Decentralization Analysis of Pooling Behavior in Cardano Proof of Stake

Aggelos Kiayias
Aggelos.Kiayias@ed.ac.uk
University of Edinburgh, IOG
Edinburgh, United Kingdom

Christina Ovezik
covezik@ed.ac.uk
University of Edinburgh
Edinburgh, United Kingdom

ABSTRACT

Blockchain protocols' main differentiator is their purported *decentralization* that unlocks various information technology applications that were supposedly impossible beforehand. The key promise is that incentive-driven participation of a large set of interested parties can lead to decentralized protocol states where no single operator can be a "single point of failure." Despite this promise, there is little systematic analysis of decentralization in blockchain systems and the sporadic theoretic and empirical investigations that exist paint a rather negative picture due to resource "pooling behaviors" that are impossible to prevent in the "permissionless" setting of such protocols where parties have no designated identities.

Motivated by this, in this paper we study the Nash dynamics of pooling in the context of Proof of Stake systems, following an agent-based modeling approach. Our focus is the Cardano blockchain as it features a number of attractive characteristics making it conducive to an in-depth analysis. We aim to answer the question of whether the incentive mechanism employed is capable of promoting decentralization. To this end, we present a simulation engine that enables strategic agents to engage in a number of actions empirically observed in the real-world deployment of the system. The engine simulates the "stake pool operation and delegation game" via successive agent actions that improve their utility as more information about their environment becomes evident in the course of the simulation. We investigate convergence to equilibrium states, and we measure various decentralization metrics in these states, such as the Nakamoto coefficient, which asks how many independent entities exist that collectively command more than 50% of the system's resources. Our results exemplify the ability of the incentive mechanism to steer the system towards good equilibria and also illustrate how the decentralization features of such equilibria are affected by different choices of the parameters used in the mechanism and the distribution of stake to participants.

CCS CONCEPTS

• **General and reference** → **Experimentation**; **Evaluation**; • **Security and privacy** → Systems security.

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11.
https://doi.org/10.1145/3533271.3561787

KEYWORDS

proof of stake, blockchains, incentives, reward sharing schemes, decentralization, simulation, cardano

ACM Reference Format:
Aggelos Kiayias and Christina Ovezik. 2022. Decentralization Analysis of Pooling Behavior in Cardano Proof of Stake. In *3rd ACM International Conference on AI in Finance (ICAIF'22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561787

1 INTRODUCTION

Blockchain protocols, introduced in Nakamoto [21], put forth a new paradigm for the deployment of information technology services. In such *permissionless* systems, agents register themselves to become system maintainers, incentivized by the provision of rewards in the form of digital tokens, which can be subsequently exchanged for offsetting costs and for profit. While the original Bitcoin system offers mainly a digital store of value and payment system, follow-up system development put forth a wide array of applications including Smart contracts [26], name registries [7] and cross border transfers [4], to name a few. The energy consumption problem of the Bitcoin protocol and related systems, see e.g., [9], led early on to the exploration of alternative disciplines for blockchain design, notably Proof of Stake (PoS), cf. [3, 18], and currently a number of systems are based on PoS, such as Cardano, Tezos, Algorand, Polkadot and Ethereum 2.0.

A key feature and consideration in all these systems is the ability of participants to "pool" resources together, (cf. [8, 20] for the case of Bitcoin), that can have the adverse effect that the system eventually centralizes to a handful (or in the extreme case just a single) operator(s), cf. [1, 17, 19]. This is a serious downside, as it removes one of the key supposed advantages of blockchain systems - their *decentralization*. The phenomenon poses a real threat since when the resources controlled by a single entity exceed a threshold (typically 50%) the entity has full control of the whole system in the sense of being capable of censoring or reverting transactions. Importantly, this is not only a theoretical consideration as it has also been observed empirically, cf. [10].

Techniques that aim to mitigate the above "pooling towards centralization" issue are scarce. One of the main problems is the lack of effective methodologies to validate mechanisms that aim to promote decentralization. The main obstacle is the complexity of the interactions between the agents. The emerging "game" is one of incomplete information, between thousands of agents, each possessing a different amount of resources and having distinct operational costs when running the protocol; the resulting strategy

space is vast and it is hard to analyze its equilibria and other game-theoretic characteristics.

Our results. We study the Nash dynamics of pooling in the context of Proof of Stake systems, adopting an empirical agent-based modeling approach. We focus on the Cardano PoS blockchain as it offers a number of attractive characteristics for the purpose of this analysis: (i) it uses an "on-chain" pooling mechanism that is formally documented [14] and leaves a trace of public information regarding how participants interact with the mechanism, (ii) it comes with an equilibrium analysis [2] that may consider a restricted set of strategies, but still offers a good theoretical foundation over which one can apply further game-theoretic analysis compared to other systems, (iii) it has been deployed in the mainnet of Cardano since August 2020 [12] allowing a sufficient time for empirical observations regarding the strategies that participants follow in the real world[1], (iv) Cardano has consistently ranked among the top ten cryptocurrencies in terms of market capitalization since August 2020 thus ensuring that the rewards allocated by the mechanism have real world value and hence participants meaningfully influence their utility by engaging with the system.

Our analysis has as its key overarching objective to answer the question: does the incentive mechanism employed by Cardano promote decentralization? To measure decentralization we consider a number of metrics and analytical tools. First, we use a metric called the *Nakamoto coefficient*, introduced in [24], that adapts well-established economic metrics to the context of cryptocurrencies. In a nutshell, the Nakamoto coefficient is defined as the minimum number of independent operators whose aggregate influence of the system exceeds the 50% barrier and hence, should they collaborate, they can have complete control over the entire system (we remark that this stems from the properties of the Bitcoin protocol, but the threshold is also shared by the Ouroboros protocol [16] employed in Cardano). The Nakamoto coefficient however is only one dimension of the problem at hand; a second, but arguably equally important metric, is how much stake is possessed in aggregate by such a set of operators. This leads to another metric, that we call *min-aggregate pledge*, which measures the resources held by the set of operators whose controlled stake exceeds 50%.

In order to estimate the above metrics, we distill a strategy for an agent using characteristics that were manifested in real world system operation: (i) agents are utility maximizers and will be modifying their strategy provided an alternative option exists that exceeds a certain threshold in terms of utility improvement, (ii) agents only have access to information available on chain (the game is of incomplete information) and hence they adapt their strategy over time as they become aware of other agents' profiles, (iii) some agents may not participate due to various barriers in participation (e.g., tax implications, engagement in DeFi smart contracts etc.) (iv) pool operators may create a large number of pools (what in the Cardano community has been referred to as "pool splitting", see [15]) in an attempt to extract higher rewards from the system.

In our model, a player is further described by two additional values: the player's stake and the player's operational cost, should they decide to become system operators. These two values are chosen at random from suitable probability distributions. We develop a simulation engine that allows us to examine the dynamic interaction of any given number of players. The engine uses a random schedule and allows the agents to engage in successive refinement of their position, adjusting their strategy as information about the other agents is revealed. We utilize domain knowledge of the Cardano system to improve the complexity of strategy selection and we allow the players a degree of far-sightedness in the way they assess the current state of the system and select their next move.

We conduct an array of experiments that examine whether the system converges to an equilibrium and we study the final configurations in terms of their decentralization properties. Of particular interest is the impact of the two parameters of the reward sharing scheme of [2], α, k, that are meant to adjust the level of decentralization offered by the mechanism. We demonstrate for the first time experimentally the impact of these parameters on the Nakamoto coefficient of the system. Our results exemplify the ability of the system to converge towards equilibria that exhibit good decentralization, given a suitable parametrization.

While the results we present paint a relatively favorable picture, they also reveal the sensitivity of the system's properties to parameter choices and the stakeholder distribution; they further hint that the current choice of Cardano's α parameter is at the bare minimum possible to produce good properties at equilibrium in terms of decentralization when we deploy the simulation with a *synthetic distribution* which is derived in part from actual data from the live system. Moreover, our results can inform the current public discourse regarding the desired setting of the k parameter, that even though was initially proposed to increase in March 2021, cf. [13], this has since being delayed, generating a debate in social media.[2]

We note that while our results are stated in the context of Cardano, they apply to any system that uses the family of reward sharing schemes from [2], such as Nym[3] [5]. Moreover, our simulation engine has been publicly released[4], offering opportunities for incorporating additional functions and comparing different schemes for rewards sharing.

Related work. The work most related to ours is [2], which put forth the mechanism used in Cardano and an initial game-theoretic and experimental analysis. Their analysis had the significant limitation in the way it excluded operators running multiple pools as a strategic option (even though they estimated the impact of Sybil attacks [6] at the state of equilibrium), assumed full participation and considered parameters fixed throughout. Our approach shares a common foundation with Empirical Game Theoretic Analysis [25], in the sense that it simulates repeated strategic interactions of agents that are sampled probabilistically from a large space. Nevertheless, our objective is not to extract an empirical game payoff matrix, but rather study the Nash dynamics of the agents and whether they converge to a "sink", cf. [11] or a sink connected component in the sense of [22] that has good decentralization properties.

Despite the proliferation of blockchain systems with various reward sharing schemes, very little is known in terms of the ability of

[1] Various third party information aggregators exist that cover how pooling is performed in Cardano, e.g. pooltool.io or adapools.org.

[2] See e.g., forum.cardano.org/t/k-parameter-to-1000-still-planned/64130 or cardanofeed.com/cardano-spo-column-hodler-coalition-hodlr-70308.

[3] See nymtech.net.

[4] See Github.com/Blockchain-Technology-Lab/Rewards-Sharing-Simulation-Engine.

such mechanisms to converge to configurations that have good decentralization properties. Moreover, given that current theoretical, cf. [17] and empirical observations, cf. [10] suggest that Bitcoin's reward sharing has strong tendency to centralize, it is important to improve the analytical toolkit and assess alternative mechanisms regarding the degree to which they promote decentralization.

2 THEORETICAL MODEL

In this section, we present the reward mechanism used in Cardano and we extend the game-theoretic model that has been used so far for its analysis.

Rewards Sharing. Brünjes et al. [2] introduce a delegative reward sharing scheme with *capped rewards* and *incentivized pledging* for PoS blockchains. In this setting, any actor that holds stake in the system can engage with the protocol, either directly by operating their own stake pool or indirectly by delegating their stake to pools of their choice. A pool's size is determined by the stake that the pool owner bonds to the pool —referred to as the pool's *pledge*— and the stake that other agents delegate to it. The rewards that a pool receives depend both on its pledge and total size, growing with them until a saturation threshold is reached, after which they level off. Specifically, the rewards of a pool with pledged stake λ and total stake σ are dictated by this piecewise function:

$$r(\sigma,\lambda) = \frac{R}{1+\alpha}(\sigma' + \lambda'\alpha\frac{\sigma' - \lambda'\frac{\beta-\sigma'}{\beta}}{\beta})$$

where $R \in \mathbb{R}$ are the total available rewards for this epoch, $\alpha \in [0,\infty)$ is the pledge influence factor, $\beta = \frac{1}{k}$ is the pool saturation threshold ($k \in \mathbb{N}$ is the target number of pools) and $\lambda' = min\{\lambda,\beta\}$, $\sigma' min\{\sigma,\beta\}$ are the pledge and total stake of the pool, but capped at β. Since R, α and k are fixed for a system, the rewards of different pools vary based on their pledge and total stake.

Observe that a pool with some fixed pledge earns the highest possible rewards when its total stake reaches the threshold ($\sigma = \beta$), in which case it is called *saturated*. An *oversaturated* pool ($\sigma > \beta$) still earns the same rewards, meaning that the excess stake does not contribute to the generation of rewards. Hence, we can view β as a *soft cap* on a pool's size, in the sense that it does not "forbid" pools to grow beyond this threshold, but it discourages them from doing so by reducing the rewards *per unit of stake* that the pool receives once the threshold is exceeded. Capping the rewards aims to prevent the formation of very large pools, which are a threat to the system's decentralization. Setting $\beta = \frac{1}{k}$ targets the formation of k pools of equal size[5].

However, a participant might circumvent this cap by operating multiple pools instead of one (a behavior that has become known as "pool splitting" in Cardano [15]), which can also be seen as a Sybil attack [6] against the system if done covertly, i.e., if an operator assumes multiple identities through their pools. This is the reason why another parameter was added to the model (α), which controls the influence of a pool's pledge (the stake that the operator has agreed to "lock" in their pool, denoted by λ here) on the rewards it receives. This parameter, when given a non-zero value, results in higher-pledged pools receiving higher rewards per unit of stake compared to lower-pledged ones (a difference that grows for higher values of α). Granting higher rewards to pools with higher pledge aims to induce operators to concentrate their stake in a single pool, thereby deterring Sybil attacks (which is why α is also referred to as the *Sybil resilience* parameter).

As a further step, the protocol also handles the distribution of rewards to the individual pool members, removing the need for additional trust in the pool operators. When creating a pool, the pool owner needs to declare its operational cost. When a pool's reward gets calculated, an amount that corresponds to the declared cost is first set aside for the pool operator, to offset that cost. If the reward is not sufficient to cover the pool's cost, then no rewards are distributed to its members. Note that, in this case, the pool operator suffers a loss (as they have to pay the operational costs anyway), whereas the pool members do not.

To compensate operators for the added risk they have to bear and to further incentivize pool creation, the protocol allows them to set a value of their choice —referred to as the pool's *margin*— that determines the fraction of the pool's profits that will be allocated to the operator before any additional distribution. The remaining fraction of the pool's profits get distributed to its members proportionally to the stake they contribute. Recall that as a pool grows, the rewards that are issued to it increase, and therefore the rewards of its members increase, too. However, if a pool's size exceeds the saturation point β, then its rewards stop growing, meaning that its rewards per unit of stake —and therefore the rewards of individual members— decrease.

Formally put, if a pool has stake σ, cost c, margin m and pledge λ, then, each delegator that contributes stake a_d to the pool receives:

$$r_d = \begin{cases} (1-m)(r(\sigma,\lambda)-c)\frac{a_d}{\sigma} & \text{if } r(\sigma,\lambda) > c \\ 0 & \text{otherwise} \end{cases}$$

And the pool owner, who contributes stake $a_o = \lambda$ to the pool, receives:

$$r_o = \begin{cases} c + m(r(\sigma,\lambda)-c) + (1-m)(r(\sigma,\lambda)-c)\frac{\lambda}{\sigma} & \text{if } r(\sigma,\lambda) > c \\ r(\sigma,\lambda) & \text{otherwise} \end{cases}$$

where $r(\sigma,\lambda)$ are the rewards that a pool with stake σ and pledge λ is entitled to.

Game description. Granted such a rewards sharing scheme, we can define the "Stake Pools Game", with the players being stakeholders of the system, their strategies describing how they engage with the protocol (operate a pool with a certain margin and pledge or delegate their stake) and their goal being the generation of profit. Importantly, we augment the definition given by Bŕunjes et al in [2], to incorporate strategies where a player operates multiple pools.

We define the strategy S_i of player i as the following quadruple:

$$S_i = (t_i, \mathbf{m}_i, \boldsymbol{\lambda}_i, \mathbf{a}_i)$$

where $t_i \in \mathbb{N}$ is the number of pools that player i wishes to operate, $\mathbf{m}_i = (m_{i,1},\ldots,m_{i,t_i})$, $m_{i,j} \in [0,1]$ are the margins that the player imposes to their t_i pools, $\boldsymbol{\lambda}_i = (\lambda_{i,1},\ldots,\lambda_{i,t_i})$, $\lambda_{i,j} \geq 0$ are the pledges that the player commits to their pools and $\mathbf{a}_i = (\mathbf{a_{i,1}},\ldots,\mathbf{a_{i,n}})$ describes the allocation of player i's stake to the

[5]Note that we consider the total stake of the system to be 1 for our purposes, i.e., all stake values used are relative to the total supply.

pools of the n players, where $\mathbf{a_{i,j}} = (a_{i,j_1}, \ldots, a_{i,j_{t_j}})$ is the allocation of player i's stake to the t_j different pools of player j. It holds that $\sum_{j=1}^{n} \sum_{k=1}^{t_j} a_{i,j_k} \leq s_i$, meaning that each player can allocate part or all of their stake (but obviously no more than that) to the pools of the system. If $t_i > 0$ then $a_{i,i_k} = \lambda_{i,k}$ for $k \in \{1, \ldots, t_i\}$ else $\mathbf{a_{i,i}}$ is a null vector (as is every $\mathbf{a_{i,j}}$ when $t_j = 0$ for $j \in \{1, \ldots, n\}$). Note that our strategy definition allows for a player to both operate pools and delegate stake to other pools at the same time. Since the game transpires over several rounds, the above definition represents player i's strategy during a snapshot of the game.

An intuitive utility function to use for evaluating a strategy is the simple addition of all rewards a player accumulates through their stake allocations, minus any costs they bear (only relevant for pool operators). We build upon this notion by allowing our players a degree of far-sightedness. To accomplish this, we employ the *non-myopic utility function* described in [2], which takes into consideration the rewards of a pool based on its expected future stake, instead of its current one. To estimate the future (non-myopic) stake of a pool, Brünjes et al define the notions of *potential profit*, *desirability* and *rank*, which we borrow for our model [6].

The *potential profit* of a pool with pledge λ_j and cost c_j is:

$$P(\lambda_j, c_j) = r(\beta, \lambda_j) - c_j$$

and it represents the highest profit that a pool with this pledge and operational cost can yield (the profit it receives at saturation).

The *desirability* of the pool is:

$$D_j = \begin{cases} (1 - m_j)P(\lambda_j, c_j) & \text{if } P(\lambda_j, c_j) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

and it represents the maximum profits that delegators can earn from this pool. Note that this value depends on the margin m_j, so by choosing lower margins, pool owners make their pools more desirable. We can rank pools based on their desirabilities, so that the pool with the highest desirability is assigned rank 1, the second highest is assigned rank 2, and so on. Then, the *non-myopic stake* of a pool with rank j, current stake σ_j and pledge λ_j is defined as:

$$\sigma_j^{NM} = \begin{cases} max(\beta, \sigma_j) & \text{if } j \leq k \\ \lambda_j & \text{otherwise} \end{cases}$$

This shows that the top k ranked pools are expected to become saturated, while the rest are expected to end up only with their owner's pledge, as delegators side with more competitive pools.

The expected utility of a stakeholder that delegates stake $a_{i,j}$ to the pool with rank j is then:

$$u_{i,j} = \begin{cases} \max((1 - m_j)(r(\beta, \lambda_j) - c_j)\frac{a_{i,j}}{\sigma_j^{NM}},\ 0) \text{ if } j \leq k \ \&\ \lambda_j \neq 0 \\ \max((1 - m_j)(r(\lambda_j + a_{i,j}, \lambda_j) - c_j)\frac{a_{i,j}}{\lambda_j + a_{i,j}}, 0) \text{ otherwise} \end{cases}$$

And the expected utility for the owner of the pool, who contributes pledge λ_j is:

$$u_{j,j} = \begin{cases} r(\sigma_j^{NM}, \lambda_j) - c_j & \text{if } r(\sigma_j^{NM}, \lambda_j) < c_j \text{ and } \lambda_j \neq 0 \\ (r(\sigma_j^{NM}, \lambda_j) - c_j)(m_j + (1 - m_j)\frac{\lambda_j}{\sigma_j^{NM}}) & \text{otherwise} \end{cases}$$

[6]For simplicity, we keep the definitions as they were presented in [2], with only one index to refer to a pool's properties (e.g. λ_j instead of λ_{j_k} for the pledge of a pool owned by player j), but they can all be trivially extended to match our multi-pool-strategy notation from the previous paragraph.

The total (non-myopic) utility of a player i is then calculated as the sum of the individual utilities from all pools they participate in: $u_i = \sum_{j=1}^{n} u_{i,j}$. Observe that a pool operator's expected payoff can be negative, while a delegator's cannot, highlighting the fact that it is only operators who risk having losses.

In the case of operating multiple pools, agents assume that an additional pool yields a lower cost than their first one when calculating their expected payoffs. This is a direct effect of the so-called *economies of scale*, which make it possible to produce more units of a good or service with lower costs per unit on average [23]. In our case, the cost of the first pool represents the first big investment (e.g. for equipment purchases) and every additional pool only increases the total cost by a certain factor (e.g. for key management). To incorporate this in our model, we define an additional parameter, the *cost factor* ϕ, which represents the cost of an additional pool as a fraction of an agent's initial operational cost. Therefore, if player i has an initial cost value c_i and operates t_i pools, then the average cost of these pools is $\frac{1+(t_i-1)\phi}{t_i} c_i$.

3 SIMULATION ENGINE

To simulate how the above game evolves for different parameter values and initial states, we develop an engine that can be configured to play out the game under any conditions of our choice. We employ an agent-based model as the basis for the simulation, defining simple actions for individual agents (micro-behavior) and observing the complex patterns that emerge through their interactions and their effect on the system's properties (macro-behavior). In this section, we explain how a simulation instance unfolds, how agents choose their strategies, and what is the reasoning behind these decisions.

First, the simulation constructs the initial conditions of the system and its participants, creating the chosen number of agents and assigning to each of them some stake and a cost value (which represents the cost of their first pool, should they choose to open one); typically, the former is sampled from a Pareto distribution, and the latter from a Uniform distribution, however the simulation supports other options as well. The system starts off without any pools and, at every step, each agent decides on a strategy for how to use their stake, based on the state of the system and any information that has been made public by the other agents —such as the operational cost of an agent that owns a pool— but without having access to the private information of other agents —such as the potential operational cost of an agent that does not own a pool at that point in time. This process goes on, until all agents have settled to their strategies, i.e., a state of equilibrium is reached, or until a predetermined number of maximum rounds is reached.

In each round, all agents are given the chance to update their strategy; to ensure fairness and make the process more realistic, players are activated in random order[7]. Upon their turn, each agent examines a handful of strategies, including delegation and pool-operation strategies, and chooses the one that yields the highest expected utility among those, or keeps their current one if the new ones do not exceed its utility by a certain threshold. If there is a tie between two moves, the one that requires less "effort" is chosen (keeping the current strategy is preferred over delegation,

[7]A seed can be set to get a reproducible order and results.

which is in turn is preferred over pool operation). In the following paragraphs, we explain how the agents narrow down the search space for potential moves and end up with a few options to compare.

Determining delegation moves. In order to choose which pool(s) to delegate their stake to, agents start by calculating the desirability of all active pools. Recall that the desirability of a pool represents the share of the profits that is allocated to delegators, so it makes sense for players to want to join pools with high desirability. Therefore, the prospective delegators rank the pools based on their desirability and choose to delegate their stake to the highest-ranked pool(s), prioritizing the ones that are not already saturated.

Determining pool-operation moves. A strategy that involves pool operation is more complex, as it requires from an agent to decide how many pools to run (given that multi-pool strategies are permissible in our model), and then also choose suitable values for the pledge and profit margin of each pool. To find a promising solution, agents employ a local search similar to hill climbing, which transpires as follows for an agent with stake s:

Algorithm 1 Determining a pool-operation strategy for an agent with stake s

1: Set the minimum number of pools $t_{min} = 1$ and the maximum number of pools $t_{max} = k$.
2: Select $t = \left\lfloor \frac{t_{min}+t_{max}}{2} \right\rfloor$ as the current number of pools to examine.
3: Allocate $\frac{s}{t}$ as pledge to each pool.
4: Calculate the average cost per pool.
5: Calculate suitable margins so that all t pools end up in the top k — if not possible for some pools then set their margin to 0.
6: Calculate the agent's expected utility given these pools.
7: Repeat steps 3 - 6 for the two neighboring solutions of this strategy, i.e., operating $t-1$ pools and operating $t+1$ pools.
8: Choose the highest utility of the three options. If that corresponds to $t-1$ (or $t+1$) then set $t_{max} = t-1$ (or $t_{min} = t+1$) and go to step 2. Otherwise, terminate with found solution (strategy with t pools that have the calculated pledges and margins).

To clarify the margin calculation process (step 5 above), the agent ranks all pools that belong to other agents based on their desirability and attempts to surpass the desirability of the k-ranked pool with their first pool, the desirability of the $(k-1)$-ranked pool with their second pool, and so on, up to pool t which attempts to beat the desirability of the $(k+1-t)$-ranked pool. Since they attempt to replace different pools from the top k, pools that belong to the same agent may end up with distinct margin values.

4 DECENTRALIZATION ANALYSIS & PARAMETER CHOICE

We use the simulation engine that we described above to run a series of experiments, and we analyze the results to gain insights about the system. In this section, we track certain behaviors that arise among the agents and explore the impact they have on the emergent properties of the system. We explore the decentralization properties of the system and investigate the relationship between its input (parameter values, stakeholder characteristics) and its eventual degree of decentralization.

For most experiments that we discuss below, we populate the system with $n = 1000$ agents, with their stake values drawn from a Pareto distribution[8] with shape parameter 2 and their cost values from a Uniform distribution so that each agent's cost is 4 to 5 orders of magnitude smaller than the total available rewards R [9]. We assume a cost factor $\phi = 0.4$, meaning that every additional pool of an agent costs only 40% the cost of their first pool. This specific value of ϕ was chosen for being presumably realistic, but later on we show that the model is not sensitive to this choice (see last paragraph of this section). Notably, and consistently with [2], all experiments that we conducted eventually arrive at equilibria, which in itself is already a positive result.

Parameter impact on decentralization. We explore how the values of the reward sharing scheme's parameters, k and α, influence the eventual decentralization of the system at equilibrium. Recall that k represents the target number of pools at equilibrium, but it is never enforced (in real life or in the simulation) that the pools are exactly k. Instead, the reward mechanism of the system incentivizes the formation of k pools by capping a pool's rewards once its stake reaches a saturation threshold that depends on k, and, because of that, the rational play we explore through the simulations typically convergences to an equilibrium of k pools. On the other hand, α was introduced as a "Sybil resilience" factor, as it defines the degree to which the pledge of a pool affects the rewards it receives. Intuitively, we can tell that the higher the value of α, the bigger influence the pledge has on a pool's reward, therefore the less likely an operator is to create multiple pools. There have been theoretical results that support this, but their scope was limited because of strict assumptions [2]. This time, we explore these relationships in a more realistic setting and gain insights on the impact of these parameters on the system's decentralization.

Before conducting any analysis, we should highlight that decentralization operates on a spectrum, hence the question is not whether a system is decentralized or not but rather to what degree it is, and for this reason, we need to define in which ways we are going to measure decentralization. Unlike the analysis of [2], we do not consider the number of pools indicative of the system's decentralization, as in our model (and in reality) it is possible for a single stakeholder to control multiple pools—for example, a configuration with k pools at equilibrium which are all controlled by the same agent should by no means be considered decentralized.

The first metric we employ is the Nakamoto coefficient, which has been widely used in the context of blockchains and represents the minimum number of independent entities that collectively control more than 50% of the system's resources, and can therefore launch a successful attack if they collude [24]. For PoS blockchains, this translates to controlling the majority of the total active stake. In our case, where we expect to have k pools of equal size at equilibrium, the *ideal Nakamoto coefficient* would be $\frac{k}{2} + 1$, implying that all k pools are independently owned, i.e. there is no pool splitting

[8] Distributions of wealth are typically modeled using a Pareto distribution.
[9] We express the costs as fractions of the total rewards for easier generalization, as the utility function we use allows for it.

at equilibrium —in practice, however, this value can diverge by a lot from the ideal one.

Figure 1: Nakamoto coefficient heat map - Pareto stake distribution

To observe this phenomenon and better understand its relationship with the parametrization of the system, we run a set of 50 experiments, with different combinations of k and α in each, while keeping everything else fixed. Specifically, we examine 5 values for k, ranging from 100 to 500, and 10 values for α spaced evenly on a logarithmic scale between 0.01 and 1 (note that Cardano currently uses $\alpha = 0.3$). We repeat this set of experiments 5 times to account for the impact of the random schedule and we combine the results of identical executions suitably. The heat map of Fig. 1 reveals the Nakamoto coefficient of the final configuration of the system, for each combination of k and α.

Our experiments confirm that the values of α and k play pivotal roles in the eventual decentralization of the system. We observe that a very low α (0.01 in our experiments) is not sufficient to deter pool splitting, resulting in extremely low Nakamoto coefficients, regardless of the value of k. Once the value of α grows to become more effective against pool splitting, we remark that the value of k becomes the decisive factor for the system's decentralization, while the increasing value of α has a marginal impact, if any, at that point.

In general, it appears that the Nakamoto coefficient grows with k, revealing that more and more stakeholders become operators, as the target number of pools rises. However, even though it is tempting to think that "the higher the k value the better", there are a few reasons why this may not always be the case. One caveat is that there exists a natural limit on the number of stakeholders that can become pool operators, imposed by the combinations of stake and operational cost they have to begin with. For some stakeholders with high costs, operating a pool may never be profitable, independent of the value of k. Note that, in the real world, there are even more barriers to entry than the lack of profitability, such as the lack of technical knowledge or time to set up and maintain a stake pool.

From our heat map, we observe another reason why a higher k is not always better. For the values of α that are on the border of being effective (0.017 - 0.046 in our example), we observe that the Nakamoto coefficient peaks for some k value, after which any increase in k only makes it drop. We can interpret this as follows: when α is not sufficiently high, a certain number of stakeholders with suitable stake and cost values may be able to profit more from operating multiple pools compared to operating a single pool; however, if their pools are not competitive enough to reach the top k, then they may end up not operating a pool at all, as they would not receive delegations; when k is stretched, it makes it more likely for this type of agents to belong in the top k, which is how we can end up with many pool splitters and a very low Nakamoto coefficient.

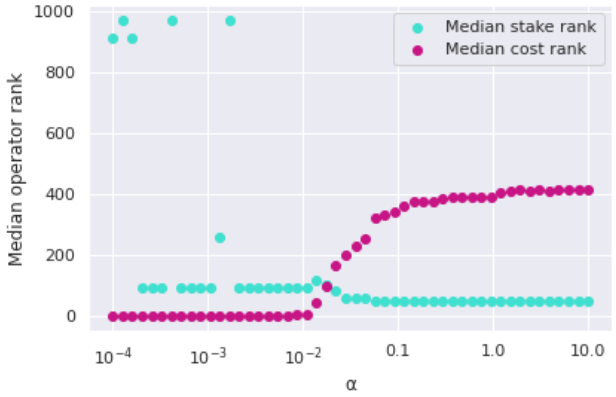

Figure 2: Median cost-rank and stake-rank of pool operators for different α values ($k = 100$)

A similar principle applies for α: while a higher value of α promises a higher degree of decentralization, there are other properties of the system that are adversely affected by it. In Fig. 2, we track the median cost rank and stake rank of the stakeholders that operate pools at equilibrium, for executions with different values of α. Note that a cost rank of 1 corresponds to the agent with the lowest cost, while a stake rank of 1 corresponds to the agent with the highest stake. From there, we conclude that for low values of α, the stakeholders that become pool operators are those with the lowest operational costs, regardless of the magnitude of their personal stake. Higher values of α, however, result in configurations where it is mainly the "richest" stakeholders that get to control the pools, and not the most cost-efficient ones.

It follows that choosing a very high value for α is not necessarily good approach for the system. Even though such a value can provide a more robust defense against Sybil attacks, it also exacerbates the "rich getting richer" phenomenon, which is an inherent issue of Proof of Stake. This trade-off between Sybil resilience and egalitarianism is vital to keep in mind when selecting the parameter values of a real-world system. Furthermore, our heat map suggests that once the right threshold for α is reached, any increase only has a marginal impact—or none at all—on the decentralization of the system, hence choosing an extremely high value would be pointless in this regard.

The second metric we use in an attempt to quantify the concept of decentralization is what is called *min-aggregate pledge*. Again,

we examine the sets of pools that exert important influence in the system (controlling more than 50% of the active stake), but instead of looking for the minimum number of stakeholders in such a set, this time we focus on the set with the lowest collective pledged stake. This aims to give an indication of how much "skin in the game" these powerful pool operators have, and therefore how much they would risk by turning against the system.

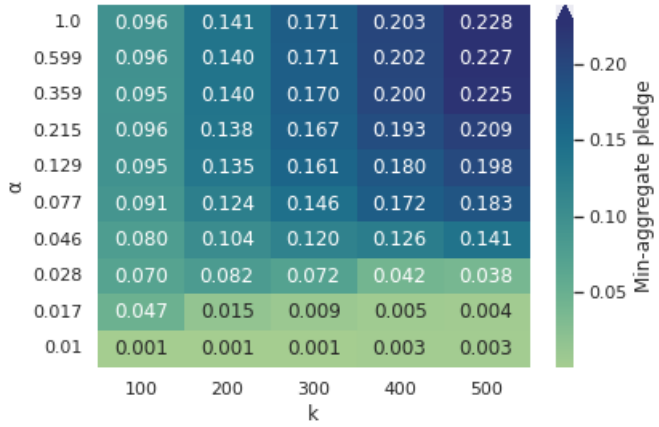

Figure 3: Min-aggregate pledge heat map

Using the same set of experiments as above, we track the min-aggregate pledge of the system at equilibrium for various combinations of k and α. We can see the results in the heat map of Fig. 3, which resembles a lot the one of Fig. 1. Note that stake values are expressed relative to the total stake, therefore a min-aggregate pledge of 0.1 means that any group of pool operators that jointly control the majority of the stake through their pools own at least 10% of the system's total stake and have pledged it to those pools.

From this angle, we observe again the patterns of a very low α being ineffective, the importance of k growing with α and the border-line-effective α values that mandate an upper bound for k. This time we remark that an increase in α continues to trigger increases in the system's min-aggregate pledge, even after the initial phase transition.

Impact of initial stake distribution. In this section, we highlight the fact that a system's output can only be as good as its input, and we explore the effects of this principle on the potential decentralization of the system. From Fig. 1, we observed that (for suitable values of α) the system's eventual decentralization was high for any value of k, and yet as k grew, the Nakamoto coefficient diverged more and more from the "ideal" one of $\frac{k}{2}+1$. This begs the question of whether the mechanism can ever achieve this ideal result, where the stake of the system is distributed evenly among k independently operated pools. Recall that for the experiments we examined so far, we sampled the agents' stake values from a Pareto distribution. Though this assumption brought the simulation closer to the real-life system, it also introduced a bias for the configuration of pools at equilibrium (for example, if an agent is so rich that they can saturate a pool with their own stake, it makes sense for them to open a second pool). To explore the effectiveness of the mechanism in a more equitable scenario, we run the same set of experiments as above but with a different initial stake distribution, making sure this time that all agents start off with the same amount of stake.

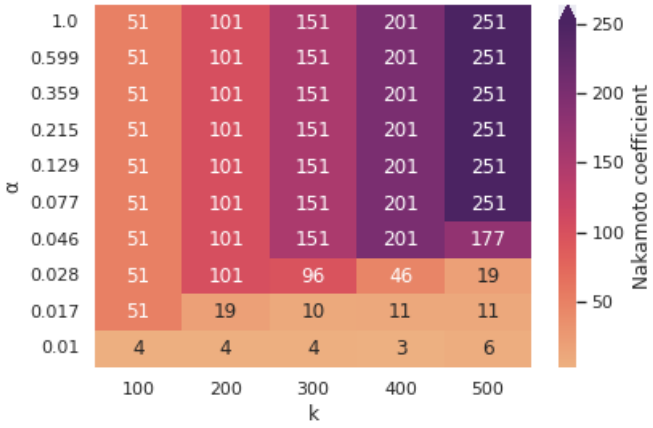

Figure 4: Nakamoto coeff. heat map - Flat stake distribution

The results, in terms of Nakamoto coefficient, can be seen in Fig. 4, where we once again witness the role of α and k in influencing the decentralization of the system. In general, a similar picture is painted as the one we saw in Fig. 1, only this time the phase transition leads to the ideal Nakamoto coefficient, where it takes $\frac{k}{2}+1$ distinct pool operators to accumulate the majority of the stake, compared to those of the experiment with the "biased" input that always fell short.

Satisfying as they may be, the results of Fig. 4 cannot be used to draw insights about the real world, as the assumption that all stakeholders start off with equal stake is highly unrealistic. In fact, even the Pareto distribution that we used before may fail to capture some complex patterns that emerge in real-world deployments, where staking involves a lot more actors, some of which have very particular characteristics, such as cryptocurrency exchanges. To achieve a more faithful representation of Cardano's deployment, we adjust the input in three ways. First, we increase the number of stakeholders from 1,000 to 10,000, allowing for higher diversity to develop within the population. Second, we provide explicit stake values for a number of stakeholders that were identified as exchanges, named entities or custody services in Cardano (the stake values of the remaining agents are still drawn from a Pareto distribution). Third, we mark close to $\frac{1}{4}$ of the total stake in the system as inactive, to account for all stakeholders that abstain from the protocol [10].

This time we also run the simulation for higher values of k, (note that Cardano's parameterization is currently at $k = 500$, while higher values are being considered by the community); because of the increased time requirements for these bigger experiments, we use fewer values for α, but the range remains the same. The results can be seen in the heat map of Fig. 5. The patterns that arise

[10]The information about the specific stake values and the fraction of inactive stake in Cardano was extracted from adapools.org and pooltool.io..

are similar to those we described in the previous paragraphs, with lower values of α failing to deter pool splitting, higher values of α pushing the "upper bound" for k and higher k values yielding higher decentralization when α is sufficiently high. However, we remark two key differences between these results and those of the Pareto distribution with 1000 stakeholders.

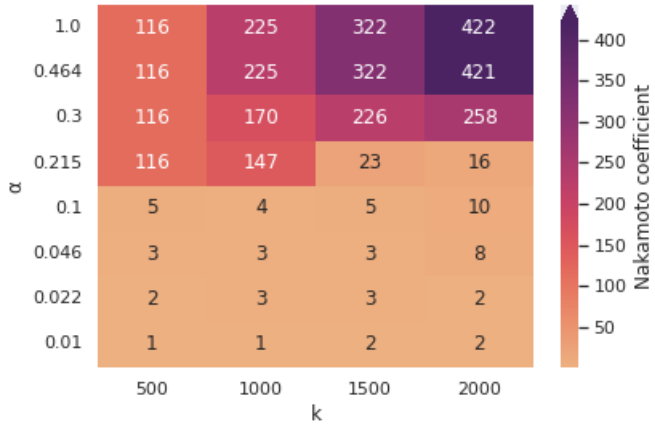

Figure 5: Nakamoto coeff. heat map - Synthetic distribution

For one thing, the peak of the Nakamoto coefficient values for $k = 500$ does not come near the corresponding peak of Fig. 1, implying that one of the modifications we made to the input lowered the ceiling for the achievable Nakamoto coefficient of the system. More importantly, we observe an overall shift towards higher values of α, meaning that α values that were considered effective in previous experiments are apparently inadequate in this more realistic setting. Notably, the value that has been chosen for Cardano's deployment, $\alpha = 0.3$ sits on the border of being effective, while higher values appear to achieve higher decentralization.

Sensitivity to cost assumptions in pool splitting. In all of the above experiments, we applied a cost factor $\phi = 0.4$, meaning that we perceived an agent's 2nd, 3rd, or any additional pool to cost 40% as much as their first one. Since this choice was based on intuition and rough estimations, we explore its potential impact on the results of the simulation: we re-run some of the experiments we conducted with the Pareto distribution for 10 additional values of the cost factor (11 in total, ranging from 0 to 1) and examine how the decentralization of the system is affected in terms of Nakamoto coefficient. For the experiments where k is kept fixed, we set it to 100 and similarly when α is fixed, we set it to 0.3. The results can be seen in Fig. 6 and 7; note that for every α value there are always 11 data points for the different cost factors, but they are not always visible, as the corresponding Nakamoto coefficients often coincide.

We observe that the properties of the system at equilibrium in terms of decentralization are not significantly influenced by the cost factor that we employ. Comparing to the results from Fig. 1 for $k = 100$, we observe that only when α is very low, the choice of ϕ has a noticeable impact. Specifically, we remark that if we assume a more optimistic cost factor (0.8 - 1), then the threshold for α being

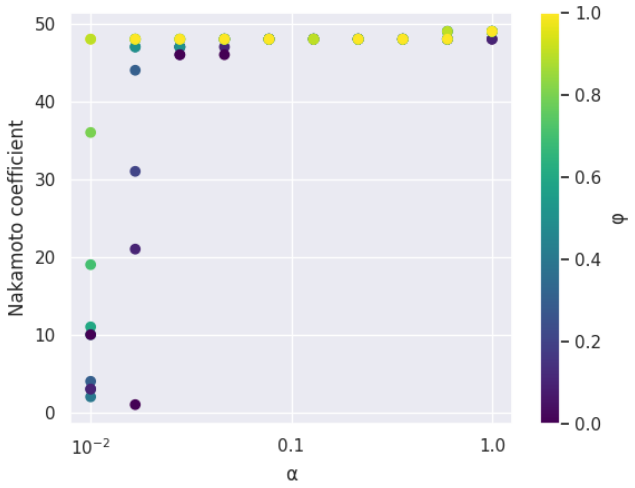

Figure 6: Nakamoto coefficient at equilibrium for executions with different α, ϕ values (with k = 100).

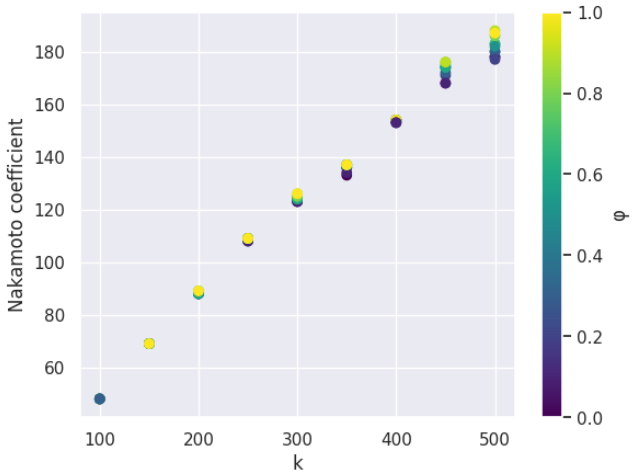

Figure 7: Nakamoto coefficient at equilibrium for executions with different k, ϕ values (with $\alpha = 0.3$).

effective is shifted down, while with a more pessimistic cost factor (0 - 0.2) the threshold is shifted up by one step. Remarkably, even with the most pessimistic assumptions, i.e. additional pools yielding no extra cost, the incentivized-pledging mechanism appears to deter pool splitting when α is not extremely low. A similar picture is painted by Fig. 7, where the value of ϕ seems to have a marginal —if any— impact on the result. From both figures, we conclude that the results are not particularly sensitive to the cost of additional pools for multi-pool operators.

5 CONCLUSIONS

In this work we study the decentralization properties that arise from staking and pooling behavior in the Cardano proof of stake

blockchain via an agent-based modeling approach. Our work studies a far more real-world relevant setting compared to the original analysis of [2] which, for the most part, was restricted to at most one pool per participant. Instead, our engine allows multi-pool strategies for all operators. Furthermore, we study the *Nakamoto coefficient* at equilibrium, a metric that captures decentralization in an essential way by disregarding nodes that are operated by the same participant. Despite the richer strategy space, our analysis reveals that the reward sharing scheme used in Cardano still converges to an equilibrium of k pools, but not necessarily independently owned ones.

Our work highlights how making suitable choices for the parameter values of the mechanism is essential for its decentralization properties. We examine the influence of the scheme's "pledge influence" parameter, α, and we present through experimental results the trade-off that it entails between egalitarianism and Sybil resilience. We showcase how the initial stake distribution of the system acts as a critical factor in the system's ability to achieve decentralization. We contrast the results between a hypothetical situation where all agents start off with the same stake and a more realistic situation using a synthetic distribution with some real-world data points. In this context, our work suggests that the choice of $\alpha = 0.3$, that is in the public Cardano implementation, is at the boundary of good decentralization behavior (see Figure 5).

Future directions include performing further experiments with even more realistic stakeholder and cost distributions and at a bigger scale (say, at 1M player levels), while expanding the simulation with additional behaviors, such as taking into account opportunity-loss due to alternative options regarding one's stake that are incongruent with staking, influencing the agents' utility by the operators' reputation and advertised "mission", modeling the effects of a volatile exchange rate, or allowing modifying the stakeholder distribution during the course of the execution. Furthermore, while our engine and outcomes are geared towards Cardano, our results immediately transfer to related schemes (e.g., NYM [5]), and the engine can be easily adapted to model other PoS systems, suggesting also the possibility for comparative analysis between reward schemes, which is an interesting research direction. Finally, formalizing the concept of decentralization, and developing novel metrics that help quantify it, presents itself as an intriguing course of action.

ACKNOWLEDGMENTS

This work was supported by Input Output (iohk.io) through their funding of the Edinburgh Blockchain Technology Lab.

REFERENCES

[1] Nick Arnosti and S. Matthew Weinberg. 2019. Bitcoin: A Natural Oligopoly. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA (LIPIcs, Vol. 124)*, Avrim Blum (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 5:1–5:1. https://doi.org/10.4230/LIPIcs.ITCS.2019.5

[2] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. 2020. Reward sharing schemes for stake pools. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 256–275.

[3] Vitalik Buterin. 2014. On stake. https://blog.ethereum.org/2014/07/05/stake/.

[4] Brad Chase and Ethan MacBrough. 2018. Analysis of the XRP Ledger Consensus Protocol. arXiv:1802.07242 [cs.DC]

[5] Claudia Diaz, Harry Halpin, and Aggelos Kiayias. 2021. The Nym Network: The Next Generation of Privacy Infrastructure. https://nymtech.net/nym-whitepaper.pdf.

[6] John R. Douceur. 2002. The Sybil Attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers (Lecture Notes in Computer Science, Vol. 2429)*, Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron (Eds.). Springer, 251–260. https://doi.org/10.1007/3-540-45748-8_24

[7] Vincent Durham. April 18, 2011. Namecoin - a distributed naming system based on Bitcoin. https://bitcointalk.org/index.php?topic=6017.

[8] Ben Fisch, Rafael Pass, and Abhi Shelat. 2017. Socially Optimal Mining Pools. In *Web and Internet Economics - 13th International Conference, WINE 2017, Bangalore, India, December 17-20, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10660)*, Nikhil R. Devanur and Pinyan Lu (Eds.). Springer, 205–218. https://doi.org/10.1007/978-3-319-71924-5_15

[9] Ulrich Gallersdörfer, Lena Klaaßen, and Christian Stoll. 2020. Energy Consumption of Cryptocurrencies Beyond Bitcoin. *Joule* 4, 9 (2020), 1843–1846. https://doi.org/10.1016/j.joule.2020.07.013

[10] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. 2018. Decentralization in Bitcoin and Ethereum Networks. In *Financial Cryptography and Data Security - 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26 - March 2, 2018, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 10957)*, Sarah Meiklejohn and Kazue Sako (Eds.). Springer, 439–457. https://doi.org/10.1007/978-3-662-58387-6_24

[11] Michel X. Goemans, Vahab S. Mirrokni, and Adrian Vetta. 2005. Sink Equilibria and Convergence. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*. IEEE Computer Society, 142–154. https://doi.org/10.1109/SFCS.2005.68

[12] Kevin Hammond. August 13, 2020. The decline and fall of centralization. IOHK Blog https://iohk.io/en/blog/posts/2020/08/14/the-decline-and-fall-of-centralization.

[13] Tim Harrison. November 5, 2020. Parameters and decentralization: the way ahead. IOHK Blog https://iohk.io/en/blog/posts/2020/11/05/parameters-and-decentralization-the-way-ahead/.

[14] Philipp Kant, Lars Brünjes, and Duncan Coutts. April 11, 2019. Engineering Design Specification for Delegation and Incentives in Cardano–Shelley. https://hydra.iohk.io/build/790053/download/1/delegation_design_spec.pdf.

[15] Aggelos Kiayias. November 13, 2020. The general perspective on staking in Cardano. IOHK Blog https://iohk.io/en/blog/posts/2020/11/13/the-general-perspective-on-staking-in-cardano/.

[16] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10401)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, 357–388. https://doi.org/10.1007/978-3-319-63688-7_12

[17] Aggelos Kiayias and Aikaterini-Panagiota Stouka. 2021. Coalition-Safe Equilibria with Virtual Payoffs. In *AFT '21: 3rd ACM Conference on Advances in Financial Technologies, 2021*.

[18] Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 19 (2012), 1.

[19] Yujin Kwon, Jian Liu, Minjeong Kim, Dawn Song, and Yongdae Kim. 2019. Impossibility of Full Decentralization in Permissionless Blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019*. ACM, 110–123. https://doi.org/10.1145/3318041.3355463

[20] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S. Rosenschein. 2015. Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind (Eds.). ACM, 919–927. http://dl.acm.org/citation.cfm?id=2773270

[21] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.

[22] Christos H. Papadimitriou and Georgios Piliouras. 2018. Game dynamics as the meaning of a game. *SIGecom Exch.* 16, 2 (2018), 53–63. https://doi.org/10.1145/3331041.3331048

[23] Adam Smith. 1937. *The wealth of nations [1776]*. Vol. 11937. na.

[24] Balaji Srinivasan and Leland Lee. 2017. Quantifying Decentralization. https://news.earn.com/quantifying-decentralization-e39db233c28e Medium.

[25] Michael P. Wellman. 2006. Methods for Empirical Game-Theoretic Analysis. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press, 1552–1556. http://www.aaai.org/Library/AAAI/2006/aaai06-248.php

[26] Gavin Wood. 2014. Ethereum yellow paper.