Journal Pre-proof

Looking for Stability in Proof-of-Stake based Consensus Mechanisms

Alberto Leporati and Lorenzo Rovida

PII: S2096-7209(24)00035-6
DOI: https://doi.org/10.1016/j.bcra.2024.100222
Reference: BCRA 100222

To appear in: *Blockchain: Research and Applications*

Received date: 15 February 2024
Revised date: 19 June 2024
Accepted date: 16 July 2024

Please cite this article as: A. Leporati and L. Rovida, Looking for Stability in Proof-of-Stake based Consensus Mechanisms, *Blockchain: Research and Applications*, 100222, doi: https://doi.org/10.1016/j.bcra.2024.100222.

Looking for Stability in Proof-of-Stake based Consensus Mechanisms

Alberto Leporati[1], Lorenzo Rovida[1]

[a] *University of Milan-Bicocca, Department of Informatics, Systems and Communication, Viale Sarca 336, 20126, Milan, Italy*

Abstract

The Proof-of-Stake (PoS) consensus algorithm has been criticized, in the literature and in several cryptocurrencies communities, due to the so-called *compounding effect*: who is richer has more coins to stake, therefore higher probability of being selected as a block validator and obtaining the corresponding rewards, thus becoming even richer. In this paper, we present a PoS simulator written in the Julia language that allows one to test several variants of PoS-based consensus algorithms, tweaking their parameters, and observe how the distribution of cryptocurrency coins among the users evolves over time. Such a tool can be used to investigate which combinations of parameters values allow to obtain a "fair" and stable consensus algorithm, in which, over the long term, no one gets richer or poorer by the mere act of validating blocks. Based on this investigation, we also introduce a new PoS-based consensus mechanism that allows the system to keep the wealth distribution stable even after a large number of epochs.

Keywords: Blockchain, Proof-of-Stake, Compounding effect, Wealth Distribution, Tokenomics

1. Introduction

The advent of Bitcoin [23] has given rise to an increasing interest in blockchains and distributed ledger technologies (DLTs), attracting many scientists, programmers, and business investors. Since then, many types of blockchains and DLTs have been proposed, both permissionless and permissioned, based on several types of consensus algorithms, among which we can find Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-Of-Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof-of-Burn (PoB), Proof-of-Capacity, and Proof of Elapsed Time (PoET). An overview of these mechanisms is given in [1].

Email addresses: alberto.leporati@unimib.it (Alberto Leporati), lorenzo.rovida@unimib.it (Lorenzo Rovida)

In the first years of DLTs, questions of technological nature received the most attention; questions about economics, cryptocurrencies distributions and tokenomics have been addressed much less thoroughly. Some papers in the literature address the issue of (lack of) decentralization in blockchain governance [11; 14; 18; 20], implicitly assuming that who is richer has more power in taking decisions about which transactions and blocks to validate. Albeit decentralization and wealth distribution among the users of a blockchain are somehow related, the two phenomena do not necessarily coincide. In fact, it is commonly believed that monetary policy concerns cryptocurrency distribution, whereas decentralization is merely a technical (infrastructural) matter. However, monetary policy is not the only important factor for wealth distribution: even technology solutions, like consensus mechanisms, might influence it. Hence, to understand the implications of wealth distribution each different type of consensus algorithm must be analyzed separately. As an example, a comparison among Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus mechanisms is very informative. In PoW, newly created units of currency are rewarded to the specialized users, called *miners*, who have access to efficient and powerful hardware. PoW miners might hold a large number of cryptocurrency units; however, a large portion of mined rewards must be sold to cover expenses like electricity bills, rent, and amortization costs of mining rigs. In PoS systems, instead, new coins are rewarded to *stakers* who hold a large number of cryptocurrency units. Unlike PoW miners, PoS stakers do not experience high costs and are encouraged not to sell their rewards as doing so increases their revenue in the future. This phenomenon, known as *compounding effect* [10; 13; 30], illustrates that even supposedly monetary-agnostic technology solutions might influence tokenomics.

Wealth distribution, decentralization, and blockchain governance, have become particularly relevant and very much discussed topics when Ethereum [34] announced their intention to switch from PoW to PoS, which they did on September 15, 2022, in the event known as "The Merge" [6]. To avoid the compounding effect of PoS – whereby the richest get even richer – and also to mitigate the negative externalities posed by Maximal Extractable Value (MEV) strategies – that include, omit, or reorder transactions when making a new block, with the aim of producing as much additional profit as possible – the community has proposed *Flashbots* [7], a (quite elaborate) infrastructure running on top of Ethereum's blockchain. Flashbots, as well as other recent similar projects, provides an off-chain marketplace to build and propose the most profitable blocks to the validators; however, this off-chain mechanism introduces some degree of opacity in the consensus mechanism, which is against the transparency and fairness principles that drive permissionless blockchains. Further details on how the Flashbots architecture works, and a preliminary analysis of how the rewards have been distributed since its birth, can be found in [21].

In this paper we partially address questions about wealth distribution in blockchains that make use of the Proof-of-Stake family of consensus algorithms. PoS was initially designed to improve the energy consumption derived from PoW [29]. Since its first implementation, PoS has evolved and many researchers have been discussing different approaches, such as *Chain-based PoS*, *Nominated PoS (NPoS)*, *BFT-based PoS*, *Delegated proof of stake (DPoS)*, and *Liquid proof of stake (LPoS)*. For a description on how these algorithms work, we refer the reader to [3; 2; 24; 9]. Even the exact definition of *stake* varies among different implementations: for instance, some cryptocurrencies use the concept of *coin age*, the product of the number of coins with the amount of time that a single user has held them, rather than merely the number of coins, to define a validator's stake [32]. In order to make our study more general, we will not focus our attention on a specific variant of PoS consensus algorithm. Instead, we have developed a PoS simulator whose behavior depends upon several parameters, and may be adapted to simulate any specific PoS-based algorithm. By tweaking these parameters, we can observe how the distribution of cryptocurrency coins evolves over time. This tool can thus be used to investigate which combinations of parameters values allow us to obtain a "fair", stable and sustainable distribution of wealth in the long term, in which no one gets richer or poorer by the mere act of validating blocks. Indeed, we believe that fairness is a necessary condition for the consensus protocol to be sustainable over time: a protocol that concentrates wealth in the hands of few makes a permissionless blockchain a centralized system, controlled by an oligarchy. This entails that users will no longer trust the system and therefore they will leave it.

Precisely, the research question that we want to address in this paper is the following:

> **Research question**: *Is it possible to define a PoS-based consensus mechanism in which, with a given set of parameters, the distribution of wealth remains relatively stable across epochs?*

We will try to answer this question by presenting a consensus algorithm that aims to keep the distribution of wealth near a given target value. This mechanism will be implemented in the proposed simulator in order to empirically show its behavior and stability.

The rest of this paper is structured as followed. In Section 2 we recall some related works from the literature, that investigate on the distribution of wealth among blockchain validators. In Section 3 we describe the two main contributions of the paper: the *Gini-Stabilized* consensus mechanism and the PoS simulator. In Section 4 we show some examples of simulations that can be performed with the latter, and we discuss the outputs of such simulations. We therefore give a qualitative analysis of the proposed Gini-Stabilized consensus algorithm. Finally, in Section 5 we draw some conclusions and delineate some directions for future research.

2. Some Related Works

In this section we recall some works taken from the literature, that are somehow related with the topic under study. However, as will be discussed shortly, our perspective is a bit different, and we believe that our approach may be of some interest to design and/or test the behavior of PoS-based consensus algorithms.

Some scientific studies about the distribution of wealth among the top richest users of PoW and PoS-based blockchains have been performed [10; 12; 27; 4]. For example, [13; 15; 19] showed that the distributions of the top richest balances might be modeled with Zipf's law. Additionally, the Gini coefficients were computed for each user to measure wealth inequality. Moreover, in [13] the authors showed that the wealth of top Bitcoin holders grows faster than the wealth of low balance accounts; this phenomenon is well known as *preferential attachment*, and it has an important impact on wealth distribution.

In [14], the authors analyze the distribution of the top richest accounts in cryptocurrencies like Bitcoin, Ethereum, and selected ERC20 tokens. Their analysis involves the data sets snapshotted at different dates with a given time interval. These data sets are used to measure different statistical and concentration metrics – Shannon entropy, Gini coefficient, Nakamoto coefficient and approximated Zipf coefficient – and to analyze their evolution over time, trying to answer the following research question: *Are there any quantitative differences between top account balances in cryptocurrency "coins" and "tokens"?*. The authors analyzed the time-dependent statistical properties of top cryptocurrency holders for 14 different distributed ledger projects. Using the above mentioned metrics, they showed that there are quantitative differences of centralization levels between cryptocurrency coins and tokens. It was thus observed that tokens are, in general, much more centralized than coins, with higher Gini coefficients and smaller Nakamoto coefficients.

All these researches focus on the top richest accounts, and hence might be of particular interest to DLTs where a group of top cryptocurrency holders fulfills a special role. Examples include Decentralized Autonomous Organizations (DAOs) – in which a committee of top token holders is responsible for DAO governance or treasury management – and Delegated Proof-of-Stake (DPoS) blockchains – where a relatively small committee of block validators issue ledger updates or distributes random number generators based on the threshold signature scheme. Since these kinds of analyses require to download and process large amounts of data, they necessarily limit their scope to the top richest users.

Other works focused on the (de)centralization of blockchains, intended as the number of players controlling them. In [20], the authors provided their analysis using three different metrics (Gini coefficient, Shannon entropy, and Nakamoto coefficient) and their evolution over time. It was found that the degree of decentralization in Bitcoin is higher and more volatile, while the degree of decentralization in Ethereum (when still adopting the PoW consensus mechanism) is smaller and more stable. Jensen et al. [11] analyzed decentralization of governance token distribution in four decentralized finance (DeFi) applications on the Ethereum blockchain using Gini and Nakamoto coefficients. Their results indicated that the token distributions for all four DeFi applications are characterized by high Gini coefficients. Similar methods were used in [19], where PoW and PoS cryptocurrencies were compared, analyzing the decentralization of Bitcoin and Steem [28] using Shannon entropy.

Other papers deal with the centralization/decentralization of PoS-based blockchains. However, it is important to note that (de)centralization and wealth distribution may be in some cases related, but they are indeed different phenomena. Further, the data about wealth distribution usually presented in the literature do not represent the wealth of individual cryptocurrency owners but rather the wealth distribution among the cryptocurrency wallets. Apart from the difficulty of establishing the owner of a wallet, a user may be in possession of multiple wallets. Clearly, all these hindrances make it difficult to interpret the results of the analyses.

Concerning simulators for blockchain consensus protocols, a good overview is given in [25]. For instance, [5] performs a comparison of rewards distribution between PoW-based and PoS-based blockchains, showing that PoS has a more fair reward distribution. However, no one of the proposed simulators allows one to compare different consensus mechanisms in PoS.

In this paper we take a different approach with respect to the above cited papers. Instead of analyzing existing data about cryptocurrencies or tokens distributions in blockchains, we study under which conditions a PoS-based consensus algorithm allows to obtain a *fair* wealth distribution over time. By *fair* we mean that who is richer has more possibilities to be chosen to be a validator, but in the long run his wealth does not significantly increase (nor decrease) due to the mere activity of validation. Stated otherwise, the validation activity *alone* should not meaningfully increase nor decrease anyone's amount of cryptocurrency. To do so, we do not look only at the top 30-50-100 richest cryptocurrency holders, but we consider the distribution of wealth among all the users of the blockchain – to be precise, all users who aspire to be selected as block validators. While the Gini coefficient can be considered a centralization measure, we will use it instead as an indicator of wealth distribution among a population (the blockchain users), as is done in economics studies. Our aim is to help researchers analyze the behavior of existing implementations of PoS consensus algorithms, and the designers of PoS-based consensus algorithms in testing variants and finding the values of parameters that eventually make the protocol fair and sustainable. We do so by proposing a PoS simulator that allows one to tweak several aspects and parameters of the consensus algorithm. Starting from an initial coins supply, the simulator computes the evolution of wealth distribution over time, measuring its fairness by means of Gini coefficient. As stated above, we believe that fairness is a necessary condition for the consensus protocol to be sustainable over time. In fact, a protocol that concentrates wealth in the hands of few makes a permissionless blockchain a centralized system, controlled by an oligarchy. This entails that users no longer trust the system, and therefore abandon it. So, in our opinion, fairness implies sustainability in the long run. Let us note that some authors have adopted a more extreme point of view about PoS: in [30], for example, it is stated that "Proof-of-stake is introducing a set of significant new flaws in both monetary and governance models. Such systems are plutocratic, oligopolistic, and permissioned". Even without being so extreme, it is true that PoS essentially means proof of wealth, since blockchain protocol's rules, upgrades, and changes are directly linked to its participants' stake (that is, wealth). Other authors have proposed significant modifications to the PoS protocol, to make it more democratic and sustainable [26; 18]. For instance, [18] propose a consensus mechanism that protects the system from the risk of coin age accumulation attack. Saad et al. [26] propose a consensus algorithm in which performing a fair mining is promoted by the nature of the algorithm itself. Both these works, though, lack of a quantitative analysis of the wealth distribution.

By using our simulator, researchers can explore all known and new proposals for PoS-based consensus algorithms, play with their parameters, and gather quantitative data about the behavior of the algorithms over a substantial number of epochs.

This paper is an extension of [16], in which an analogous PoS simulator written in the R language was proposed. While such a previous work was focused on the proposal of the simulator, we found that the performances obtained were not enough to allow us to perform qualitative or quantitative analyses on a substantial number of blockchain users, and a sufficiently high number of epochs. Hence, to perform the experiments described in this work we completely rewrote (and extended) the simulator in Julia, so that more extensive experiments can be executed, in more realistic scenarios. In this paper, to measure fairness we use the Gini coefficient, whose formal definition is recalled in the next section. As commonly agreed in the economic literature, a fair economic system is such that its Gini coefficient is less than 0.3 (while a value greater than 0.5 is considered dangerous and divisive). Hence, in what follows we also propose a new PoS-based consensus algorithm that, given a desired target value θ for the Gini coefficient, adapts its behavior in order to approach it.

3. A Proof-of-Stake Simulator

In order to address the research question posed in the previous section, we have developed a Proof-of-Stake based simulator using the Julia language. As stated above, the simulator computes the wealth distribution in terms of *Gini coefficient*, which is formally defined as follows:

$$G = \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{N} |x_i - x_j|$$

where N is the number of elements in the population (in our case, the number of blockchain users) and x_i, for $1 \leq i \leq N$, is the monetary value associated to the i-th element (in our case, the number of cryptocurrency coins or tokens available to the i-th user). The Gini coefficient is an inequality measure widely used in economics and social statistics. For example, it is used to measure the inequality of incomes – or of wealth – among the citizens of a country. More in general, it is a measure of the inequality of a statistical distribution. It takes values from 0, which corresponds to a complete decentralization (that is, a fair distribution) of wealth, to 1, that corresponds to absolute centralization. As commonly agreed in the literature, a fair economic system is such that its Gini coefficient is less than 0.3, while a value greater than 0.4 indicates a risk of social and political instability. Indeed, 0.4 was indicated in 2013 as a desired target by United Nations to reduce economic inequalities in the world [8]. Let us note, however, that these values are not strongly supported by theoretical analyses; on the contrary, based on some data analysis, [31] indicates 0.5 as a better threshold value. For this reason, in what follows we will not set a fixed value for the Gini coefficient; instead, we will propose a variant of PoS that allows the system to never deviate from a pre-set value θ, whatever it may be.

The role of the simulator is to help studying the existence of a set of rules and parameters that, with a given consensus mechanism, maintain the Gini coefficient relative to a set of peers near a desired target value. Before going into the details of the simulator, we first define two types of consensus mechanisms that will be useful:

- *Weighted PoS*: the probability of a participant being chosen to validate a block, and to earn the associated reward, is proportional to the number of cryptocurrency coins they are willing to *stake*. More formally, the probability P_i for the participant i to be chosen is defined as follows:

$$P_i = S_i / \sum_{j=1}^{N} S_j$$

where S_i is the stake held by participant i.

- *Opposite-Weighted PoS*: the probability of a participant being chosen is computed as the complement of the previous consensus mechanism:

$$P_i = 1 - S_i / \sum_{j=1}^{N} S_j$$

The *Weighted PoS* mechanism causes for sure the *compounding effect*, as it is also shown in Section 4; in fact, it can be observed that the Gini coefficient tends to grow towards 1. On the other hand, the *Opposite-Weighted PoS* behaves the opposite, resulting in a tendency for the Gini coefficient to decrease towards 0. More in general, it may be desirable to achieve an equilibrium near a given target value; hence, in the next section we define a new PoS-based consensus algorithm which achieves this goal.

3.1. Gini-Stabilized PoS

We propose *Gini-Stabilized PoS*, an *adaptive* consensus mechanism that regulates the choice on the validator by interpolating between the previous two mechanisms according to the current value g of the Gini coefficient. Given a target value θ, the probability of a participant being chosen to validate a block in this consensus mechanism could be defined as:

$$P_i = \begin{cases} S_i / \sum_{j=1}^{N} S_j & \text{if } g < \theta \\ 1 - S_i / \sum_{j=1}^{N} S_j & \text{otherwise} \end{cases} \tag{1}$$

This mechanism pushes the Gini coefficient towards one when $g < \theta$, and towards zero otherwise. Nevertheless, we must introduce a way to interpolate between the two distributions to smoothly transition between them, in order to avoid an alternating back-and-forth motion (i.e., a "zig-zag" effect) around θ. Given two real values $x, y \in \mathbb{R}$, the interpolation function $\ell(x, y, t)$ is defined as follows:

$$\ell(x, y, t) = x + t(y - x)$$

where the parameter t is defined at each epoch j over the current Gini coefficient g as:

$$\begin{cases} t_0 = d(g, \theta) \\ t_j = \ell(t_{j-1}, d(g, \theta), s) \end{cases} \tag{2}$$

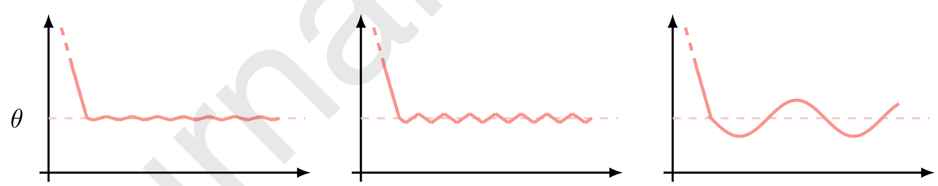
where the function $d(g, \theta)$ controls the "direction" of the transition and is defined as:

$$d(g, \theta) = \frac{\operatorname{sgn}(g - \theta) + 1}{2}$$

where $\operatorname{sgn}(x)$ is the sign function, defined as:

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

and $0 < s \leq 1$ controls the "speed" of the transition. When $s = 1$, the value of t will always be equal to $d(g, \theta)$, generating the zig-zag effect. Smaller values of s, on the other hand, will relax the value of g and allow for smoother changes in the distribution. Ideally, we want to achieve something like it is shown in Figure 1.



(a) $s \approx 1$ (b) $s < 1$ (c) $s \ll 1$

Figure 1: Desired behavior for the *Gini-Stabilized PoS*: the value of Gini coefficient (g, in red) is always "pushed" towards the target value θ by the function $d(g, \theta)$ like a sinusoidal wave

We are now able to assemble the above components and define the probability of a participant i to be chosen by the *Gini-Stabilized* consensus mechanism:

$$P_i = \ell(P_w, 1 - P_w, t_j) \tag{3}$$

where P_w is the probability distribution defined in the *Weighted PoS* and j is the index of the current epoch, required to compute the value of t with respect to the previous one. The difference from Equation (1) is that the interpolation function, whose speed is defined by the parameter s, relaxes the transition between the two distributions.

We will explore different approaches (that is, functions) to assigning values to s. The first, and most straightforward, is to assign a fixed value k:

- *Constant smoothing*: given a fixed value k, evaluate $s = k$

However, we also propose computing the value of s proportionally to the difference $|g-\theta|$ between the current Gini coefficient (g) and the target Gini coefficient (θ), using the following two approaches:

- *Linear smoothing*: given a fixed value k, evaluate $s = |g - \theta|/k$
- *Quadratic smoothing*: given a fixed value k, evaluate $s = |g - \theta|^2/k$

Each type of smoothing will have a strong impact on the smoothness of the trend of g. The experiments presented in Section 4 will further clarify this point.

One may observe that, with our *Gini-Stabilized PoS*, when the current value g of the Gini coefficient exceeds the threshold value θ, we are increasing the probability of choosing, as validator, those who have put less money on stake. This could incentivize malicious behavior on the part of those with little to lose. On the other hand, the mantra on which PoS is based is that those who stake a lot of money have every interest in behaving honestly, given that they are also the ones who would lose the most if the system failed. While these reasonings are correct, we must note that - although we are increasing the probability of choosing the poorest participants - the fact remains that the choice of validator is essentially made proportionally to the amount of money staked. The probabilities are changed only *slightly*, so as to bring the value of the Gini coefficient back towards the threshold as soon as it deviates.

Notwithstanding, we can think of several possibilities to discourage malicious behavior on the part of those who stake little money. For example, one possibility is to introduce a minimum threshold of money to stake to become a validator. While on the one hand this constitutes a disincentive, since the amount of money lost in case of incorrect behavior is not negligible, on the other hand the threshold value should not be too high, otherwise only a few wealthy individuals can become validators. Another possibility is to increase the punishment for those who behave incorrectly, i.e. the amount of money slashed from the stake. This seems to be a good solution, since it does not change the basic assumption that those who are richer have no interest in cheating. Yet another possibility is that whoever has validated a block cannot do so for a certain number of subsequent time slots. In the simulator, we have called such number of epochs the *standby period*. Under the assumption that few people will try to cheat, this mechanism makes their participation in the block validation activities more difficult.

We have performed several experiments, using each of these three disincentives. In the long run, each of them does not influence significantly the wealth distribution. Among all, the most effective mechanism seems to be the *Gini-Stabilized PoS*. Indeed, the probability of being selected as a validator remains basically proportional to the amount of money put in stake; thus, if a cheater puts a small amount of money, they will be selected very rarely. And this is what seems to count in the long run.

3.2. The parameters of the simulator

The proposed simulator implements various types of consensus mechanisms, including the just introduced *Gini-Stabilized PoS*, along with many parameters that let the user control the simulation; a complete list is presented in Table 1.

Table 1: Parameters of the simulator

Parameter	Description
`n_epochs`	Number of epochs to be simulated, corresponding to the number of blocks validated
`proof_of_stake`	Type of PoS consensus mechanism: {*Weighted*, *OppositeWeighted*, *GiniStabilized*}
`initial_stake_volume`	Initial number of coins distributed among peers, based on `initial_distribution`
`initial_distribution`	Initial distribution of coins among validators: {*Uniform*, *Gini*, *Random*}
`n_peers`	Number of participants (same as N) aiming to be validators
`n_corrupted`	Number of validators that could exhibit corrupted behavior
`p_fail`	Probability that a corrupted peer, if selected as validator, fails to validate correctly
`penalty_percentage`	Percentage of coins slashed from the stake of corrupted validators
`p_join`	Probability of a new user joining validators, at each epoch
`p_leave`	Probability of any validator quitting the pool, at each epoch
`join_amount`	Amount of coins owned by a newly joined peer: {*Average*, *Random*, *Max*, *Min*}
`reward_type`	Type of reward: *Constant* or *Dynamic*. If the latter is chosen, the reward is a percentage of the current stake volume
`minimum_stake`	Minimum amount of coins required to join the pool
`standby_period`	Number of epochs that a previously chosen validator must wait before being considered to validate the next block

The tool requires additional parameters, in case some options have been selected. These parameters are reported in Table 2.

Table 2: Optional parameters of the simulator

Parameter	Description
`initial_gini`	In case *Gini* has been selected as initial distribution, this indicates the value of the Gini coefficient at the beginning of the simulation
`gini_threshold`	The value of θ, to be defined in case of *GiniStabilized* consensus mechanism
`initial_stake_volume`	Initial number of coins distributed among peers, based on `initial_distribution`
`s_funct`	The smooting function applied to s (refer to Equation (2)). It takes a value among *Constant*, *Linear* and *Quadratic*
`k`	The value of k, given as input to the smoothing function used to compute s

The simulator, whose source code is available at [17], allows one to track the distribution of cryptocurrency coins over time, for a given PoS consensus algorithm, under a specified choice of parameters. It is not intended as a complete solution, but rather as a generic skeleton to be customized according to the precise implementation of the consensus algorithm to be analyzed. Notice that modifications to the simulator are simple to make, as it is designed in a modular manner. In the next section we provide some examples of analyses performed on some hypothetical implementations of PoS.

3.3. Simulator workings

In this section, we provide an overview of how the proposed simulator operates. We will offer some insights from a high-level perspective by annotating the pseudocode that summarizes its functionality in Algorithm 1.

In the following, we comment each line of the algorithm; each item refers to the corresponding line number:

1. Generate a set of peers with specified parameters such as `n_peers`, `initial_stake_volume` and `initial_distribution`.
2. Select a random subset of peers to be corrupted, with the size specified as `n_corrupted`. Usually this value should be much less than the number of peers; for example, a reasonable assumption seems to be that the number of corrupted peers is $< 5\%$.
3. Initialize an empty list history which will store the values for variable g over time.

Algorithm 1 Pseudo-code of the PoS simulator

```
1: peers ← generatePeers(n_peers, initial_distribution)
2: corruptedPeers ← random subset of peers of size n_corrupted
3: history ← {}                                   ▷ This will contain all the values for g
4: g ← Gini(peers)
5: t ← d(g, θ)
6: append g to history
7: for i ← 1 to n_epochs do
8:     peers ← quit(p_quit)          ▷ A random set of peers might leave the pool
9:     peers ← join(p_join)          ▷ A new set of peers might join the pool
10:    g ← Gini(peers)
11:    s ← funct(s_funct, k)
12:    t ← ℓ(t, d(g, θ), s)                                  ▷ See Equation (2)
13:    append g to history
14:    v ← consensus(proof_of_stake, peers, t)          ▷ Computes the index
                                                        of the chosen validator
15:    ▷ The selected validator must not be in its standby period
16:    while v is in standby do
17:        v ← consensus(proof_of_stake, peers, t)
18:    end while
19:    if v ∈ corruptedPeers ∧ r < p_fail then
20:        peers_v ← peers_v · penalty_percentage
21:    else
22:        peers_v ← peers_v + reward
23:    end if
24:    reduce all standby periods by 1
25: end for
26: plot(history)                               ▷ Plots g over the different epochs
```

4. Calculate the Gini coefficient g for the current set of peers.
5. Calculate the interpolation factor t based on the initial value t_0. Refer to Equation (2) for a detailed explanation.
6. Append the current value of g to the history list.
7. Begin a loop iterating over a specified number of epochs. Within each epoch run the following:
    8. Attempt to remove a set of random peers. In particular, each peer may be removed with probability `p_leave`.
    9. Attempt to add a new set of peers. In particular, a new peer is added with probability `p_join`; if it is added, a new peer can join the pool with the same probability. Otherwise, the procedure stops. The stake of the new peers is defined according to the `join_amount` parameter. Lastly,

the new added peer might be a corrupted one with a probability equal to the initial ratio of corrupted peers over the total number of peers.

10. Calculate the Gini coefficient g for the current set of peers.
11. Compute the value of s according to the selected function `s_funct`.
12. Update the value of t_i according to the previously defined Equation (2).
13. Append the current value of g to the history list.
14. Determine the consensus among peers using the chosen Proof-of-Stake mechanism with the current set of peers. The value of t is only used when the `proof_of_stake` parameter is set to *GiniStabilized*.

16-18. If the selected validator is in standby (i.e., he was selected less than `standby_period` epochs before as a validator), re-do the selection.

19-20. If the chosen validator is in the corruptedPeers list, and a random value $r \in [0, 1)$ is less than `p_fail`, then the validator is slashed and their stake is reduced by a factor `penalty_percentage`.

21-23. Otherwise, reward the validator and set their standby period to `standby_period`

24. All greater than zero standby periods are reduced by one.

4. Some Experiments

In this section we illustrate some experiments performed using our simulator. As stated in previous sections, we have not focused on a particular real implementation of PoS. The simulator generates a list of peers, users of the blockchain that aim to be selected as validators, which are numbered from 1 to `n_peers`. This list is not static: at each epoch new peers can join the set of prospective validators according to a probability `p_join`; in this case, the number of coins owned by the new peer depends on the value of `join_amount`. Moreover, some peers might leave the blockchain with a probability `p_leave`. Among the possible validators there is a subset of corrupted peers, meaning that for some reason, if selected as validators, these peers will fail to correctly validate the current block. In such a case a penalty is applied, and the stake of the corrupted validator gets slashed by a factor `penalty_percentage`. Each peer receives an initial supply of cryptocurrency coins from the total volume `initial_stake_volume`, distributed among the peers according to the selected `initial_distribution`.

For each experiment, we briefly introduce the most important parameters; refer to Table 3 for a complete presentation of all the values of the parameters used in all the experiments. Notice that the value of `minimum_stake` is always equal to `reward`. We remark that all the experiments are available as Jupyter notebooks, based on Julia, in our open-source repository [17].

In what follows we will analyze the results obtained in each experiment from two perspectives: the trend of the Gini coefficient to analyze the compounding

Table 3: Set of parameters used in the experiments. *Multiple* denotes that multiple values of that parameter have been used

	Experiments				
Parameters	1	2	3	4	5
n_epochs	20.000	400.000	300.000	300.000	4.000.000
proof_of_stake	Weighted	Opposite	GiniSt.	GiniSt.	GiniSt.
initial_stake_volume	5.000	5.000	5.000	5.000	50000
initial_distribution	Gini	Gini	Gini	Gini	Gini
n_peers	10.000	10.000	10.000	10.000	10.000
n_corrupted	-	-	-	-	50
p_fail	-	-	-	-	0.7
penalty_percentage	-	-	-	-	0.5
p_join	-	-	-	-	0.01
p_leave	-	-	-	-	0.01
join_amount	-	-	-	-	Random
initial_gini	*Multiple*	*Multiple*	*Multiple*	0.5	0.5
reward	20	20	5	10	10
theta (θ)	-	-	0.3	0.3	0.3
s_funct	-	-	Constant	Linear	*Multiple*
k	-	-	1/100	10	*Multiple*
minimum_stake	20	20	5	10	10
standby_period	2	2	2	2	2

effect, and the percentage of stake held among peers to analyze the preferential attachment effect.

4.1. Experiment 1: Weighted PoS

As a first experiment, we simply want to assess the tendency of *Weighted* consensus to push the Gini coefficient towards 1, thus generating the compounding effect. We define the number of peers to be equal to 10.000 and we run the simulator for 20.000 epochs. We performed five different runs, each of them starting from a different initial distribution, defined by different values of `initial_gini`. As shown in Figure 2, the Gini coefficient g tends to grow towards 1, independently from the choice of the initial distribution of coins.

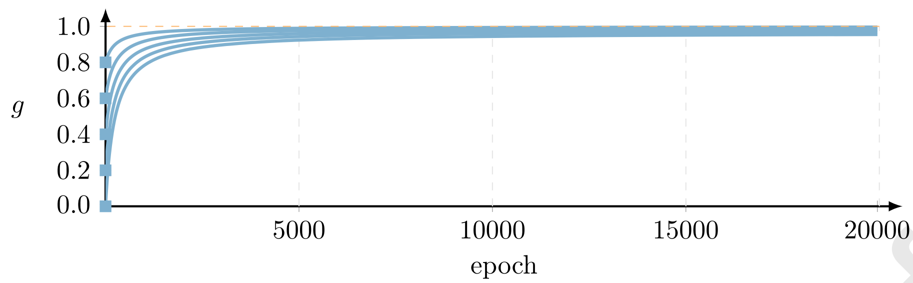


Figure 2: Trend of the Gini coefficient g using the *Weighted* consensus algorithm, on five different starting distributions

Additionally, we want to study the effect that this consensus mechanism has over the difference between the distribution of stake held among the peers at the beginning and at the end of the simulation. We thus generate a set of peers, each having a random amount of coins (initial distribution set to *Random*), and we present the percentage of stake held by each peer across iterations in Figure 3.

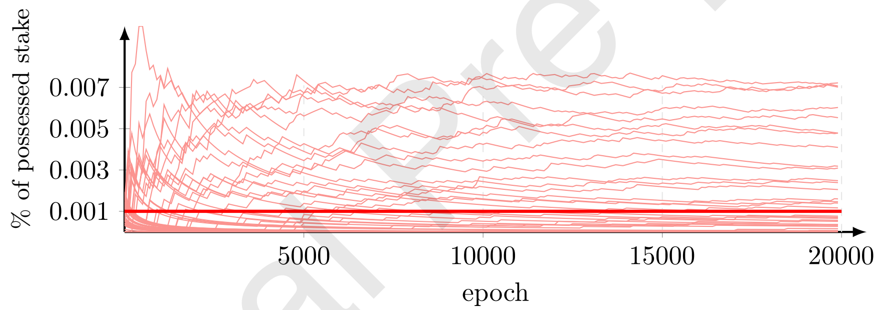


Figure 3: Percentage of stake held across iterations using the *Weighted* PoS, with a random starting distribution. The red line indicates the value of 1/`n_peers`

It is clear from the figure that the *Weighted* PoS tends to create a strong division between peers that are above and below the mean percentage of possessed stake. Notice the red horizontal line; if a peer's position on the plot is above the red line, it means that the percentage of stake it possesses is greater than what at least half of the peers possess. Conversely, if a peer's position is below the red line, it means that the percentage of stake it possesses is smaller than what at least half of the peers possess.

In the considered case, at the first epoch 50.4% of the peers was below the average, and the other 49.6% was above. After the simulation, this ratio changed to 76.7% over 23.3%. This implies that rewards for validating blocks are collected by an increasingly smaller amount of peers, making the blockchain monopolized by them. This experiment indeed showed two simple but effective facts: the compounding effect, and the preferential attachment caused by using the *Weighted* consensus mechanism.

4.2. Experiment 2: Opposite-Weighted PoS

The second experiment aims to assess the tendency of *OppositeWeighted*. As stated at the beginning of Section 3, the implementation of this consensus mechanism causes the Gini coefficient to tend toward zero. Apart from the mechanism and the number of epochs, the set of parameters is the same as the previous experiment. As presented in Figure 4, the experiment shows that the value of g tends to zero, regardless of the initial distribution of coins. The

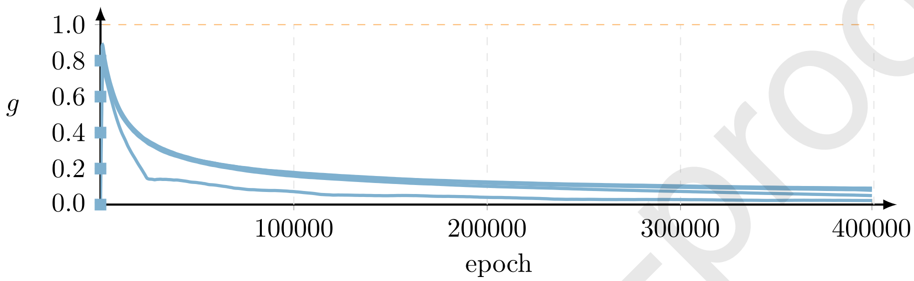


Figure 4: Trend of the Gini coefficient g using the *Opposite-Weighted* consensus algorithm, on five different starting distributions

number of epochs has been increased because we empirically noticed that this consensus mechanism converges more slowly with respect to the previous experiment. In order to complete the analysis with respect to the percentage of stake held by each peer, in Figure 5 we present its trend across iterations.

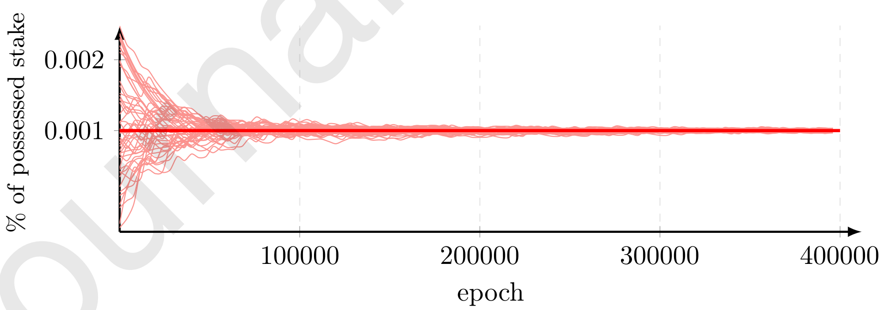


Figure 5: Percentage of stake held across iterations using the *OppositeWeighted* PoS, with uniform starting distribution

Notice that, as opposed to the previous experiment, the stake becomes well distributed among all the peers. The problem with this approach is that, of course, peers are not encouraged to put coins in stake, since the possessed percentage will become, eventually, the same for everyone.

These two experiments aimed to empirically prove the behavior of two basic consensus algorithms. Additionally, we pointed out a problem related to the percentages of stake held among peers. Ideally, we would like to have a consensus mechanism that preserves such percentages over iterations, because such a consensus mechanism would better distribute the rewards among all peers.

The following experiments will explore the behavior of a more complex consensus algorithm, the *Gini-Stabilized* PoS, under different possible ways of assigning values to parameter s.

4.3. Experiment 3: Gini-Stabilized with Constant update

We start by showing the occurrence of the "zig-zag" effect centered in a fixed value $\theta = 0.3$. This is done by setting a constant value for s equal to 1. As a consequence, the PoS behavior will switch from *Weighted* to *OppositeWeighted* as soon as g crosses the θ threshold, without any smooth interpolation. The result is shown in Figure 6.

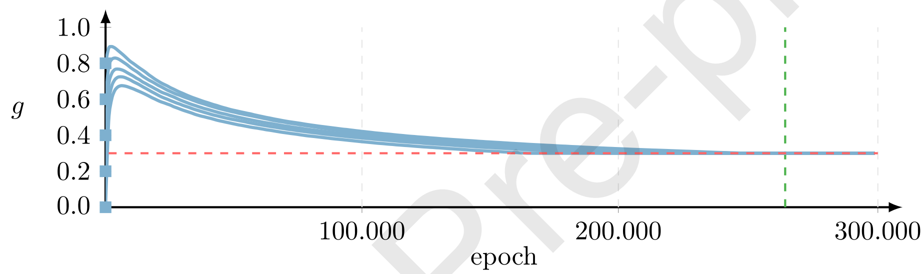


Figure 6: Trend of the Gini coefficient g using the *Gini-Stabilized* consensus algorithm with $\theta = 0.3$, on five different starting distributions

It is interesting to notice that, in the very first epochs, before starting to descend towards 0, the value of g has an initial growth. The interval of epochs in which this phenomenon happens depends on the volume of the initial stake, with respect to the quantity of coins given as a reward. The first rewards will have a bigger impact on g, with respect to the subsequent ones, since the reward is constant and does not change over time. The system, therefore, requires some epochs to "stabilize" its trend. Notice that, as g, for any starting configuration, approaches $\theta = 0.3$, its value starts to rigidly follow the target value. This point has been informally highlighted in Figure 6 with a vertical green line. It is possible to visually recognize the zig-zag phenomenon by observing the trend of g in small intervals of epochs, after it has stabilized near θ (see Figure 7). In particular, we observed that the region in which g lies, after a number of epochs, is small.

Now we wonder if it is possible to obtain a more "relaxed" trend of g by using a smaller value for s, for instance $s = 1/100$. As before, we are interested to plot the trend of g after it has stabilized near θ. We show the results in Figure 8 on a larger number of epochs, to highlight the differences with $s = 1$. As it can

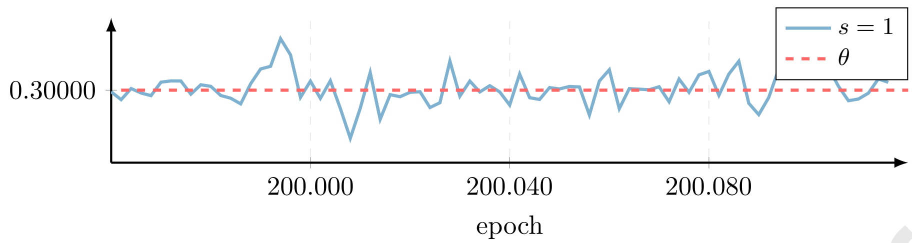


Figure 7: A zoom-in of the previous figure, in 80 epochs, that highlights the zig-zag effect caused by $s = 1$. The y-axis represents the value of g

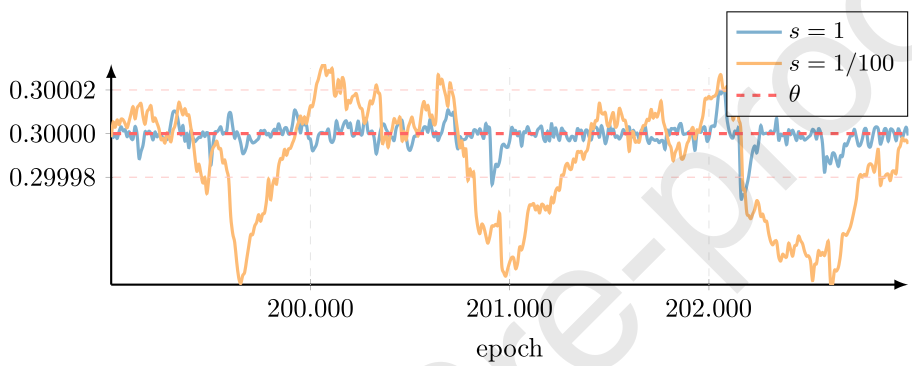


Figure 8: A comparison between $s = 1$ and $s = 1/100$, analyzed in 2000 epochs, after a stabilization near $\theta = 0.3$. The y-axis represents the value of g

be seen, when using a smaller value of s, the amplitude of the trend is larger, and the frequency is smaller, although we are still lacking a smoother transition. We conclude the experiment by showing the behavior of the proposed consensus mechanism, with $s = 1/100$ and a random initial distribution of stake volume, on the percentage of stake possession in Figure 9.

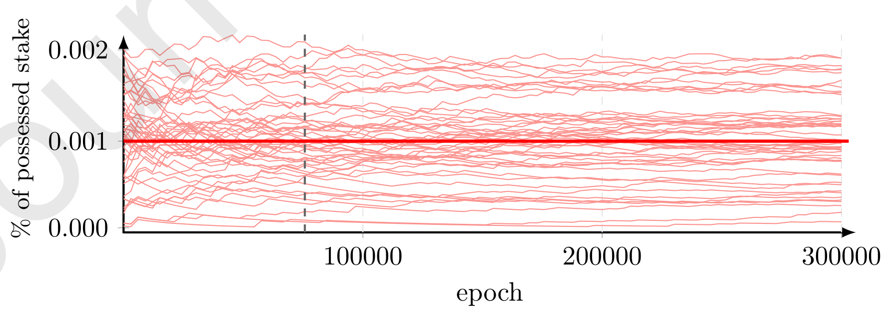


Figure 9: Percentage of possessed stake across iterations using the *GiniStabilized* PoS, with a random starting distribution. The black dashed line indicates the point in which the Gini target value $\theta = 0.3$ has been reached

Notice an unpredictable behavior in the initial phase, before the Gini coefficient reaches the target value θ. Nevertheless, this behavior could be avoided by fixing a custom initial distribution of coins. After that, the percentage of stake held among peers slightly fluctuates, but there are no significant drops or losses. As a consequence, there is no significant change in the percentage of possessed stake from the peers. This distribution is fairer, compared to the one shown by the *Weighted* consensus. For instance, now the ratio between the amount of people above and below the average goes from 50%/50% in the first iteration, to 46.4%/53.6%, which is fairer with respect to Experiment 1, and consistent with the choice of the target value $\theta = 0.3$ of the Gini coefficient.

4.4. Experiment 4: Gini-Stabilized with Linear update

To achieve a smoother trend, we require an additional ingredient. Specifically, the value of s will be dynamically adjusted based on the current difference $|g - \theta|$. As this difference grows larger, s should increase proportionally; conversely, as the difference decreases, s should decrease accordingly. We therefore set `s_funct` to *Linear*. The function used to control s will be $s = |g - \theta|/k$, with $k = 10$. We plot the results following a stabilization near θ in Figure 10.

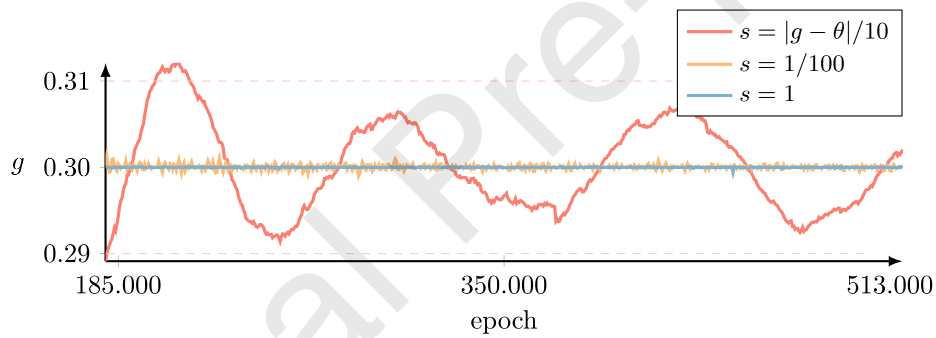


Figure 10: A comparison between *Constant* and *Linear* update strategies for s, analyzed following a stabilization near the target value θ

Here a comparison is made between the trends of the Gini coefficient for three update strategies, corresponding to the constant values $s = 1$ and $s = 1/100$, and to $s = |g - \theta|/10$. It is now clear that, by changing s according to the current value of $|g - \theta|$, we obtain a smoother trend, with a larger amplitude. It is possible to apply different functions to this difference in order to “relax” the trend as preferred.

4.5. Experiment 5: Comparison between Constant, Linear and Quadratic update strategies

As a last experiment, we execute the simulator using all the functions defined in `s_funct`: a constant, a linear, and a quadratic function. Plus, we set the probabilities for new peers to join and to quit the pool at each epoch equal to 0.1%, thus simulating a more realistic scenario. The number of epochs will be equal to 4.000.000 and, initially, the pool will consist of 10.000 potential validators. Newly added validators will stake a number of coins equal to the number of coins staked by another randomly chosen validator. As before, we set $\theta = 0.3$. The results are presented in Figure 11.

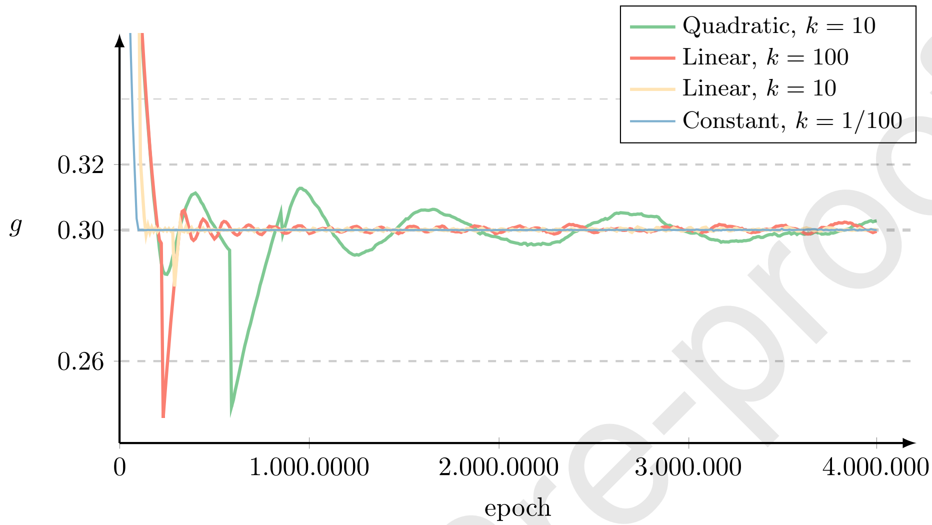


Figure 11: A comparison between *Constant*, *Linear* and *Quadratic* update strategies

Looking at the obtained results, we can make the following considerations. First of all, the *Gini-Stabilized* consensus seems to confirm robustness with respect to penalties applied to corrupted peers and to external events, such as peers leaving the blockchain and new peers joining. These events may cause unexpected changes in g, such as sudden peaks, but the trend is eventually brought back near θ. The choice of the function `s_funct` depends on many factors, and making good choices is important for determining the final behavior of the system. Indeed, the latter experiment shows that it is possible to "control" the trend of g with the desired level of flexibility, by choosing appropriately the function applied to s and the value of k.

5. Conclusions and Directions for Future Work

In this paper we have described a simulator of PoS-based consensus algorithms, and we have performed some experiments to investigate the behavior of some PoS-based consensus algorithms in terms of wealth distribution, measured using the Gini coefficient. Furthermore, we have introduced a new PoS-based consensus algorithm that makes the value of the Gini coefficient converge to a desired target value θ.

The simulator is not intended to be a complete solution but rather as a generic skeleton to be customized according to the precise implementation of the consensus algorithm to be analyzed. Adopting this point of view, we have provided some examples of analyses performed on some hypothetical implementations of PoS. The simulator allows to tweak several parameters of the consensus algorithm and observe how the distribution of cryptocurrency coins among the users evolves over time. The final aim is to help researchers analyze the behavior of existing implementations of PoS consensus algorithms, and the designers of such algorithms to find the values of parameters that make the protocol fair and sustainable in the long term. With respect to this latter goal, we proposed an adaptive mechanism that makes the value g of the Gini coefficient converge to (and then smoothly oscillate around) a desired value θ. Fixing $\theta = 0.3$, which is the value indicated by many economists for a fair wealth distribution, we obtain a behavior which seems to be sustainable in the long run.

The presented work has several limitations, and can be extended in several ways. First of all, our simulator assumes that the peers of the blockchain put all their cryptocurrency coins in the stake at each epoch. An enhanced simulator could handle peers that put a certain amount of coins in the stake according to a certain distribution, depending on the epoch. A possible extension concerns the possibility to make the peers join and leave the system according to user-defined probability distributions. The same applies to the amount of coins given to the new peers. Another possible extension concerns the indices and coefficients used to analyze wealth distributions: as we have seen, many papers in the literature use also Shannon entropy and Nakamoto coefficient to perform their analyses; at the moment our simulator only uses the Gini coefficient – we have started from this one because it is widely adopted in economics studies about wealth distributions. Adding the computation of further coefficients is not difficult, and will certainly be done in a future version of the simulator. Similarly, it would be possible to determine, by linear regression, the Zipf's law coefficients that best approximate the wealth distribution under study.

When designing a cryptocurrency, initial supply and subsequent distribution of coins are fundamental problems to tackle and consider. Due to Proof-of-Stake's intrinsic initial supply requirements, blockchain networks implementing PoS as a distributed consensus mechanism present an important pre-mined initial distribution, in terms of coin percentage of the entire network. In this paper we have ignored this aspect, and we have simply assumed that each user initially obtains a number of coins according to some predefined fixed distribution, which may be uniform, random, or with a fixed Gini coefficient. While this means that our simulator is only able to analyze situations where the blockchain has already been running for some time, it is not clear to the authors whether the simulator should really consider also the start-up period in which the creators of the blockchain distribute cryptocurrency coins to the prospective users, according to some political, monetary, and marketing strategy.

Finally, a clear direction for future research is to use more elaborated versions of the proposed simulator for investigating which combinations of parameters, and which policies – implementing forces that increase or decrease the number of coins in the system, and their assignment to the peers – make it possible to obtain a variant of PoS that is fair and hence sustainable in the long run. A comparison with other approaches taken in the literature could also be helpful [**? ? ?**].

Further, we would like to simulate more realistic scenarios. For example, in the real world, external events can determine sudden drops or surges in the value of a cryptocurrency, resulting in a large number of users leaving or joining in a short period of time. The blockchain should resist even these "catastrophic events", converging towards a new point of equilibrium. It is well known that markets are complex and chaotic systems [33]; some studies even show fractal properties [22]. Therefore, such a study will probably involve mathematical tools commonly used in the theory of complex systems, and will require the implementation of much more sophisticated simulators than the one presented in this paper.

References

[1] Shubhani Aggarwal and Neeraj Kumar. Cryptographic consensus mechanisms. In *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, volume 121 of *Advances in Computers*, pages 211–226. Elsevier, 2021.

[2] Imran Bashir. *Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition*. Packt Publishing, 2020.

[3] Imran Bashir. *Blockchain Age Protocols*, pages 331–376. Apress, Berkeley, CA, 2022. ISBN 978-1-4842-8179-6. doi: 10.1007/978-1-4842-8179-6_8.

[4] Nicola Dimitri. Monetary Dynamics With Proof of Stake. *Frontiers in Blockchain*, 4:443966, 2021. doi: 10.3389/fbloc.2021.443966.

[5] Georgios Drakopoulos, Eleanna Kafeza, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. Simulating Blockchain Consensus Protocols in Julia: Proof of Work vs Proof of Stake. In Ilias Maglogiannis, Lazaros Iliadis, John Macintyre, and Paulo Cortez, editors, *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops*, pages 357–369. Springer International Publishing, 2022. ISBN 978-3-031-08341-9.

[6] Ethereum.org. The Merge. `https://ethereum.org/en/upgrades/merge/`, 2022. Accessed: January 18, 2023.

[7] Flashbots. `https://www.flashbots.net/`, 2020. Accessed: March 4, 2023.

[8] United Nations Research Institute for Social Development (UNRISD). Inequalities and the Post-2015 Development Agenda. Concept Note. 2013. URL `https://www.files.ethz.ch/isn/159691/02%20-%20Inequalities.pdf`.

[9] Lina Ge, Jie Wang, and Guifen Zhang. Survey of Consensus Algorithms for Proof of Stake in Blockchain. *Security and Communication Networks*, 2022, 2022. doi: 10.1155/2022/2812526.

[10] Felix Irresberger. Coin Concentration of Proof-of-Stake Blockchains. *SSRN Electronic Journal*, 01 2018. doi: 10.2139/ssrn.3293694.

[11] Johannes Rude Jensen, Victor von Wachter, and Omri Ross. How Decentralized is the Governance of Blockchain-based Finance: Empirical Evidence from four Governance Token Distributions. Paper 2102.10096, arXiv.org, February 2021.

[12] Leonid Kogan, Giulia C. Fanti, and Pramod Viswanath. Economics of Proof-of-Stake Payment Systems. *MIT Sloan Research Paper No. 5845-19*, 5 2021. doi: 10.2139/ssrn.4320274.

[13] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network. *PLOS ONE*, 9(2):1–10, 02 2014. doi: 10.1371/journal.pone.0086197.

[14] B. Kusmierz and R. Overko. How centralized is decentralized? Comparison of wealth distribution in coins and tokens. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society. doi: 10.1109/COINS54846.2022.9854972.

[15] Bartosz Kuśmierz, Sebastian Müller, and Angelo Capossele. Committee Selection in DAG Distributed Ledgers and Applications. In Kohei Arai, editor, *Intelligent Computing*, pages 840–857, Cham, 2021. Springer International Publishing. ISBN 978-3-030-80126-7.

[16] Alberto Leporati. Studying the compounding effect: The role of proof-of-stake parameters on wealth distribution. In Paolo Mori, Ivan Visconti, and Stefano Bistarelli, editors, *Proceedings of the Fifth Distributed Ledger Technology Workshop (DLT 2023), Bologna, Italy, May 25-26, 2023*, volume 3460 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL `https://ceur-ws.org/Vol-3460/papers/DLT_2023_paper_2.pdf`.

[17] Alberto Leporati and Lorenzo Rovida. PoS Simulator. `https://github.com/narger-ef/PoS-Simulator`, 2024. Accessed: April, 2024.

[18] Aiya Li, Xianhua Wei, and Zhou He. Robust Proof of Stake: A New Consensus Protocol for Sustainable Blockchain Systems. *Sustainability*, 12(7):1–15, 2020. URL `https://ideas.repec.org/a/gam/jsusta/v12y2020i7p2824-d340545.html`.

[19] Chao Li and Balaji Palanisamy. Comparison of Decentralization in DPoS and PoW Blockchains. In Zhixiong Chen, Laizhong Cui, Balaji Palanisamy, and Liang-Jie Zhang, editors, *Blockchain – ICBC 2020*, pages 18–32, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59638-5.

[20] Q. Lin, C. Li, X. Zhao, and X. Chen. Measuring Decentralization in Bitcoin and Ethereum using Multiple Metrics and Granularities. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, pages 80–87, Los Alamitos, CA, USA, apr 2021. IEEE Computer Society. doi: 10.1109/ICDEW53142.2021.00022.

[21] Davide Mancino, Alberto Leporati, Marco Viviani, and Giovanni Denaro. Exploiting Ethereum after "The Merge": The Interplay between PoS and MEV Strategies. In Francesco Buccafurri, Elena Ferrari, and Gianluca Lax, editors, *Proceedings of the Italian Conference on Cyber Security (ITASEC 2023), Bari, Italy, May 2-5, 2023*, volume 3488 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3488/paper24.pdf.

[22] Benoit Mandelbrot and Richard. L. Hudson. *The (Mis)Behaviour of Markets: A Fractal View of Risk, Ruin and Reward*. Profile, 2010. ISBN 9781847651556.

[23] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[24] Cong T. Nguyen, Dinh Thai Hoang, Diep N. Nguyen, Dusit Niyato, Huynh Tuong Nguyen, and Eryk Dutkiewicz. Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. *IEEE Access*, 7:85727–85745, 2019. doi: 10.1109/ACCESS.2019.2925010.

[25] Remigijus Paulavičius, Saulius Grigaitis, and Ernestas Filatovas. A systematic review and empirical analysis of blockchain simulators. *IEEE Access*, 9:38010–38028, 2021. doi: 10.1109/ACCESS.2021.3063324.

[26] Muhammad Saad, Zhan Qin, Kui Ren, Daehun Nyang, and David Mohaisen. e-PoS: Making Proof-of-Stake Decentralized and Fair. *IEEE Transactions on Parallel and Distributed Systems*, 32(8):1961–1973, 2021. doi: 10.1109/TPDS.2020.3048853.

[27] Yenatfanta Shifferaw and Surafel Lemma. Limitations of proof of stake algorithm in blockchain: A review. *Zede Journal*, 39(1):1961–1973, 2021.

[28] Steem blockchain. https://steem.com/, 2018. Accessed: March 4, 2023.

[29] Scott Nadal Sunny King. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. 2012. URL http://www.peercoin.net/.

[30] Vicent Sus. Proof-of-Stake Is a Defective Mechanism. *IACR Cryptol. ePrint Arch.*, 2022:409, 2022. URL https://eprint.iacr.org/2022/409.pdf.

[31] Yong Tao, Xiangjun Wu, and Changshuai Li. Rawls' fairness, income distribution and alarming level of Gini coefficient. *Economics: The Open-Access, Open-Assessment E-Journal*, (2017-67), 2017. URL http://www.economics-ejournal.org/economics/discussionpapers/2017-67.

[32] Paolo Tasca and Claudio J. Tessone. A Taxonomy of Blockchain Technologies: Principles of Identification and Classification. *Ledger*, 4, 2019. doi: 10.5195/ledger.2019.140.

[33] James O. Weatherall. *The Physics of Wall Street: A Brief History of Predicting the Unpredictable*. Houghton Mifflin Harcourt, 2013. ISBN 9780547317274.

[34] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. 2014. URL `https://gavwood.com/paper.pdf`.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for *[Journal name]* and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Alberto Leporati: Conceptualization, Methodology, Investigation, Writing- Reviewing and Editing
Lorenzo Rovida: Methodology, Investigation, Software, Visualization, Writing- Reviewing and Editing