DPOS consensus algorithm optimized based on index dynamic balancing algorithm

Abstract—**With the development of the new electricity market, the related parties continue to increase, and the traditional power market management needs to rely on the third party trust center. Blockchain has the advantage of decentralization, and does not need a third party trust center, which can be well applied to the data management of power market. DPOS (Delegated Proof of Stake) consensus protocol is widely used for blockchain data management in smart grids due to its high throughput, low latency, and low power consumption. In the original DPOS consensus, the voting is carried out according to the node share weight, so that the node with a large share weight is easier to be elected as the agent node, resulting in the reduction of the enthusiasm of the other nodes to participate in the consensus, resulting in the reduction of the enthusiasm of ordinary nodes in DPOS to participate in the consensus, which reduces the performance of the consensus. This paper proposes DPOS+IDB algorithm (DPOS with weighted Index Dynamically Balancing), which transforms the network configuration optimization problem into the problem of finding the overall optimal solution of multiple indexes. The dynamic balancing algorithm was used to solve the problem, and the solution results were used to optimize the configuration of the blockchain network to improve the enthusiasm of nodes. Simulation results show that compared with the original DPoS algorithm, CE-DPOS (Comprehensive Election-DPoS) algorithm and RC-DPOS (Reputation Classification-DPOS) algorithm, DPOS+IDB algorithm has better performance in delay and throughput, and reduces the communication overhead.**

Index Terms—**Blockchain; DPOS; Performance optimization; Index dynamically balancing algorithm**

I. Introduction

Ref. [1] points out that in recent years, the number of trading participants in the electricity market has been increasing. The traditional centralized management mode of electric power management depends on the third party, and the general information transmission is not transparent. Ref. [2] points out that DPOS consensus, as a widely used consensus mechanism, does not need to consume computing power to generate blocks, which has the advantage of saving resources. Ref. [3,4] points out that there is an imbalance of node status in DPOS consensus, which reduces the enthusiasm of ordinary nodes to participate in consensus and affects the overall performance of consensus algorithm. Therefore, an algorithm to improve the enthusiasm of nodes in DPOS consensus is needed, so as to improve the consensus performance.

At present, a variety of methods are proposed to improve the positivity of nodes. The node Comprehensive Election algorithm (Composition.Election-DPOS,CE-DPOS) was proposed in Ref. [5]. Selecting appropriate agent nodes improves the enthusiasm of nodes. Reputation value in the Ref. based on node classification algorithm (Reputation) ClassificationDPOS, RC - DPOS), preferred node as an agent, high Reputation value, encourage node right to vote, to promote its Reputation values. Thus, it improves the enthusiasm of nodes. Ref. Diao proposed a method to evaluate the performance of blockchain. It is pointed out in the Ref. that configuring an appropriate block size is beneficial to reduce the system delay.

The above DPOS consensus optimization algorithm introduces a node scoring mechanism to improve the enthusiasm of nodes. Before the consensus starts, pairwise scoring between nodes needs to be performed. Let the number of nodes be, and the node scoring mechanism requires an additional network overhead of size be, resulting in a large network overhead. This kind of optimization algorithm can not run when the network bandwidth of nodes is small. Therefore, this paper proposes a weighted Index Dynamically Balancing Algorithm (IDB). The algorithm transforms the network configuration optimization problem into the problem of solving the overall optimal solution of multiple indicators. The dynamic equilibrium search algorithm was used to solve the problem, and the solution results were used to optimize the configuration of the blockchain network to improve the enthusiasm of nodes. Simulation results show that compared with the original DPOS algorithm, the proposed algorithm has better performance indicators in delay and throughput. At the same time, compared with CE-DPOS algorithm and RC-DPOS algorithm, the network overhead of the algorithm operation is reduced. The main contributions of this paper are as follows:

1) A new algorithm optimization strategy is proposed to avoid the problem that the existing optimization algorithms are mainly based on node scoring and voting, which introduces additional communication overhead. This paper effectively reduces the communication overhead of DPOS algorithm, making the algorithm more suitable for edge networks.
2) The concept of weighted index is introduced, which can adjust the weight of each index according to the specific situation, so that the algorithm can adapt to different situations and improve the adaptability of the algorithm;
3) The optimization algorithm based on node scoring or evaluation is avoided, and the situation that nodes collude with each other in scoring is avoided, which makes the scoring cannot reflect the real situation of nodes and leads to performance fluctuations. The performance of the algorithm is more stable.

II. RELATED WORK

The process of DPoS mechanism does not rely on any computing power, and it is a good solution to the waste of computing resources in PoW mechanism and the centralization of interests that may occur in PoS mechanism. This mechanism significantly reduces the number of nodes involved in block generation and block verification, and can achieve consensus verification at the second level. In this mechanism, the nodes are not in the direct mechanism, but use the proxy nodes to act as witness nodes in turn to keep accounts. But the ordinary nodes who elect the proxy nodes often have the phenomenon of not active or dishonest election, which makes the election cycle too long and reduces the overall performance.

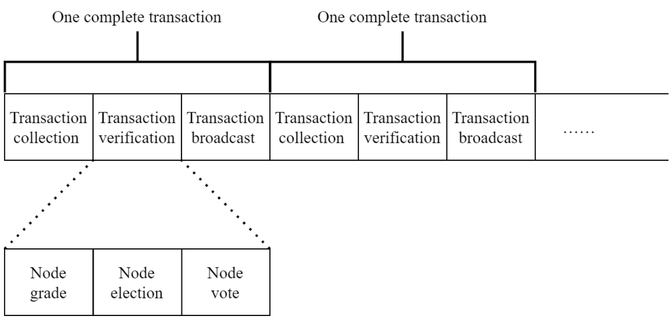

Fig. 1. The illustration of DPOS consensus optimization.

In order to solve this problem, many scholars have further optimized the DPoS consensus algorithm from the perspective of optimizing agent node selection. As shown in Figure 1, a complete transaction can be divided into three phases: transaction collection phase, transaction verification phase, and block broadcast phase. The three stages cycle continuously to complete the transaction. Among them, the transaction verification phase can be further divided into three stages: verification node scoring stage, verification node selection stage, and verification node voting stage. The existing optimization algorithms mainly focus on the verification node scoring stage and the verification node election stage in the transaction verification stage

Ref. [9] introduces the dynamic election of honest nodes on the basis of node rise and fall, which effectively reduces the probability of selecting malicious nodes, but does not significantly improve the voting activity of nodes. Ref. [10] introduces the credit value, which sets a credit value for each node, and the credit value is an important reference index for the elected principal node. After each round of consensus, the credit value will change, but the accumulation of credit value may lead to the situation that the higher the credit value, the higher the credit value. In Ref. [11], nodes in DPoS are divided into different reputation states, which may also lead to the problem that nodes with high reputation value are always in high reputation. From the perspective of optimizing voting, Ref. [12] puts forward the idea of refined voting, which includes support, abstention and negative vote, and measures the willingness of nodes to vote in three ways. Ref. [13] further optimizes the voting method by adding 10 points of support vote and 10 points of negative vote, which reflects the willingness of voting nodes in a more fine-grained way.

III. CLOUD-EDGE FUSION ARCHITECTURE

The decentralization of blockchain, as well as the immutability of tampering, can be used as a trusted platform, and people hope to combine the advantages of blockchain with edge computing. At the same time, blockchain technology, combined with consensus mechanism, smart contract for automatic transaction processing, identity authentication, information encryption and other security protection mechanisms, can not only provide a safe and reliable environment for external market transactions of the energy Internet, improve the agility of transactions, but also eliminate information barriers between internal members of the energy Internet, and improve the collaborative efficiency of the energy Internet.

Ref. [14] points out that in the process of transmission of power grid related data, there are problems such as high transmission cost and unsafe transmission. Edge computing has the advantage of reducing the load of data centers and fast response. At the same time, the blockchain platform has the advantages of high data security, tamper resistance, and no need for third party trust. Therefore, this paper combines edge computing with the advantages of blockchain. Ref. [15] proposed a way to combine edge computing with blockchain. Ref. [16] proposed a smart city model based on the hybrid architecture of blockchain and edge computing. In this model, nodes such as transportation, factory, home and medical treatment were deployed in the edge layer. The blockchain network composed of high-performance servers was deployed in the cloud. The edge processing nodes provide real-time processing with low latency and high network bandwidth to provide real-time services to the entity nodes that depend on the edge layer.

According to the position of blockchain deployment in edge computing, the edge computing blockchain fusion architecture can be divided into three categories, namely:

1) Deploying the blockchain system in the cloud network to make full use of the hardware advantages of the cloud network. This strategy is usually used in scenarios where the edge network is under-resourced. The overall architecture is divided into two parts: core network and edge network. Among them, the edge network nodes have limited storage and computing power, and are responsible for collecting and filtering data. The core network nodes have high computing and storage capabilities, and are responsible for saving and maintaining data. To ensure the integrity and availability of data.
2) Deploying the blockchain system in the edge network. This strategy is usually used in scenarios where edge network resources are abundant. The data is connected to the chain at the source of data generation to ensure the fast processing of edge network requests. The cloud is responsible for storing the data of the blockchain and auditing the data in the edge network regularly.

3) Deploying different blockchain systems by combining the characteristics of devices at different levels of the whole edge computing system. This strategy is often used in complex Internet of things scenarios, which can not only ensure the data security of the edge network, but also ensure the data security of the cloud network. However, due to the deployment of multiple blockchain systems, the storage redundancy of the overall architecture is large, which increases the storage burden of the architecture.

As shown in Table 1, the comparison is as follows:

Considering the advantages and disadvantages of various architectures, this paper adopts the architecture of deploying blockchain in the cloud to ensure the security of the overall data. To this end, a consensus algorithm with low latency and low communication overhead is needed to ensure the fast response of edge nodes and save network bandwidth resources.

The architecture flow as shown in Figure 2 can be divided into the following steps:

1) Starting from the edge network, the local server regularly records the transactions among wind power users, photovoltaic power users, energy storage users, and new energy vehicle users in the edge network;
2) After recording, the block data is packaged and compressed by the local server, and the block data is labeled (regular, emergency, safe) and uploaded to the cloud server network.
3) The cloud server verifies the received block data. After the verification passes, the block data is stored in different blockchain channels according to the different marks of the block data;
4) After the block data is received by different blockchain channels, the administrator reads the channel for configuration, constructs a multi-index weighted index on demand, and selects the best agent node from all nodes in the channel according to the index. The block data received by the channel was blocked and uploaded to the chain by the agent node to ensure the integrity and tamper resistance of the data.
5) The local server regularly downloads data from the blockchain network and compares it with local transaction data to ensure the correctness of local transaction data.

The above steps are a typical cycle of the architecture. In the whole cycle, the blockchain network ensures the security of the cloud data. Ref. [17] points out that the performance of blockchain network often becomes the performance bottleneck of the integration architecture of blockchain and edge computing, so it is necessary to optimize the performance of the blockchain network in the cloud to improve the performance of the whole architecture.

IV. WEIGHTED INDEX EQUALIZATION ALGORITHM

A. Weighted index generation algorithm

Considering that the transaction types of the whole blockchain system are diverse. In order to make the algorithm adapt to more scenarios, the algorithm is divided into two parts: the transaction sorting part and the weighted index generation part. Firstly, the sorting algorithm was used to estimate the processing time of each transaction and inform the data sender. According to the label of the transaction, the transaction was put into different blockchain channels. Then, a comprehensive index is generated in each channel, which is used by the algorithm in the next part.

Consider that transactions in a blockchain system have different priorities. The m/m/1 model with priority proposed in ref. [18] was used to deduce and analyze the delay of each transaction. It is assumed that the transaction processing rate of the blockchain proxy node is greater than the sum of the rates at which the individual nodes send transactions to the blockchain proxy node. When the priority of each transaction is the same, the delay of each transaction is analyzed, and the analysis results are as follows:

$$S_i = \frac{1}{\mu - \sum_{n=1}^{N} \lambda_i} \tag{1}$$

When the priority of each transaction is different, taking the i^{st} transaction as an example, the analysis is carried out, where R_i and B_i are the average processing time of each transaction before processing the transaction and the average remaining time of processing the transaction. The analysis result is as follows:

$$\begin{aligned} S_i =& \frac{\sum_{n=1}^{i} \lambda_i R_i}{\left(1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_i}{\mu}\right)\right)\left(1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_{i-1}}{\mu}\right)\right)} \\ &+ \frac{H_i}{1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_{i-1}}{\mu}\right)} \end{aligned} \tag{2}$$

In the m/m/1 model, the probability of each node sending a transaction to the blockchain agent node follows an exponential distribution, and $H_i = R_i = \frac{1}{\mu}$ can be obtained by putting it in (3) and simplifying it, the following can be obtained:

$$\begin{aligned} S_i =& \frac{\frac{1}{\mu}\sum_{n=1}^{i} \lambda_i}{\left(1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_i}{\mu}\right)\right)\left(1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_{i-1}}{\mu}\right)\right)} \\ &+ \frac{\frac{1}{\mu}}{1-\left(\frac{\lambda_1}{\mu}+\cdots+\frac{\lambda_{i-1}}{\mu}\right)} \end{aligned} \tag{3}$$

It adopts multi-channel structure, so that each transaction can choose different channels according to different priorities. The transactions in each channel are isolated from each other, which further improves the security of the framework

Ref. [19,20] points out that the optimization indicators of blockchain mainly include the following: transaction delay L, security n , and transaction verification cost C . In this algorithm, the following optimization objectives are considered:

TABLE I
COMPARISON OF ADVANTAGES AND DISADVANTAGES OF DIFFERENT CLOUD-EDGE-END FUSION ARCHITECTURES

Architecture type	Advantages	Disadvantages
The blockchain is deployed at the edge	Fast response to transaction requests at the edge	The security of data in the cloud cannot be guaranteed
The blockchain is deployed at the cloud	Make full use of the high computingpower resources of the cloud, and ensure the security of data	The response to transaction requests on the edge side is slow
Blockchain is deployed both at the edge and in the cloud	Can respond quickly to transaction requests at the edge and ensure the security of cloud transactions	The deployment is complex, and the simultaneous deployment of blockchain at the edge end and the cloud leads to large system redundancy and increased storage pressure

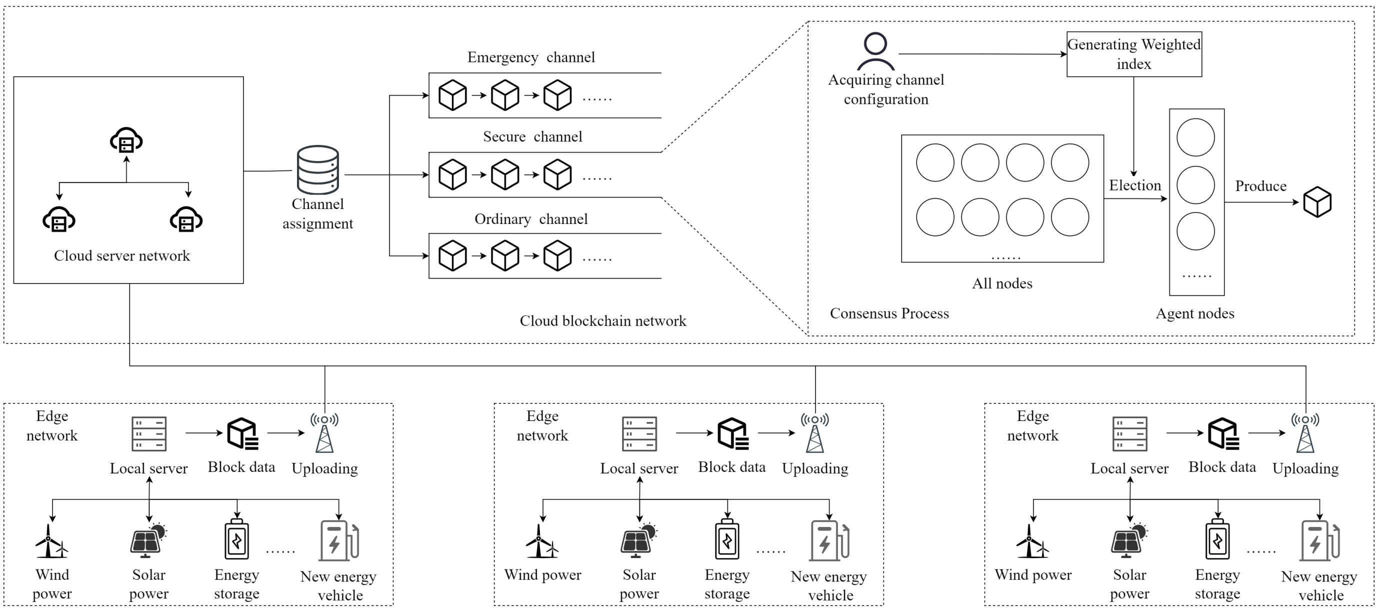

Fig. 2. The illustration of the edge computer and blockchain fusion architecture.

transaction delay, security, and transaction verification cost. The calculation formulas of the above optimization objectives are given respectively.

Transaction latency: By decomposing the various steps of a transaction, it can be concluded that the latency of a transaction is composed of the following factors :

1) The transmission of an unverified block from a normal node to a verified node;
2) Block verification;
3) Comparison between the verification result broadcast and the verification node;
4) Verification nodes transmit verification results to ordinary nodes.

Summing up the above, namely:

$$L = \frac{n \cdot B}{r_d} + \frac{K}{x} + \psi(n \cdot B)m + \frac{O}{r_u} \tag{4}$$

Security: The security of the DPOS algorithm mainly comes from the security of the verification node. The more the number of verification nodes, the stronger the security of the blockchain network. Security is defined as the likelihood of a verification node failing. Namely:

$$\eta = \theta \cdot m^q \tag{5}$$

Transaction verification cost: In order to make the model more general, the verification node is logically regarded as not having the verification ability itself, and it needs to pay a certain price to the cloud service provider to purchase the verification service. Ref.[21] points out that the verification node and the cloud service provider will eventually reach the game equilibrium, that is, the cost paid to the verification node can at least cover the cost of purchasing the verification service from the cloud service provider. The transaction verification cost is defined as the sum of the costs spent by each verification node to provide the purchased verification resources to the cloud service. Namely:

$$C = \sum_{i=1}^{m} \rho_i \cdot T_i \tag{6}$$

Each index is summed according to the given weight to obtain the comprehensive index of the corresponding channel. Considering the different specific requirements of different channels, different weights are set for different indicators. At the same time, in order to exclude the influence of dimension, each index is standardized. The weighted index U is defined as follows.

$$U = \alpha \cdot \frac{L}{l_m} + \beta \cdot \frac{\eta_m}{\eta} + \gamma \cdot \frac{C}{c_m} \tag{7}$$

B. Dynamic balance search algorithm

When the comprehensive index of each channel is generated, the optimal value needs to be solved by solving the comprehensive index, and the corresponding channel of the blockchain is configured with the optimal value.

Firstly, the verification node expenditure constraint is considered: the cost given to the verification node for computation can cover the cost of the verification node's own purchase of verification services from the third party. Namely:

$$C_i \geq p_i \cdot T_i, \quad \forall i \in \{1, \ldots, m\} \tag{8}$$

Secondly, the constraint on the number of verification nodes is considered: the number of verification nodes used per block is between the maximum and minimum values. Namely:

$$v \leq m \leq M \tag{9}$$

Finally, consider the verification block record number constraint: the number of transactions recorded per block is between the maximum and minimum values. Namely:

$$\chi \leq n \leq N \tag{10}$$

Summing up the above constraints, the objective function and the corresponding constraints are obtained as follows:

$$\begin{cases} P : \min(U) \\ s.t. \text{ (8-10)} \end{cases} \tag{11}$$

Considering the practical meaning of m, n ,this optimization model problem belongs to the nondeterministic polynomial problem. Reference [22] points out that this kind of problem is difficult to be solved by conventional methods, and the iterative search method should be used to find the solution that satisfies the conditions. The original optimization problem is divided into two sub-problems, and the two sub-problems are independent of each other, so as to prune the problem search space. Consider the number n of transactions recorded in each block, and when n is determined, consider the number m of verification nodes required for each block. Meanwhile, is a local variable in the preparation phase, while m is a global variable. Therefore, it can be considered that m, n are independent of each other.

The specific work flow is as follows: Solve n, let m be a constant, and the process of solving n is as follows: Let the partial derivative of U with respect to n be zero, then:

$$\frac{\partial U}{\partial n} = 0 \tag{12}$$

Substituting (7) into (12), the following is obtained.

$$\frac{\partial \left[\alpha \cdot \frac{L}{l_m} + \beta \cdot \frac{\eta_m}{\eta} + \gamma \cdot \frac{C}{c_m}\right]}{\partial n} = 0 \tag{13}$$

Substituting (4-6) into (13), the following is obtained.

$$\alpha \left(\frac{B}{r_d} + \psi \cdot B \cdot m \right) - \gamma \frac{\sum_{i=1}^{m} p_i \cdot x_i}{n^2} = 0 \tag{14}$$

The solution is:

$$n^2 = \frac{\gamma \sum_{i=1}^{m} p_i \cdot T_i}{\alpha \left(\frac{B}{r_d} + \psi \cdot B \cdot m \right)} \tag{15}$$

Considering the practical significance n is non-negative, and the theoretical optimal value obtained by solving n is as follows.

$$n = \sqrt{\frac{\gamma \sum_{i=1}^{m} p_i \cdot T_i}{\alpha \left(\frac{B}{r_d} + \psi \cdot B \cdot m \right)}} \tag{16}$$

After solving the optimal value of n, the m is solved in the following steps:

1) The verification nodes are sorted according to their verification speed.
2) The iteration starts from the minimum allowed number of verification nodes, and uses (17) to calculate n and obtain the theoretical optimal value of n. Within the range of n values, the closest value is found to be n^*.
3) After adding a new verification node, the value of the optimization objective function is calculated again, and compared with the value of the original optimization objective function, whether the value of the updated optimization objective function is smaller. If the result is "yes", there is room for further optimization of the objective function, add this verification node to the selected verification node and skip to step (2); Otherwise, the objective function is optimized to the optimal state, and the verification node will be discarded, exit the loop and obtain m^*.
4) Output the calculated m^*, n^*.

The dynamic balance search algorithm and the brute force search algorithm are compared according to the number of iterations required to solve the optimal value, and the results are shown in Fig. 3.

In fig. 3, it is shown that the dynamic balance search algorithm converges faster and more stably while achieving the same minimum value as the brute force search algorithm. It can be proved that the dynamic equilibrium search algorithm is correct, and compared with the brute force search algorithm, the number of iterations required by the algorithm is smaller, and the algorithm is more efficient.

C. Algorithm analysis

For the blockchain system, the algorithm in this paper firstly sorts the transactions according to their priority, and assigns the transactions with different priorities to different blockchain channels. Then, a comprehensive target is generated in each blockchain channel. Finally, under certain constraints, the comprehensive objective was solved to obtain the optimal solution of blockchain configuration. The overall pseudocode of the algorithm is as follows:

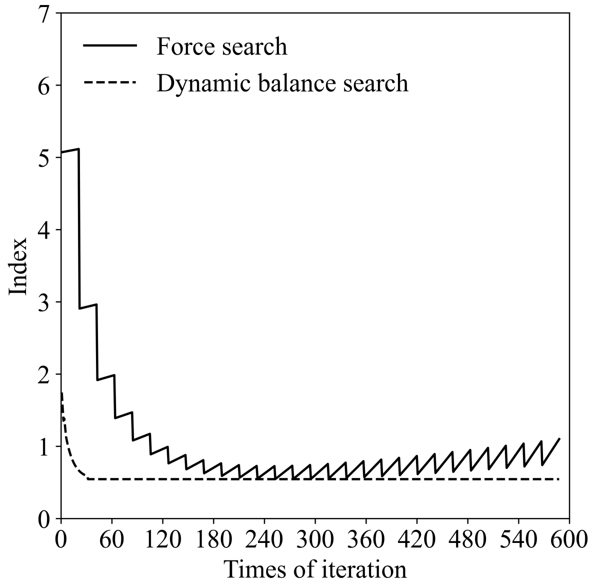

Fig. 3. The delay analysis results.

Algorithm 1 Weighted index Dynamic balance algorithm

Require: U, P

Ensure: m^*, n^*

Block validity check

Assign transactions to different blocks based on their labels Blockchain channel

Read the metric weight for that channel

Each item of the weighted index is constructed according to (4-6)

The weighted index is constructed according to (7)

while $U(m+1, n^*) < U(m, n^*)$ **do**

  Calculate the theoretical optimal value of n according to (17)

  Find the closest value in the range of n values and denoted as n^*

end while

$m^* \leftarrow m$

return m^*, n^*

In Algorithm 1, transactions with different priorities are assigned to different blockchain channels according to their priorities in one execution cycle of the algorithm to prevent interference with each other. The optimization focus of the blockchain is set by the corresponding blockchain channel, and the weights of different indicators are configured. The weighted index is obtained by accumulating the indicators after configuring the weights. By iteratively solving the weighted index, the optimal value of the parameter is solved and the blockchain channel is configured.

In order to verify the effectiveness of the algorithm, this paper uses the following three indicators to analyze the performance of the algorithm and the resources consumed when running the algorithm. The communication cost metric is used to analyze the resources consumed by the algorithm

Communication Cost: The blockchain communication cost C_c is defined as the average network resource consumed for each block confirmation. This index is used to measure the consumption of bandwidth resources in the blockchain network.

V. SIMULATION AND PERFORMANCE ANALYSIS

A. Simulation conditions

In this paper, GO language is used for simulation analysis, and the experimental environment is as follows: processor i5-12400F, 2.5GHz, 16GB, operating system windows 10 64bit. docker is used to configure the running environment to ensure the consistency of the environment. The Ethereum Geth client was used to simulate the operation of the blockchain and the hyper bench automated testing tool was used to test the performance of the blockchain. In this section, the performance of the algorithm is first analyzed in terms of delay and throughput, and then the network resources consumed by the algorithm when it is running are analyzed in terms of communication overhead.

The parameters used in the simulation analysis of the algorithm are as follows: $q = 4$, $O = 0.5Mb$, $\theta = 1$,$r_d = 1.2Mb/s$,$r_u = 1.3Mb/s$,$K = 100$,$B = 0.5kb$, the number of super nodes in the blockchain network is set to 15, the number of ordinary nodes, and the length of the blockchain can be adjusted. The comparison algorithms are: DPOS algorithm, CE-DPOS algorithm and RC-DPOS algorithm.

B. Blockchain performance analysis

In terms of algorithm performance, the proposed DPOS+IDB algorithm is compared with the original algorithm, CE-DPOS algorithm and RC-DPOS. In CE-DPOS algorithm, positive nodes are elected as agent nodes by voting to improve the performance of blockchain. In the RC-DPOS algorithm, the original selection of agent nodes from all nodes was changed to the classification based on node reputation scores, and the highest level and the second highest level were selected as the candidate nodes of the agent node to improve the performance of the blockchain.

FIG. 4 shows that although CE-DPOS algorithm, RC-DPOS algorithm and the proposed algorithm have a great improvement in delay performance compared with the original algorithm. However, when the consensus algorithm runs for a period of time, there may be nodes in CE-DPOS algorithm and RC-DPOS algorithm that use the scoring rules to score each other high and elect unsuitable nodes as agent nodes, resulting in a decrease in network delay performance. The algorithm in this paper can still effectively configure the network after running for a period of time, and the network delay performance is better.

FIG. 5 shows that the CE-DPOS algorithm, RC-DPOS algorithm and the proposed algorithm have a great improvement in throughput compared with the original DPOS algorithm. However, the throughput of the proposed algorithm is better

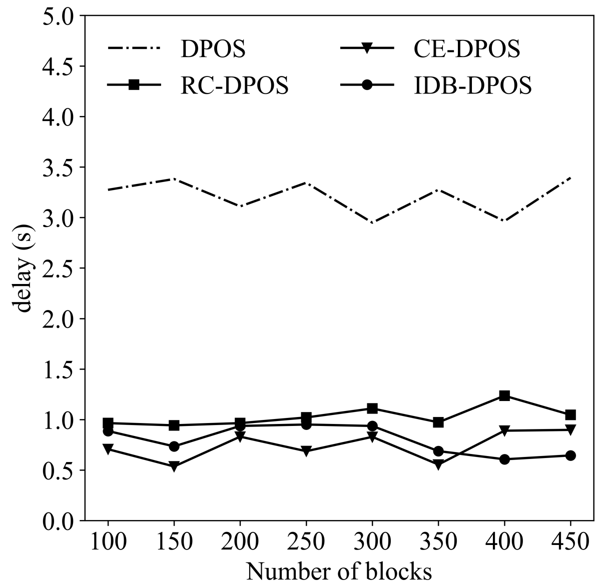

Fig. 4. The delay analysis results.

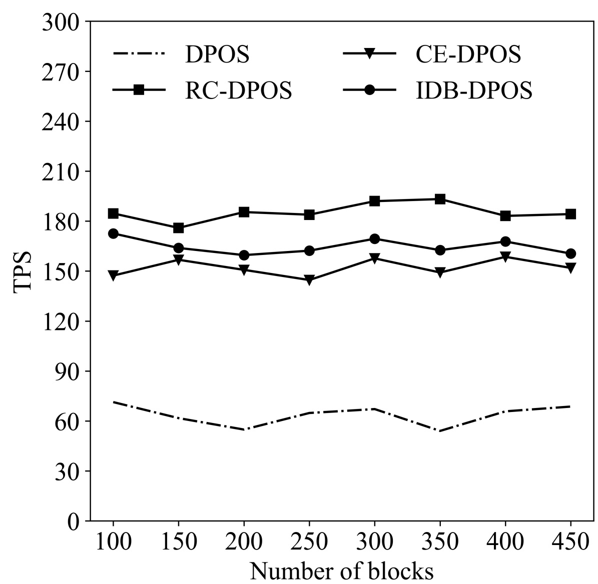

Fig. 5. The delay analysis results.

than CE-DPOS algorithm, and the stability is better than RC-DPOS algorithm.

C. Blockchain network cost analysis

In terms of network resource cost, the proposed algorithm is compared with the DPOS algorithm, CE-DPOS algorithm and RC-DPOS algorithm.

DPOS algorithm: the information that needs to be transmitted includes that ordinary nodes first transmit blocks to each agent node, and then agent nodes confirm the blocks and broadcast them to the whole network. The communication cost is:

$$C_c = j \times b + m \times b \tag{17}$$

CE-DPOS algorithm: the algorithm changes the original algorithm from the delegating nodes to generate blocks, and the nodes vote with weights to select the top m nodes with the most votes, and the communication cost includes the voting cost to re-select the proxy nodes. And the cost of broadcasting the block to the whole network after the agent node confirms it. Assuming that the size of the voting message is 0.1 of the normal block size, the communication cost is:

$$C_c = 0.1 \times j^2 \times b + j \times b + m \times b \tag{18}$$

RC-DPOS algorithm: the algorithm changed the original algorithm from the entrusted node to generate blocks, to rank nodes based on the correctness of node behavior. The level of nodes is divided into four levels from high to low. The highest first 2 levels are taken as the candidate nodes of the agent node, and within the range of candidate nodes, the score is scored pairwise. The node level is automatically given by the blockchain, which does not occupy communication overhead, and the first two level nodes account for 0.5 of the total number of nodes. The communication overhead includes the voting overhead to re-select the agent node, and the communication cost is:

$$C_c = 0.1 \times (j \times 0.5)^2 \times b + j \times b + m \times b \tag{19}$$

IDB-DPOS algorithm: the algorithm in this paper divides the blockchain into different channels, and different transactions are added to different accesses according to different needs. Through the reasonable allocation of channels, the performance of the algorithm is improved. Thus, a large amount of network resource consumption during voting is avoided. At the same time, it avoids the situation that the nodes brush each other in order to seek common interests, and ensures the stability of the algorithm. The communication overhead is similar to the original algorithm. Due to the reasonable configuration of the blockchain, the overall overhead is smaller. The communication cost is:

$$C_c = j \times b^* + m^* \times b^* \tag{20}$$

The algorithm in this paper divides the blockchain into different channels, and different transactions are added to different accesses according to different needs. Through the reasonable allocation of channels, the performance of the algorithm is improved. Thus, a large amount of network resource consumption during voting is avoided. At the same time, it avoids the situation that the nodes brush each other in order to seek common interests, and ensures the stability of the algorithm. The communication overhead is similar to the original algorithm. Due to the reasonable configuration of the blockchain, the overall overhead is smaller. The communication overhead is:

FIG. 6 shows that when the number of nodes increases, the communication overhead of CE-DPOS algorithm and RC-DPOS algorithm increases greatly. The proposed algorithm

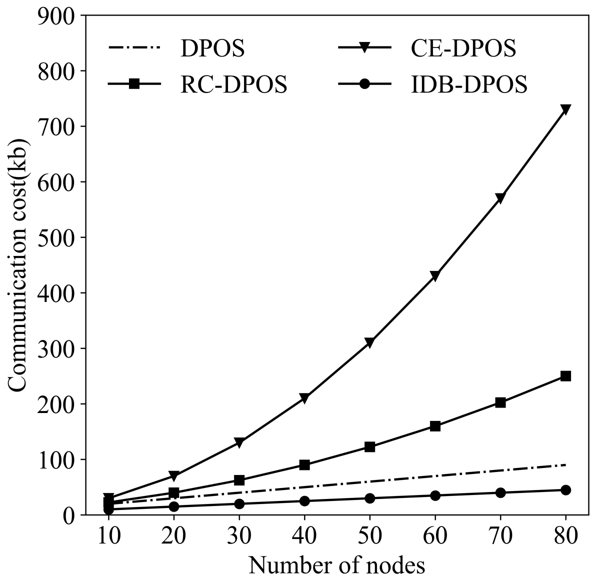

Fig. 6. The delay analysis results.

TABLE II
COMPARATION OF ALGORITHM PERFORMANCE

Algorithm	Delay	TPS	Network cost
DPOS	high	low	$\Theta(j)$
CE-DPOS	low,unstable	high,stable	$\Theta(j^2)$
RC-DPOS	low,unstable	high,unstable	$\Theta(j^2)$
IDB-DPOS	low,stable	high,stable	$\Theta(j)$

effectively reduces the communication overhead and is suitable for the situation of tight bandwidth of nodes. Figs. 6 and 7 show that the proposed algorithm has the advantages of low latency and high throughput.

As shown in Table 2, the comparison results of each index are given, and it can be found that the proposed algorithm reduces the communication overhead. In summary, the proposed algorithm is suitable for the situation that the network bandwidth resources are scarce and the delay requirement is high.

VI. CONCLUSION

In this paper, aiming at the data throughput optimization of the electricity market, a weighted index dynamic equilibrium algorithm for blockchain consensus performance optimization is proposed. Transactions are allocated to different blockchain channels through queuing theory, and then weighted indicators are obtained by weighted synthesis of different indicators. The solution results are used to configure the blockchain network, so that the algorithm in this paper has the advantages of low delay, high throughput and low communication overhead, and the overall consensus performance of the DPOS algorithm is improved. The following work will further optimize the definition of each index in the weighted index and the iterative solution algorithm.

REFERENCES

[1] Jian W, Fu B C, Wu Z T, ”Dynamic Game of Multi-domain Microgrid Power Market Based on Blockchain,” Control Engineering of China,vol. 29,no. 8, pp. 1505–1513, 2022.

[2] Jin S X,Zhang X D,Ge J G,”Overview of blockchain consensus algorithm,”Journal of Cyber Security,vol. 6,no. 2,pp. 85–100, 2021.

[3] Sharma M.,Pant S.,Kumar S. D.,”Enabling security for the Industrial Internet of Things using deep learning, blockchain, and coalitions,”Transactions on Emerging Telecommunications Technologies,vol. 32,no. 7,pp. 4137,2020.

[4] Bing W,Hui L,Li P.,”DPoS sconsensus strategy: Credit-weighted comprehensive election,”Ain Shams Engineering Journal,vol. 14,no. 2,pp.101874,2023 .

[5] Wang B,LI H L,Niu X Z.,”DPoS Consensus Algorithm Based on Comprehensive Election,”Computer Engineering,vol. 48,no. 6,pp.50–56,2022.

[6] Chen Y T,Liu F M.,”Research on improvement of DPoS consensus mechanism in collaborative governance of network public opinion,”Peer-to-peer networking and applications,vol. 15,no. 4,pp. 1849-1861,2022.

[7] Mazzoni M,Corradi A,Di N V,”Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study,”Blockchain: Research and Applications,vol. 3,no. 1,pp. 10026,2022.

[8] Shiva R P,Jinho C.”Federated Learning with Blockchain for Autonomous Vehicles: Analysis and Design Challenges,”IEEE Transactions on Communications,vol. 68,no. 8,pp. 4734–4746,2020.

[9] Feng J Y,Zhao X Y,CHEN K X,”Towards random-honest miners selection and multi-blocks creation:proof-of-negotiation consensus mechanism in blockchain networks,”Future Generation Computer Systems,vol. 105,pp. 248–258,2020.

[10] SUN Y Y,Yan B W,Yao Y,”DT-DPoS:a delegated proof of stake consensus algorithm with dynamic trust,”Procedia Computer Science,vol. 187,pp. 371–376,2021.

[11] Hu Q,Yan B W,Han Y B,”An improved delegated proof of stake consensus algorithm,”rocedia Computer Science,vol. 187,pp. 341-346,2021.

[12] Xu G X,Liu Y,Kjam P W,”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.

[13] Liu J,Xie M Y,Chen S Y,” An improved DPoSconsensus mechanism in blockchain based on PLTS for the smart autonomous multi-robot system,”Information Sciences,vol. 575,pp. 528–541,2021.

[14] Zhang H J,DING P G,Peng Y X,”State Grid Electricity Data Sharing Scheme Based on CKKS and CP-ABE,”Journal of Information Security Research,vol. 9,no. 3,pp. 58–66,2023.

[15] Bai S Z,Chen M J,”Research on Hierarchical and Sharding Blockchain for Industrial Internet,”Computer Engineering,vol. 49,no. 3,pp. 58–66,2023.

[16] Pradip K S,Jong H P,”Blockchain based hybrid network architecture for the smart city,”Future Generation Computer Systems,vol. 86,pp. 650–655,2018.

[17] Thippa R. G,Dinh C N,Praveen K R,”Blockchain for Edge of Things: Applications, Opportunities, and Challenges,”IEEE Internet of Things Journal,vol. 9,no. 2,pp. 964–988,2022.

[18] Zhang Y T,Xu X L,”Optimization analysis of M/M/1 queue with two types of priority customers,”Chinese Journal of Applied Probability and Statistics,vol. 37,no. 5,pp. 449–460,2021.

[19] Jia W K,Ze H X,Duist N,”Toward Secure Blockchain-Enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory,”IEEE Trans. Vehicular Technology,vol. 68,no. 3,pp. 2906–2920,2019.

[20] Wu Z,Huang H,Zhou Y,” secure and efficient data deduplication framework for the internet of things via edge computing and blockchain,”Connection Science,vol. 34,no. 1,pp. 1999–2025,2022.

[21] Ze H X,Shao H F,Wen B W,”Cloud/Fog Computing Resource Management and Pricing for Blockchain Networks,”IEEE Internet of Things Journal,vol. 6,no. 3,pp. 4585–4600,2022.

[22] Zhu X P,Fu Q,Wen H,”Optimal allocation model of multi-energy entity energy storage from perspective of blockchain,”Electric Power Automation Equipment,vol. 40,no. 8,pp. 47–56,2020.