



## Research Article

## Looking for stability in proof-of-stake based consensus mechanisms

Alberto Leporati<sup>\*</sup>, Lorenzo Rovida

Department of Informatics, Systems and Communication, University of Milano-Bicocca, Viale Sarca 336, Milan 20126, Italy

## ARTICLE INFO

## Keywords:

Blockchain  
Proof-of-stake  
Compounding effect  
Wealth distribution  
Tokenomics

## ABSTRACT

The Proof-of-Stake (PoS) consensus algorithm has been criticized in the literature and in several cryptocurrency communities, due to the so-called compounding effect: who is richer has more coins to stake, therefore a 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, tweak their parameters, and observe how the distribution of cryptocurrency coins among users evolves over time. Such a tool can be used to investigate which combinations of parameter 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.

## 1. Introduction

The advent of Bitcoin [1] 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 Ref. [2].

In the first few years of DLTs, questions of a technological nature received the most attention; questions about economics, cryptocurrency distributions, and tokenomics have been addressed much less thoroughly. Some papers in the literature address the issue of (lack of) decentralization in blockchain governance [3–6], implicitly assuming that who is richer has more power in making decisions about which transactions and blocks to validate. Although 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: also 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 the PoW and 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 the 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 the compounding effect [7–9], 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 [10] announced their intention to switch from PoW to PoS, which they did on September 15, 2022, in the event known as “The Merge” [11]. 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 which 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 [12], a (quite elaborate) infrastructure running on top of Ethereum’s blockchain. Flashbots, as well as other recent sim-

<sup>\*</sup> Corresponding author.

E-mail addresses: [alberto.leporati@unimib.it](mailto:alberto.leporati@unimib.it) (A. Leporati), [lorenzo.rovida@unimib.it](mailto:lorenzo.rovida@unimib.it) (L. Rovida).

ilar 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 Ref. [13].

In this paper, we partially address questions about wealth distribution in blockchains that make use of the PoS family of consensus algorithms. PoS was initially designed to improve the energy consumption derived from PoW [14]. 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 PoS (DPoS), and Liquid PoS (LPoS). For a description of how these algorithms work, we refer the reader to Ref. [15–18]. 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 [19]. In order to make our study more general, we will not focus our attention on a specific variant of the 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 parameter 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 people makes the permissionless blockchain a centralized system controlled by an oligarchy. This entails that users will no longer trust the system and therefore will leave it.

Precisely, the research question that we want to address in this paper is as follows.

**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 follows. In Section 2, we recall some related works from the literature, that investigate the distribution of wealth among blockchain validators. In Section 3 we describe the two main contributions of this 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 the conclusions and delineate the directions for future research.

## 2. Related works

In this section we recall some works taken from the literature, that are related to the topic under study. However, as discussed below, our perspective is slightly different, and we believe that our approach may be of some interest for designing and/or testing the behavior of PoS-based consensus algorithms.

Some scientific studies on the distribution of wealth among the top richest users of PoW and PoS-based blockchains have been performed [20,7,21,22]. For example, Ref. [8,23,24] show that the distributions

of the top richest balances might be modeled with Zipf's law. Additionally, the Gini coefficients are computed for each user to measure wealth inequality. Moreover, in Ref. [8], 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 Ref. [4], the authors analyzed 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 abovementioned metrics, they showed that there are quantitative differences in the centralization levels between cryptocurrency coins and tokens. It is thus observed that tokens are, in general, much more centralized than coins, with higher Gini coefficients and smaller Nakamoto coefficients.

All these studies focus on the top richest account, and hence might be of particular interest to DLTs where a group of top cryptocurrency holders fulfill 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 PoS (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 Ref. [6], the authors provided their analysis using three different metrics (Gini coefficient, Shannon entropy, and Nakamoto coefficient) and their evolution over time. It is found that the degree of decentralization in Bitcoin is higher and more volatile, while the degree of decentralization in Ethereum (when the PoW consensus mechanism was still adopted) is lower and more stable. Jensen et al. [3] analyzed the decentralization of governance token distribution in four Decentralized Finance (DeFi) applications on the Ethereum blockchain using Gini and Nakamoto coefficients. Their results indicate that the token distributions for all four DeFi applications are characterized by high Gini coefficients. Similar methods are used in Ref. [24], where PoW and PoS cryptocurrencies are compared, analyzing the decentralization of Bitcoin and Steem [25] 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. Furthermore, the data concerning wealth distribution usually presented in the literature do not represent the wealth of individual cryptocurrency owners but rather the wealth distribution among 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 Ref. [26]. For instance, Ref. [27] performs a comparison of reward distributions between PoW-based and PoS-based blockchains, showing that PoS has a more fair reward distribution. However, none of the proposed simulators allows for a comparison between different consensus mechanisms in PoS.

In this paper, we take a different approach with respect to the abovementioned papers. Instead of analyzing existing data about cryptocurrency or token distributions in blockchains, we study under which

conditions a PoS-based consensus algorithm allows us 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 (or 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, we instead 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 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 coin supply, the simulator computes the evolution of wealth distribution over time, measuring its fairness by means of the 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 a few people makes the permissionless blockchain a centralized system controlled by an oligarchy. This entails that users no longer trust the system and therefore abandon it. Therefore, 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 Ref. [9], 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 [28,5]. For instance, Ref. [5] proposes a consensus mechanism that protects the system from the risk of coin age accumulation attacks. Saad et al. [28] proposed a consensus algorithm in which performing a fair mining is promoted by the nature of the algorithm itself. Both of these works, however, lack 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 Ref. [29], in which an analogous PoS simulator written in the R language is proposed. While such a previous work focused on the proposal of the simulator, we find that the performances obtained are 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 upon 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  $\theta$  for the Gini coefficient, adapts its behavior in order to approach it.

### 3. A PoS simulator

In order to address the research question posed in the previous section, we have developed a PoS-based simulator using the Julia language.

As stated above, the simulator computes the wealth distribution in terms of the 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 with 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 wealth among the citizens of a country. More generally, 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, which corresponds to absolute centralization. As commonly agreed upon 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 the United Nations to reduce economic inequalities in the world [30]. Let us note, however, that these values are not strongly supported by theoretical analyses; on the contrary, based on some data analysis, Ref. [31] indicates 0.5 as a better threshold value. For this reason, we do 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  $\theta$ , whatever it may be.

The role of the simulator is to help study 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  $i$ .

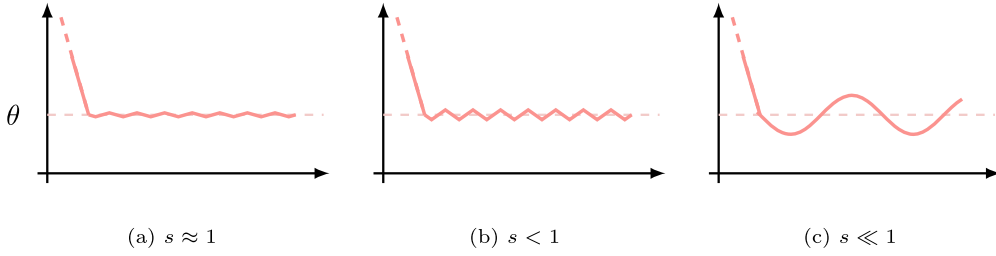
- **Opposite-Weighted PoS:** The probability of a participant being chosen is computed as the complement of the previous consensus mechanism:

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

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

#### 3.1. Gini-Stabilized PoS

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



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

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

This mechanism pushes the Gini coefficient toward 1 when  $g < \theta$ , and toward 0 otherwise. Nevertheless, we must introduce a way to interpolate between the two distributions, allowing a smooth transition between them to avoid an alternating back-and-forth motion (i.e., a “zig-zag” effect) around  $\theta$ . 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:

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

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

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

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

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

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 such as the one shown in Fig. 1.

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

$$P_i = \ell(P_w, 1 - P_w, t_j) \quad (3)$$

where  $P_w$  is the probability distribution defined in the Weighted PoS and  $j$  is the index of the current epoch which is 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 explore different approaches (that is, functions) to assigning values to  $s$ . The first, and most straightforward, approach 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 ( $\theta$ ), 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 has 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  $\theta$ , we increase 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 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 toward 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. On the one hand, this constitutes a disincentive, since the amount of money lost in the 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. However, another possibility is that whoever has validated a block cannot do so for a certain number of subsequent time slots. In the simulator, we call this number of epochs the standby period. Under the assumption that few people will try to cheat, this mechanism makes their participation in 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 significantly influence 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 into stake; thus, if a cheater puts in a small amount of money, they will be selected very rarely. Ultimately, 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.

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

The simulator, whose source code is available at Ref. [32], allows one to track the distribution of cryptocurrency coins over time, for a



**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: { <i>Weighted</i> , <i>OppositeWeighted</i> , <i>GiniStabilized</i> }
initial_stake_volume	Initial number of coins distributed among peers, based on initial_distribution
initial_distribution	Initial distribution of coins among validators: { <i>Uniform</i> , <i>Gini</i> , <i>Random</i> }
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: { <i>Average</i> , <i>Random</i> , <i>Max</i> , <i>Min</i> }
reward_type	Type of reward: <i>Constant</i> or <i>Dynamic</i> . 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

**Table 2**  
Optional parameters of the simulator.

Parameter	Description
initial_gini	In case <i>Gini</i> 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 $\theta$ , to be defined in case of <i>GiniStabilized</i> consensus mechanism
initial_stake_volume	Initial number of coins distributed among peers, based on initial_distribution
s_funct	The smoothing function applied to $s$ (refer to Equation (2)). It takes a value among <i>Constant</i> , <i>Linear</i> and <i>Quadratic</i>
k	The value of $k$ , given as input to the smoothing function used to compute $s$

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 on 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.
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

**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

```

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 newly 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 PoS 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., it 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 its 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 0 standby periods are reduced by 1.

#### 4. 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, and 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`, which are 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 [32].

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

##### 4.1. Experiment 1: Weighted PoS

As the first experiment, we simply want to assess the tendency of Weighted consensus to push the Gini coefficient toward 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 Fig. 2, the Gini coefficient  $g$  tends to grow toward 1, independent of the choice of the initial distribution of coins.

Additionally, we want to study the effect that this consensus mechanism has on the difference between the distribution of stakes 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 number of coins (initial distribution set to Random), and we present the percentage of stake held by each peer across iterations in Fig. 3.

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

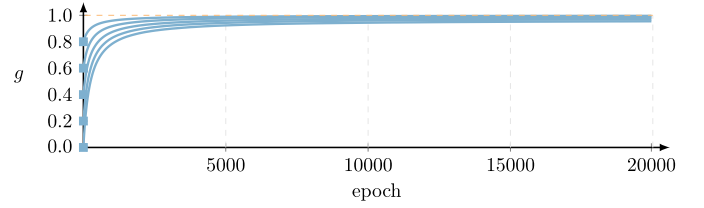


Fig. 2. Trend of the Gini coefficient  $g$  using the Weighted consensus algorithm, on five different starting distributions.

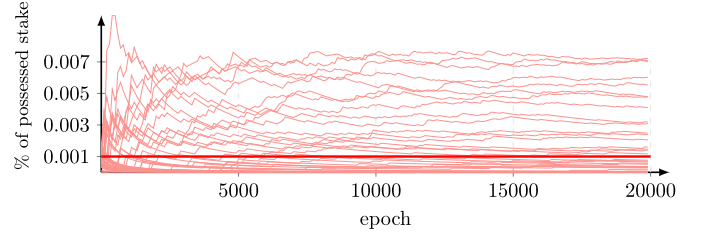


Fig. 3. Percentage of stake held across iterations using the Weighted PoS, with a random starting distribution. The red line indicates the value of  $1/n_{\text{peers}}$ .

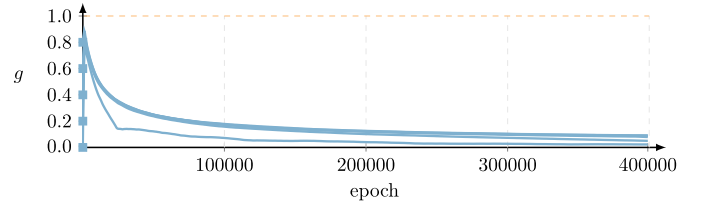


Fig. 4. Trend of the Gini coefficient  $g$  using the Opposite-Weighted consensus algorithm, on five different starting distributions.

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 are below the average, and the other 49.6% are above. After the simulation, this ratio changes to 76.7% over 23.3%. This implies that rewards for validating blocks are collected by an increasingly smaller number of peers, making the blockchain monopolized by them. This experiment indeed shows 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

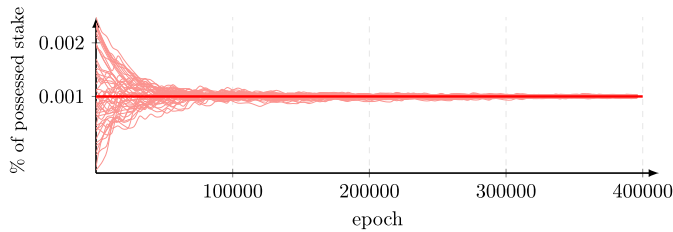
The second experiment aims to assess the tendency of Opposite-Weighted. As stated at the beginning of Section 3, the implementation of this consensus mechanism causes the Gini coefficient to tend toward zero. Apart from the mechanism and the number of epochs, the set of parameters is the same as in the previous experiment. As presented in Fig. 4, the experiment shows that the value of  $g$  tends to zero, regardless of the initial distribution of coins. The number of epochs has 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 Fig. 5, we present its trend across iterations.

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 eventually become the same for everyone.

**Table 3**

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

Parameters	Experiments				
	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	50.000
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 ( $\theta$ )	-	-	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



**Fig. 5.** Percentage of stake held across iterations using the Opposite-Weighted PoS, with uniform starting distribution.

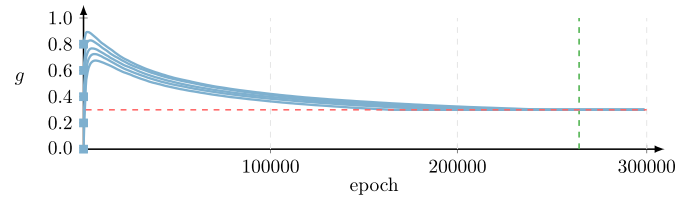
These two experiments aim to empirically prove the behavior of two basic consensus algorithms. Additionally, we point out a problem related to the percentages of stakes held among peers. Ideally, we would like to have a consensus mechanism that preserves such percentages over iterations because such 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  $s$ .

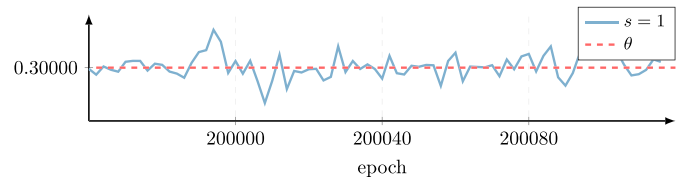
#### 4.3. Experiment 3: Gini-Stabilized with constant update

We start by showing the occurrence of the “zig-zag” effect centered at 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 Opposite-Weighted as soon as  $g$  crosses the  $\theta$  threshold, without any smooth interpolation. The results are shown in Fig. 6.

It is interesting to notice that, in the very first epochs, before starting to descend toward 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 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 Fig. 6 with a vertical green line. It is possible to visually



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



**Fig. 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  $g$ .

recognize the zig-zag phenomenon by observing the trend of  $g$  in small intervals of epochs after it has stabilized near  $\theta$  (see Fig. 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 in plotting the trend of  $g$  after it has stabilized near  $\theta$ . We show the results in Fig. 8 for a larger number of epochs to highlight the differences with  $s = 1$ . As it can be seen, when using a smaller value of  $s$ , the amplitude of the trend is larger, and the frequency is smaller, although we still lack 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 Fig. 9. Notice an unpredictable behavior in the initial phase before the Gini coefficient reaches the target value  $\theta$ . 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.

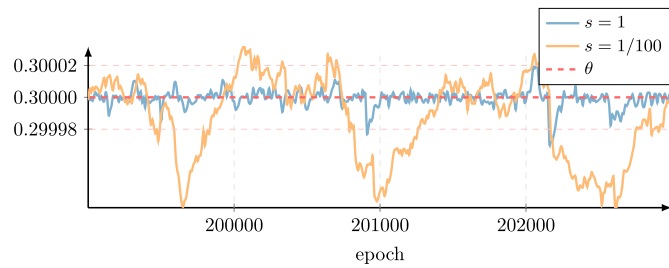


Fig. 8. A comparison between  $s = 1$  and  $s = 1/100$ , analyzed in 2000 epochs, after a stabilization near  $\theta = 0.3$ . The y-axis represents the value of  $g$ .

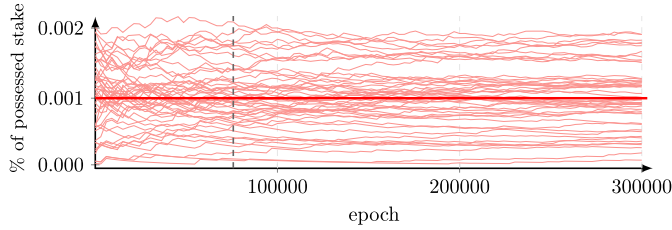


Fig. 9. Percentage of possessed stake across iterations using the Gini-Stabilized PoS, with a random starting distribution. The black dashed line indicates the point in which the Gini target value  $\theta = 0.3$  has been reached.

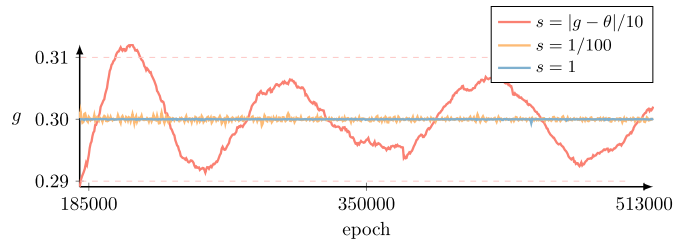


Fig. 10. A comparison between Constant and Linear update strategies for  $s$ , analyzed following a stabilization near the target value  $\theta$ .

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, the ratio between the number 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 that in Experiment 1 and is 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  $s$  is 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  $s_{\text{funct}}$  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  $\theta$  in Fig. 10. Here a comparison is made between the trends of the Gini coefficient for three update strategies, corresponding to the constant values  $s = 1$ ,  $s = 1/100$ , and  $s = |g - \theta|/10$ . 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.

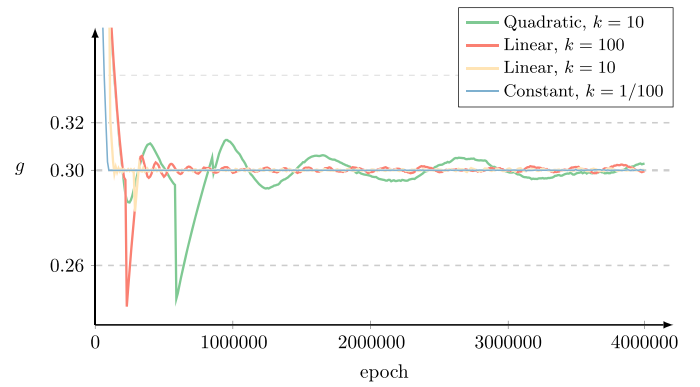


Fig. 11. A comparison between *Constant*, *Linear* and *Quadratic* update strategies.

#### 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  $s_{\text{funct}}$ : *Constant*, *Linear* and *Quadratic* functions. Additionally, we set the probabilities for new peers to join and quit the pool at each epoch equal to 0.1%, thus simulating a more realistic scenario. The number of epochs is equal to 4,000,000, and initially, the pool consists 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 Fig. 11.

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 the 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  $\theta$ . The choice of the function  $s_{\text{funct}}$  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  $\theta$ .

The simulator is not intended to be a complete solution but rather 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 one 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 propose an adaptive mechanism that makes the value  $g$  of the Gini coefficient converge to (and then smoothly oscillate around) a desired value  $\theta$ . Fixing  $\theta = 0.3$ , which is the value indicated by many economists for a fair wealth distribution, we obtain a behavior that 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 can handle peers that put a certain number of coins in the stake according to a certain distribution, depending on the epoch. A possible extension concerns the possibility of making the peers join and leave the system according to user-defined probability distributions. The same applies to the number 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 also use 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, the initial supply and subsequent distribution of coins are fundamental problems to tackle and consider. Due to PoS's intrinsic initial supply requirements, blockchain networks implementing PoS as a distributed consensus mechanism present an important pre-mined initial distribution, in terms of the 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 also consider 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 elaborate 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 [33–35].

Furthermore, 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 toward a new point of equilibrium. It is well known that markets are complex and chaotic systems [36]; some studies even show fractal properties [37]. 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.

## CRedit authorship contribution statement

**Alberto Leporati:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Lorenzo Roviða:** Writing – review & editing, Visualization, Software, Methodology, Investigation.

## Declaration of competing interest

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.

## References

- [1] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, <http://www.bitcoin.org/bitcoin.pdf>, 2009.

- [2] S. Aggarwal, N. Kumar, Cryptographic consensus mechanisms, in: S. Aggarwal, N. Kumar, P. Raj, et al. (Eds.), *The Blockchain Technology for Secure and Smart Applications Across Industry Verticals*, Elsevier, 2021, pp. 211–226, <https://doi.org/10.1016/bs.adcom.2020.08.011>.
- [3] J.R. Jensen, V. von Wachter, O. Ross, How Decentralized is the Governance of Blockchain-based Finance: Empirical Evidence from four Governance Token Distributions, arXiv, 2021, preprint, arXiv:2102.10096.
- [4] B. Kusmierz, R. Overko, How centralized is decentralized? Comparison of wealth distribution in coins and tokens, in: *Proceedings of the 2022 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, IEEE, 2022, pp. 1–6, <https://doi.org/10.1109/COINS54846.2022.9854972>.
- [5] A. Li, X. Wei, Z. He, Robust Proof of Stake: a new consensus protocol for sustainable blockchain systems, *Sustain.* 12 (7) (2020) 1–15, <https://doi.org/10.3390/su12072824>.
- [6] Q. Lin, C. Li, X. Zhao, et al., Measuring decentralization in bitcoin and Ethereum using multiple metrics and granularities, in: *Proceedings of the 2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, IEEE, 2021, pp. 80–87, <https://doi.org/10.1109/ICDEW53142.2021.00022>.
- [7] Felix Irresberger, Coin concentration of Proof-of-Stake blockchains, SSRN, 2019, preprint, <https://ssrn.com/abstract=3293694>.
- [8] D. Kondor, M. Pósfai, I. Csabai, et al., Do the rich get richer? An empirical analysis of the bitcoin transaction network, *PLOS ONE* 9 (5) (2014) e97205, <https://doi.org/10.1371/journal.pone.0086197>.
- [9] V. Sus, Proof-of-Stake is a defective mechanism, <https://eprint.iacr.org/2022/409.pdf>, 2022. (Accessed 4 March 2023).
- [10] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, <https://gavwood.com/paper.pdf>, 2014. (Accessed 4 March 2023).
- [11] Ethereum.org, The merge, <https://ethereum.org/en/upgrades/merge/>. (Accessed 18 January 2023).
- [12] Flashbots, <https://www.flashbots.net/>. (Accessed 4 March 2023).
- [13] D. Mancino, A. Leporati, M. Viviani, et al., Exploiting Ethereum after “the merge”: the interplay between PoS and MEV strategies, in: *Proceedings of the Italian Conference on Cyber Security (ITASEC 2023)*, CEUR Workshop Proceedings, 2023, <https://eur-ws.org/Vol-3488/paper24.pdf>.
- [14] S. Nadal, S. King, PPKoin: peer-to-peer crypto-currency with Proof-of-Stake, <https://decred.org/research/king2012.pdf>, 2012. (Accessed 4 March 2023).
- [15] I. Bashir, Blockchain Age Protocols, in: *Blockchain Consensus*, Apress, Berkeley, CA, 2022, pp. 331–376, [https://doi.org/10.1007/978-1-4842-8179-6\\_8](https://doi.org/10.1007/978-1-4842-8179-6_8).
- [16] I. Bashir, Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum, and More, 3rd ed., Packt Publishing, Birmingham, UK, 2020.
- [17] T.C. Nguyen, D.T. Hoang, D.N. Nguyen, et al., Proof-of-Stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities, *IEEE Access* 7 (2019) 85727–85745, <https://doi.org/10.1109/ACCESS.2019.2925010>.
- [18] L. Ge, J. Wang, G. Zhang, Survey of consensus algorithms for Proof of Stake in blockchain, *Secur. Commun. Netw.* (2022) 2812526, <https://doi.org/10.1155/2022/2812526>.
- [19] P. Tasca, C.J. Tessone, A taxonomy of blockchain technologies: principles of identification and classification, *Ledger* 4 (2019), <https://doi.org/10.5195/ledger.2019.140>.
- [20] N. Dimitri, Monetary dynamics with Proof of Stake, *Front. Blockchain* 4 (2021) 443966, <https://doi.org/10.3389/fbloc.2021.443966>.
- [21] Leonid Kogan, Giulia C. Fanti, Pramod Viswanath, Economics of Proof-of-Stake payment systems, SSRN, 2021, preprint, <https://ssrn.com/abstract=4320274>.
- [22] Yenatfanta Shifferaw, Surafel Lemma, Limitations of proof of stake algorithm in blockchain: a review, *Zede J.* 39 (1) (2021) 1961–1973.
- [23] B. Kusmierz, S. Müller, A. Caposelle, Committee selection in DAG distributed ledgers and applications, in: K. Arai, et al. (Eds.), *Intelligent Computing*, Springer, Cham, 2021, pp. 840–857, [https://doi.org/10.1007/978-3-030-80126-7\\_59](https://doi.org/10.1007/978-3-030-80126-7_59).
- [24] C. Li, B. Palanisamy, Comparison of decentralization in DPoS and PoW blockchains, in: Z. Chen, L. Cui, B. Palanisamy, et al. (Eds.), *Blockchain—ICBC 2020*, Springer, Cham, 2020, pp. 18–32, [https://doi.org/10.1007/978-3-030-59638-5\\_2](https://doi.org/10.1007/978-3-030-59638-5_2).
- [25] Steem, Steem blockchain, <https://steem.com/>. (Accessed 4 March 2023).
- [26] R. Paulavičius, S. Grigaitis, E. Filatovas, A systematic review and empirical analysis of blockchain simulators, *IEEE Access* 9 (2021) 38010–38028, <https://doi.org/10.1109/ACCESS.2021.3063324>.
- [27] G. Drakopoulos, E. Kafeza, I. Giannoukou, et al., Simulating blockchain consensus protocols in Julia: Proof of Work vs Proof of Stake, in: I. Maglogiannis, L. Iliadis, J. Macintyre, et al. (Eds.), *Artificial Intelligence Applications and Innovations, AIAI 2022 IFIP WG 12.5 International Workshops*, Springer, Cham, 2022, pp. 357–369, [https://doi.org/10.1007/978-3-031-08341-9\\_29](https://doi.org/10.1007/978-3-031-08341-9_29).
- [28] M. Saad, Z. Qin, K. Ren, et al., e-PoS: making Proof-of-Stake decentralized and fair, *IEEE Trans. Parallel Distrib. Syst.* 32 (8) (2021) 1961–1973, <https://doi.org/10.1109/TPDS.2020.3048853>.
- [29] A. Leporati, Studying the compounding effect: the role of Proof-of-Stake parameters on wealth distribution, in: *Proceedings of the Fifth Distributed Ledger Technology Workshop (DLT 2023)*, CEUR Workshop Proceedings, 2023, [https://eur-ws.org/Vol-3460/papers/HLT\\_2023\\_paper\\_2.pdf](https://eur-ws.org/Vol-3460/papers/HLT_2023_paper_2.pdf).
- [30] United Nations Research Institute for Social Development (UNRISD), Inequalities and the Post-2015 Development Agenda, Concept Note, 2013, <https://www.files.ethz.ch/isn/153554/RPB15e.pdf>.

- [31] Y. Tao, X. Wu, C. Li, Rawls' fairness, income distribution and alarming level of Gini coefficient, *Econ. Discuss. Pap.* 2017 (67) (2017), <http://www.economics-ejournal.org/economics/discussionpapers/2017-67>.
- [32] A. Leporati, L. Roviida, PoS simulator, <https://github.com/narger-ef/PoS-Simulator>, 2024. (Accessed 1 April 2024).
- [33] H. Afzaal, N.A. Zafar, A. Tehseen, S. Kousar, M. Imran, Formal verification of justification and finalization in Beacon chain, *IEEE Access* 12 (2024) 55077–55102, <https://doi.org/10.1109/ACCESS.2024.3389551>.
- [34] B. Kraner, N. Vallarano, C. Schwarz-Schilling, C.J. Tessone, Agent-based modelling of ethereum consensus, in: 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2023, pp. 1–8, <https://doi.org/10.1109/ICBC56567.2023.10174948>.
- [35] L. Galletta, C. Laneve, I. Mercanti, A. Veschetti, Resilience of hybrid Casper under varying values of parameters, *Distributed Ledger Technol. Res. Pract.* 2 (1) (2023) 5, <https://doi.org/10.1145/3571587>, pp. 1–25.
- [36] J.O. Weatherall, *The Physics of Wall Street: A Brief History of Predicting the Unpredictable*, 1st ed., Houghton Mifflin Harcourt, Boston, Massachusetts, United States, 2013.
- [37] B. Mandelbrot, R.L. Hudson, *The (Mis)behaviour of Markets: A Fractal View of Risk, Ruin, and Reward*, 1st ed., Basic Books, New York, USA, 2004.