An Improved Model on the Vague Sets-Based DPoS's Voting Phase in Blockchain

Lin You, *Member, IEEE*, Zhuobiao Wang, Gengran Hu, Chengtang Cao, Liang Li

***Abstract*—As a common consensus mechanism used in Blockchain, DPoS uses voting to select the committee members to generate blocks. In order to elect the committee members more fairly, vague sets are introduced into the voting phase of DPoS. In the original vague sets-based model proposed in 2020, the voting nodes can vote yes, no or abstain. In this paper, we improve this model by introducing a new mapping from the vague set to the fuzzy set and assigning weights to each node. In our new model, we consider that the influence factor of abstention vote increases with the increase of effective votes. Moreover, several nice properties of the improved model are proved which make our model conform to the real situation. Through our experiments, the fuzzy value's distribution of our scheme is more uniform and reduce the probability of the same fuzzy value. If the ratio of non-abstention votes is higher, the influence of abstention vote will be greater, and the chance of being selected will increase. This makes the voting phase of DPoS fairer.**

***Index Terms*—Blockchain, DPoS, Voting, Weight Value, Fuzzy Set**

I. INTRODUCTION

Since Satoshi Nakamoto proposed Bitcoin [1], its core technology, blockchain, has been highly valued. The emergence of blockchain technology solves the security issues of untrusted third parties and data tampering. In blockchain, each block contains transaction data generated by the network over a period of time. By consensus mechanism, all the network nodes can verify the validity of a new block and participate in the mining to generate the next block [2]. There are currently three types of blockchains: public chains, private chains, and alliance chains. The consensus mechanisms used in the alliance chain [3] are BFT and PBFT. Bitcoin and Ethereum [4] is a typical representative of public chain. Common consensus mechanisms [5] in public chain are PoW [6] and PoS [7]. DPoS is an improvement on PoS, which uses proxy voting and delegates to vote on nodes. DPoS has high efficiency and can be applied in public and alliance chains.

At present, there are not many researches on DPoS. The main improvement is to improve its voting process. In the literature [8], it was proposed to use the reputation mechanism to dynamically the select nodes. The agent can vote according to the node's reputation score, and set a certain reward and punishment mechanism to make the voting process of DPoS more fair. In the literature [9], an incentive mechanism of clustering algorithm is proposed, which distributes rewards to different nodes according to the classification results, and improves the enthusiasm of the nodes through this scheme. In the literature [10], it proposed an algorithm to identify attacks, and use game theory to design incentive mechanism. Most of these improvements have designed a reward and punishment mechanism to improve the efficiency of the voting algorithm. However, in Xu's work [11], the concept of fuzzy set was used to make the voting more fair. Each node can vote for, against and abstain. Here in this paper, we will improve their work.

Consensus mechanism is the key to the decentralization and security of blockchain technology. There are many existing consensus mechanisms [12], such as PBFT [13], [14], PoW, PoS [15], and DPoS [16]. This paper is focused on the voting phase of DPoS and is organized as follows: In Section II, several typical consensus mechanisms are introduced, among which DPoS is an important one. In Section III, the concepts of fuzzy set and vague set are briefly explained. The existing vague sets-based DPoS voting model is introduced in Section IV. Then an improved model and the reason for the improvement are presented in Section V. And this improved model is analyzed and its several nice properties are proved In Section VI. The experimental simulation of our improved model has been done and analyzed in the next section.

In this paper, we have improved the conversion formula from vague set to fuzzy set in the DPoS voting process proposed by Xu [11]. The formula in [11] transforming vague set $[t_V(u), 1-f_V(u)]$ into a fuzzy value $\mu_{F(V)}(u)$ is:

$$\mu_{F(V)}(u) = t_V(u) + \frac{1}{2}[1 + \frac{t_V(u) - f_V(u)}{t_V(u) + f_V(u) + 2\lambda}] \times (1 - t_V(u) - f_V(u)).$$

However, this conversion formula cannot easily prove that the fuzzy value increases as $t_V(u)$ increases or $f_V(u)$ decreases. In Xu's model, the multiplication ratio $\alpha = 1/(t_V(u) + f_V(u) + 2\lambda)$. We notice that this α decreases on $t_V(u) + f_V(u)$. Considering that in reality, α should increase as the $t_V(u) + f_V(u)$ value increases, which means the influence of abstention votes should increase. In our scenario, we set $\alpha = t_V(u) + f_V(u)$. Our improved conversion formula can obviously prove that the fuzzy value increases as $t_V(u)$ increases or $f_V(u)$ decreases, and α obviously increases on the $t_V(u) + f_V(u)$. What's more, our model introduces the concept of weight to avoid the same fuzzy-set value as much as possible, which makes the voting process fairer. In this way, the system can select committee members more fairly based

This work is supported in part by the National Natural Science Foundation of China (No.61602143, No.61772166) and in part by the Key Program of the Natural Science Foundation of Zhejiang Province of China under Grant LZ17F020002. (*Corresponding Author: Gengran Hu*)

Lin You, Zhuobiao Wang, Gengran Hu, Chengtang Cao and Liang Li are with the School of Cyberspace, Hangzhou Dianzi University, 310018, China (e-mail: mryoulin@gmail.com, wangzhuobiao@hdu.edu.cn, grhu@hdu.edu.cn, caochengtang@163.com, liangli@hdu.edu.cn).

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3279571

on the fuzzy value. That is to say, We make the following contributions.

1) The conversion formula from the vague set to the fuzzy set has been improved to make the formula more reasonable and the node election process fairer.
2) The weight is introduced to make the voting process more in line with the working process of DPoS.
3) By using the improved fuzzy set transformation formula after adding weight, we can get that the distribution of fuzzy set values of our new model is more scattered.

All symbols in this article are listed in TABLE I.

TABLE I
NOTATIONS

Notations	Descriptions
$t_V(u)$	Proportion of yes votes
$f_V(u)$	Proportion of no votes
$\mu_{F(V)}(u)$	fuzzy value
α	Influence factor of abstention vote
t	Simplified $t_V(u)$
f	Simplified $f_V(u)$
μ	Simplified $\mu_{F(V)}(u)$

II. Related Work

A. Blockchain's Consensus Mechanism and DPoS

The consensus mechanism [18] of blockchain is a technology that allows the irrelevant nodes to reach a consensus on the transactions. Due to the existence of the consensus mechanism, each transaction does not need to be processed by an untrusted third party. Instead, all the network nodes use the consensus mechanism to reach the agreement on each transaction, and it improves the security of the transaction and avoids transaction tampering. The main phases of the consensus mechanism are as follows [19]:

- *1) Election of block producers:* Select the node which is responsible for generating blocks. A node needs to complete certain tasks to become block producers.
- *2) Block generation:* Pack the transaction data generated on the network within a period of time into the current block. The block header contains the hash value of the previous block, time-stamp and other contents.
- *3) Node verification and blockchain update:* Once the current block is generated, it will be broadcast across the entire network. The nodes that receive the information verify the correctness of the block and update the blockchain.

The most commonly used consensus mechanisms are the Proof-of-Work mechanism (PoW) and the Proof-of-Stake mechanism (PoS). The Proof-of-Work is the earliest consensus mechanism, which is used in the Bitcoin. In order to solve the problem of power consumption in PoW, PoS has been proposed. PoS selects nodes based on the size of the tokens they hold. PoS has to some extent improved consensus efficiency. Subsequently, DPoS was proposed by combining PoS and proxy voting systems. Compared to PoS, DPoS is more efficient. DPoS achieves consensus in a more equitable manner.

As a relatively new consensus mechanism, Delegated Proof of Stake (DPoS) was proposed in 2014. In DPoS, the token holders vote to elect the nodes that generate blocks. The size of the equity held by the holders determine their votes. The node owing the greater equity has more votes. After the voting phase, the fixed number of the nodes with the most votes will become the committee members to generate blocks. Each node will generate a block in turn. If the node does not generate a block during a specific period of time, it will be delisted. And the network will select a new node to replace it. DPoS also has some drawbacks, such as inability to prevent malicious nodes from entering the committee, low enthusiasm of voting nodes, or only voting for a few nodes. There is a possibility of collusion voting among nodes.

In order to solve the problems of DPoS, many improvement schemes are proposed. In the literature [11], Xu proposed using the Vague Set voting method to reduce the possibility of node collusion voting and to some extent improve the enthusiasm of node voting. In the literature [20], a credit scoring mechanism was proposed to rate the behavior of nodes and vote based on the credit score. At the same time, an incentive mechanism is introduced to reward nodes who actively vote and punish malicious nodes. Different incentive mechanisms have also been proposed in the literature [21] to enhance DPoS voting enthusiasm. In this paper, we improve the vague sets-based voting phase based on Xu's scheme to make the election more reasonable. Such an improvement ensures the fairness of the election and keeps the members entering the committee more reliable, and so it can reduce the possibility of the selecting malicious nodes.

B. Fuzzy Set and Vague Set

In the traditional set theory [22], given a set U, for any element a, there are only two cases related to U and a: a belongs to U or does not belong to U, which refers to only two distinct values: 0 and 1. Fuzzy set theory introduces the concept of the membership degree. The membership degree refers to the certainty of an element in this set. For a set U, each element in the set has a corresponding membership value $\mu_F : U \to [0,1]$ and this value is unique. For example, in a set of some tall persons, μ_F maps a person of 2-meters height to 0.7, and maps a person of 1.7-meters height to 0.4. With the emergence of fuzzy set theory, the possibilities are expressed in numbers. Subsequently, the vague set was proposed. Compared with fuzzy set, vague set proposed the interval more accurately. Moreover, the vague set combines certainty and uncertainty, the fuzzy set has only a single certainty or uncertainty.

Vague set can also describe the membership degree in the set [23]. An interval instead of a single value is used to represent a vague set. Given a set U, $t_V(u)$ represents the membership degree that V truly belongs to U, and $t_V(u)$ is the lower bound of the supporting membership derived from the supporting evidence. At the same time, the membership degree $f_V(u)$

represents the membership degree that V falsely belongs to U. As a result, $f_V(u)$ is the lower bound of the opposing membership derived from the opposing evidence. Therefore, the membership interval of V is $[t_V(u), 1 - f_V(u)]$.

III. The Vague Set-based DPoS's Voting Phase

The DPoS voting model [11] recently proposed is based on the conversion from the vague set to the fuzzy set [24]. In this model, each node can vote favor, oppose or abstention. And the ratio of the favor votes to the total votes is the true membership degree, namely $t_V(u)$. The percentage of oppose notes is the false membership degree $f_V(u)$. Because of the existence of abstention votes, it is obvious that $t_V(u) + f_V(u) \leq 1$, matching the requirements of the vague set. Through the transformation from the vague set to the fuzzy set, the interval value of vague set is transformed into the unique value of the fuzzy set. As a result, vague set can be transformed into a fuzzy set. In [21], Y. Liu et al. proposed several relatively simple transformation methods and discussed their advantages and shortcomings. In [13], G. Xu et al. formally presented a more complicated conversion from $[t_V(u), 1 - f_V(u)]$ to $\mu_{F(V)}(u)$:

$$\mu_{F(V)}(u) = t_V(u) + \frac{1}{2}[1 + \frac{t_V(u) - f_V(u)}{t_V(u) + f_V(u) + 2\lambda}] \times (1 - t_V(u) - f_V(u)), \tag{1}$$

where $\lambda = 1$ by default. After applying this conversion, the $\mu_{F(V)}(u)$ for each node can be computed. Then all the nodes can be sorted by the $\mu_{F(V)}(u)$, which means the node with higher fuzzy value is in the front rank. Suppose that there are m nodes to be selected among n nodes, according to $\mu_{F(V)}(N_{i_1}) > \mu_{F(V)}(N_{i_2}) > \cdots > \mu_{F(V)}(N_{i_n})$, the first m nones $N_{i_1}, N_{i_2}, \cdots, N_{i_m}$ are selected. If $\mu_{F(V)}(N_{i_m}) = \mu_{F(V)}(N_{i_{m+1}})$, a lottery algorithm is performed for node selection. First, uniformly choose a random number r from $[0, 1)$. Then the m-th node is selected if $r \in [0, 0.5)$, and the the $m+1$-th node is selected if $r \in [0.5, 1)$. Moreover, it is natural to generalize the lottery algorithm to the case that more nodes share the same $\mu_{F(V)}(u)$.

IV. The Improved Model on the Vague Set-Based DPoS's Voting Phase

In this section, we present FW-DPoS (a new weighted vague to fuzzy mapping DPoS) which has two improvements on the vague set based DPoS's voting phase.

A. Improvement on the vague to fuzzy mapping

In practice, the abstention vote will favor the party with a larger number according to the number of affirmative votes and negative votes. Therefore, the influence factor of abstention will increase with the increase of affirmative votes and negative votes. First, we rewrite the mapping from the vague set to the fuzzy set in [23] as follows:

$$\mu_{F(V)}(u) = t_V(u) + \frac{1}{2}(1 + \alpha(t_V(u) - f_V(u))) \times (1 - t_V(u) - f_V(u)),$$

where

$$\alpha = 1/(t_V(u) + f_V(u) + 2\lambda).$$

The multiplication ratio α clearly represents the trend of the abstention part being counted into the final fuzzy value of the node.

We notice that this α decreases on $t_V(u) + f_V(u)$ for the fixed $\lambda > 0$. However, we consider that this property of α is not reasonable. If a node has high $t_V(u) + f_V(u)$, then its participation rate is also high, which means that the abstention part should contribute more when computing the fuzzy value. From this perspective, α should increase on $t_V(u) + f_V(u)$.

We now present our improvement, which is based on such a hypothesis: α is increasing on $t_V(u) + f_V(u)$. For simplicity, we just set

$$\alpha = t_V(u) + f_V(u)$$

which leads to an improved mapping:

$$\begin{aligned} &\mu_{F(V)}(u) \\ =\ & t_V(u) + \frac{1}{2}[1 + (t_V(u) + f_V(u))(t_V(u) - f_V(u))] \\ &\times (1 - t_V(u) - f_V(u)). \end{aligned} \tag{2}$$

Remark 1. *Notice that when $t_V(u) = f_V(u)$, we have $\mu_{F(V)}(u) = t_V(u) + \frac{1}{2}(1 - 2t_V(u)) = 0.5$, which is a reasonable case.*

B. Improvement on the voting weights of each node

In the previous voting model, each node can only cast one vote and the weight of each vote is set to be 1, 0 and -1 for favor votes, abstention votes and opposing votes, respectively. Thus the fuzzy value of each node can be computed by calculating the ratio of favor votes and the ratio of opposing votes.

Under the improved model, the weight of the node voting is considered [25]. In DPoS, the number of the votes each node can vote is different due to the different rights and interests held. Here we use voting weights to represent the equity of each node. For an elector, the favor votes received is the sum of the weights of all nodes that voted in favor, and we set $t_V(u)$ to be the ratio of this value in the sum of the weights of all nodes voted. Similarly, we set $f_V(u)$ to be the ratio of the weight summation of the opposing votes to the one of the all nodes. Then we use Formula 2 to calculate each node's fuzzy value and sort all the nodes by decreasing $\mu_{F(V)}(u)$.

Example: In order to see the changes of the fuzzy value before and after weighting, we present a simple example. Assume that there are 10 nodes in total, and the voting status of one node is shown in Table II. The node received 5 favor votes, 3 abstention votes and 2 opposing votes. The distribution of weights for voting is listed in Table III. For simplicity, the weights of 10 nodes are set to be ranging from 1 to 10. The weight vector of the favor, abstention and opposing votes are $[1,\ 4,\ 6,\ 7,\ 9]$, $[3, 5, 10]$ and $[2, 8]$, respectively. In this way, the vague value and the fuzzy value before and after weighting can be calculated.

TABLE II
NODES' VOTING RESULTS BEFORE WEIGHTING

Total votes	Favor votes	Abstention votes	Negative votes
10	5	3	2

TABLE III
WEIGHTS FOR VOTING

Total weights	Favor weights	Abstention weights	Negative weights
[1,2,3,4,5,6,7,8,9,10]	[1,4,6,7,9]	[3,5,10]	[2,8]

TABLE IV
VAGUE VALUES AND FUZZY VALUES BEFORE AND AFTER WEIGHTING

	Vague value interval	Fuzzy value
Before weighting	[0.5, 0.8]	0.6815
After weighting	[0.49, 0.82]	0.6886

From Table II, the vague value before weighting can be calculated as

$$t_V(u) = \frac{5}{10} = 0.5, \qquad f_V(u) = \frac{2}{10} = 0.2.$$

Then we use Table III to calculate the weighted vague values:

$$t_V(u) = \frac{1+4+6+7+9}{\sum_{i=1}^{10} i} = \frac{27}{55} = 0.49,$$

$$f_V(u) = \frac{2+8}{\sum_{i=1}^{10} i} = \frac{10}{55} = 0.18.$$

This means that the vague value interval before weighting is $[0.5, 0.8]$, and the weighted vague value is $[0.49, 0.82]$. Then we use Eq. (2) to calculate the corresponding fuzzy value. The results are shown in Table IV. For the vague value before and after weighting, notice that the requirements on $t_V(u)$ and $f_V(u)$ are always $t_V(u) \in [0,1]$, $f_V(u) \in [0,1]$ and $t_V(u) + f_V(u) \in [0,1]$. This also means that after the introduction of weights, it is still feasible to vote through the proposed model in Section V.

The improvement of the introduced weights is necessary and meaningful. It makes our model closer to the real situation of DPoS's voting phase since each node in DPoS has different voting rights. Moreover, the effect of weighting to the model will be explained in Section VII's experimental analysis.

V. MODEL ANALYSIS

Recall that for the vague set V, the improved model mapping the vagues set to the fuzzy set is defined by

$$\begin{aligned} &\mu_{F(V)}(u) \\ = \;& t_V(u) + \frac{1}{2}[(1 + (t_V(u) - f_V(u)))(t_V(u) + f_V(u))] \\ &\times (1 - t_V(u) - f_V(u)), \end{aligned}$$

where $[t_V(u), 1 - f_V(u)]$ is the vague value of $u \in U$.

After checking the properties of $\mu_{F(V)}(u)$, it can be proved that $\mu_{F(V)}(u)$ **increases** on $t_V(u)$ and **decreases** on $f_V(u)$.

Theorem 1. *For any $u \in U$ and the mapping 2, we have*

$$\frac{\partial \mu_{F(V)}(u)}{\partial t_V(u)} \geq 0, \quad \frac{\partial \mu_{F(V)}(u)}{\partial f_V(u)} \leq 0$$

on the conditions with $t_V(u)$, $f_V(u)$ and $t_V(u) + f_V(u) \in [0,1]$.

Proof. For simplicity, we replace $t_V(u)$, $f_V(u)$ and $\mu_{F(V)}(u)$ by t, f and μ to respectively rewrite the model 2 as

$$\mu = t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f).$$

We have to prove

$$\frac{\partial \mu}{\partial t} \geq 0 \;\; and \;\; \frac{\partial \mu}{\partial f} \leq 0$$

for conditions t, f and $t + f \in [0,1]$, It can be directly computed as follows:

$$\begin{aligned} \frac{\partial \mu}{\partial t} &= 1 + \frac{1}{2}(1 + t^2 - f^2)'_t \cdot (1 - t - f) \\ &\quad + (1 + t^2 - f^2) \cdot ((1 - t - f)'_t)) \\ &= 1 + \frac{1}{2}(2t \cdot (1 - t - f) - (1 + t^2 - f^2)) \\ &= \frac{1}{2}(2 + 2t - 2t^2 - 2tf - 1 - t^2 + f^2) \\ &= \frac{1}{2}(-3t^2 + (2 - 2f)t + (1 + f^2)). \end{aligned}$$

Since $t, f, t + f \in [0,1]$, the domain of t is actually $t \in [0, 1 - f]$ for some $f \in [0,1]$. Moreover, we have

$$\frac{\partial \mu}{\partial t}|_{t=0} = \frac{1}{2}(1 + f^2) > 0$$

and

$$\begin{aligned} \frac{\partial \mu}{\partial t}|_{t=1-f} &= \frac{1}{2}(-3(1-f)^2 + (2 - 2f)(1 - f) + (1 + f^2)) \\ &= f \geq 0. \end{aligned}$$

It follows that $\frac{\partial \mu}{\partial t} \geq 0$ at two endpoints. Since $\frac{\partial \mu}{\partial t}$ is a parabola opening down with respect to the variable t, it is true that

$$\frac{\partial \mu}{\partial t} \geq 0$$

for any $t \in [0, 1 - f]$ with $f \in [0,1]$, which completes the first part of the proof.

For the second part, similarly, we have

$$\begin{aligned} \frac{\partial \mu}{\partial f} &= \frac{1}{2}(1 + t^2 - f^2)'_f \cdot (1 - t - f) \\ &\quad + (1 + t^2 - f^2) \cdot ((1 - t - f)'_f)) \\ &= \frac{1}{2}(-2f \cdot (1 - t - f) - (1 + t^2 - f^2)) \\ &= \frac{1}{2}(-2f + 2tf + 2f^2 - 1 - t^2 + f^2) \\ &= \frac{1}{2}(3f^2 + (2t - 2)f - (1 + t^2)). \end{aligned}$$

For t, f and $t + f \in [0,1]$, the domain of f is $f \in [0, 1 - t]$ for some $t \in [0,1]$. Then we have

$$\frac{\partial \mu}{\partial f}|_{f=0} = -\frac{1}{2}(1 + t^2) < 0$$

and

$$\begin{aligned} \frac{\partial \mu}{\partial f}|_{f=1-t} &= \frac{1}{2}(3(1-t)^2 + (2t - 2)(1 - t) - (1 + t^2)) \\ &= -t \leq 0. \end{aligned}$$

It follows that $\frac{\partial \mu}{\partial f} \geq 0$ at two endpoints. Similarly, since $\frac{\partial \mu}{\partial f}$ is a parabola opening up with respect to the variable f, it is true that

$$\frac{\partial \mu}{\partial f} \leq 0$$

for any $f \in [0, 1-t]$ with $t \in [0, 1]$, which finishes the second part of the proof. □

Remark 2. *These properties of $\mu_{F(V)}(u)$ are reasonable. Since larger $t_V(u)$ or smaller $f_V(u)$ means that the possibility of the node u to be chosen into the next phase is larger, corresponding to the larger $\mu_{F(V)}(u)$.*

Theorem 2. *Given a vague set V in U, for any $u \in U$, if $t_V(u)$, $f_V(u)$ and $t_V(u) + f_V(u) \in [0, 1]$, then we have*

$$t_V(u) \leq \mu_{F(V)}(u) \leq 1 - f_V(u)$$

and the value distribution table about $\mu_{F(V)}(u)$ as follows.

Values of $\mu_{F(V)}(u)$	*Conditions on* $t_V(u)$ *and* $f_V(u)$
1	$t_V(u) = 1, f_V(u) = 0$
$(0.5, 1]$	$t_V(u) > f_V(u)$
0.5	$t_V(u) = f_V(u)$
$[0, 0.5)$	$t_V(u) < f_V(u)$
0	$t_V(u) = 0, f_V(u) = 1$

.

Proof. Similarly, we replace $t_V(u)$, $f_V(u)$ and $\mu_{F(V)}(u)$ respectively by t, f and μ to simplify the model as

$$\mu = t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f).$$

For the first part, notice that for $t, f, t+f \in [0, 1]$, it is easy to obtain

$$1 + t^2 - f^2 \in [0, 2].$$

Since $1 - t - f \geq 0$, we have

$$\mu \geq t + \frac{1}{2} \cdot 0 \cdot (1 - t - f) \geq t$$

and

$$\mu \leq t + \frac{1}{2} \cdot 2 \cdot (1 - t - f) = 1 - f,$$

which proves the first part of the theorem.

For the second part, we prove it in the following cases:

1) **Case 1**: $\mu = 1 \Leftrightarrow t = 1, f = 0$

$\Leftarrow$:

For $t = 1, f = 0$,

$$\begin{aligned}\mu &= t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f) \\ &= 1 + \frac{1}{2}(1 + 1^2 - 0^2)(1 - 1 - 0) = 1.\end{aligned}$$

$\Rightarrow$:

If $\mu = 1$, since $t \leq \mu \leq 1 - f$ from the first part, we know that $1 - f = 1$, implying $f = 0$. Then in fact $\mu = t + \frac{1}{2}(1 + t^2)(1 - t) = 1$. Simplify this equation to obtain

$$(1 - t^2)(t - 1) = 0.$$

Consequently, $t = 1$ and $f = 0$.

2) **Case 2**: $\mu = 0 \Leftrightarrow t = 0, f = 1$

$\Leftarrow$:

For $t = 0, f = 1$,

$$\begin{aligned}\mu &= t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f) \\ &= 0 + \frac{1}{2}(1 + 0^2 - 1^2)(1 - 0 - 1) = 0.\end{aligned}$$

$\Rightarrow$:

If $\mu = 0$, since we know that $t \leq \mu \leq 1 - f$, then $t = 0$. Thus we have $\mu = 0 + \frac{1}{2}(1 - f^2)(1 - f) = 0$, which can be simplified as

$$(1 - f^2)(1 - f) = 0.$$

As a result, $f = 1$ and $t = 0$.

3) **Case 3**: $\mu = 0.5 \Leftrightarrow t = f$

$\Leftarrow$:

For $t = f$,

$$\begin{aligned}\mu &= t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f) \\ &= t + \frac{1}{2}(1 + t^2 - t^2)(1 - t - t) \\ &= t + \frac{1}{2}(1 - 2t) = 0.5.\end{aligned}$$

$\Rightarrow$:

If $\mu = 0.5$, then

$$\begin{aligned}0 &= \mu - 0.5 \\ &= t + \frac{1}{2}(1 + t^2 - f^2)(1 - t - f) - \frac{1}{2} \\ &= \frac{1}{2}(2t + (1 + t^2 - f^2)(1 - t - f) - 1) \\ &= \frac{1}{2}(t - f - (t^3 - f^3) + (t^2 - f^2) - tf(t - f)) \\ &= \frac{1}{2}(t - f)(1 - (t^2 + tf + f^2) + t + f - tf) \\ &= \frac{1}{2}(t - f)(1 - (t + f)^2 + (t + f)).\end{aligned}$$

Since $t + f \in [0, 1]$, we know that

$$1 - (t + f)^2 + (t + f) \geq 1,$$

which implies that $t - f = 0$.

4) **Case 4** and **Case 5**:

$$\mu \in (0.5, 1] \Leftrightarrow f < t,$$

$$\mu \in [0, 0.5) \Leftrightarrow t < f.$$

From the proof in **Case 3**, we know that

$$\mu - 0.5 = \frac{1}{2}(t - f)(1 - (t + f)^2 + (t + f)).$$

Since $t + f \in [0, 1]$, we have

$$1 - (t + f)^2 + (t + f) \geq 1.$$

As a result, it follows

$$\mathbf{sgn}(\mu - 0.5) = \mathbf{sgn}(t - f).$$

Thus we have

$$\mu > 0.5 \Leftrightarrow f < t,$$

$$\mu < 0.5 \Leftrightarrow t < f.$$

From the first part of the proof for this theorem, we have $t \le \mu \le 1 - f$, implying $\mu \in [0, 1]$, which completes the proof of **Case 4** and **Case 5**.

□

VI. Experiments and Analysis

For our FW-DPoS voting model, we conduct experiments to verify its effectiveness, which are performed in the two versions: the small-scale version and the large-scale version. In the experiments, we simulate the voting process of 30 nodes in the small-scale version and 1000 nodes in the large-scale version. We use "0" for abstention vote, "1" for favor vote, and "-1" for opposing vote. We count the number of the votes received by each node and calculate the fuzzy values of each node by the new conversion formula (2). We use Python to run the experiments. The processor of the computer is Intel (R) Core (TM) i5-6500 CPU @ 3.20GHz with 8GB RAM. The operating system is Windows 10.

A. Small-scale Experiments

In this version, we conduct the experiments in small scale according to the following steps:

- *1) Generate nodes and assign weights.* We assume that **5** nodes are elected among **30** nodes. First we generate **30** nodes in the personal computer. In order to simplify the experiment, we regard all of these nodes as both the voters and the candidates. Then we randomly assign weight to each node. Here we choose a random integer in $[1, 100)$ for each node as its weight.
- *2) Make each node vote and calculate vague values.* We make each node randomly do the voting, where it can vote **yes**, **no** or abstain with the same probabilities equal to $1/3$. After the voting, we calculate the voting results. For each node u, Let the ratio of the favored weights to the total weights be $t_V(u)$, and the ratio of the opposed weights to the total weights be $f_V(u)$.
- *3) Compute fuzzy values to sort nodes and do election.* After calculating $t_V(u)$ and $f_V(u)$, we use the new mapping (2) to obtain the fuzzy value $\mu_{F(V)}(u)$ for each node, and sort the **30** nodes by their fuzzy values from large to small. The first **5** nodes will be selected as the committee members. In the case that the different nodes share the same fuzzy value, the node with the highest weight will be chosen. If their weights are still the same, we apply the lottery algorithm to randomly select the nodes.

TABLE V
Distribution of the weights of each node

Nodes	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}
Weights	55	97	38	19	49	70	61	68	23	19	70	78	31	51	81
Nodes	N_{16}	N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}	N_{23}	N_{24}	N_{25}	N_{26}	N_{27}	N_{28}	N_{29}	N_{30}
Weights	8	61	2	70	22	15	91	93	17	77	1	61	87	97	34

TABLE V shows the weight distribution of 30 nodes. This also represents the size of the node's power, which has an impact on the subsequent voting. It can be easily calculated that the weight sum of all nodes is $\sum_{i=1}^{30} w(N_i) = 1546$.

TABLE VI
The comparison of the fuzzy values' distribution in our model and Xu's model

Nodes	Yes votes	No votes	Abstention votes	The fuzzy values of our model with weight	The fuzzy values of our model without weight	The fuzzy values of Xu's model[13]
N_{11}	17	7	6	0.7331	0.6933	0.6786
N_{18}	14	7	9	0.6917	0.6412	0.6269
N_{30}	13	6	11	0.6508	0.6438	0.6329
N_{28}	16	6	8	0.6503	0.6993	0.6829
N_3	14	10	6	0.6312	0.5773	0.5714
N_{22}	11	8	11	0.5662	0.5616	0.5567
N_{20}	12	8	10	0.5642	0.5815	0.5750
N_{13}	13	10	7	0.5403	0.5589	0.5542
N_8	9	7	14	0.5386	0.5416	0.5395
N_{17}	11	7	12	0.5299	0.5827	0.5769
N_{23}	11	11	8	0.5163	0.5	0.5
N_7	10	9	11	0.5065	0.5205	0.5190
N_2	10	10	10	0.5063	0.5	0.5
N_{29}	7	7	16	0.5020	0.5	0.5
N_5	10	9	11	0.4808	0.5205	0.5190
N_1	9	9	12	0.4671	0.5	0.5
N_9	7	10	13	0.4576	0.4377	0.4416
N_{25}	11	9	10	0.4525	0.5407	0.5375
N_{27}	7	13	10	0.4447	0.3778	0.3875
N_4	9	10	11	0.4273	0.4795	0.4810
N_{24}	7	11	12	0.4179	0.4173	0.4231
N_{15}	9	15	6	0.4146	0.384	0.3929
N_{21}	11	11	8	0.4082	0.5	0.5
N_{10}	5	10	15	0.4073	0.3958	0.4
N_{14}	9	14	7	0.4051	0.4018	0.4096
N_{26}	6	9	15	0.4010	0.4375	0.44
N_{19}	10	13	7	0.3938	0.4411	0.4458
N_{16}	8	13	9	0.3324	0.3992	0.4074
N_{12}	5	14	11	0.3314	0.3152	0.3291
N_6	5	14	11	0.2916	0.3152	0.3291

In TABLE VI, we count the votes of all nodes and calculate the fuzzy values of our model and Xu's model. Then we sort the nodes according to our weighted fuzzy values in the descending order. Since 5 nodes needs to be chosen, from Table VI, the nodes N_{11}, N_{18}, N_{30}, N_{28} and N_3 will be selected. However, in the case of Xu's model, the nodes N_{28}, N_{11}, N_{30}, N_{18} and N_{17} will be selected. By comparing our model with Xu's model, we can see that in our model there are no situations that two different nodes share the same fuzzy value. However, in Xu's model, there are several nodes with the same fuzzy value. For example, the nodes N_2, N_{23} and N_{29} share the same fuzzy value 0.5 in Xu's model. But these three nodes have different fuzzy values in our model, which allows us to directly sort them and therefore improve the sorting efficiency. By comparing the experimental results of our unweighted fuzzy value with Xu's fuzzy value, it can be found that when the yes votes are more than the no votes, the unweighted fuzzy value in our model is greater than that in Xu's model. And when the yes votes are less than the no votes, the fuzzy value in our model is smaller than that in Xu's model. This means that our proposed conversion formula will bias the abstention votes in favor of the side with the larger number of the votes.

Through the TABLE VI, we can also analyze that when the number of the yes votes of the nodes is the same, the fuzzy value of nodes with more abstention votes is also high. This shows that although the sum of the yes votes and the no votes is the influence factor of the abstention votes, the main determining factor in the calculation is the number of the abstention votes. At the same time, we also note that the number of abstained votes is a part of the calculation of fuzzy value, but not a decisive factor. Even though the number of the abstention votes accounts for a large proportion, the size of its fuzzy value depends on the number of the yes votes. This is consistent with the actual situation.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3279571

B. Large-scale Experiments

To analyze the influence that assigning weights to the nodes has in the distribution of the fuzzy values, a large-scale version is conducted. We generate **1000** nodes with each node's weight randomly chosen from $[1, 1000000)$. Then we also make all of the nodes vote favor, oppose or abstention in random and compute the vague values for each node in our model and Xu's model. Finally, we use the new mapping Eq.(2) to compute the corresponding fuzzy value of each node and study the distributions of the fuzzy values in our model and Xu's model, respectively. Fig. 1 and Fig. 2 show the analyzed results.

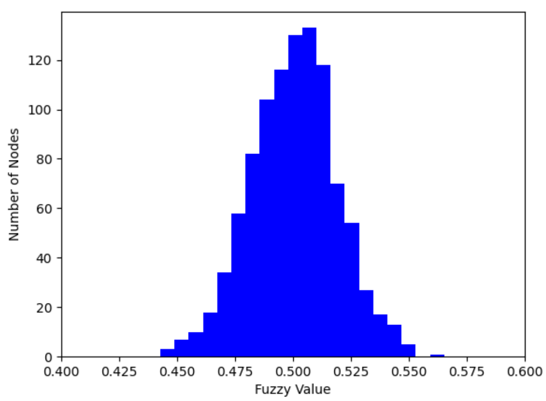

Fig. 1. Distribution of the fuzzy values in our FW-DPoS model.

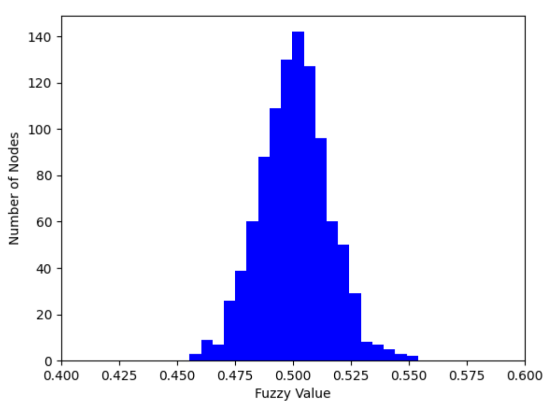

Fig. 2. Distribution of the fuzzy values in Xu's model.

It can be seen from the above figures that the distributions of the fuzzy values in our model and Xu's model are both subject to a normal distribution. However, the variances of the two distributions seem to be different. To be more specific, we use the hypothesis testing in statistics to study the two distributions of the fuzzy values, denoted by X and Y:

X : samples from the fuzzy values in our FW-DPoS model,

Y : samples from the fuzzy values in Xu's model.

C. Analysis of Hypothesis Testing

The improved model has been verified through the experimental results. From Fig. 1 and 2 and the hypothesis testing in the large-scale experiments, it can be concluded that

$$X \sim \mathcal{N}(0.5, \sigma_1^2), \quad Y \sim \mathcal{N}(0.5, \sigma_2^2),$$

where

$$\sigma_1^2 > \sigma_2^2.$$

We use several hypothesis testing methods to prove X and Y are normally distributed, and the variance of X is greater than the variance of Y, which implies this conclusion.

1) Apply KS-test to ensure the normality of X and Y: Kolmogorov-Smirnov test (KS-test) can be used to determine whether one dataset comes from a certain type of the distribution. In this case, we intend to clarify that both X and Y are subject to normal distribution. Since the KS-test has the advantage of making no assumption about the distribution of data, we now apply it to ensure the normality of X and Y, or

$$X, Y \sim \text{Normal Distribution}.$$

First, we calculate the sample mean and the sample standard deviation of X and Y:

$$\bar{X} = 0.50055, \qquad S_X = 0.01863,$$

$$\bar{Y} = 0.50053, \qquad S_Y = 0.01499.$$

Then we standardize X and Y by setting $X_{std} = (X - \bar{X})/S_X$ and $Y_{std} = (Y - \bar{Y})/S_Y$, set the significance level $\alpha = 0.05$ and the hypotheses to be as follows:

$$H_0(X) : X_{std} \sim \mathcal{N}(0, 1),$$

$$H_1(X) : X_{std} \nsim \mathcal{N}(0, 1),$$

$$H_0(Y) : Y_{std} \sim \mathcal{N}(0, 1),$$

$$H_1(Y) : Y_{std} \nsim \mathcal{N}(0, 1).$$

After using the *scipy.stats.kstest* module in Python to apply the KS-test to the standardized X and Y, we obtain the testing results:

$$\text{KstestResult}(\text{statistic} = 0.01792, \text{p-value} = 0.89925),$$

$$\text{KstestResult}(\text{statistic} = 0.02292, \text{p-value} = 0.66099).$$

The two p-values are both greater than 0.05, which indicates the strong evidence for the null hypothesis. Thus we retain the null hypotheses $H_0(X)$ and $H_0(Y)$ and reject the alternative hypotheses $H_1(X)$ and $H_1(Y)$, and it means $X_{std}, Y_{std} \sim \mathcal{N}(0, 1)$.

2) Apply T-test to estimate the means of X and Y: T-test is a type of the inferential statistic used to determine if there is a significant difference between the means of the two groups. It is mostly used when the data sets have unknown variances. In this case, we use T-test to show that X and Y are both subject to the normal distributions with the mean $= 0.5$, or

$$X \sim \mathcal{N}(0.5, \sigma_1^2), \quad Y \sim \mathcal{N}(0.5, \sigma_2^2).$$

First, we use $\bar{X} = 0.50055$, $S_X = 0.01863$, $\bar{Y} = 0.50053$ and $S_Y = 0.01499$ to construct T-Statistics T_X and T_Y:

$$T_X = \frac{\bar{X} - 0.5}{S_X/\sqrt{n}} = \frac{0.50055 - 0.5}{0.01863/\sqrt{1000}} = 0.93358,$$
$$T_Y = \frac{\bar{Y} - 0.5}{S_Y/\sqrt{n}} = \frac{0.50053 - 0.5}{0.01499/\sqrt{1000}} = 1.11808.$$

Then we set the significance level $\alpha = 0.05$ and list the hypotheses as follows:

$$H_0(X): X \sim \mathcal{N}(0.5, \sigma_1^2),$$
$$H_1(X): X \nsim \mathcal{N}(0.5, \sigma_1^2),$$
$$H_0(Y): Y \sim \mathcal{N}(0.5, \sigma_2^2),$$
$$H_1(Y): Y \nsim \mathcal{N}(0.5, \sigma_2^2).$$

Finally, we apply the T-test to X and Y to obtain the testing results:

$$|T_X| = 0.93358 < 2.24479 = T_{0.025}(999) = T_{\alpha/2}(n-1),$$
$$|T_Y| = 1.11808 < 2.24479 = T_{0.025}(999) = T_{\alpha/2}(n-1).$$

Since T_X and T_Y are both smaller than $T_{\alpha/2}(n-1)$, we retain the null hypothesis $H_0(X), H_0(Y)$ and reject the alternative hypothesis $H_1(X), H_1(Y)$, which means $X \sim \mathcal{N}(0.5, \sigma_1^2)$ and $Y \sim \mathcal{N}(0.5, \sigma_2^2)$.

3) Apply F-test to show the difference between the variances of X and Y: F-test is used to test if the variances of the two fuzzy values X and Y are equal. And the two-tailed version test method will be used to replace the F-test if the variances are not equal. In this case, we use F-test to show that there is a strong difference between the variances of X and Y, or $X \sim \mathcal{N}(0.5, \sigma_1^2)$ and $Y \sim \mathcal{N}(0.5, \sigma_2^2)$, where

$$\sigma_1^2 > \sigma_2^2.$$

First, we use $S_X = 0.01863$ and $S_Y = 0.01499$ to construct F-Statistics F:

$$F = \frac{S_X^2}{S_Y^2} = 1.54359.$$

Then we also set the significance level $\alpha = 0.05$ and suppose as follows:

$$H_0: \sigma_1^2 \leq \sigma_2^2,$$
$$H_1: \sigma_1^2 > \sigma_2^2.$$

Similarly, we apply the F-test to X and Y to obtain

$$F = 1.54359$$
$$> 1.10975 = F_{0.5}(999, 999) = F_\alpha(n-1, n-1).$$

Since F is larger than $F_\alpha(n-1, n-1)$, we accept the alternate hypothesis H_1 and reject the null hypothesis H_0, which means $\sigma_1^2 > \sigma_2^2$.

D. Further Analysis of our FW-DPoS model

To make the results more obvious, we add Fig. 3 and Fig. 4 for comparison. Fig. 3 shows the distribution map of our scheme after removing the weights, while Fig. 4 shows the distribution map of Xu's scheme after adding weights.

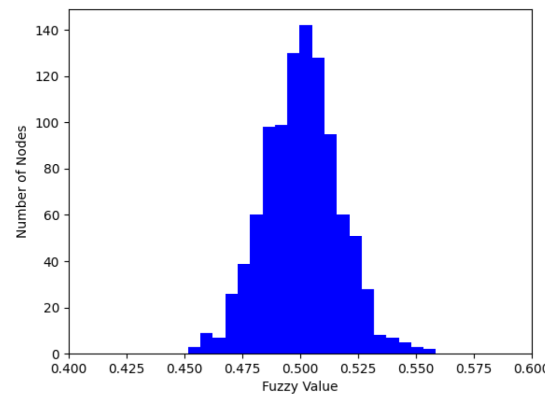

Fig. 3. Distribution of the fuzzy values in our model without weight.

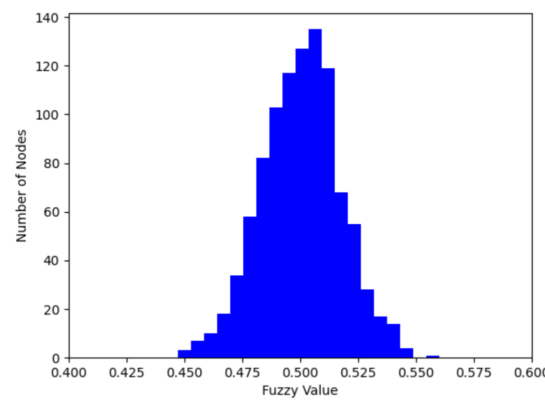

Fig. 4. Distribution of the fuzzy values in Xu's model with weight.

TABLE VII
THE COMPARISON OF FUZZY VALUES' MEAN AND STANDARD DEVIATION IN OUR AND XU'S MODELS FOR DIFFERENT CASES

	Fig. 1	Fig. 2	Fig. 3	Fig. 4
Mean	0.50055	0.50053	0.50057	0.50051
Standard deviation	0.01863	0.01499	0.01628	0.01715
Improved mapping	√	×	√	×
Weight	√	×	×	√

We compute the mean and standard deviation of Fig. 3 and Fig. 4. All results are shown in the TABLE VII

By comparing second and fourth columns of TABLE VII, it can be concluded that the fuzzy value standard deviation of our model with weight is greater than that of our model without weight. This means that by adding weights to our scheme, the distribution of fuzzy values becomes more dispersed, which is beneficial for more fair voting. Comparing third and fifth columns simultaneously, the standard deviation of Xu's model with weight is greater than that of Xu's model without weight. After adding weights to Xu's model, its fuzzy values will also be more uniform.

We compare second and fifth columns in TABLE VII, as well as third and fourth columns. The standard deviation in our model with weight is greater than that in Xu's model with weight, and the standard deviation in our model without weight is greater than that in Xu's model without weight. It can be seen that regardless of whether weights are added or not, the fuzzy value distribution of our scheme is more uniform than that of Xu's model. So our proposed scheme has better fairness and is conducive to DPoS voting.

E. Analysis of Experimental Results

Through the proof of the hypothesis testing, we get the following conclusion:

$$X \sim \mathcal{N}(0.5, \sigma_1^2), \quad Y \sim \mathcal{N}(0.5, \sigma_2^2),$$

where

$$\sigma_1^2 > \sigma_2^2.$$

This means that both X and Y have the average values close to 0.5. However, the variance of X is greater than that of Y, which implies that the distribution of X is more spread out than that of Y.

This indicates that the strategy of adding weights makes the distribution of the fuzzy value more uniform, which is helpful in selecting the nodes by the sorting the fuzzy value. The probability of the occurrence of the same value will become less, and it will reduce the necessity of applying the lottery algorithm. Thus, the improved model definitely leads to the efficiency growth in the voting phase.

We further analyzed the data and found that in the context of 1000 nodes, there was no situation where the fuzzy values of nodes were the same in our model, while in Xu's model, there were 182 nodes with the same fuzzy values. This will be detrimental to the sorting of nodes. Compared with Xu's model in the large-scale experiment, our scheme's fuzzy values will not be the same when there are a large number of nodes. That is to say, as the number of nodes increases, we can sort them directly based on the fuzzy values, without using the lottery algorithm. After adding weights, the probability of each node having the same fuzzy value is greatly reduced. As long as the node weights are sufficiently random, an increase in the number of nodes will not have a significant impact on the scheme's efficiency.

The advantages of our improved voting model can be seen from the following aspects.

- *1) Better simulation of the real DPoS's voting phase.* After adding weight values, our voting model is closer to the real voting mechanism of DPoS, since the number of the votes for each node in DPoS is different. To be elected in the voting phase, the nodes need not only to get the favor votes, but also to get the favor votes with the high weights. This undoubtedly increases the fairness of our voting model.
- *2) More theoretical analyses on the new mapping.* The new mapping Eq. (2) remains the fuzzy value 0.5 when the favor votes and opposing votes are the same. We also prove its monotonicity with $t_V(u)$ and $f_V(u)$.
- *3) More efficient selection of the nodes.* We can see from the experimental results that by adding weight values, the variance of the distribution of the fuzzy value becomes larger, which makes the fuzzy value closer to the uniform distribution. As a result, this will improve the efficiency of the node selection.

VII. Conclusion

An improved model called FW-DPoS on DPoS's voting phase in Blockchain is established. In this FW-DPoS model, a more reasonable conversion formula from the vague set to the fuzzy set is proposed. If there are more **yes** votes and **no** votes, the participation rate will be higher, which implies that the abstention votes can correctly contribute more in computing the fuzzy value. In addition, in order to make the model closer to the real voting situation, we add weights for voting. The larger the weight is, the more significant its vote will be. By introducing the concept "weight", the probability of the same fuzzy value will be reduced, and the use of random lottery algorithm will be reduced. This makes the node selection process more fair.

By analyzing the experimental results, our FW-DPoS model is more efficient than Xu's model. When the yes votes are greater than the no votes, our fuzzy value will be greater than Xu's fuzzy value; when the yes votes are less than the no votes, our fuzzy value will be less than Xu's fuzzy value. In this way, the nodes with more yes votes are easier to enter the committee. When the affirmative votes remain the same, the more abstention votes, the greater the fuzzy value of the node. When the no votes are the same. The more abstention votes the node has, the smaller the fuzzy value will be. Our new model strengthens the influence factor of abstention vote, making it increase with the increase of effective votes. That is to say, abstention will be more likely to favor the party with more votes. This is in line with the actual situation. In real life, the nodes will judge their identity according to the number of votes, which will affect the voting bias.

Our proposed model can be applied to the blockchain system, especially in the alliance chain. Our scheme can alleviate the decentralized problem of DPoS, make the members of the committee more random, and improve the fairness of the block. Meanwhile, our FW-DPoS model can be applied to industrial blockchain systems, such as IoV Blockchain, to improve the efficiency of the consensus algorithm by replacing the originally used DPoS with our FW-DPoS. At the same time, our FW-DPoS model can also be used in the IoT blockchains to improve the security of the consensus algorithms. In detail, the fuzzy values in our model are more scattered, which makes our FW-DPoS fairer and reduces the probability of the malicious nodes entering the committee compared with the classical DPoS used in some

IoT blockchains. However, we leave the problem of how to further improve the decentralization of DPoS's voting phase as an open issue.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org//bitcoin.pdf.

[2] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin Meets Strong Consistency," In: *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pp.1-10, 2016.

[3] K. Li, L. Hui, H. Hou, K. Li and Y. Chen, "Proof of Vote: A High-Performance Consensus Protocol Based on Vote Mechanism Consortium Blockchain," In: *HPCC/SmartCity/DSS 2017*, pp. 466-473, 2017.

[4] V. Buterin, and V. Griffith, "Casper the Friendly Finality Gadget," 2017. [Online]. Available: https://arxiv.org//abs//1710.09437.pdf.

[5] S. Zhang, J. Lee, "Analysis of the main consensus protocols of blockchain", *ICT Express*, vol. 6, no. 2, pp. 93-97, 2020.

[6] C. Boyd, and C. Carr, "Valuable Puzzles for Proofs-of-Work," In: *Proceedings of DPM/CBT 2018*, LNCS, vol. 11025, pp. 130-139, 2018.

[7] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of financial studies* vol. 34, pp. 11561190, 2021.

[8] Y. Gao, and X. Tan, "Improvement of DPoS consensus mechanism," *Application Research of Computers*, vol. 37, no. 10, pp. 3086-3090, 2020.

[9] L. Wang, P. Xu, W. Su, Y. Li, and X Chen, "Research on Improvement of Blockchain DPOS Consensus Mechanism Based on HK Clustering", *Automation Congress (CAC)* 2021 China, pp. 1167-1172, 2021.

[10] Y. Liu, and G. Xu, "Fixed degree of decentralization DPoS consensus mechanism in blockchain based on adjacency vote and the average fuzziness of vague value", *Computer Networks*, vol. 199, 2021.

[11] G. Xu, Y. Liu, and P. W. Khan, "Improvement of the DPoS Consensus Mechanism in Blockchain Based on Vague Sets," *IEEE Transactions on Industrial Informatics*, vol.16, no. 6, pp. 4252-4259, 2020.

[12] Y. Liu, J. Liu, and Z. Zhang, "Overview On Blockchain Consensus Mechanism," *Journal of Cryptologic Research*, vol.6, no. 4, pp. 395-432, 2019.

[13] M. Castro, and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *ACM Transactions on Computer Systems*, vol.20, no.4, pp. 398-461, 2002.

[14] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Ling, and P. Verissimo, "Efficient Byzantine Fault-Tolerance," *IEEE Tansactions on Computers*, vol.62, no.1, pp. 16-30, 2013.

[15] S. King, and S. Nadal, "PPcoin: Peer-to-peer crypto-currency with proof-of-stake," 2012. [Online]. Available: https://bitcoin.peryaudo.org/vendor/peercoin-paper.

[16] I. Grigg, "EOS-An introduction," 2017. [Online]. Available: http://eos.io//documents//EOS-An introduction.pdf.

[17] G. Nguyen, and K. Kim, "A Survey about Consensus Algorithms Used in Blockchain," *Journal of Information Processing Systems*, vol.14, no.1, pp. 101-128, 2018.

[18] X. Han, and Y. Liu, "Research on the consensus mechanisms of Blockchain technology," *Netinfo Security 2017*, vol. 9, pp. 147-152, 2017.

[19] R. Ta, and M. Z. Tanrver, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting," *Symmetry*, vol. 12, no. 8, pp. 1328, 2020.

[20] M. Liu, M. Tan, P. Lei, "Research on DPOS Consensus Mechanism Improvement", *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pp. 84-88, 2022.

[21] Y. Wei, L. Liang, B. Zhou, and X. Feng, "A Modified Blockchain DPoS Consensus Algorithm Based on Anomaly Detection and Reward-Punishment", *Communication Software and Networks (ICCSN) 2021 13th International Conference on*, pp. 283-288, 2021.

[22] Y. Liu, G. Wang, and F. Lin, "A general model for transforming vague sets into fuzzy setsm," *Transactions on Computational Science*, vol. 2, pp. 133-144, 2008.

[23] Y. Shi, and H. Wang, "The method criteria of Vague value into the fuzzy value," *Computer Engineering and Application*, vol. 41, no. 24, pp. 169-171, 2005.

[24] J. Liu, Z. Liu, S. Wu, and Y. Zhang, "New Method for Approximating Vague Sets to Fuzzy Sets based on Voting Model," In: *Proceedings of CIS 2008*, pp. 869-872, 2008.

[25] S. Leonardos, D. Reijsbergen, and G. Piliouras, "Weighted Voting on the Blockchain: Improving Consensus in Proof of Stake Protocols," *International Journal of Network Management*, vol. 30, no. 5, 2020.

Lin You is currently a Professor in the School of Cyberspace, Hangzhou Dianzi University. He is also the Director of the Institute of Cryptography and Information Security. His research interests include cryptography, biometric recognition, and their combinations and applications. He is also a member of IACR and CACR.

Zhuobiao Wang is currently studying for the M.S. degree in electronic information at Hangzhou Dianzi University. His research interests include blockchain consensus algorithms and cryptography.

Gengran Hu is currently a Lecturer in the School of Cyberspace, Hangzhou Dianzi University. His research interests include lattice-based cryptography, blockchain, and their applications. He is also a member of CACR.

Chengtang Cao is a PhD student majored in cryptography at Hangzhou Dianzi University. His research interests include lattice-based cryptography and blockchain.

Liang Li is currently a lecturer in the School of Cyberspace, Hangzhou Dianzi University. His research interests include cryptography, blockchain and finance.