Weighted Voting on the Blockchain: Improving Consensus in Proof of Stake Protocols

Stefanos Leonardos*, Daniël Reijsbergen*, Georgios Piliouras*
*Singapore University of Technology and Design

***Abstract*—Proof of Stake (PoS) protocols rely on voting mechanisms to reach consensus on the current state. If an enhanced majority of *staking nodes*, also called validators, agree on a proposed block, then this block is appended to the blockchain. Yet, these protocols remain vulnerable to faults caused by validators who abstain either accidentally or maliciously.**

To protect against such faults while retaining the PoS selection and reward allocation schemes, we study *weighted voting* in validator committees. We formalize the block creation process and introduce validators' *voting profiles* which we update by a *multiplicative weights* algorithm relative to validators' voting behavior and aggregate blockchain rewards. Using this framework, we leverage *weighted majority voting rules* that optimize collective decision making to show, both numerically and analytically, that the consensus mechanism is more robust if validators' votes are appropriately scaled. We raise potential issues and limitations of weighted voting in trustless, decentralized networks and relate our results to the design of current PoS protocols.

***Index Terms*—Proof of Stake, Consensus, Weighted Voting, Multiplicative Weights Update**

I. Introduction

In Nakamoto or Bitcoin-type Proof of Work (PoW) [36], a single miner claims the right to append the next block to the blockchain after solving a secure cryptographic puzzle. The strengths and vulnerabilities of this mechanism are well understood, see [23]–[25] and [22], [28], [46]. Two fundamental properties satisfied by PoW selection are the following: (i) Miners holding $p\%$ of computational power will create on average $p\%$ of blocks that will be part of the blockchain assuming that all miners follow the protocol. This property relies on the randomness of the cryptographic puzzle and ensures *fairness* in the allocation of mining rewards [40], [41]. (ii) Miners can be selected as the next block creators only when they actually create and submit the block to the network. This property says that the random selection of the next block creator and the creation of the next block occur simultaneously.

Proof of Stake (PoS) or virtual mining protocols [14] constitute alternative selection mechanisms that aim to retain PoW's benefits while improving on its weaknesses [11], [15]. While different PoS protocols propose different schemes [12], [17], [26], [29], [31], [42], in general, the block proposal mechanism is the following: blocks are created by staking nodes, also called *validators* [32], [44], who are granted the right to participate in the block creation process by locking capital in protocol currency, called *stake*. Subsequently, a pseudo-random mechanism selects nodes proportionally to their stake to form committees that will decide on the validity of a block proposed by a selected node, called *block proposer* or *leader*. At the core of the consensus mechanism, the selected validators cast approval or disapproval votes on the proposed block. If an enhanced majority of approval votes, usually $2/3$ of the committee size [33], is reached, then the proposed block is accepted and appended to the blockchain.

However, maliciously or accidentally abstaining validators can cause consensus failures and stall the blockchain[1]. The reason is, that unlike Nakamoto consensus [49], PoS protocols decouple the selection of block creator(s) from the actual block creation and hence do not satisfy property (ii) above. Since consensus on the "valid" state of the blockchain relies on the voting behavior of all participating validators, this also implies that the actual rate of blocks that a validator gets to create may differ from the times that she gets selected in a committee. Hence, while the PoS selection mechanism aims to ensure fair allocation of "mining" rewards in line with PoW's property (i) above, in practice, it may fail to do so. These observations motivate the following question

> *How can we improve the efficiency and robustness of the consensus mechanism, i.e., how can we use information on validators' past voting patterns to enforce that with overwhelming probability the selected validator committees will reach consensus on the "valid" state of the blockchain.*

The problem of improving consensus has been treated in the context of fixed-size committee voting by a rich stream of literature [10], [37], [38], [47], [52]. The derived solutions hinge on quantifying voters' abilities to make correct decisions and apply *weighted voting rules* in which votes are weighted according to voters' profiles. Our goal is to apply these results in the setting of PoS protocols. The link is immediate, since, as [10] remarks, the assumptions of their original model imply both decentralized information processing and limited communication. The additional challenges that have to be treated

This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2016NCR-NCR002-028) and administered by the National Cybersecurity R&D Directorate. Georgios Piliouras acknowledges SUTD grant SRG ESD 2015 097, MOE AcRF Tier 2 Grant 2016-T2-1-170 and NRF 2018 Fellowship NRF-NRFF2018-07.

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

[1] Malicious absention refers to non-voting or incorrect voting to attack the protocol, whereas accidental abstention refers to a variety of reasons, such as dropping offline, experiencing network latency, bugs in client updates or being a victim of a censoring/eclipse attack.

in the blockchain context, concern updating these profiles and protecting against their manipulation by adversaries.

To address this problem, we formulate the proper mathematical framework and develop a model to quantify validators' *voting profiles*. The proposed scheme is applied *once* voting committees have formed and does not modify the underlying PoS selection and reward mechanisms. Each staking node (validator) is assigned a score based on her so-far contribution to protocol execution. When selected to a committee, her vote is weighted relative to her profile and consensus is decided according to a *weighted majority rule* that maximizes the expected collective rewards [10], [47]. Finally, based on her vote and the overall consensus outcome, her voting profile is revised according to a fully parametrisable *multiplicative weights update algorithm* [2], [8]. Supported by numerical examples and simulations, our findings demonstrate that weighted voting renders the consensus mechanism more efficient, even if more than $1/3$ of nodes are not properly voting. In this way, the proposed scheme restores fairness without compromising other PoS features. Additionally, since it does not modify the underlying PoS mechanism, it can be tested, implemented and reverted with minimal cost to existing protocol users.

Although weighted voting in distributed systems is known to increase efficiency, incentive compatibility [4], [6], and network reliability [7], it also raises additional risks [3], [51], [54]. Similar to *proof of reputation* systems [53], weighted voting deviates from the principle *one node – one vote* and hence, is vulnerable to manipulation by adversaries. We raise issues that pertain to weighted voting such as loss of anonymity & centralisation and discuss their relevance to protocol design and implementation. Weighted voting becomes particularly relevant in less anonymous [30], private or permissioned blockchains [1], [48], [50], in *delegated* PoS protocols [27], [34] and in PoS protocols with low targeted number of *staking pools* [16].

A. Outline

In Section II, we abstract the PoS consensus mechanism as a voting game. In Section III, we design the improved voting scheme and illustrate our findings with numerical examples and simulations. We conclude with a discussion about limitations and implementation issues in Section IV and summarise our findings in Section V.

II. The Model

Our terminology is based on the Ethereum 2.0 PoS protocol specification, [17]. Yet, our model remains as general as possible in an effort to capture similar voting mechanisms implemented by related PoS platforms[2].

Time: Time is divided into *time slots* $t \in \{0, 1, 2, \dots\}$ of fixed duration d. Each time slot is dedicated to the proposal and creation of a new *block* B_t. Time slot $t = 0$ is the time of creation of the *genesis block* B_0.

Validators: The main actors in the block proposal and creation mechanism are the *staking nodes*, also called *validators*, denoted by $i \in I$, where $I \subseteq \mathbb{N}$ is the *set* of all validators. $\mathcal{B}_{i,t}$ will denote the set of blocks for which validator i is aware of at time $t \geq 0$.

Stake: The deposit or *stake* is the amount of the underlying cryptocurrency that a potential validator locks as *Proof of Stake* (PoS) to participate in the block creation process. Such deposits may change over time. Accordingly, let $v_{i,t} \geq 0$ denote the stake of validator $i \in I$ and $v_t := \sum_{i \in I} v_{i,t}$ the total stake at time slot $t \in \mathbb{N}$. If $v_{i,t} > 0$, then validator i is called *active* at time slot t. Validators who have withdrawn from I or who have not entered it yet, can be thought of as validators with stake $v_{i,t} = 0$. Thus, although the set of validators is dynamic, we may write I instead of I_t to denote the set of validators at any time $t \geq 0$,

Block proposer & committees: To create blocks and extend the blockchain, active validators are selected *proportionally to their stake* by a pseudo-random mechanism which assigns to each time slot t a *leader* or *block proposer* and a fixed-sized *committee* $N_t = \{1, 2, \dots, n\}$ of validators[3]. The block proposer is assigned the task to propose a block B_t to the committee. In turn, the committee votes on whether the proposed block should be appended to the blockchain or not. This process constitutes the core of the consensus mechanism and will be the focus of the present paper.

Validators' strategies: The set of strategies of a validator who has been selected in a committee will be denoted by $S = \{-1, 1\}$, where -1 stands for rejecting and 1 for approving the proposed block. In particular, -1 also corresponds to *not* casting a vote, either deliberately or accidentally. Accordingly, let $X_{i,t}$ denote the indicator random variable

$$X_{i,t} = \begin{cases} 1, & \text{if validator } i \text{ voted on the approval of the} \\ & \text{proposed block } B_t \text{ in time slot } t \\ -1, & \text{otherwise} \end{cases}$$

We will write $\mathbf{x}_t := (x_{1,t}, x_{2,t}, \dots, x_{n,t}) \in \{-1, 1\}^n$ to denote the *decision* or *action* profile of the committee at time $t \geq 0$.

Decision rule: A *decision rule* $f : \{-1, 1\}^n \to \{-1, 1\}$, also called *social choice function* or *aggregation of preferences rule*, is a function that receives as input the action profile $\mathbf{x}_t$ and outputs a decision in $\{-1, 1\}$, where -1 and 1 stand for dissaproval and approval of the proposed block, respectively. We will focus on *(simple or enhanced) majority rules* $f_q, q \in [0.5, 1]$ defined by

$$f_q(\mathbf{x}_t) := \begin{cases} 1, & \text{if } \sum_{i=1}^{n} x_{i,t} \geq (2q - 1)\, n, \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

[2] Claiming that *all* PoS protocols fit under this model would be oversimplifying and wrong. Yet, most PoS protocols that we are aware of involve voting mechanisms and hence, may benefit – to a larger or lesser extent – from the present proposal. An incomplete list includes [12], [26], [29], [31], [42]. For more extensive details of PoS protocols, we refer to [9], [15], [19], [21].

[3] The mechanics of the pseudo-random mechanism vary between different protocols. Here, we are not interested in risks associated with manipulating this mechanism and focus on the mechanics of the voting process *once* a random committee has been formed.

If at least $q \in [0.5, 1]$ of the selected validators approve the proposed block, then this block is appended to the blockchain. Otherwise, the time slot remains empty and the mechanism progresses to the next time slot.

If block proposers follow the protocol and do not behave maliciously, then all proposed blocks are valid. Moreover, if the network is not partitioned and network latency is insignificant (lower than the time slot duration during which votes are expected to appear), then there is no controversy about which blocks are valid, since all validators view essentially the same blockchain (state), i.e. $\mathcal{B}_{i,t} := \mathcal{B}$ for all $i \in I, t \geq 0$. Under these conditions, the required majority should be reached and valid blocks should be regularly approved and appended [26], [42]. However, in practice, two main reasons may lead to failures on the consensus mechanism

- adversarial behavior: a malicious node abstains from voting or censors other validators' votes to block the required majority and stall the block creation process.
- accidental behavior: validators drop offline accidentally or due to negligence, they experience bugs on client updates or bad network connectivity, their votes do not propagate through the network in the expected time slot or they are victims themselves of a censoring/eclipse attack.

Our goal is to study how existing results on optimizing aggregation of preferences in committees, in particular [10], [37], [47] can be applied to the PoS blockchain setting and improve the underlying consensus mechanism. The application of these results will be immediate once we have defined the proper framework.

III. An Improved Voting Rule

To optimize the consensus process from an aggregative perspective, we quantify the collective benefits and losses (payoffs) from correct and wrong decisions respectively. This is done in Table I. The benefit from making a correct decision, i.e., approving a valid block or rejecting an invalid block, is scaled to 1. Here, -1 denotes an invalid and 1 a valid block B_t, cf. Section IV. If a valid block is rejected, then a loss of $\ell_r > 0$ is incurred which corresponds to the waste of computational resources and the failure of the system to process pending transactions. On the other hand, if validators vote for an invalid block, then $\ell_a > 0$ represents the losses from validating a conflicting history[4]. Determining the exact values of ℓ_r and ℓ_a is a matter of protocol parametrisation. Finally, let $\alpha \in (0, 1)$ denote the prior probability that a proposed block is *invalid*, e.g., blocks that an adversarial is trying to create.

		Proposed Block B_t	
		Valid (1)	Invalid (-1)
Committee	Approve	1	$-\ell_a$
	Reject	$-\ell_r$	1

TABLE I
COLLECTIVE WELFARE FROM CONSENSUS OUTCOME.

Given the above, we seek to maximize the *expected collective welfare* E_t at time slot $t > 0$. E_t depends on the probabilities of accepting a valid block and of rejecting an invalid block under the decision rule f_q,

$$\pi_1(f_q) := P(f_q(\mathbf{X}_t) = 1 \mid B_t = 1)$$

and $\pi_{-1}(f_q) := P(f_q(\mathbf{X}_t) = -1 \mid B_t = -1)$, respectively. Using this notation,

$$E_t(f_q) = (1-\alpha)(1+\ell_r)\pi_1(f_q) + \alpha(1+\ell_a)\pi_{-1}(f_q)$$

To estimate $\pi_1(f_q)$ and $\pi_{-1}(f_q)$, and hence, to maximize E_t, we need to reason about the decision rule f_q and particularly, about the distribution of the decision variables $X_{i,t}$. Fortunately, this can be done by retrieving existing information about validators' past votes that have been stored as messages on the blockchain. This is captured by the notion of validators' *voting profiles*.

Voting profiles: Each validator $i \in I$ is assigned a *score* $p_{i,t} \in [0, 1]$ that corresponds to their *voting profile* at the start of time slot t. The value $p_{i,t}$ can be thought of as validator i's *decision ability* or *probability* that i will vote correctly, i.e.,

$$p_{i,t} := P(X_{i,t} = 1 \mid B_t = 1) = P(X_{i,t} = -1 \mid B_t = -1)$$

for $i \in I, t \geq 0$. In its simplest form, $p_{i,t}$ is equal to the fraction of validators i's correct votes to the number of slots in $\{0, 1, \ldots, t-1\}$ that i was selected in a committee. This expression is only given to provide some intuition and in what follows, we will examine a different scheme that depends on the collective welfare of the consensus outcome, cf. Table I.

Initializing & suspending profiles: We will set a newly entering validator i's voting profile at $p_{i,0} := 0.5$ and will require that $p_{i,t} \in [0.5, 1)$, for any $t \geq g$, where $g \geq 1$ denotes an initial grace period. If $p_{i,t} < 0.5$, for some $t \geq g$, then validator i will be suspended from I. The reasoning behind these choices is detailed in Section IV.

Updating scheme: In general, an *updating scheme* is given by a function $h : [0, 1] \times \{-1, 1\} \to [0, 1]$ which revises validator i's voting profile after time slot t based on i's prior voting profile $p_{i,t}$, the correctness of her voting decision $x_{i,t}$ and the current state $\mathcal{B}_t$, i.e.,

$$p_{i,t+1} = h(p_{i,t}, x_{i,t}, \mathcal{B}_t)$$

for $t > 0$. The current state $\mathcal{B}_t$ may include all relevant information, such as the collective welfare parameters, cf. Table I, the validity of the proposed block and the consensus outcome. If a validator $i \in I$ has not be selected for slot t, then simply $p_{i,t+1} = p_{i,t}$. A concrete updating scheme that fits in this description is developed in Section III.

We now return to the problem of maximizing the collective welfare E_t. For a selected validator committee $N = \{1, 2, \ldots, n\}$ at time slot t, we may condition on the vector of validators' voting profiles $\mathbf{p}_t = (p_{1,t}, p_{2,t}, \ldots, p_{n,t})$ and write

[4] This may include approval votes for a block that a malicious node is trying to create in order to double-spend or perform some other kind of attack. It may also involve votes that get wasted on blocks that will be subsequently reverted or abandonded.

the probability $\pi_1(f_q \mid \mathbf{p}_t)$ of approving a correct block with the decision rule f_q as

$$\pi_1(f_q \mid \mathbf{p}_t) = \sum_{\mathbf{x}_t : f_q(\mathbf{x}_t)=1} \left(\prod_{i:x_{i,t}=1} p_{i,t} \prod_{j:x_{j,t}=-1} (1-p_{j,t}) \right)$$

and similarly for $\pi_0(f_q \mid \mathbf{p}_t)$. This expression for $\pi_1(f_q \mid \mathbf{p}_t)$ is derived as follows: The summation ranges over all action profiles $\mathbf{x}_t$ for which the decision rule f_q approves the proposed block, i.e., $f_q(\mathbf{x}_t) = 1$. The double product inside the parenthesis is precisely the likelihood of each of these profiles given that validators' decisions are independent. Specifically, the first product ranges over all validators $i \in I$ who vote correctly, i.e., $i : x_{i,t} = 1$, and multiplies each one's probability of a correct vote, i.e., $p_{i,t}$, and the second ranges over the remaining validators $j \in I$ who vote incorrectly, i.e., $j : x_{j,t} = -1$, and multiplies each one's probability of voting incorrectly, i.e., $(1 - p_{j,t})$. Given these expressions, the problem of maximizing $E_t(f_q)$ can now be studied as an instant of the *committee-voting* models in [10] and [37], [47].

Example 1 ([47])**.** Consider a committee of 5 validators with voting profiles, i.e., empirical probabilities of voting correctly, $\mathbf{p} = (0.9, 0.9, 0.6, 0.6, 0.6)$, as in [47]. In the unweighted case, or equivalently in the case in which all votes are weighted equally, and under the $2/3$-majority decision rule $f_{2/3}$ that is commonly used in consensus protocols [33], [43], the probability of reaching consensus on the correct block is equal to the probability that at least 4 out of the 5 validators vote correctly. This is

$$0.9^2 0.6^3 + 2 \cdot 0.6^3 (0.9)(0.1) + 3 \cdot 0.9^2 0.6^2 (0.4) \approx 0.56$$

which is lower even than the lowest voting profile of 0.6. A naive improvement would be to only consider the vote of the validator with the highest voting profile, i.e., of either the first or the second validator. This would increase the probability of correct voting to 0.9, however at a toll on decentralisation.

In the naive improvement of the previous example, the vote of the best validator received a *weight* of 1 and the vote of all others a *weight* of 0. This raises the question of whether we can assign non-trivial *weights* (scale factors) to all validators and still improve the probability of a correct decision. The answer is affirmative and hinges on the notions of *weighted voting* and *weighted majority rules*.

Weighted majority rule: For a set of n validators, let $\mathbf{w}_t := (w_{1,t}, w_{2,t}, \ldots, w_{n,t})$ denote a vector of non-negative *weights* or *scaling factors*, with $w_t := \sum_{i=1}^n w_{i,t}$. The *weighted majority rule*, $f_q(w)$, or simply f_q, is defined as

$$f_q(\mathbf{x}_t) := \begin{cases} 1, & \text{if } \sum_{i=1}^n w_{i,t} x_{i,t} \geq (2q-1) w_t, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

for $q \in [0.5, 1]$. If all votes are equally weighted, i.e., if $w_{i,t} = 1$ for all $i = 1, \ldots, n, t > 0$, then (2) reduces to (1).

Using this notation, our goal is to determine the weights $\mathbf{w}_t$ and the weighted majority rule $f_{\hat{q}_t}$ – or equivalently the quota $\hat{q}_t$ – that optimize the collective welfare E_t given the selected committee N_t of validators at time slot t, i.e.,

$$\max_{q, \mathbf{w}_t} \{E_t(f_q, w_t)\} \tag{3}$$

This is the statement of the following Theorem which is due to [10] and in special instances due to [37], [47].

Theorem 2 (Optimal Weighted Voting Scheme, [10])**.** *Consider a committee $N_t = \{1, 2, \ldots, n\}$ of validators with voting profiles $\mathbf{p}_t = (p_{i,t}, p_{2,t}, \ldots, p_{n,t})$ that have been selected to vote on the proposed block in time slot $t > 0$. Then, given α and the collective welfare parameters ℓ_a, ℓ_r in Table I, the decision rule that maximizes the collective welfare, cf.* (3)*, is given by the weighted majority rule $f_{\hat{q}_t}$, with quota*

$$\hat{q}_t := \frac{1}{2}\left[1 - \left(\ln\left(\frac{1-\alpha}{\alpha}\right) + \ln\left(\frac{1+\ell_r}{1+\ell_a}\right)\right) w_t^{-1}\right] \tag{4}$$

and individual weights

$$w_{i,t} := \ln\left(\frac{p_{i,t}}{1-p_{i,t}}\right), \qquad \textit{for } i = 1, 2, \ldots, n. \tag{5}$$

Remark 3. According to Theorem 2, the optimal quota (4) depends on the validators' profiles and hence, it may vary between different time slots $t > 0$. Also, it may vary according to the values of the parameters α, ℓ_r, ℓ_a. For instance, the selection $\alpha = 1/2$ neutralizes the bias due to assumptions on the distribution of valid versus invalid blocks whereas the selection $\ell_r = \ell_a$ neutralizes the bias due to differences in perceived network costs/rewards. In this way, Theorem 2 maximises the collective welfare – or equivalently, the probability of consensus on the correct decision – by an easily adaptable and *dynamic* decision rule.

Yet, in blockchain applications, it may be desirable to enforce certain restrictions, as for example that the required weighted majority will be no less than $2/3$ of the total weights or that each individual weight will be no less than some threshold value. As [47, p.332] explains, even in such cases of additional restrictions and/or perturbed assumptions, selecting weights that are proportional (or equal) to the optimal ones (5) will improve the probability of a correct outcome compared to unweighted decision making.

Example 1 (Continued)**.** Assume for simplicity that $\alpha = 1/2$ and that $\ell_r = \ell_a$. In this case, the optimal rule is simple weighted majority, i.e., $\hat{q} = 1/2$, or by substituting in (2), $f_{\hat{q}}(\mathbf{x}) = 1$ if $\sum_{i=1}^n w_i x_i \geq 0$ (dependence on t is omitted to simplify the notation). Using (5) and normalizing the weights to sum up to 1, we obtain $\mathbf{w} = (0.392, 0.392, 0.072, 0.072, 0.072)$. With these choices, the probability of approving a valid block is approximately 0.927 as shown in Table II. The weighting has resulted in a voting rule which approves a block if the two high-profile (0.9) validators agree on its validity (first column of decision profiles). The votes of the remaining validators come into play only if these two disagree. In this case, it suffices that a majority (2 out of 3) of the remaining validators approve the block (remaining 6 columns of decision profiles). The probabilities of decision

Profiles	Weights	Decision profiles $\mathbf{x}$, with $f_{\hat{q}}(\mathbf{x}) = 1$.						
0.9	0.392	1	1	1	1	-1	-1	-1
0.9	0.392	1	-1	-1	-1	1	1	1
0.6	0.072	-1,1	1	1	-1,1	1	1	-1,1
0.6	0.072	-1,1	1	-1,1	1	1	-1,1	1
0.6	0.072	-1,1	-1,1	1	1	-1,1	1	1
$\sum_{i=1}^{5} w_i x_i$		> 0	> 0	> 0	> 0	> 0	> 0	> 0

TABLE II

profiles $\mathbf{x}_t$ with $f_{\hat{q}_t}(\mathbf{x}_t) = 1$ sum up to approximately 0.927 as claimed

$$0.9^2 + \binom{2}{1} 0.9 \cdot 0.1 \left(\binom{3}{3} 0.6^3 + \binom{3}{2} 0.6^2 \cdot 0.4 \right) \approx 0.927$$

The weights $\mathbf{v}_t = (1/3, 1/3, 1/9, 1/9, 1/9)$ yield the same result and are in that sense, equivalent to the optimal ones. In fact, there may be several other optimal choices. As mentioned above, if we impose additional restrictions such as a de facto $2/3$-weighted majority, the weights given by (5) may not be optimal anymore. However, as [47] remarks, they will still yield an improved probability compared to the unscaled case. In this example, a similar calculation as in Table II shows that the probability of reaching the $2/3$-majority is 0.81 with the $\mathbf{w}_t$ weights and approximately 0.85 with the $\mathbf{v}_t$ weights.

A. Multiplicative Weights Update Algorithm

We now turn to one of the main challenges of implementing the voting profile scheme in the dynamic blockchain setting, which is the *update* of voting profiles after every time slot. The updating scheme may considerably vary depending on the desired result: [53] propose a reputation system in which reputation increases according to a *sigmoid function* when nodes vote correctly and decreases sharply (to 0) after a single violation. While this approach adheres to intuition and comes with certain merits, practical applications may call for more flexibility in the updates. To develop a parameterizable scheme, we utilize the approach of [2] who generalize the standard *multiplicative weights update* (MWU) algorithm to a non-binary setting in which experts' scores are revised according to the impact of their decision on the social welfare.

Algorithm 1 Weighted Voting in Committees

```
1:  procedure PoS SELECTION($(v_i)_{i \in I}, \mathcal{B}$)
2:      return $N \leftarrow \{v_1, v_2, \ldots, v_n\}$
3:      return $B$ (proposed block)
4:  procedure COLLECT VOTES($N, B$)
5:      for $i \leftarrow 1, n$ do
6:          if $x_i == B$ then            (▷) $x_i = i$'s vote message
7:              vote($i$) $\leftarrow 1$
8:          else
9:              vote($i$) $\leftarrow -1$
10: procedure WEIGHTED VOTING($(p_i)_{i \in N}, (\text{vote}(i))_{i \in N}$)
11:     for $i \leftarrow 1, n$ do
12:         $w_i \leftarrow \ln(p_i) - \ln(1 - p_i)$
13:         sum $\leftarrow$ sum $+ w_i \cdot$vote($i$)
14:     $q = $ OPTIMAL QUOTA($(w_i)_{i \in N}, \alpha, \ell_r, \ell_a$)
15:     if sum $\geq 2q - 1$ & $B == 1$ then
16:         append.block($B$)
17:         $\mathcal{B} \leftarrow \mathcal{B} \cup B$
18: procedure PoS REWARDS($(v_i)_{i \in N}, (\text{vote}(i))_{i \in N}, \mathcal{B}$)
```

Using Table I, the corresponding MWU algorithm for the present application is given in Table III. Here, $\delta > 0$ is a (small) number subject to the exact protocol parametrisation. Apart from the efficiency properties of the MWU algorithm that are well known, see [2], [8], [39] and references therein, this scheme can leverage the prevailing network conditions and adjust the updates accordingly. This can be realized by replacing δ and/or ℓ_r, ℓ_a with *sequences of updating parameters*, $(\delta_t, \ell_{r,t}, \ell_{a,t})_{t>0}$.

		Proposed Block B_t	
		Valid (1)	Invalid (-1)
Committee	Approve	$p_{i,t}(1+\delta)$	$p_{i,t}(1-\delta)^{\ell_a}$
	Reject	$p_{i,t}(1-\delta)^{\ell_r}$	$p_{i,t}(1+\delta)$

TABLE III
MULTIPLICATIVE WEIGHTS UPDATES.

Algorithm 1 summarises the weighted voting procedure. Validators are selected and rewarded according to the underlying PoS protocol (lines 1 and 18). The weighted voting procedure[5] is applied *once* the committee has been formed (lines 10 to 17) without modifying the rest of the protocol. In this way, it contributes towards a more efficient and fair consensus mechanism while remaining decoupled from the PoS selection mechanism. This results in a two-layered scheme that on the one hand improves the efficiency of the consensus mechanism and restores the fairness property of the PoS protocol and on the other hand can be implemented and reverted with minimal cost to the users. The proposed scheme for updating validators' voting profiles is given for completeness in Algorithm 2. The $\max$ and $\min$ expressions (lines 8,10,12) ensure that the profiles remain in $[0, 5, 1)$.

Algorithm 2 Validators' Voting Profiles

```
1:  procedure MWUPDATE($I, \delta, \ell_r, \ell_a$)
2:      initialize: $p_i \leftarrow 0.5, t \leftarrow 1$
3:      while $t > 0$ do
4:          if $p_i < 0.5$ & $t \geq g$ then          (▷) grace period $g$
5:              $i \leftarrow \emptyset$                 (▷) suspend validator $i$
6:          else if $v_i \in N$ then
7:              if $x_{i,t} == B_t$ then
8:                  $p_i \leftarrow \min\{1 - \epsilon, (1+\delta)\}$     (▷) $\epsilon = 10^{-5}$
9:              else if $B_t == 1$ then
10:                 $p_i \leftarrow \max\{0.5 - \epsilon, (1-\delta)^{\ell_r}\}$
11:             else
12:                 $p_i \leftarrow \max\{0.5 - \epsilon, (1-\delta)^{\ell_a}\}$
13:         $t \leftarrow t + 1$
```

[5] The function that determines the optimal quota is given in a general form (line 14) to account for implementations with a rule different from the one proposed in (4), as e.g., a constant $2/3$-majority rule.

B. Numerical results & Simulations

To visualize the proposed scheme, we study some instantiations of the weighted voting and MWU algorithms. Before going into the details of each case, the following hold in general

- **Adversarial model:** an adversary blocks $v\%$ of the votes, either by abstaining (accidentally or intentionally) or by censoring other validators. To demonstrate the efficiency of the model in extreme conditions, we show simulations for $v = 40\%, 50\%$ and 60%, but the results are similar for any value of $v > 0$.
- **Parameter choice:** α, the prior probability that a proposed block is invalid, is set to $1/2$. This choice neutralizes the bias due to assumptions on the distribution of valid-invalid blocks in (4) and corresponds to the most general model. To capture that costs from approval of invalid blocks are higher than costs from rejection of valid blocks, we select $\ell_a = 12 \gg 10^{-2} = \ell_r$. The results are robust in a wide range of these parameters. Yet, very low values of ℓ_a may lead to unwanted behavior: validators who are commonly off-line may still improve their profiles despite not voting on valid blocks. Finally, the updating parameter δ has been choosen according to the related literature [2], [8]. Different values of δ affect the rate of convergence of the profiles.
- **Simulation environment:** The numerical results have been established in MATLAB R2018b. The simulations validate the algorithms for resuming consensus, cf. Figures 1 and 2 and for updating a validator's profile, cf. Figures 3 and 4. Further issues related to scalability and computational complexity on a proper – or simulated – blockchain are discussed in Section IV.

Figure 1 illustrates a scenario with an adversary blocking 40% of the votes. At the start of the attack, all n validators[6] have a voting profile $p = 0.9$. We examine two choices of the updating parameter $\delta = 10^{-3}$ (red) and $\delta = 2 \times 10^{-3}$ (blue). In both cases, $\alpha = 1/2$ and $\ell_r = 10^{-2}, \ell_a = 12$. The depicted curves indicate that the weighted approval votes (vertical axis) rise above the $2/3$ majority threshold[7] for both cases. The pace is different and depends on the selection of δ. The knicks in both lines correspond to the point in which the scores of voting validators numerically reach 1. After this point, the majority of the voting validators increases at a very slow pace which is a desirable property that allows for a more smooth recovery in case that the abstaining 40% resume voting.

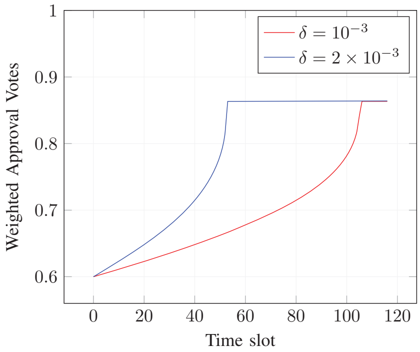

Fig. 1. Time slots to resuming consensus on valid blocks with 40% non-voting nodes under mild (red line) and aggressive (blue line) updating parameter δ. In both cases, $\ell_r = 10^{-2}$.

As a comparison, Figure 2 illustrates the process of resuming approval of blocks in the same scenario but with an adversary controlling 50% of the stake (left panel) and 60% of the stake (right panel). The results indicate a very similar recovery pattern, cf. Figure 1, independently of the initial stake of the non-voting validators. The evolution of a validator's voting profile is illustrated in Figure 3. In the depicted scenario, the validator's initial profile is 0.9. She votes correctly 80% of the time, but drops 10% of the time off-line, and votes on invalid blocks another 10% of the time. The exact formula for updating her profile is

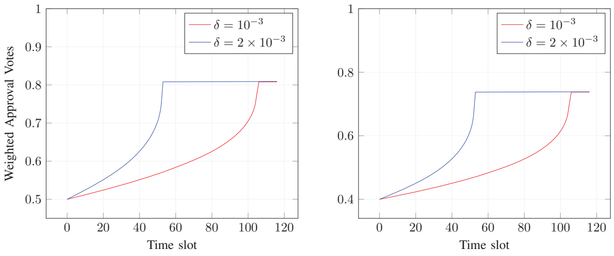

Fig. 2. Time slots to resuming consensus on valid blocks with 50% (left panel) and 60% (right panel) non-voting nodes under mild (red line) and aggressive (blue line) updating parameter δ. In both cases, $\ell_r = 10^{-2}$.

$$p_{i,t+1} = \min\{1 - 10^{-5}, \max\{0.5, \tilde{p}_{i,t+1}\}\}$$

where $\tilde{p}_{i,t+1}$ is the value calculated by Table III. This formula ensures that $p_{i,t}$ remains in $[0.5, 1)$. Again, we examine two cases for different values of the update parameter, $\delta = 2\times10^{-2}$ (lefta panel) and $\delta = 10^{-2}$ (right panel). While the patterns differ, in both depicted panels the voting profile falls due to the regular incorrect votes. We remark, that lower values of ℓ_a would result in upwards sloping curves (not depicted here)

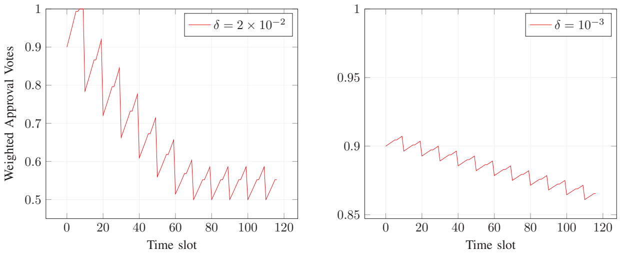

Fig. 3. Evolution of a validator's voting profile who periodically drops offline and periodically votes on an invalid block for different values of parameter δ. In both panels, the validator's initial score is 0.9, $\ell_a = 12$ and $\ell_r = 10^{-2}$.

[6] The exact number does not change the results. For simplicity, we used $n = 100$.

[7] While the optimal quota, cf. (4), remains slightly above $1/2$, we consider the $2/3$-majority rule which is currently implemented in PoS protocols.

implying that a validator could regularly vote incorrectly and still improve her voting profile. Similarly, higher values of δ would allow validators to quickly recover their profiles after pitfalls which is an undesirable property. The depicted patterns in the evolution of the voting profile are robust in the choice of the initial score and the value of ℓ_r.

Finally, Figure 4 extends the above scenarios to a period in which the validator resumes proper voting. Specifically, we assume that the validator votes correctly on every block except of occasional time slots – less than 10% of the time – in which she goes offline. Again, the two panels correspond to different values of δ. In both cases, the pattern is linear (the knicks correspond to the occasional drops) with a slope that can be adjusted by the choice of δ. In sum, the simulations support the versatility of the proposed scheme and leave the exact parametrisation (static or dynamic) subject to each protocol's implementation and scope.

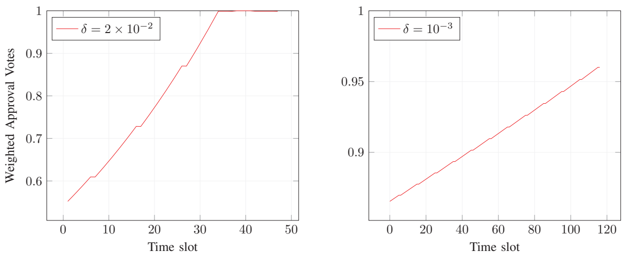

Fig. 4. Recovery of the validator's voting profile after resuming proper voting (with only occasional offline-drops) in the scenarios of Figure 3. Recovery exhibits linear pattern for both choices of δ. It is fast for $\delta = 2 \times 10^{-2}$ (aggresive adjustment) and slow for $\delta = 10^{-3}$ (mild adjustments).

IV. Design & Limitations

The introduction of validators' voting profiles and the improvement in the consensus mechanism come with a trade-off in terms of security. Since the system becomes reliable on information that can be retrieved from the blockchain – validators' votes are stored as messages [18] – this raises new risks on adversarial manipulation of this information. In the current section, we try to address these risks alongside implementation issues and limitations.

Staking nodes & anonymity: A profiling system reduces or eliminates the anonymity of staking nodes which is at odds with the design philosophy of permissionless blockchains. However, with blockchain governance yet to be determined, introducing less anonymity may be a desired feature. Recent results support relative low desired numbers of staking nodes [18] or stake pools [16]. In such schemes, reputation will implicitly or explicitly influence protocol execution. Moreover, stake pools can retain anonymity at a user level, i.e., while the pool becomes identifiable by its voting profile, the users remain anonymous. In any case, the introduction of voting profiles and weighted voting seems particularly relevant to permissioned blockchains or delegated PoS mechanisms.

Defense against known attacks: To defend against consensus level attacks, current PoS proposals leverage the fact that non-voting nodes can be identified and penalized [5], [17]. Yet, these actions are ineffectual against adversaries who can censor other validators or replenish their own stake and withstand the penalties to retain more than the required consensus-quota of the total stake [26], [42]. Although pessimistic, the scenario in which adversaries sustain an attack despite suffering losses on their own stake gains credence in the presence of potential *out-of-protocol* profits. Further, consensus mechanisms that rely on PoS selection are vulnerable to *flash* or *blindsiding* attacks conducted by entering nodes [13], [20] or to accidental faults such as network latency, bad connectivity or simple negligence. Weighted voting provides lines of defense (in an obvious way) against these kinds of attacks or faults while retaining the benefits of the underlying PoS design. In addition, the preserved reliance on PoS for the selection of validators in committees and the allocation of rewards, protects against adversarial nodes that maintain small stake but high voting profile or vice versa.

Valid-invalid blocks: A likely contentious assumption of the present model is that proposed blocks can be indentified as valid or invalid[8]. In practice, different nodes may have different views of the blockchain and hence perceive proposed block differently. Yet, on closer inspection, this assumption can still be supported in the current framework: if a node is honest but has a (completely) different view of the canonical chain due to (say) poor connection to the network, then her votes do not contribute to extending the canonical chain and indeed can be regarded as incorrect. For instance, in Ethereum, which is the base case for this paper, *valid–invalid* votes are well-defined and identified [44], [45].

Updating scheme: Dealing with the issue of valid-invalid blocks becomes more relevant in the design of the updating scheme. Clearly, faults that can be identified as deliberate should be treated differently than accidental ones. For instance, a validator who has honestly participated in the protocol for a long period of time and drops offline for a short period of time, should be able to quickly recover her previous voting profile. This motivates updating profiles by two variables $s_{i,t}$ and $l_{i,t}$ representing *short-term* and *long-term* voting respectively. In general, the advantages of the here employed generalized MWU algorithm – e.g., convergence rates and tight bounds on its overall performance [2] – can be further exploited alongside stability properties of opinion dynamics in networks [35] to yield more robust results also in the present context.

Entry & threshold voting profiles: The levels at which voting profiles are initialized and suspended are crucial, since they can incentivize or prevent *Sybil attacks*. The exact parametrization can be case-depend, justified by simulations or incorporate prior information for each entering validator, e.g., reputable financial institution versus unkown private staker.

[8] This is a chicken-egg problem: if we can identify the "canonical" blocks, then we can improve consensus with a scheme as the one proposed here. But in blockchains, the "canonical" blocks are precisely the ones for which consensus was reached.

The present choice to initialize voting profiles at 0.5 and to require that they remain in $[0.5, 1)$ for all $t \geq g > 0$, where g denotes a potential initial grace period, is supported by both intuitive and theoretical arguments. From a mathematical perspective, the initial choice of 0.5 represents an uninformative prior on an entering's validators voting profile. Similarly, the reason for the upper bound is purely numerical, namely to avoid the instability in $\ln(p_{i,t}/(1-p_{i,t}))$ if $p_{i,t} = 1$. In contrast, the suspension of validators with voting profile – or probability of making a correct decision – lower than 0.5 is more fundamental. [10] provide a detailed probabilistic argument to explain that such voters are harmful to consensus and their votes should not be considered. Moreover, in the specific context of distributed networks, allowing nodes to participate with scores lower than their initial one triggers *Sybil attacks*, since it motivates switching to new or creating multiple accounts. Finally, suspending validators in terms of their voting behavior relaxes the need for frequent economic penalties [18]. This makes the PoS ecosystem more secure and appealing to investors who would otherwise be concerned of suffering losses due to accidental misbehavior, e.g., dropping off-line or being censored.

Future implementation: Currently, the proposed scheme seems more attractive for systems with low numbers of staking nodes: these may be permissioned and delegated PoS blockchains or blockchains with fixed number of stake pools. More closely related to this spirit are the proposals of [1], [17], [29], [34]. In these settings, the computational complexity of implementing a profiling system is not an issue. However, this is also expected to remain true in the general case of permissionless blockchains, since extra data storage is limited to one additional value per validator and computations to update profiles are linear in the size of the committees. Light on this issues will be shed by future implementations on simulated blockchains and if successful, on properly designated testnets of these blockchains. In the core of these studies will be the understanding of issues related to network conditions (latency, scalability), computational complexity to store and retrieve profiles and the testing of different updating schemes in terms of their convergence rates and efficiency bounds.

V. Summary & Conclusions

Existing PoS protocols select staking nodes proportionally to their stake to form block-creating committees. Yet, they do not guarantee that selected committees will create blocks, since consensus may fail due to accidental or adversarial behavior. Thus, the perceived fairness in the distribution of rewards in proportion to the stake of participating nodes is actually violated. Motivated by this observation, we studied *weighted voting* as a way to improve the consensus mechanism. We introduced validators' voting profiles – that quantify the probability that a validator will cast a correct vote based on her so far contribution to the protocol – and defined the proper mathematical framework to apply the results of [10] on optimal decision rules in committee voting. Using the approach of [2], we designed a multiplicative weights algorithm to update individual validator's profiles according to their voting behavior, the consensus outcome and collective blockchain welfare. The result is a two-layered scheme in which selection of nodes and allocation of rewards are performed by the underlying PoS mechanism whereas blocks are decided by a weighted majority voting rule. This scheme improves consensus *within* selected committees by scaling votes according to validators' profiles without interfering with the PoS execution. Hence, it can be tested, implemented and reverted with minimal cost to existing users. On the negative side, the introduction of a profiling scheme in a distributed network raises new risks associated with manipulation of the relevant information. We discussed such risks along with actions that should be considered in the design of future PoS protocols.

References

[1] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, pages 30:1–30:15, New York, NY, USA, 2018. ACM.

[2] S. Arora, E. Hazan, and S. Kale. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8(6):121–164, 2012.

[3] H. Aziz and M. Paterson. False Name Manipulations in Weighted Voting Games: Splitting, Merging and Annexation. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 409–416, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[4] H. Aziz, M. Paterson, and D. Leech. Efficient Algorithm for Designing Weighted Voting Games. In *2007 IEEE International Multitopic Conference*, pages 1–6, Dec 2007.

[5] S. Azouvi, P. McCorry, and S. Meiklejohn. Betting on Blockchain Consensus with Fantomette. *ArXiv e-prints*, 2018.

[6] Y. Azrieli and S. Kim. Pareto Efficiency and Weighted Majority Rules. *International Economic Review*, 55(4):1067–1088, 2014.

[7] Y. Bachrach, J. Rosenschein, and E. Porat. Power and Stability in Connectivity Games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '08, pages 999–1006, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[8] J. P. Bailey and G. Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 321–338, New York, NY, USA, 2018. ACM.

[9] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis. Consensus in the age of blockchains. *arXiv preprint arXiv:1711.03936*, 2017.

[10] R. C. Ben-Yashar and S. I. Nitzan. The Optimal Decision Rule for Fixed-Size Committees in Dichotomous Choice Situations: The General Result. *International Economic Review*, 38(1):175–186, 1997.

[11] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies without proof of work. In J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, editors, *Financial Cryptography and Data Security*, pages 142–157, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[12] I. Bentov, R. Pass, and E. Shi. Snow white: Provably secure proofs of stake. Cryptology ePrint Archive, Report 2016/919, 2016. https://eprint.iacr.org/2016/919.

[13] J. Bonneau, E. Felten, S. Goldfeder, J. Kroll, and A. Narayanan. Why buy when you can rent ? Bribery attacks on Bitcoin consensus, 2015.

[14] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121, 2015.

[15] J. Brown-Cohen, A. Narayanan, C.-A. Psomas, and S. M. Weinberg. Formal Barriers to Longest-Chain Proof-of-Stake Protocols. *ArXiv e-prints*, 2018.

[16] L. Brünjes, A. Kiayias, E. Koutsoupias, and A.-P. Stouka. Reward Sharing Schemes for Stake Pools. *ArXiv e-prints*, page arXiv:1807.11218, July 2018.

[17] V. Buterin. Ethereum 2.0 spec – Casper and sharding, 2018. Available [online]. [Accessed: 30-10-2018].

[18] V. Buterin and V. Griffith. Casper the Friendly Finality Gadget. *ArXiv e-prints*, 2017.

[19] C. Cachin and M. Vukolić. Blockchain Consensus Protocols in the Wild. *CoRR*, abs/1707.01873, 2017.

[20] V. Chia, P. Hartel, Q. Hum, S. Ma, G. Piliouras, D. Reijsbergen, M. van Staalduinen, and P. Szalachowski. Rethinking blockchain security: Position paper. In *Proceedings of IEEE Blockchain 2018*, 2018.

[21] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385, 2018.

[22] I. Eyal and E. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In N. Christin and R. Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 436–454, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[23] J. Garay, A. Kiayias, and N. Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310. Springer Berlin Heidelberg, 2015.

[24] J. Garay, A. Kiayias, and N. Leonardos. The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In *CRYPTO*, pages 291–323. Springer, 2017.

[25] J. A. Garay, A. Kiayias, N. Leonardos, and G. Panagiotakos. Bootstrapping the Blockchain, with Applications to Consensus and Fast PKI Setup. Cryptology ePrint Archive, Report 2016/991, 2016. https://eprint.iacr.org/2016/991.

[26] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 51–68, New York, NY, USA, 2017. ACM.

[27] I. Grigg. EOS – An Introduction, 2017. Available [online]. [Accessed: 17-12-2018].

[28] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 365–382, 2016.

[29] A. Kiayias, A. Russell, B. David, and R. Oliynykov. "lboros: A provably secure proof-of-stake blockchain protocol". In J. Katz and H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 357–388. Springer International Publishing, 2017.

[30] P. Koshy, D. Koshy, and P. McDaniel. An Analysis of Anonymity in Bitcoin Using P2P Network Traffic. In N. Christin and R. Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 469–485. Springer Berlin Heidelberg, 2014.

[31] J. Kwon. Tendermint: Consensus without mining, 2014. Available [online]. [Accessed: 17-12-2018].

[32] J. Kwon. Tendermint Core, 2014. Available [online]. [Accessed: 17-12-2018].

[33] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[34] Lisk. Lisk's Consensus Algorithm, 2018. Available [online]. [Accessed: 17-12-2018].

[35] T. Mai, I. Panageas, and V. V. Vazirani. Opinion dynamics in networks: Convergence, stability and lack of explosion. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 140:1–140:14, 2017.

[36] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. Available: [online]. [Accessed: 14-11-2018].

[37] S. Nitzan and J. Paroush. Optimal Decision Rules in Uncertain Dichotomous Choice Situations. *International Economic Review*, 23(2):289–297, 1982.

[38] S. Nitzan and J. Paroush. Are Qualified Majority Rules Special? *Public Choice*, 42(3):257–272, 1984.

[39] G. Palaiopanos, I. Panageas, and G. Piliouras. Multiplicative weights update with constant step-size in congestion games: Convergence, limit cycles and chaos. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5872–5882. Curran Associates, Inc., 2017.

[40] R. Pass, L. Seeman, and A. Shelat. "analysis of the blockchain protocol in asynchronous networks". In J.-S. Coron and J.-B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673. Springer International Publishing, 2017.

[41] R. Pass and E. Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 315–324, New York, NY, USA, 2017. ACM.

[42] R. Pass and E. Shi. "thunderella: Blockchains with optimistic instant confirmation". In J.-B. Nielsen and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 3–33, 2018.

[43] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, Apr. 1980.

[44] D. Ryan. Validator Implementation Guide, 2018. Available [online]. [Accessed: 17-12-2018].

[45] D. Ryan and C.-C. Liang. Ethereum improvement proposal 1011. Available: [online]. [Accessed: 3-9-2018].

[46] A. Sapirshtein, Y. Sompolinsky, and A. Zohar. Optimal selfish mining strategies in bitcoin. In J. Grossklags and B. Preneel, editors, *Financial Cryptography and Data Security*, pages 515–532, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[47] L. Shapley and B. Grofman. Optimizing group judgmental accuracy in the presence of interdependencies. *Public Choice*, 43(3):329–343, 1984.

[48] C. Stathakopoulou. On Scalability and Performance of Permissioned Blockchain Systems. 2018.

[49] N. Stifter, A. Judmayer, P. Schindler, A. Zamyatin, and E. R. Weippl. Agreement with Satoshi – On the Formalization of Nakamoto Consensus. *IACR Cryptology ePrint Archive*, 2018:400, 2018.

[50] G. Vizier and V. Gramoli. ComChain: Bridging the Gap Between Public and Consortium Blockchains. In *Proceedings of the IEEE International Conference on Blockchain (Blockchain'18)*, Jul 2018.

[51] Y. Bachrach and E. Elkind. Divide and conquer: false-name manipulations in weighted voting games. In *AAMAS*, 2008.

[52] P. Young. Optimal Voting Rules. *Journal of Economic Perspectives*, 9(1):51–64, March 1995.

[53] J. Yu, D. Kozhaya, J. Decouchant, and P. Verissimo. RepuCoin: Your Reputation is Your Power. Cryptology ePrint Archive, Report 2018/239, 2018. https://eprint.iacr.org/2018/239.

[54] M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind. Manipulating the quota in weighted voting games. *Artificial Intelligence*, 180–181:1–19, 2012.