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

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

***Abstract*—As a common consensus mechanism used in blockchain systems, DPoS uses voting to select committee members who will generate the corresponding blocks. In order to elect committee members as fairly as possible, the vague sets are introduced into the voting phase of DPoS. In the vague sets based model proposed by Xu et al., the voting nodes can vote for, oppose or abstain from it. In this paper, we improve this vague set based model by introducing a new mapping from the vague set to fuzzy set and considering the case that each voting node is assigned a weight. In addition, several nice properties of our improved model are proved and it makes the voting phase of DPoS more fair.**

***Index Terms*—Blockchain, DPoS, Voting, Vague Set, Consensus Mechanism**

I. Introduction

Since Satoshi Nakamoto proposed Bitcoin [1], its core technology, blockchain, has been highly valued. Blockchain technology can solve the security issues of untrusted third parties and data tampering. Blockchain can be seen as a non-tamperable distributed ledger and a decentralized database. In blockchain, each block contains transaction data generated by the network over a period of time. By consensus mechanism, all the network nodes in the network can verify the validity of a new block and participate in the mining to generate the next block [2]. As a result, the continuous blocks form a chain structure in the chronological order, which is the origin of the term ”blockchain”. Blockchain technology has been constantly improving these years. Up to now, Blockchain has successively experienced the three obvious eras. In the era of Blockchain 1.0, the main research topic is digital cryptocurrencies [3], among which Bitcoin is the typical representative. Blockchain 2.0 introduces the concept of smart contracts [4], by which users can customize contracts and expand the blockchain. In this era, Ethereum [5] is a typical programmable blockchain system. Blockchain 3.0 is supposed to be a more secure and complete smart contract. However, there is currently no blockchain platform meeting such requirements.

There are currently three types of blockchains: public chains, private chains, and consortium chains [6]. The public chains are completely decentralized, in which any user can freely enter and exit. The private chains are usually the blockchains within a company, and only the members of this company can join in. The consortium chains are a type of blockchains between the public chains and the private chains. The members of the consortium can not enter or exit the system without the consent of the consortium.

Blockchain uses technologies such as p2p networks, digital signatures and smart contracts to achieve openness, decentralization, immutability and permanence. These nice properties make blockchains suitable for applications in the fields of medicine, finance and data storage. At present, blockchain has been showing a trend of continuous development. However, Before blockchain can be widely used and improve our daily life, the research on blockchain security issues still needs more our attentions.

Consensus mechanism is the key to the decentralization and security of blockchain technology. There are many existing consensus mechanisms [7], such as PBFT [8], [9], PoW, PoS [10], and DPoS [11]. 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 set-based DPoS voting model is introduced in Section IV. Then an improved model and the reason for the improvement are presented in Section V. Later, the improved model is analyzed and several nice properties of it are proved in Section VI. In the next section, the experiment simulation on our improved model is done and described. The last section summarizes the article.

II. Blockchain's Consensus Mechanism and DPoS

The consensus mechanism [12] of blockchain is a technology that allows irrelevant nodes to reach a consensus on the transactions. Due to the existence of the consensus mechanism, any 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 any transaction. This improves the security of the transaction and avoids tampering of any transaction. The main phases of the consensus mechanism are as follows [13]:

- *1) Election of block producers:* Select the node which

This work was supported in part by the National Natural Science Foundation of China under the Grants (61602143, 61772166) and in part by the Key Program of the Nature Science Foundation of Zhejiang province of China under Grant LZ17F020002.(*Corresponding Author: Gengran Hu*)

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

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 broadcasted across the entire network. The nodes that receive the information can 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 firstly used in Bitcoin. The term ”proof-of-work” refers to that a node has to prove how much work it has done, meaning to solve a difficult mathematical problem, for which the problem difficulty can be dynamically adjusted. For example, in Bitcoin system, each block header contains the hash value of the previous block (prev-hash), a random number (nonce), the time stamp (time-stamp), and the root of the Merkle tree (Merkle-root). The node packs the header block of the block producer and the random number together for hash output, where the hash function used in most of the blockchain systems is SHA-256. Given the difficulty level k, the node keeps guessing the value of the nonce until it satisfies that $hash(\text{block data}||\text{nonce})$ starts with k consecutive 0. The node that finds the proof of work broadcasts the new block together with its hash value. Then the other nodes receiving the data verify the legitimacy of the new block and update the local blockchain. Since the hash function has strong collision-resistance and nearly uniform output, it is very difficult to find a valid nonce, which means that the node needs to pay a great deal of computing power to find such a random number. Such a process is the so-called ”mining”, and only the first node completing the mining can obtain rewards.

Ethereum, as a typical system of the blockchain 2.0 era, also adopted a proof-of-work mechanism at the beginning of its release. However, compared with the previous Bitcoin's proof-of-work mechanism, Ethereum has a huge difference. In Ethereum, the interval time among the block productions is about 15 seconds, which is much faster than the block production by Bitcoin (about 10 minutes per block). In Bitcoin, only the main chain is used and chain forks are not allowed. In contrast, chain forks are legal in Ethereum. Even the forked miners can get a certain percentage of remuneration. However, the workload proof mechanism requires huge resource consumption. Thus, Ethereum began to do the transition from a proof-of-work mechanism to a proof-of-stake mechanism. Specifically, the Casper adopted by Ethereum is a consensus mechanism that combines PoW and PoS. In Casper, a checkpoint is generated and verified for every 100 blocks to avoid the resulting forks by the previous PoW method. To become a verifier, a node needs to bet its own ”ether”(Ethereum's token) into a smart contract. A successful verification will bring a reward to the verifier, while the verification failure will also penalize the corresponding ether. The verifier needs to broadcast a message, including the block hashes and the block heights of the source checkpoint and the destination checkpoint, together with the signature of the verifier. The checkpoint will only be confirmed if it receives $2/3$ of the valid votes of the verifier.

As a relatively new consensus mechanism, Delegated Proof of Stake(DPoS) [14] was proposed in 2014. In DPoS, token holders vote to elect the nodes that generate blocks. The size of the equity held by the holders determine their votes [15]. The nodes owing greater equity have more votes. After the voting phase, the fixed number of the nodes with the most votes will become committee members to generate blocks. Each node will generate a block in turn. If the node does not generate a block during a specific time period, it will be delisted. And the network will select a new node to replace it. In this paper, we improve the vague set-based voting phase 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 reducing the possibility of selecting malicious nodes.

III. Fuzzy Set and Vague Set

In the traditional set theory [16], 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 membership degree. The degree of membership 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 tall people, 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, 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 degree of membership in the set [17]. However, an interval instead of a single value is used to represent a vague set. Given a set U, $t_V(u)$ represents the degree of membership that V truly belongs to U, and $t_V(u)$ is the lower bound of support membership derived from the support of evidence. At the same time, the membership degree $f_V(u)$ represents the degree of membership that V falsely belongs to U. As a result, $f_V(u)$ is the lower bound of opposing membership derived from the opposing of evidence. Therefore, the membership interval of V is $[t_V(u), 1 - f_V(u)]$.

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

The DPoS voting model recently proposed is based on the conversion from vague set to fuzzy set [18]. In this model [19], each node can vote for, against or to abstain. And the ratio of affirmative votes to the total votes is the true membership degree, namely $t_V(u)$. The percentage of negative 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 vague set. Through the transformation from vague set to fuzzy set, the interval value of vague set is transformed into the unique value of fuzzy set. As a result, vague set can be transformed into a fuzzy set. In [16], Y. Liu et al. proposed several relatively simple transformation methods and discussed their advantages and shortcomings. In [19], G. Xu et al. formally presented a more complicated conversion from $[t_V(u), 1 - f_V(u)]$ to $\mu_{F(V)}(u)$:

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

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 there are m nodes to select among n nodes, according to $\mu_A(N_{i_1}) > \mu_A(N_{i_2}) > \cdots > \mu_A(N_{i_n})$, the first m nones $N_{i_1}, N_{i_2}, \cdots N_{i_m}$ are selected. If $\mu_A(N_{i_m}) = \mu_A(N_{i_{m+1}})$, 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 $\mu_{F(V)}(u)$.

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

In this section, we present two improvements on the vague-set based DPoS's voting phase.

A. Improvement on the vague to fuzzy mapping

First, we rewrite the mapping function from vague set to fuzzy set in [19] as follows.

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

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 is decreasing on $t_V(u) + f_V(u)$ for fixed $\lambda > 0$. However, we consider that this property of α is not reasonable. If $t_V(u) + f_V(u)$ is higher, then its percentage of abstentions is smaller, which means the abstention part should contribute more when computing the fuzzy value. From this perspective, α should be increasing 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 function:

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

***Remark** 1:* Notice that when $t_V(u) = f_V(u)$, $\mu_{F(V)}(u) = t_V(u) + \frac{1}{2}(1 - 2t_V(u)) = \frac{1}{2}$ as required.

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 affirmative votes, abstention votes and negative votes, respectively. Thus the fuzzy value of each node can be computed by calculating the ratio of YES votes and the ratio of NO votes.

Under the improved model, the weight of node voting is considered [20]. In DPoS, the number of 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 affirmative 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 all nodes that voted against. 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 visually see the changes of fuzzy value before and after weighting, we present a simple example. Assuming there are 10 nodes in total, the voting status of one node is shown in TABLE 1. The node received 5 supporting votes, 3 abstention votes and 2 negative votes. The distribution of weights for voting is listed in TABLE 2. For simplicity, the weights of 10 nodes are set to be ranging from 1 to 10. The weight vector of the supporting, abstention and negative votes are $[1,\ 4,\ 6,\ 7,\ 9]$, $[3, 5, 10]$ and $[2, 8]$, respectively. In this way, the vague value and fuzzy value before and after weighting can be calculated.

TABLE I: Node voting results before weighting

Total votes	YES votes	Abstention votes	NO votes
10	5	3	2

TABLE II: Weights for voting

Total weights	YES weights	Abstention weights	NO weights
[1,2,3,4,5,6,7,8,9,10]	[1,4,6,7,8]	[3,5,10]	[2,8]

TABLE III: vague value and fuzzy value before and after weighting

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

From TABLE I, 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 II to calculate the weighted vague value:

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

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

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

The improvement of introducing 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 further explained in Section VII's experimental analysis .

VI. Model Analysis

Recall that for vague set V, the improved model mapping vagues sets to fuzzy sets 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 $\mu_{F(V)}(u)$ is **increasing** on $t_V(u)$ and **decreasing** on $f_V(u)$.

***Theorem** 1:* For any $u \in U$ and the mapping function (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 that $t_V(u)$, $f_V(u)$, $t_V(u) + f_V(u) \in [0,1]$.

Proof: For simplicity, we replace $t_V(u)$, $f_V(u)$, $\mu_{F(V)}(u)$ by t, f, μ to rewrite the model (3) as

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

We have to prove that on conditions $t, f, t + f \in [0,1]$,

$$\frac{\partial \mu}{\partial t} \geq 0, \quad \frac{\partial \mu}{\partial f} \leq 0.$$

It can be directly computed that

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

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

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

which means 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 variable t, it is true that

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

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

For the second part, similarly, we compute

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

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

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

and

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

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

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

for any $f \in [0, 1 - t]$ where $t \in [0,1]$, finishing 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 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 $u \in U$, on the conditions that $t_V(u)$, $f_V(u)$, $t_V(u) + f_V(u) \in [0,1]$, we have

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

and

value of $\mu_{F(V)}(u)$	relation 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 $t_V(u)$, $f_V(u)$, $\mu_{F(V)}(u)$ by t, f, 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 **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 know

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

As a result,

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

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 $t \leq \mu \leq 1 - f$, implying $\mu \in [0, 1]$, which completes the proof of **Case 4** and **5**.

■

VII. Experiments and Analysis

After proposing the new Blockchain's DPoS voting model, we conducted experimental simulations to verify its effectiveness. And the simulations are in two version: small-scale and large-scale.

A. Small-scale Simulation

In this version, we conducted the simulation in small scale according to the following steps:

- *1) Generate nodes and assign weights.* We assume that **5** nodes are to elect among **30** nodes. First we generate **30** nodes in personal computer. In order to simplify the experiment, we regard all of these nodes as voters and electors. 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 for, against, and abstention with same probabilities equal to $\frac{1}{3}$. After the voting, we calculate the voting results for each node. 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 function (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 committee members. In the case that 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 nodes.

TABLE IV: Distribution of weights of each node

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}
55	97	38	19	49	70	61	68	23	19	70	78	31	51	81
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}
8	61	2	70	22	15	91	93	17	77	1	61	87	97	34

Table IV 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 V: Fuzzy set value sorting of each node

Nodes	YES votes	No votes	Abstention votes	unweighted fuzzy value	weighted fuzzy value
N_{11}	17	7	6	0.6933	0.7331
N_{18}	14	7	9	0.6412	0.6917
N_{30}	13	6	11	0.6438	0.6508
N_{28}	16	6	8	0.6993	0.6503
N_3	14	10	6	0.5773	0.6312
N_{22}	11	8	11	0.5616	0.5662
N_{20}	12	8	10	0.5815	0.5642
N_{13}	13	10	7	0.5589	0.5403
N_8	9	7	14	0.5416	0.5386
N_{17}	11	7	12	0.5827	0.5299
N_{23}	11	11	8	0.5	0.5163
N_7	10	9	11	0.5205	0.5065
N_2	10	10	10	0.5	0.5063
N_{29}	7	7	16	0.5	0.5020
N_5	10	9	11	0.5205	0.4808
N_1	9	9	12	0.5	0.4671
N_9	7	10	13	0.4377	0.4576
N_{25}	11	9	10	0.5407	0.4525
N_{27}	7	13	10	0.3778	0.4447
N_4	9	10	11	0.4795	0.4273
N_{24}	7	11	12	0.4173	0.4179
N_{15}	9	15	6	0.384	0.4146
N_{21}	11	11	8	0.5	0.4082
N_{10}	5	10	15	0.3958	0.4073
N_{14}	9	14	7	0.4018	0.4051
N_{26}	6	9	15	0.4375	0.4010
N_{19}	10	13	7	0.4411	0.3938
N_{16}	8	13	9	0.3992	0.3324
N_{12}	5	14	11	0.3152	0.3314
N_6	5	14	11	0.3347	0.2916

In Table V, we counted the votes of all nodes and used the votes to compare the fuzzy value of each node before and after weighting. Then we sort the nodes according to their weighted fuzzy values in descending order. Since 5 nodes needs to be chosen, from Table V, the nodes N_{11}, N_{18}, N_{30}, N_{28} and N_3 will be selected. However, in the case of no weights, the nodes N_{28}, N_{11}, N_{30}, N_{18} and N_{17} will be selected. Compared to node N_3, node N_{17} has higher unweighted fuzzy value but lower weighted fuzzy value. After adding weights, the probability of obtaining the same fuzzy value seems to become smaller. For example, node N_2 and N_{29} share the same fuzzy value 0.5 before weighting, but their fuzzy values after weighting are different, which allows us to directly sort them and therefore improve the sorting efficiency.

B. Large-scale Simulation

In this version, to study the influence that assigning weights to nodes has in the distribution of fuzzy values, a large-scale simulation 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 for, against or to abstain in random and compute the unweighted and weighted vague values for each node. Finally, we use the new mapping function (2) to compute the corresponding fuzzy value of each node and study the distributions of unweighted fuzzy values and weighted fuzzy values, which can be seen in Figure 2 and Figure 3.

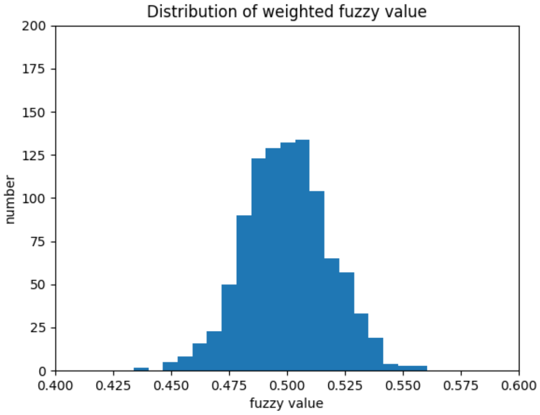

Fig. 1: Distribution of weighted fuzzy values.

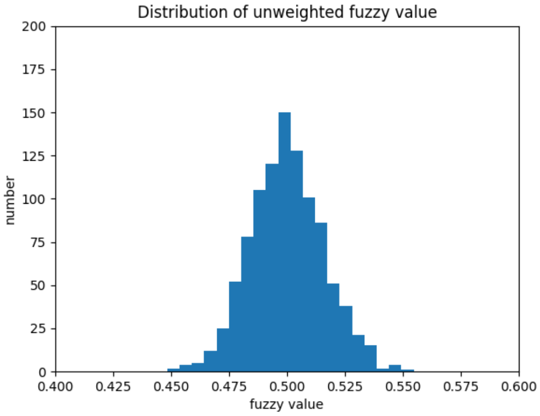

Fig. 2: Distribution of unweighted fuzzy values.

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

$$X: \quad \text{samples from weighted fuzzy value,}$$

$$Y: \quad \text{samples from unweighted fuzzy value.}$$

1) Apply KS-test to ensure the normality of X and Y: Kolmogorov-Smirnov test (KS-test) can be used to determine if one dataset comes from a certain type of 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 sample standard deviation of X and Y:

$$\bar{X} = 0.50001, \qquad S_X = 0.01833,$$

$$\bar{Y} = 0.50025, \qquad S_Y = 0.01533.$$

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 null hypothesis and the alternate hypothesis 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:

$$\text{KstestResult}(\text{statistic} = 0.02288, \text{pvalue} = 0.66323),$$

$$\text{KstestResult}(\text{statistic} = 0.02738, \text{pvalue} = 0.43371).$$

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 $H_0(X), H_0(Y)$ and reject the alternative hypotheses $H_1(X), H_1(Y)$, which means $X_{std}, Y_{std} \sim \mathcal{N}(0, 1)$ and therefore X and Y are subject to normal distribution.

2) Apply T-test to estimate the means of X and Y: T-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups. It is mostly used when the data sets have unknown variances. In this case, we use T-test to show that X, Y are both subject to normal distributions with 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 $\bar{X} = 0.50001, S_X = 0.01833, \bar{Y} = 0.50002, S_Y = 0.01533$ to construct T-Statistics T_X and T_Y:

$$T_X = \frac{\bar{X} - 0.5}{S_X/\sqrt{n}} = \frac{0.50001 - 0.5}{0.01833/\sqrt{1000}} = 0.01725,$$

$$T_Y = \frac{\bar{Y} - 0.5}{S_Y/\sqrt{n}} = \frac{0.50025 - 0.5}{0.01533/\sqrt{1000}} = 0.51570.$$

Then we set the significance level $\alpha = 0.05$ and the null hypothesis and the alternate hypothesis 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:

$$|T_X| = 0.01725 < 2.24479 = T_{0.025}(999) = T_{\alpha/2}(n-1),$$

$$|T_Y| = 0.51570 < 2.24479 = T_{0.025}(999) = T_{\alpha/2}(n-1).$$

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 variances of X and Y: F-test is used to test if the variances of two populations are equal. The two-tailed version tests against the alternative that 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 $S_X = 0.01833$ and $S_Y = 0.01533$ to construct F-Statistics F:

$$F = \frac{S_X^2}{S_Y^2} = 1.42969.$$

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

$$H_0: \sigma_1^2 \le \sigma_2^2,$$

$$H_1: \sigma_1^2 > \sigma_2^2.$$

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

$$\begin{aligned} F &= 1.42969 \\ &> 1.10975 = F_{0.5}(999, 999) = F_\alpha(n-1, n-1). \end{aligned}$$

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

C. Analysis of Experimental Results

The improved model was verified through experimental results. From Figure 2 and 3 and the hypothesis testing in the large-scale simulation, it can be concluded that under the new mapping function, for weighted fuzzy value X and unweighted fuzzy value Y, we have

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

where

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

This 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 fuzzy value. The probability of appearing same fuzzy value will become less, reducing the necessity of applying lottery algorithm. Thus the improved model definitely leads to the efficiency growth in voting phase.

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

- *1) Better simulation to real DPoS's voting phase.* After adding weights, our voting model is closer to the real voting mechanism of DPoS, since the number of votes for each node in DPoS is different. To be elected in the voting phase, nodes need not only to get the yes votes, but also to get the yes votes with high weights. This undoubtedly increases the fairness of voting model.
- *2) More theoretical analyses on the new mapping function.* The new conversion function 2 retains the fuzzy value of 0.5 when the approval and disapproval are the same. We also prove its own unique characteristics in Section VI.
- *3) More efficient selection of nodes.* We can see from the experimental results that by padding weights, the variance of the distribution of fuzzy value is becoming larger, which makes the fuzzy value closer to uniform distribution. As a result, this will improve the efficiency of node selection.

VIII. Conclusion

In this paper, an improved model on DPoS's voting phase in blockchains has been proposed. In the previous voting model, the voters can only vote for, oppose or abstain from a single vote. By applying the conversion of the vague set to the fuzzy set in voting, the nodes will be elected more fairly. In our new model, a more reasonable conversion formula between the vague set and fuzzy set is proposed. If there are more pros and cons, the participation rate will be higher, and it implies that the abstention part should contribute more in computing the fuzzy value. Our new conversion formula is proposed by following this idea. In addition, in order to make the model closer to the real voting situation, we add weights to the voting. In this way, what each node has is no longer a single vote, but it is a weighted one. The larger the weight is, the more significant its vote will be. Our experimental results show that the new model is both valid and efficient. However, we leave the problem of how to further improve the decentralization of the DPoS's voting phase as a open question.

References

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

[2] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin Meets Strong Consistency," 2014. [Online]. Available: https://arxiv.org/pdf/1412.7935.pdf.

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

[4] S. Xuan, L. Zheng, I. Chung, et al., "An incentive mechanism for data sharing based on blockchain with smart contracts," *Computers & Electrical Engineering*, 2020, 83:106587.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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

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

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

[18] J. Liu, Z. Liu, S. Wu, et al., "New Method for Approximating Vague Sets to Fuzzy Sets based on Voting Model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4252-4259, 2020.

[19] G. Xu, Y. Liu, and P. W. Khan, "Improvement of the DPoS Consensus Mechanism in Blockchain Based on Vague Sets," *In: Proceedings of CIS 2008*, pp. 869-872, 2008.

[20] 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.