and acceptance of blocks can be modelled as an Ultimatum Game, where miners' strategies represent their fairness sentiment. Extending previous Literature, our study focuses on the effect of the transaction arrival rate on global consensus in the system and whether local consensus is formed under certain assumptions about the strategies of miners. We find that consensus is threatened when the supply of transactions is low and stable consensus only emerges when the amount of unconfirmed transactions remains sufficient. In addition, when miners are set with randomised strategies, it is more difficult for the system to achieve consensus. Our research suggests that transitioning from a block reward incentive to a transaction fee incentive may weaken and even destroy the consensus of PoW-based systems.**

***Index Terms*—Blockchain consensus, Bitcoin, Agent-based Model, Ultimatum Game, Gillespie algorithm**

I. Introduction

Miners in a Proof-of-Work (PoW) based blockchain system are in charge of the confirmation of transactions. They commit computing resources to find blocks, which extend the existing blockchain and allow transactions to be recorded in the public ledger. For example, *Bitcoin* incentivises miners to commit computing resources by rewarding each block with a fixed block reward and a variable fee collected from the transactions they confirm in a block [1]. The current design of Bitcoin and many derivatives (its forks, Litecoin, etc.) is chosen such that a fixed amount of cryptocurrency can be issued. Therefore, when the last Bitcoin will be mined, only transaction fees will fund the incentive to continue mining Bitcoin [2]. It will then resemble a sequential bargaining game. The absence of block rewards leads to a situation in which miners may have an incentive to strategically adjust the amount of transactions they confirm within a block. The transition from a block reward regime to a transaction fee regime could increase the forking of the blockchain and threaten efficiency and consensus. Miners may end up in disagreement.

In order to study a blockchain consensus system in the absence of block rewards, we expand the stochastic model of blockchain consensus provided by Tessone et al. [3], incorporating transactions and modeling the strategies of miners as an Ultimatum Game. We model transactions as objects which can be processed by miners in blocks. Additionally, miners are endowed with strategies representing their perception of block fairness. In our agent-based model of Bitcoin mining, the construction of its blockchain is modelled as an Ultimatum Game. Miners of a block act as proposers and miners who receive a block act as responders. We use the seminal work of Teixeira et al. [4], where they analyse fairness in multiplayer Ultimatum Games as a motivation.

As consensus is key to the efficient functioning of a blockchain system, the main purpose of this work is to analyse the impact that miners' strategies will have on consensus in the absence of block rewards. Firstly, our study focuses on what effects the transaction arrival rate and the network delay have on global consensus in the system. Secondly, we analyse whether the global or local consensus is formed under the random strategies of miners. To sum up, we are able to conclude that when transitioning from block reward incentive to transaction fee incentive, the consensus of the system may be at risk. In order to ensure a stable consensus in the absence of block rewards, PoW-based blockchain, like Bitcoin, will have to maintain a plenty of unconfirmed transactions in the

978-1-6654-6104-7/22/$31.00 ©2022 IEEE
DOI 10.1109/Blockchain55522.2022.00015

system.

The paper is organised as follows: Section II provides a literature review; Section III introduces the modelling approach and simulation; Section IV defines measures for global and local consensus; Section V shows the results of consensus under miners' global and random strategies. Finally, Section VI draws conclusions and poses venues for future research.

II. Literature Review

We present related work along the two conceptual dimensions discussed above: **(i)** Bitcoin and blockchain systems and **(ii)** Ultimatum game. The literature that analyses the limit case of diminishing block rewards in Bitcoin is scarce. Differently the literature on the Ultimatum Game is vast, so we focus on the literature that inspired this paper.

The issues with a public blockchain in which miner's verification is incentivised are multiple and have given raise to a growing body of Literature [5], [6]. Some examples of miners' strategic behaviour include Li et al. [7], [8] who conducted an empirical study of miners' selfish mining that was proposed as an attack on Bitcoin by Eyal and Sirer [9] in 2014, and miner's dilemmas induced by which mining pool to join [10]. The selfish mining strategy shows that miners (respectively mining pools) could increase their revenue by withholding blocks intentionally. These analysis is bound to the case where miners are incentivised by a block reward. Carlsten et al. [11] studied how different strategies might threaten the efficiency of Bitcoin when transaction fees are the only incentive left to encourage mining. They show that the security of the blockchain is threatened by miners who may undercut blocks of other miners, by including less transactions in their own blocks. Furthermore, they explored the possibility of a *mining gap*, where miners would stop to mine Bitcoin, when not enough transactions are available to cover the (electricity) cost miners face in order to discover blocks. These results were extended in the so-called *gap game* [12], which shows that miners may select different gap sizes to optimise their utility, even when their operating costs are identical and that the system creates incentives for miners to reduce decentralisation through the creation of coalitions.

About the modeling of blockchain evolution, Tessone et al. [3] proposed a minimalist stochastic model of blockchain systems in order to study the effect of network delay in such systems. They identified two regimes, a functional regime in which consensus is given and a non-functional regime in which the blockchain disperses into a branched state. They observed a phase transition from a non-consensus state to a state of consensus as network delay drops. Their model serves as the foundation of our analysis.

Teixeira et al. [4] analysed fairness, measure by the average proposal amount and average fitness in a population, in a multiplayer ultimatum game. Nodes of a network are interacting with each other based on an implementation of the ultimatum game in a multiplayer setting. They studied the effect of a network-based role assignment of nodes and find that low-degree proposers increase the fairness in such a setting. Although their concepts are not fully applicable in the case of Bitcoin, our model is motivated by their train of thought.

III. Model and Simulation

A. Model Elements

1) Basic Notions: The stochastic model proposed by Tessone et al. [3] consists of a network, which simulates the structure of the P2P network in Bitcoin and blockchain systems in general. The nodes of the network represent the miners of the blockchain, each of whom is endowed with a computational power, π_i, which, as Bitcoin is a PoW system, ultimately determines with what probability a node discovers a new block. The edges of the network represent whether the miners are connected with each other and thus may exchange their state of the blockchain. They can only communicate with their peers and their view of the blockchain is dictated by what information they receive. Each miner possesses its own local copy of the blockchain, denoted by $\mathcal{B}_i(t)$, which represents his view of the current state of the system.

Tessone et al. [3] also model the discovery of blocks and the diffusion of blocks over the network. Since the mining of blocks are independent and occur at a fixed rate η_i, the mining of new blocks is described by a Poisson process. Hence the time it takes for miner i to find a block is distributed according to an exponential distribution with parameter η_i. In addition, according to PoW protocol, the interval in which blocks are mined is adjusted on the global level, namely difficulty adjustment. The Bitcoin mining interval as a design choice is 10 minutes and denoted by: $\tau = 10'$. Thus, we can derive that the global rate at which blocks are mined by all the miners in a given interval can be denoted by $\sum_i \eta_i = \tau^{-1}$. After mined, the blocks will be diffuse over the network. To model block propagation with a *network delay*, we assume that block propagation follows a Poisson process and we define the parameter of the network delay as τ_{nd}. It follows, as in the case of the mining process, that the average time it takes for a block to be propagated from a node to another follows an exponential distribution with parameter τ_{nd}^{-1}.

Each block b in the chain has a certain height h_b (number of blocks between block b and the genesis block). As mentioned above, each miner has at any point time t a local copy of the blockchain $\mathcal{B}_i(t) = \{b_0, b_1, b_2...\}$, and it is worth noting that miners usually count the longest chain valid in current Bitcoin protocol. Differently, we assume that there is no block reward, thus implying that the longest chain rule might not hold. Instead of holding any information of the reward, we will allow the blocks to bear information about the number of transactions each block contains, denoted as θ_b. This will be the important point at which miners can apply their (offering) strategies. Meanwhile, blocks will have a finite size in respect to transactions, to which we will refer to as θ^{max}, representing the equivalent to the block size maximum.

2) Transactions: To retain the simplicity of the model, we refrain from modelling the size and fees of transactions explicitly. Instead we assume all transactions to be of the same

size and to bear the same positive transaction fee which are the ultimate incentive for miners to keep on mining after no block rewards are obtainable anymore. We further assume that each transaction will be verified, i.e., there are no transactions that would be refused, thus transactions are only awaiting their confirmation into a block. Finally, we make the assumption that transactions flow into the system at a average constant rate τ_t^{-1}, named as *transaction arrival rate*.

When a transaction tx enters the system, the transaction is stored in a *global pool* $\mathcal{T}(t) = \{tx_1, tx_2...\}$. We denote $\Theta_g(t) = |\mathcal{T}(t)|$ as a *counter* of the total number of transactions in the system. By defining the pool of transactions in a global way (all nodes connected to it), we assume that transactions, upon entering the system, are immediately known to all nodes. In reality, transactions are being received by a specific node, where they are verified, and then propagated throughout the network. Nonetheless, we make this assumption based on the research of Decker et al. [13] as transactions are much smaller in size than blocks and hence face a shorter propagation time than blocks.

Even though we have defined a global counter of transaction Θ_g, it is important to note that each miner would have a distinct memory pool which contains different amount of unconfirmed transactions. Only when the blockchain is in consensus will the memory pools of nodes coincide. Specifically, since each node has a local blockchain copy, $\mathcal{B}_i(t)$, when the blockchain is forked, the local blockchain copies do not have to agree on the same longest block (main-chain) and, thus, the amount of unconfirmed transactions may differ. We denote the transaction that are incorporated in a block b as $T_b = \{tx_1, tx_2...\}$. Thus, the number of transactions included in that block is $\theta_b = |T_b|$. The total amount of transactions which are confirmed in a miner i's main-chain $V_i(t)$ then equals: $V_i(t) = \sum_{b \in \mathcal{B}_i^M(t)} \theta_b$, where $\mathcal{B}_i^M(t)$ represents the chain of blocks originating from the genesis block (b_0) to the currently highest block in miner i's blockchain. Equipped with the knowledge of $V_i(t)$, a miner is able to calculate the amount of transactions that are awaiting confirmation by setting

$$U_i(t) = \Theta_g(t) - V_i(t), \tag{1}$$

where $U_i(t)$ refers to the memory pool of a miner i, the stock of unconfirmed transactions specific to miner i's blockchain. $U_i(t)$ also refers to the memory pool in Bitcoin, as participants only verify a transaction, which has not yet been written (confirmed) in the blockchain. Fig. 1 shows how different local blockchain copies lead to differences in the amount of transactions that can be confirmed. Both miners have the same information about the total number of transactions that entered the system, Θ_g, but miner j has already received a different fork of the blockchain, which is longer than miner i's copy. When calculating the number of transactions confirmed in their respective main-chain, both miner i and j will encounter different amounts.

3) Ultimatum Game Strategies: Mining in Bitcoin resemble the ultimatum game [14] in the absence of block rewards. To effectively describe the decisions a miner faces in the context

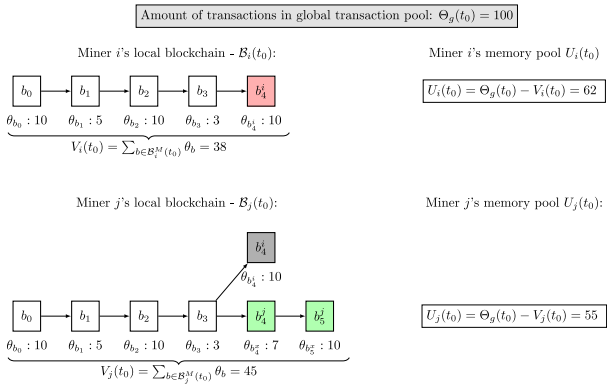

Fig. 1. Distinct memory pools: Differences in the memory pool can be witnessed if miners local blockchain copies are not the same.

of the ultimatum game, we have to define when miners act as proposers or responders, and how they decide what share of the memory pool $U_i(t)$ they consider fair, to propose and to accept. As in the Ultimatum Game our agents will possess idiosyncratic strategies representing their strategic choices on how to divide their amount of unconfirmed transactions $U_i(t)$ and decide whether a division of $U_i(t)$ is fair. Therefore, each miner is also endowed with a strategy set S_i, containing two parameters: $S_i = (p_i, q_i)$, where p_i denotes the *offering strategy*, the strategy miner i will use when acting as a proposer and q_i denotes the *accepting strategy* of miner i, which he applied when acting as a responder.

Offering strategy: The offering strategy, p_i, represents the case when a miner discovers a block and it is his turn to propose a share of the unconfirmed transactions included in his block to the network. p_i dictates the fraction of transactions the miner would want to include in a block he mines – p_i is bounded in the interval between zero and one, $p_i \in [0, 1]$. When a block is discovered, miner i will check the current memory pool size with regards to his local blockchain copy $U_i(t) = \Theta_g(t) - V_i(t)$. Given his offering strategy, the miner will decide how many transactions he would like to include by calculating $\lfloor p_i U_i(t) \rceil$, where $\lfloor . \rceil$ indicates the nearest integer function. Given that the block size is limited number of transactions, he will include θ_b transaction, which is given by:

$$\theta_b = \min(\lfloor p_i U_i(t) \rceil, \theta^{max}) \tag{2}$$

where θ^{max} indicates the block maximum. (We will keep the θ^{max} fixed at 100 throughout our analysis.)

Accepting strategy: The accepting strategy, q_i, dictates what share of transactions in a block a miner i considers fair and is thus willing to accept, acting as a responder. The evaluation of fairness is not straightforward. The simplest idea would be to let the miner only consider the share of the highest block he receives: when a miner i receives a block b, he will accept it only if the transactions included in the block θ_b are below the accepting strategy q_i of a miner i respective to the size of the memory pool from i's perspective:

$U_i(t) = \Theta_g(t) - V_i(t)$ at time t. He will evaluate a block and *accept* if

$$q_i \geq \frac{\theta_b}{U_i(t)}. \tag{3}$$

Until now we have implied that the block a node receives is the direct continuation of its local blockchain (direct as the height of the block is larger by one). In fact, this may not always be the case, especially when we deviated from the longest chain rule. We may consider a block that is several blocks higher than a miner's local blockchain or it may be a block on a different fork. Thus, we need a more general concept on how a block can be evaluated by a node to accept or not, even if it is not a direct continuation of his local blockchain. To do so, a node has: **(i)** to consider not only the last block that will be added to the local blockchain, but also all preceding blocks that will have to be added in order for the last block to be accepted; **(ii)** to account for the change of the memory pool induced by continuing on another fork of the blockchain.

We can tackle both points by allowing agents to not only evaluate a block based on the last block they receive, but rather based on the whole set of blocks they have to accept in order to add that block. For instance, a miner i who is currently mining on block b_i is receiving a block b_j from miner j, and both miners have their respective local blockchain copy, $\mathcal{B}_i(t)$ and $\mathcal{B}_j(t)$. In order to evaluate the fairness of a block and to decide whether to accept it, miner i will search for the highest block that both he and miner j agree on: $b_r = \text{argmax}_{b \in \mathcal{B}_i \cap \mathcal{B}_j}(h_b)$, thus they also agree on the transactions confirmed in the blockchain up until this block b_r that is called as *root* block. Given that each block possesses exactly one parent block, we can define the sequence of blocks from block b_j backward to block b_r (including b_j and b_r) as $\mathcal{B}_j^{b_j \to b_r}$, and similarly the sequence of blocks from b_i to b_r as $\mathcal{B}_i^{b_i \to b_r}$. Finally, as the root block b_r is included in both forks, we may subtract it from both sets in order to get the non-consensus part of the local blockchain: We denote $\mathcal{B}_j^{b_j \to b_{r+1}} = \mathcal{B}_j^{b_j \to b_r} \setminus \{b_r\}$ and $\mathcal{B}_i^{b_i \to b_{r+1}} = \mathcal{B}_i^{b_i \to b_r} \setminus \{b_r\}$.

By defining the non-consensus part of blocks, $\mathcal{B}_j^{b_j \to b_{r+1}}$ and $\mathcal{B}_i^{b_i \to b_{r+1}}$, we can effectively account for the amount of transactions that miner i would have to give up (because those transactions were already added to miner j's local blockchain copy) in order to accept the block b_j. Thus, in order to evaluate a block, the miner i will consider the effective share of transactions ϕ_{b_j} that he will have to give up, when accepting that block b_j

$$\phi_{b_j} = \frac{\Theta_j}{\Theta_i + U_i(t)}, \tag{4}$$

where $\Theta_j = \sum_{b \in \mathcal{B}_{j(t)}^{b_j \to b_{r+1}}} \theta_b$ is the amount of transactions have been consumed by the fork of miner j, and $\Theta_i = \sum_{b \in \mathcal{B}_{i(t)}^{b_i \to b_{r+1}}}$ is the amount of transactions processed in miner i's specific fork that he will add back to his memory pool. Miner i compares ϕ_{b_j} to his accepting strategy q_i and accepts the block b_j and all preceding blocks if

$$q_i \geq \phi_{b_j} \tag{5}$$

It is immediate to see that the formula above will reduce to evaluating a single block (as Eq. 3) when $b_i = b_r$, as ϕ_{b_j} collapses to $\phi_{b_j} = \theta_b / U_i(t)$.

B. Simulation

To simulate our model, we rely on an agent-based modelling approach. The nodes in our network are the agents and their strategies as described above. The agent-based model evolves as time goes by and different events occur. We have defined above the three different processes and events that drive the evolution of our model: **(i)** a process for the discovery of the blocks, dictating the inter block time, **(ii)** a process that controls how long it takes blocks to propagate between two nodes and **(iii)** a transaction process, which dictates at what rate transactions enter the system. Given the nature of our proposed parameters, the event times can be calculated efficiently in an implementation of the *Gillespie Algorithm* [15], [16]. We will revise it briefly and show how they are bound together in terms of the Gillespie Algorithm.

1) Block creation: a miner discovers blocks at a constant rate η_i. As each node represents an independent Poisson process, the aggregate rate at which blocks are discovered is given by $\sum_i \eta_i = \tau^{-1}$. The design of Bitcoin fixes this at the global rate to $\tau^{-1} = 10'$.
2) Block diffusion: the block propagation rate at which a local blockchain is sent from node i to node j is τ_{nd}^{-1}. We define the edge between a miner i and one of its neighbours j, $i \to j$, as active, if miner i possesses a block, which i has not yet shared with j. We assume that the latency τ_{nd}^{-1} between any active edge is the same, therefore, the aggregate rate at which a the diffusion of a blockchain happens is given by $E_a \tau_{nd}^{-1}$, where the number of active edges is given by E_a.
3) Incoming transaction: the rate at which transactions enter the global transaction pool is given on a global level by τ_t^{-1}.

We assume that when a node rejects a block, it will not accept the same block in the future, and the exchange can only happen if a block is newly found or received and added to a miners local blockchain copy. Our definition ensures that a block is sent only once to a miner from one of his peers, allowing the decision whether a block is fair or not to happen only once per edge and block. As all events are independent from each other, the total rate of transition is defined as:

$$\xi = \tau^{-1} + E_a \tau_{nd}^{-1} + \tau_t^{-1}. \tag{6}$$

The Gillespie algorithm selects the event that will occur next, proportional to the total rate of transitions. After selection, the corresponding event will be triggered in the system. The events in the system are described hereafter at some arbitrary time $t_0 \geq 0$ and the updated blockchain $\mathcal{B}_i(t_0 + t')$ and time increment t' are defined below.

1) With probability τ^{-1}/ξ, the next event will be *block discovery*: a miner i will be selected proportional to the computing power he commits to the system, $\pi_i/\sum_j \pi_j$. He has chosen a number of transactions θ_b (assuming a candidate block was composed immediately before), according to his offering strategy and the current amount of unconfirmed transactions: $\theta_{b_i^{new}} = \min(\lfloor p_i U_i(t) \rceil, \theta^{max})$. He will discover a block b_i^{new} and append it to his local blockchain copy: $\mathcal{B}_i(t_0 + t') = \mathcal{B}_i(t_0) \cup b_i^{new}$.
2) With probability $E_a \tau_{nd}^{-1}/\xi$, the next event will be *block diffusion*: a miner i will be selected randomly from the set of nodes connected to the network, which have at least one active directed link originating from i. Then one of the active links of node i is selected randomly and the blockchain ledger is sent to node j. The node j will evaluate the highest block of miner i, $b_i^h = \arg\max_{b \in \mathcal{B}_i}(h_b)$, and if the height of $H_i(t_0) = max_{b \in \mathcal{B}_i(t_0)}(h_b)$ is below the height of miner j local blockchain ledger ($H_i(t_0) \leq H_j(t_0)$), the block will not be evaluated and is not accepted. If $H_i(t_0) > H_j(t_0)$, then the block will be evaluated based on the accepting strategy, q_j of miner j as it is defined in section of miners' *acceptiong strategy*. The miner will calculate $\phi_{b_i}(t_0) = \frac{\Theta_i(t_0)}{\Theta_j(t_0)+U_i(t_0)}$ as the fraction of transactions confirmed in the subset of blocks he has to accept in order to accept block b_i, adjusting his amount of unconfirmed transactions in case of a switch to another fork and accepts the block if $q_i \geq \phi_{b_i}(t_0)$.
If block b_i is accepted, miner j will update his local blockchain copy with the blocks he did not yet have in it: $\mathcal{B}_j(t_0 + t') = \mathcal{B}_j(t_0) \cup \mathcal{B}_i^{b_i \to b_{r+1}}$.
3) With probability τ_t^{-1}/ξ, the next event will be an *incoming transaction* to the system. The amount of transactions in the global transaction pool will be increased by one: $\Theta_g(t_0 + t') = \Theta_g(t_0) + 1$.

After the execution of an event, the time in the model is increased by t', which follows an exponential distribution with parameter ξ, $t' \sim \exp(\xi)$.

IV. Emergent Properties

To study whether consensus can be achieved when miners are not strictly obeying to the longest chain rule, but following the notion of fairness embedded in their strategies, we define two relevant measures: 1) orphan rate Ξ, which measures on how many blocks the system has achieved consensus; 2) a measure for the relative efficiency E_r based on the height of the longest chain.

A. Global: Orphan Rate

Due to the latency of the system or when blocks are deemed unfair, miners may produce blocks which will not be included in the blockchain copies of other miners. The computational resources a miner invested for such a block are wasted. The more such blocks are produced the higher the inefficiency of a system. When a block is not included in all the other miners' local blockchain copies, it is deemed to be *orphaned*. In return, blocks which end up in all of the local blockchains are said to be in the main-chain or consensus chain. We define the set of all blocks existing in the blockchain system as $B = \bigcup_i \mathcal{B}_i$. The set of main-chain blocks is given by $M = \bigcap_i \mathcal{B}_i$ and the set of orphaned blocks is given by: $O = B - M$. The orphan rate Ξ is then defined as the ratio of the number of orphaned blocks among all the blocks:

$$\Xi = \frac{|O|}{|B|} \tag{7}$$

where the lower bound of 0 defines a state of complete consensus, and the upper bound of 1 indicates a state in which consensus is not reached on any block.

B. Local: Relative Efficiency

When the blockchain stays in a forked state and consensus is not reached, the orphan rate Ξ is not a sufficient measure of the system efficiency. Our model introduces a new component that can lead miners to not be in consensus, which is the perceived fairness of a block based on the strategies of the miners, even when the block is higher. This deviation from the longest chain rule leads to a situation where some nodes may never accept a chain of blocks, regardless of its height, but rather in terms of the amount of transactions that were confirmed in that chain. Consider the case where a miner has an accepting strategy $q_i = 0$. Given his perception of block fairness, he will never accept any block (or a chain thereof) that contains a strictly positive amount of transactions.

Nonetheless, even in a system without consensus, we may be interested to what degree miners are willing to work together on a common blockchain fork. In order to do so we introduce a measure of the relative efficiency E_r in absence of consensus. We define the relative efficiency as:

$$E_r = \frac{h_B}{|B|} \tag{8}$$

where $h_B = \max_{b \in B}(h_b)$ denotes the height of the longest block in the set of all block. In an extreme case: when all blocks are aligned in a single chain and the height of it is equal to the amount of block produced in the system, E_r will go to one. In the other extreme case, if miners are not cooperating at all and they would simply mine on their individual blockchain fork, every miner is expected to produce a share of blocks proportional to his share of computational power. Thus, the longest chain will be given by the the miner who possesses the maximum of computational power $E_r = \max(\pi_i/(\sum_j \pi_j))$. These two extreme cases will give the upper and lower bound for E_r respectively. For values of E_r in between the lower and upper bound, miners show some cooperation and blocks are accepted and mined on by other miners, but global consensus is not necessary reached. Thus, the relative efficiency E_r may be interpreted as a measure of the local consensus between miners in the network.

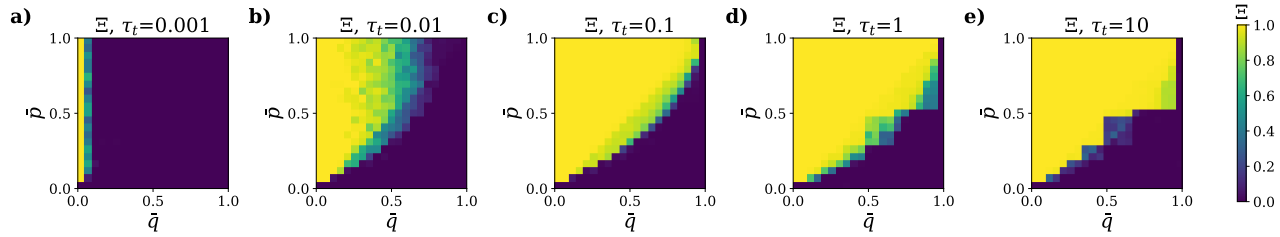

Fig. 2. The orphan rate Ξ for different global strategy ($p_i = \bar{p}, q_i = \bar{q}$) combinations, a fixed network delay $\tau_{nd} = 0.01$. The different panels account for different values of the transaction arrival rate τ_t. Averages of 30 realisations.

V. Results

A. Baseline Parameters

In our simulations we consider a network of one hundred miners ($N = 100$), while small we think that such a number of miners is sufficient to represent the core network of Bitcoin miners [17]. As a baseline we consider the Barabási-Albert topology of the peer-to-peer network with parameter $m = 3$ of edges attached to new nodes such that nodes are on average connected to 6 peers. The computational power of miners is distributed according to an exponential distribution with $\pi_i \sim exp(0.05)$. According to previous researches [3], [18], we also assume that an exponential distribution is a reasonable approximation for the hashing power distribution in Bitcoin. The maximum amount of transactions in a block is set to one hundred ($\theta^{max} = 100$), so that one transaction represents one percent of the size of a block. To avoid situations in which nodes in the beginning would have to mine blocks with a very low amount of transactions, we set the initial amount of transactions in the system to one hundred, $\Theta_g(0) = 100$, when not mentioned otherwise. Miners have a common *genesis* block as a foundation for the blockchain. The model simulates a time of 1000 in the system, which translates to 10'000 minutes (as the mining rate is given by $\tau = 1$) and thus on average the creation of 1000 blocks is expected. The data presented hereinafter, are averaged results of 50 realisations of the system, unless otherwise indicated.

B. Global Strategies

In order to analyse the model with regards to consensus for different combinations of strategies, we allow different strategies (keeping them fixed at first) for all nodes ($p_i = \bar{p}, q_i = \bar{q}\ \forall q_i$). Then, we analyse under which conditions consensus on the global scale is possible. We expect that when the strategies are aligned, such that $\bar{q} > \bar{p}$, consensus should emerge.

Fig. 2 shows the orphan rate Ξ for different combinations of global strategies $\bar{p}$ and $\bar{q}$. The panels of Fig. 2 from left to right show the orphan rate for increasing values of the transaction arrival rate. For higher values of τ_t, it is a slow rate of newly appearing transaction, which also means less unconfirmed transactions left in the memory pool. We identify two different regimes in each panel, a regime in which consensus is achieved ($\Xi \approx 0$) indicated by the dark shading and a regime in which consensus is not achieved ($\Xi \approx 1$) indicated by the light shading. The transition of the regimes between the panels shows how the transaction arrival rate influences both regimes. With increasing values of τ_t, and thus less transactions in the system, consensus is achieved in a smaller and smaller region of the strategy space defined by $\bar{p}$ and $\bar{q}$.

The panel (a) of Fig. 2 shows that given a very small transaction arrival rate, representing abundant transactions in the system, the system could achieves consensus for almost all strategy combinations. Then, in the panel (b) of Fig. 2 shows a situation at which the transaction arrival rate meets the confirmation capacity of the blockchain in terms of transactions. We can find that, in the upper right corner where values of $\bar{p}$ and $\bar{q}$ close to one, the offering strategy is higher than the accepting strategy ($\bar{q} < \bar{p}$), and consensus still prevails. The next three panels (c) (d) (e) show Ξ for high values of τ_t and in a state where transactions are scarce relative to the capacity of the blockchain. In this situation, the atomic property of transactions, its indivisibility, reduces the strategy space in which consensus can prevail.

The influence of the network delay τ_{nd} on the consensus of a blockchain system with global strategies can be seen in Fig. 3. Given a fixed transaction arrival rate, $\tau_t = 0.01$, the panels from left to right show the results for different network delays. In the panel (c) we observe that an increase in the network delay enables regions of the strategy space, where $\bar{p} > \bar{q}$, to reach consensus on a subset of the blocks in the system ($1 > \Xi > 0$). The increased network delay leads to more wasted resources, as blocks are not communicated fast enough over the network, leading to forks. When we take a look at the most right panel (e) of Fig. 3, we observe that the fork introduced by a high network delay leads to a decay of consensus.

C. Random Strategies

To study consensus in Bitcoin, a more general in PoW blockchain system, we analyse the setting in which agents' strategies are not given on a global level, but rather are

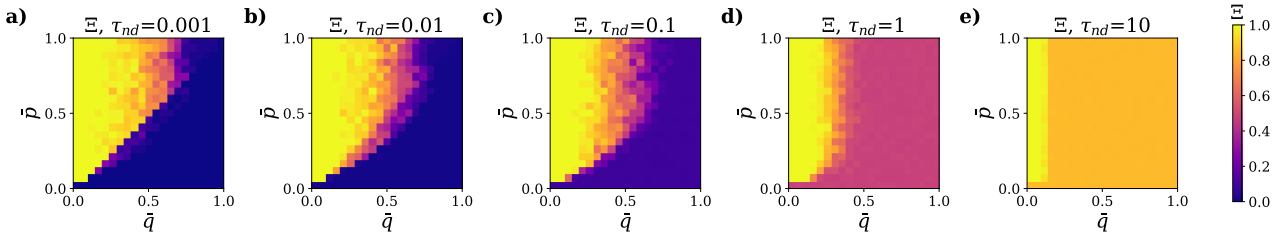

Fig. 3. The orphan rate Ξ for different global strategy ($p_i = \bar{p}, q_i = \bar{q}$) combinations, a fixed transaction arrival rate $\tau_t = 0.01$. The different panels account for different values of the network delay τ_{nd}. Averages of 30 realisations.

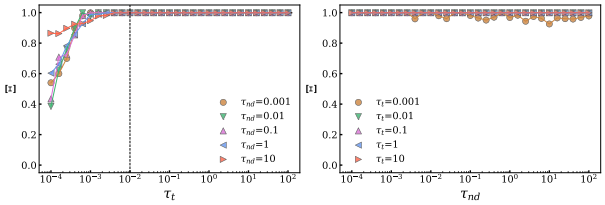

Fig. 4. The orphan rate Ξ in a system with random strategies ($p_i \sim U(0,1)$ and $q_i \sim U(0,1)$) as a function of the transaction arrival rate τ_t (left) and the network delay τ_{nd} (right).

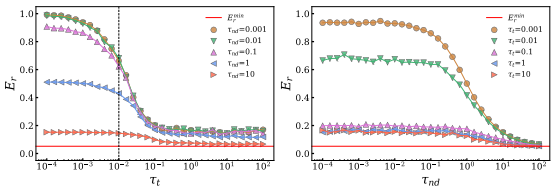

Fig. 5. The relative efficiency E_r in a system with random strategies ($p_i \sim U(0,1)$ and $q_i \sim U(0,1)$) as a function of the transaction arrival rate τ_t (left) and the network delay τ_{nd} (right).

randomly chosen by the agents. As such we assume that each strategy p_i and q_i is chosen by each miner uniformly $p_i \sim U(0,1), q_i \sim U(0,1)$.

Fig. 4 shows the orphan rate in a setting with randomised strategies, for both the transaction arrival rate τ_t and the network delay τ_{nd}. We observe the disappearance of consensus. In the left panel, only in the extreme case, where the transaction arrival rate is substantial low bringing abundant of transaction in the pool, the system achieves consensus on a subset of blocks ($\Xi < 1$). In addition, the right panel shows that the network delay is not responsible for the lack of consensus. Although we may not encounter an accepting strategy of zero, strategies close to zero are possible. And as such miners may exist, that are very reluctant to evaluate a block as fair. Therefore, they will never take part in a chain of blocks mined by miners with offering strategy greater than zero. Consensus on a global scale is ruled out.

Given that *no* global consensus emerges, one may nonetheless be interested to what degree the system is able to cooperate on a common fork of the blockchain. Given this, we resort to the relative efficiency measure, E_r, as discussed in Section IV-B. By measuring the longest chain of blocks relative to the amount of all blocks discovered, we evaluate how many resources are committed to the single longest chain and also to what degree miners are willing to cooperate in the setting with randomised offering and accepting strategies.

In Fig. 5, we show E_r as a function of τ_t and τ_{nd}, when miners have distinct strategies for the offering and acceptance of blocks, under the assumption that strategies are randomly chosen. Observing E_r in the left panel of Fig. 5, we can see that the concentration of computational power in the longest chain can almost equal to one when transactions are arriving at the system at a fast pace, $\tau_t = 10^{-4}$, indicating that most miners are willing to cooperate in a *long* chain they deem fair. The relative efficiency then steadily drops as less transactions enter the system, and stabilises when a transaction arrival rate exceed about 0.1. We notice that the minimum E_r indicated by the red line E_r^{min} is still slightly greater than 0, which may indicate that, even when transactions are very scarce, some cooperation among miners remain: they accept blocks with zero transactions, i.e., they might follow the longest chain rule in the absence of transactions.

Looking to the right panels of Fig. 5 reveals the role of the network delay τ_{nd}. Only for high values of network delay τ_{nd} does the hindered communication between the nodes detain them from concentrating their computational power in one long chain. When τ_{nd} reaches very high values, the network enters a completely branched state, in which the computational resources cannot be concentrated. Even when a miner may accept a block, the high network delay makes it impossible to know the other blocks that are mined until they will reach the miner. We find no indication that a high τ_{nd}, by allowing more transactions to enter the system in the time between block creation and reception, increases E_r. It is possible that since miners may only decide once per block and peer, the negative effect of the delay overshadows the positive effects (of more transactions entering in the meanwhile).

VI. Conclusion

Currently, miners of Bitcoin are incentivised mainly by block rewards. By design, block rewards in Bitcoin are diminishing and, thus, the incentives for miners will shift to a transaction fee regime. Indeed, at some point only transactions fees will motivate miners to mine Bitcoin. As the stability of Bitcoin relies on consensus, it is important to study how it will be affected by such change. In this work, we argue that Bitcoin mining will resemble a variation of an Ultimatum Game when the block rewards vanish, and study the impact on consensus in the Bitcoin system by modelling the PoW protocol and miners' strategies.

The core innovations of our model are along three dimensions: 1) We modelled the arrival of transactions within the system, 2) extended the miners of the system with distinct memory pools, and 3) endowed agents with a strategy set similar to the Ultimatum Game. Furthermore, we define a measure for the relative efficiency in a non-consensus environment.

In this paper, we study the existence of consensus for miners' global strategies and find that consensus is generally possible, when such strategies are favourably aligned. We show that a lower transaction arrival rate (abundant transaction in the system) may loosen such constraints on the strategies. We then allowe miners to possess random strategies representing their fairness sentiment and show that no global consensus exists in such a setting. We analyse the situation of random strategies and discuss the influence of the transaction rate regarding the relative efficiency (local consensus) and show how a low supply of transactions hampers limited consensus. We believe that in order for Bitcoin to ensure a stable consensus in the absence of block rewards, Bitcoin will have to maintain a stable stock of unconfirmed transactions and thus prevent a situation in which the capacity of the blockchain exceeds the appearance of new transactions.

Further research can extend our current work in many meaningful directions. First, our simulations could be directly extended by allowing different strategy regimes, as we have only focused on global and random strategies, possible considering the payoff of miners to evaluate and update their strategies. Second, the simulation time of our models could be extended and enhanced with long term processes such as nodes entering or leaving the network and the adjustment process of the hashing difficulty. Third, the modelling of transactions may be extended to incorporate distinct transactions fees and sizes [19]. This could be tied with an more realistic model of blocks, where block rewards are made explicit and the block size limit is modelled realistically [20]. Additionally, the attack strategies of malicious agents and the response of rational agents could be further considered in our model [21], [22].

The study of the incentive regime transition in Bitcoin will increase both the technological and the economic importance of Bitcoin. Our research suggests that the transition from a block reward to a transaction fee incentive system is not without disturbances and consensus may be at risk. Only under particular conditions consensus may arise.

References

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

[2] P. Tasca and C. J. Tessone, "A taxonomy of blockchain technologies: Principles of identification and classification," *Ledger*, vol. 4, Feb. 2019. [Online]. Available: https://ledger.pitt.edu/ojs/ledger/article/view/140

[3] C. J. Tessone, P. Tasca, and F. Iannelli, "Stochastic modelling of blockchain consensus," *Available at SSRN 3865040*, 2021.

[4] A. S. Teixeira, F. C. Santos, A. P. Francisco, and F. P. Santos, "Eliciting fairness in n-player network games through degree-based role assignment," *Complexity*, vol. 2021, 2021.

[5] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[6] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A survey on blockchain: A game theoretical perspective," *IEEE Access*, vol. 7, pp. 47 615–47 643, 2019.

[7] S.-N. Li, Z. Yang, and C. J. Tessone, "Mining blocks in a row: A statistical study of fairness in bitcoin mining," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2020, pp. 1–4.

[8] ——, "Proof-of-work cryptocurrency mining: a statistical approach to fairness," in *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*. IEEE, 2020, pp. 156–161.

[9] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 436–454.

[10] C. Li, F. Spychiger, and C. J. Tessone, "The miner's dilemma with migration," in *2021 3rd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, 2021, pp. 97–104.

[11] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 154–167.

[12] I. Tsabary and I. Eyal, "The gap game," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 713–728. [Online]. Available: https://doi.org/10.1145/3243734.3243737

[13] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*. IEEE, 2013, pp. 1–10.

[14] M. A. Nowak, K. M. Page, and K. Sigmund, "Fairness versus reason in the ultimatum game," *Science*, vol. 289, no. 5485, pp. 1773–1775, 2000.

[15] D. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of computational physics*, vol. 22, no. 4, pp. 403–434, 1976.

[16] ——, "Exact stochastic simulation of coupled chemical reactions," *The journal of physical chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[17] C. Schwarz-Schilling, S.-N. Li, and C. J. Tessone, "Agent-based modelling of strategic behavior in pow protocols," in *2021 Third International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 2021, pp. 111–118.

[18] A. E. Gencer, S. Basu, I. Eyal, R. v. Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," in *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 439–457.

[19] D. Zhou, N. Ruan, and W. Jia, "A robust throughput scheme for bitcoin network without block reward," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019, pp. 706–713.

[20] A. Meynkhard, "Fair market value of bitcoin: Halving effect," *Investment Management and Financial Innovations*, vol. 16, no. 4, pp. 72–85, 2019.

[21] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 706–719.

[22] K. Qin, L. Zhou, and A. Gervais, "Quantifying blockchain extractable value: How dark is the forest?" *arXiv preprint arXiv:2101.05511*, 2021.