applied sciences

MDPI

Article

Improved PBFT Algorithm Based on Comprehensive Evaluation Model

Wangxi Jiang, Xiaoxiong Wu, Mingyang Song, Jiwei Qin * and Zhenhong Jia *

College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
* Correspondence: jweiqin_xju@163.com (J.Q.); jzhh@xju.edu.cn (Z.J.)

Abstract: Blockchain technology is well known due to the advent of Bitcoin. With the development of recent years, blockchain technology has been widely used in medicine, digital currency, energy, etc. The practical Byzantine fault-tolerant (PBFT) algorithm is a consensus algorithm widely used in consortium blockchains. Aiming to address the problems of the PBFT algorithm, low consensus efficiency due to high communication complexity, and malicious behavior of the primary node leading to consensus failure, an improved PBFT algorithm based on a comprehensive evaluation model (TB-PBFT) is proposed. First, nodes are divided into several groups based on the multi-formation control strategy of an unmanned aerial vehicle (UAV) cluster, which significantly reduces the communication complexity. Second, a comprehensive evaluation model combining the entropy method, TOPSIS method, and Borda count is proposed, which uses the behavior of nodes as an evaluation index, and the comprehensive score of nodes is obtained according to the preferences of other nodes. Finally, the highest ranking node is selected as the primary node through the comprehensive evaluation model to ensure the security and stability of the blockchain network. We analyze TB-PBFT algorithms and compare them with other Byzantine fault tolerance algorithms. Theoretical analysis and simulation results show that the TB-PBFT algorithm can improve node scalability and fault tolerance and reduce communication complexity and view switching probability. We also prove that the comprehensive evaluation model can improve the consensus success rate of the algorithm, and the feasibility and effectiveness of the improved consensus algorithm are verified. Hence, it can be applied to the consortium blockchain system effectively and efficiently.

Keywords: PBFT; scalability; communication complexity; voting mechanism; comprehensive evaluation model

check for updates

Citation: Jiang, W.; Wu, X.; Song, M.; Qin, J.; Jia, Z. Improved PBFT Algorithm Based on Comprehensive Evaluation Model. *Appl. Sci.* **2023**, *13*, 1117. https://doi.org/10.3390/app13021117

Academic Editor: Arcangelo Castiglione

Received: 30 November 2022
Revised: 8 January 2023
Accepted: 9 January 2023
Published: 13 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Since "an electronic cash currency completely realized through peer-to-peer technology" (i.e., Bitcoin) [1] was proposed in 2008, blockchain technology has received attention and developed rapidly. The essence of the blockchain is a decentralized distributed ledger database [2], which integrates peer-to-peer (P2P) networks, consensus algorithms, cryptology principles [3], smart contracts, and other technologies. Due to its security, decentralization, tamper-proofing, and traceability, the blockchain is widely used in digital currency, finance, health care, energy, IoT, and other fields [4].

The consensus algorithm, as the core technology of the blockchain, enables participants to verify the validity of data in the blockchain system, ensuring that each node of the blockchain can record and store data according to the same uniform standard of behavior, thus reaching a consensus of the whole network. The "CAP theorem" [5] indicates that even if a feasible consensus algorithm is designed in engineering, it cannot perfectly satisfy the consistency, availability, and partition tolerance at the same time, and all nodes in the blockchain network need to follow consistent rules. The proposed theorem provides a guiding principle for the design of a consensus algorithm.

Bitcoin uses a fully decentralized proof of work (PoW) algorithm to reach a consensus through computing competition to ensure data consistency, but at the same time, it creates

a waste of resources. Proof of stake (PoS) [6] is an efficient consensus mechanism that introduces coin days to dynamically adjust the difficulty of mining to reduce computational consumption. Delegated proof of stake (DPoS) [7] uses a scheme based on node weights to elect proxy nodes, effectively reducing the number of participant nodes. However, most of these algorithms do not take Byzantine fault tolerance into account. In a distributed blockchain system, different nodes reach a consensus by exchanging information. However, malicious nodes may tamper with or fake data in the network, which affects the consensus among nodes and destroys the consistency of the distributed system. In this highly secure network environment, the design of consensus algorithms must take into account the "Byzantine general problem". Therefore, the BFT algorithm [8], which can tolerate Byzantine faults, has become a trend in the development of the consortium blockchain, but the earlier Byzantine fault-tolerant protocol overemphasized the theoretical nature of the algorithm while ignoring its practicability, and due to the high complexity of the BFT algorithm, it has not been practically applied. Castro and Liskov et al. [9] proposed a practical Byzantine fault tolerance (PBFT) algorithm in 1999, which improved the inefficiency of the BFT algorithm and was able to achieve fault tolerance for a certain number of Byzantine nodes to provide security and activity in asynchronous distributed systems. The biggest difference between blockchain systems and traditional distributed systems is that the environment is complex, and Byzantine nodes will be introduced even in strict consortium blockchains, so research on the Byzantine fault-tolerant consensus algorithm has been increasing since the birth of blockchain technology. However, there are still some problems in the PBFT algorithm, such as arbitrary selection of primary nodes, high communication complexity, poor scalability of nodes and inefficient consensus.

At present, researchers have improved PBFT algorithms from different aspects. Some algorithms change the network topology of the PBFT algorithm to star network topology. For example, HotStuff [10] uses threshold signature technology to reduce the communication complexity to $O(n)$, and the primary node collects signatures instead of using a many-to-many broadcast communication mode, but the primary node will bear a greater communication burden and easily cause network congestion. Zyzzyva [11] uses a speculative BFT algorithm that can reduce the communication overhead to near-optimal if all nodes are normal, otherwise, it will adopt a reconfirmation mechanism to reach consensus, but the decision-making with two mechanisms will lead to higher delay. In addition, the number of nodes participating in the consensus has a direct impact on the communication complexity of the PBFT consensus algorithm, so researchers try to reduce the number of nodes participating in the consensus. Algorithms such as DBFT [12] and Tendermint [13] are based on PoS to elect committees, these algorithms associate nodes with margins to solve the noninterest issues of native PoS. The T-PBFT algorithm [14] combines the EigenTrust model with a consensus algorithm to evaluate the trust level of nodes, selects some nodes with higher credit to participate in the consensus, and uses the nodes with higher quality in the network to build a consensus group instead of a single primary node. To prevent the selected nodes from being too concentrated to pose a threat to the security of the system, the Algorand algorithm [15] uses VRF [16] to randomly select a subset of network nodes as committees to ensure the unpredictability of committee members and improve the security of the system. These consensus algorithms of the "committee + PBFT" model improve the performance and scalability of the algorithm due to the reduction of the consensus network scale. However, the functions of the committee node are higher than those of the general node, and the power is too concentrated in the committee node, which violates the decentralized nature of the blockchain. The Byzantine fault tolerance algorithms are improved by increasing the number of tolerable malicious nodes in the communication network, using methods including timely detection and hardware assistance to improve the fault tolerance of the blockchain. Minbft [17] combines the BFT consensus algorithm with trusted hardware and constrains the evil node with trusted hardware so that only non-Byzantine faults can be generated by Byzantine nodes. With the introduction of trusted hardware, the fault tolerance rate of the BFT algorithm is increased from $1/3$ to $1/2$.

SDMA-PBFT [18] proposes a scalable dynamic multi-agent hierarchical PBFT algorithm, which divides nodes into several groups. Each group selects an agent and then executes the PBFT algorithm in each group. Multi-layer PBFT [19] proposes a scalable multi-layer PBFT-based consensus mechanism, which significantly reduces the communication complexity and provides guidance for the design and performance analysis of hierarchical systems. However, the instability of the primary node will lead to poor fault tolerance of the system and higher delay in the multi-layer structure. It can be seen from the previous work that the design of the consensus algorithm is a compromise and balance among efficiency, security, and stability on the premise of specific application scenarios.

However, whether it is the PoW algorithm, PoS algorithm, or BFT algorithm, the design of the blockchain consensus algorithm needs to face a complex environment, and there are many difficulties and problems to be solved. How to adaptively design a consensus mechanism for a specific performance evaluation target in combination with specific requirements and application scenarios to achieve the purpose of optimizing the algorithm [20,21] is an urgent problem to be solved. Therefore, we solve the problems of the PBFT algorithm, such as high communication complexity, arbitrary selection of primary nodes, and poor node scalability, and improve the applicability of the blockchain in large-scale scenarios. In this paper, an improved PBFT algorithm based on a comprehensive evaluation model is proposed. Specifically, the main contributions of this paper are as follows:

First, nodes are divided into several groups based on the multi-formation control strategy of a UAV cluster. Each group implements the PBFT algorithm to reach local consensus, and the primary node of each group as the representative reaches global consensus with the primary nodes of other groups.

Second, to evaluate the performance and behavior of nodes more scientifically and comprehensively, we propose a comprehensive evaluation model that combines the entropy method [22], TOPSIS method [23], and Borda count [24]. In the comprehensive evaluation model, the entropy method is used to determine the evaluation index weight of nodes, and then the TOPSIS method is used to obtain the ranking of nodes to be evaluated. Finally, the results are combined with a Borda count to obtain a comprehensive score of nodes that is more in line with the other nodes' preferences.

Third, to reduce the possibility of malicious nodes being selected as primary nodes, we combine the comprehensive evaluation model with the consensus algorithm and use the comprehensive evaluation model to select the node with the highest comprehensive score as the primary node in each group.

2. Preliminaries

2.1. Protocol Overview of PBFT

PBFT is a state machine replication algorithm, which mainly consists of three parts: The consistency protocol, view change protocol, and garbage collection. When the primary node fails, other nodes trigger the view change mechanism by timeout to ensure the activity of consensus. In addition to deleting expired logs, the garbage collection protocol is used to update the valid range of request sequence numbers. The consistency protocol is the core of the PBFT algorithm, which is mainly used to ensure the correctness and consistency of messages in the blockchain system. Assume that the total number of nodes in the consensus system is $n = 3f+1$, where f is the maximum number of Byzantine nodes that the system can tolerate. The consensus process of the PBFT algorithm can be divided into five phases: The request phase, pre-prepare phase, prepare phase, commit phase, and reply phase. The consensus process is shown in Figure 1. The algorithm process is divided into the following parts:

- Request phase: The client sends a *request* to the primary node.
- Pre-prepare phase: The primary node receives *requests* from the client, then packages the transactions sequentially into blocks and broadcasts them to all replicas.
- Prepare phase: After receiving the *pre-prepare* message, all replicas broadcast each other to verify the validity of the message.

- Commit phase: Similar to the prepare phase, in the commit phase, each node will perform the all node broadcast mode to verify the validity of the message again.
- Reply phase: The committed node sends the consensus result to the client. If the client receives more than f valid *reply* messages from all nodes, the system reaches a consensus.

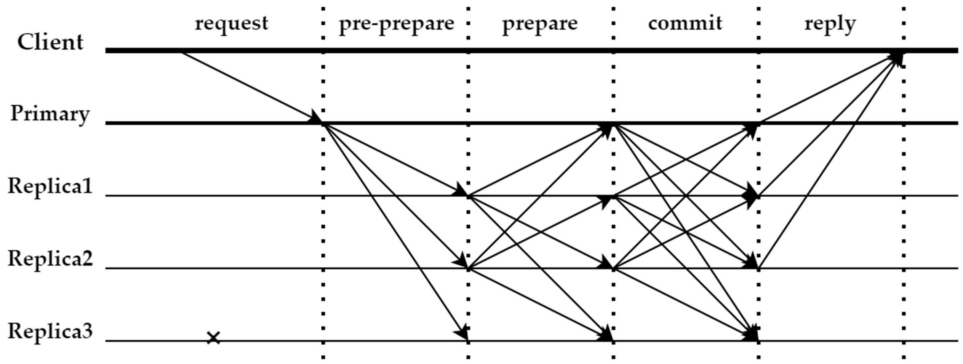


Figure 1. The consensus process of PBFT algorithm.

2.2. Entropy Weight Method

There are two commonly used methods for calculating index weights: Subjective weighting and objective weighting, and the entropy weight method avoids the interference of subjective factors and has strong objectivity [25]. Entropy is a thermodynamic concept. Shannon first introduced it to information theory to measure the uncertainty of a system [22]. The entropy weight method determines the weight according to the information entropy of the evaluation index. The smaller the information entropy of the evaluation index is, the greater the fluctuation of the data of the evaluation index, and the greater the corresponding weight. In contrast, the greater the information entropy of the evaluation index is, the smaller the corresponding weight. Therefore, compared with the subjective weighting method, the entropy weight method has higher credibility and accuracy. The calculation steps are as follows:

Step 1: Original data pre-processing. The evaluation expert K establishes an initial evaluation matrix according to the original data of the evaluation index, and the evaluation matrix X_k is as follows:

$$X_k = (x_{ij})_{m\times d} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} \end{bmatrix}. \quad (1)$$

Noteworthily, X_k represents the initial evaluation matrix of evaluation expert k, x_{ij} represents the original data of the j-th index of the i-th evaluation object, m represents the number of evaluation objects, and d represents the number of evaluation indices, where $0 \le k \le m$, $0 \le i \le m$, $0 \le j \le d$. For positive index and negative index, the standardized treatment formulas are obtained as follows:

$$y_{ij} = \frac{x_{ij} - \min\limits_{1\le i\le m}\{x_{ij}\}}{\max\limits_{1\le i\le m}\{x_{ij}\} - \min\limits_{1\le i\le m}\{x_{ij}\}}, \quad (2)$$

$$y_{ij} = \frac{\max\limits_{1\le i\le m}\{x_{ij}\} - x_{ij}}{\max\limits_{1\le i\le m}\{x_{ij}\} - \min\limits_{1\le i\le m}\{x_{ij}\}}. \quad (3)$$

where y_{ij} represents the standardized evaluation index data. The calculation process of y_{ij} normalization is as follows:

$$p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{m} y_{ij}}. \tag{4}$$

p_{ij} represents the normalized value of the evaluation object corresponding to index j, and obtains a normalized matrix P_k:

$$P_k = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1d} \\ p_{21} & p_{22} & \cdots & p_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{md} \end{bmatrix}. \tag{5}$$

where P_k denotes the standardized matrix of evaluation expert k.

Step 2: Calculate the information entropy of the evaluation index. The information entropy e_j of the evaluation index j is obtained:

$$e_j = -\frac{1}{ln(m)} \sum_{i=1}^{m} p_{ij} ln(p_{ij}). \tag{6}$$

Step 3: Calculate the entropy weight of the evaluation index.

$$\beta_j = \frac{1 - e_j}{\sum_{j=1}^{d} (1 - e_j)}. \tag{7}$$

β_j represents the entropy weight of the evaluation index j in Equation (7), and then the weight vector β_k of the evaluation index d set of the evaluation objects m by evaluation expert k is:

$$\beta_k = (\beta_1, \beta_2, \cdots, \beta_d). \tag{8}$$

2.3. TOPSIS Method

The TOPSIS method [23] calculates the approach degree between the evaluation vector of each evaluation object and the optimal target under the index system, and uses it as the basis for evaluating the quality. Therefore, the TOPSIS method is used to analyze multiple evaluation objects, calculate the closeness of each evaluation object to the optimal target, and sort the evaluation objects according to their relative closeness [26]. The specific steps are as follows:

Step 1: The evaluation matrix is constructed. The evaluation matrix S_k is obtained from the original data:

$$S_k = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1d} \\ s_{21} & s_{22} & \cdots & s_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{md} \end{bmatrix}. \tag{9}$$

Step 2: Determine the positive ideal solution and the negative ideal solution. The best scheme S^+ and the worst scheme S^- are obtained:

$$S^- = \max_{1 \le i \le m} s_{ij} = \left\{ s_1^-, s_2^-, \cdots, s_d^- \right\}, \tag{10}$$

$$S^+ = \max_{1 \le i \le m} s_{ij} = \left\{ s_1^+, s_2^+, \cdots, s_d^+ \right\}. \tag{11}$$

Step 3: Calculate the relative proximity. The relative proximity between each evaluation object and the optimal value is obtained:

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, i = 1, 2, 3, \cdots, m. \tag{12}$$

Step 4: Sort the evaluation objects. Sorted according to the relative proximity, the larger C_i is, the closer evaluation object i is to the optimal level. Therefore, we can obtain the preference voting ranking of evaluation expert k for each evaluation object.

2.4. Borda Count

The Borda count [24] is a classical voting method, which is a sorted voting mechanism, and has been employed in the elections in Nauru [27] and Slovenia [28]. Unlike the plurality voting mechanism, the Borda count not only lets voters choose the best candidates, but also to let voters rank these candidates. That is, voter k expresses the preference order for each candidate by voting. Then, the total score of each candidate is obtained according to different rankings, and the final ranking results are obtained according to the comprehensive scores. The specific steps are as follows:

Step 1: Assume the number of voters is m, the number of candidates is n, and the candidates are $x_1, x_2, \cdots, x_n$. For voter k, we can obtain:

$$r_{ab}^k = \begin{cases} 1, \text{voter } k \text{ prefers } x_a \succ x_b \\ 0, \text{voter } k \text{ does not prefer } x_a \succ x_b \end{cases}. \tag{13}$$

If $a = b$, then $r_{ab}^k = 1$, where $0 \leq a \leq n$, $0 \leq b \leq n$.

Step 2: Construct the preference matrix R_k of voter k:

$$R_k = \begin{bmatrix} r_{11}^k & r_{12}^k & \cdots & r_{1n}^k \\ r_{21}^k & r_{22}^k & \cdots & r_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1}^k & r_{n2}^k & \cdots & r_{nn}^k \end{bmatrix}. \tag{14}$$

Step 3: Calculate the Borda score of candidate a:

$$r_a^k = \sum_{b=1}^{n} r_{ab}^k. \tag{15}$$

Step 4: Construct the Borda score matrix $R^\dagger$ of all voters for all candidates:

$$R^\dagger = \begin{bmatrix} r_1^1 & r_1^2 & \cdots & r_1^m \\ r_2^1 & r_2^2 & \cdots & r_2^m \\ \vdots & \vdots & \ddots & \vdots \\ r_n^1 & r_n^2 & \cdots & r_n^m \end{bmatrix}. \tag{16}$$

Step 5: Therefore, the cumulative score of all voters for candidate a is:

$$r_a = \sum_{k=1}^{m} r_a^k. \tag{17}$$

Finally, the ranking results of candidates are obtained through cumulative scores.

3. TB-PBFT Algorithm

Blockchain uses the consensus mechanism to solve consistency problems. However, the consensus mechanism also has many shortcomings and is vulnerable to attack by malicious nodes. How to reduce the influence of malicious nodes in the system has always

been an important research topic in the field of consensus. Therefore, we propose an improved TB-PBFT algorithm based on a comprehensive evaluation model, which changes the mesh topology of the original PBFT algorithm into a grouped blockchain architecture. It is assumed that there are n nodes in the system, which are divided into n/m groups on average, and each group has m nodes. Our consensus algorithm mainly consists of four parts: A grouping strategy based on a UAV cluster, a comprehensive evaluation model, the selection of the primary node, and the consensus process of the TB-PBFT algorithm. The consensus system framework is shown in Figure 2.

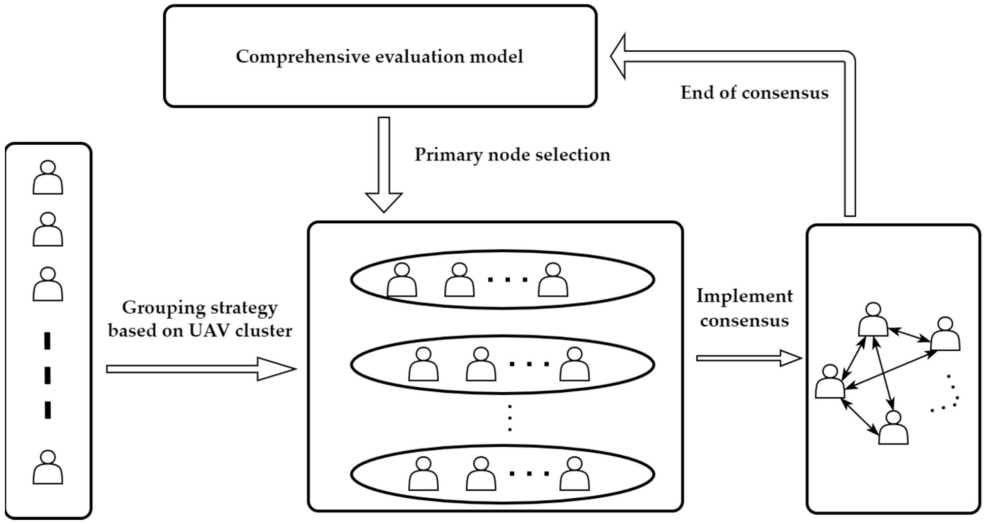


Figure 2. TB-PBFT Algorithm Framework.

3.1. Grouping Strategy Based on UAV Cluster

The UAV cluster formation with a distributed control strategy has strong adaptability, good scalability, and fault tolerance. However, the UAV cluster may face the situation of multiple targets when performing tasks. It is difficult to achieve encirclement and tracking of dispersed targets only by a single formation. At this time, it is necessary to group the clusters and complete the encirclement task through collaboration between groups [29], which can significantly improve the operational efficiency. To maintain the formation, each group can adopt the leader–follower method [30]. The leader is responsible for controlling the formation and managing the team, and other UAVs, as followers, follow the leader's movement. Multiple leaders reduce the dependence on the control center, reduce the amount of information interaction, and shorten the calculation time. Therefore, this study adopts the grouping strategy based on UAV cluster formation, and the blockchain nodes are divided into several groups according to the formation results of the UAV cluster. The grouping process is shown in Figure 3. Each formation selects a leader as the primary node, so the consensus mode of multiple primary nodes can improve the degree of decentralization of the consortium blockchain, and also reduce the impact of a single primary node on the consensus system. Each primary node participates in intergroup communication on behalf of each group, thus greatly reducing the communication complexity. However, the consensus mechanism is similar to the UAV cluster based on the leader–follower formation method, which relies too much on the leader in the system structure. Failure of the leader will easily lead to failure of the whole formation control [31]. Therefore, to reduce the probability that malicious nodes are selected as the primary node, we evaluate the reliability of the node through a comprehensive evaluation model and select the node with the highest comprehensive score as the primary node. Details are described in Section 3.3.

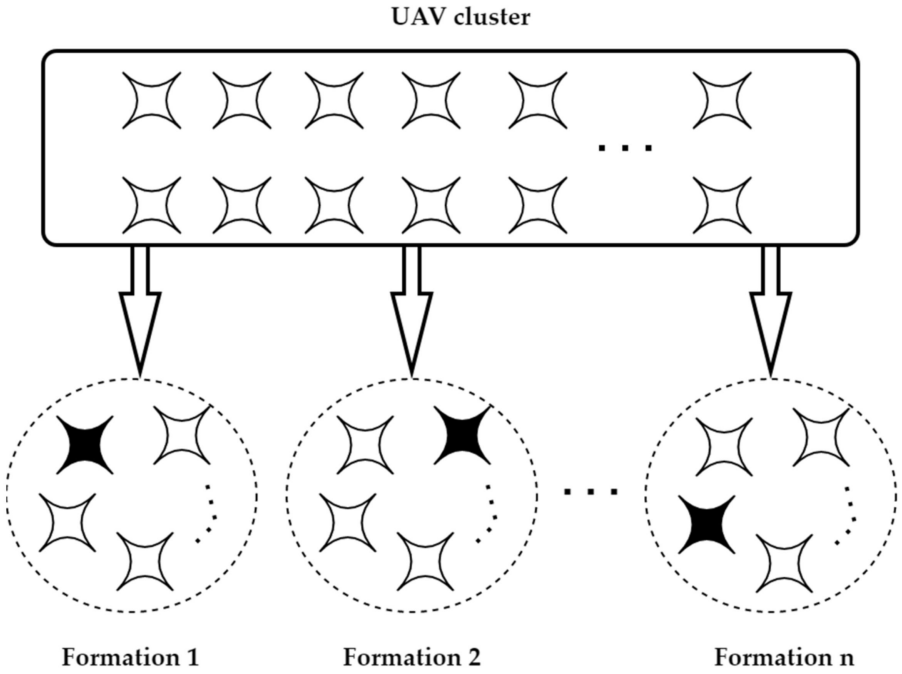


Figure 3. Grouping strategy based on UAV cluster.

3.2. Comprehensive Evaluation Model

The purpose of the comprehensive evaluation model is to select the node with good comprehensive performance and obtain the preference of most nodes. This section is divided into two parts: The construction of evaluation index and the design of the comprehensive evaluation model.

3.2.1. Construction of the Evaluation Index

To better describe our model, considering the security and computing performance of nodes, we refine and quantify the performance and behavior of nodes [32]. Some node behaviors such as communication data, availability of nodes, and historical behavior will be recorded by the system, and various behaviors of nodes will be refined and used as evaluation indices, with specific scoring standards for each index. The node reliability evaluation can be established by referring to the node behavior performance [33]. The detailed evaluation index is defined as follows:

Malicious behavior: The behavior of the consensus node directly affects system performance. A node is judged as malicious if it has tampering behavior or does not send a message within the specified time. If the behavior of the node is honest, the score is high, and if it is malicious, the score is the lowest.

Lazy behavior: The behavior of nodes with network latency or intentional latency when sending messages is defined as lazy behavior. The lower the transmission delay of the node is, the higher its evaluation score. If the node delays beyond the specified time, it is labeled as malicious behavior.

Historical performance: When malicious nodes quickly gain the trust of most nodes by using normal behaviors, or because of network failure and other reasons, honest nodes are mistaken for malicious nodes. Therefore, it is necessary to refer to the previous comprehensive scores of nodes, which requires nodes to have good behavior in each round.

Computing performance: The computing performance of nodes includes transmission capacity, storage capacity, and CPU performance.

Voting behavior: Each node is a decision node (evaluation experts), so the voting behavior of decision nodes on honest nodes or malicious nodes will also be used as scoring criteria.

3.2.2. Design of The Comprehensive Evaluation Model

The entropy weight method, TOPSIS, and other evaluation methods can achieve more efficient and accurate results through mutual combination, and have been applied in agricultural research, medical data analysis, and other scenarios [34,35]. At the same time, the generalized Khater (GK) technique and Atangana's conformable fractional (ACF) are implemented for nonlinear fractional population biology (FBP) models [36]. Not only are these techniques and theories, but also some mathematical models are crucial for many practical applications [37].

This part introduces how to combine the comprehensive evaluation model with a consensus mechanism and how the three evaluation methods play corresponding functions and roles in the comprehensive evaluation model. In this model, each node is not only an evaluation object but also a voter.

(1) Determine the index weight of nodes. For a blockchain system, the smaller the information entropy of the evaluation index of node behavior, the greater its weight in the evaluation of node behavior. In contrast, if the information entropy of the index is greater, its weight will be smaller. In the consortium blockchain, nodes are strictly screened before entering the system. Therefore, most nodes in the system are excellent and honest nodes. If some nodes have poor performance according to the evaluation index, such as malicious behavior, the weight of the index obtained by the entropy weight method is larger.

(2) Construction of the preference matrix. TOPSIS is an effective evaluation method in multi-objective decision analysis, that can objectively rank evaluation objects scientifically and reasonably. However, TOPSIS does not consider the weight of each index, which will lead to a certain research deviation between the evaluation results and the objective actual values. In this study, on the basis of the above method, we obtain the EM-TOPSIS model by combining the entropy weight method with the TOPSIS method. For voter k, the weight β_k calculated according to the entropy weight method is multiplied by the normalized matrix P_k to obtain the weighted decision-making evaluation matrix S_k. Then, the matrix S_k is used as an evaluation matrix, and the nodes are sorted according to relative proximity by the TOPSIS method. The preference matrix R_k is established according to the preference order of voters.

(3) Comprehensive score of nodes. The Borda count is a voting mechanism combining ranking order. The preference matrix R_k of voter k is obtained from the EM-TOPSIS model, and the cumulative score of all voters on each node is calculated through the Borda count. Finally, the nodes are comprehensively ranked according to the cumulative scores.

3.3. Primary Node Selection

The PBFT algorithm uses rotation to select the primary node. If the primary node has malicious behavior, it will lead to the failure of consensus and seriously affect the efficiency of the system consensus. Therefore, this study divides the consensus node into many groups, and each group selects the primary node as the group representative to participate in the global consensus. In addition, the multi-primary node approach is adopted to reduce the impact on the consensus algorithm when a single primary node is a malicious node, and the collaborative consensus of multi-primary nodes is more in line with the purpose of decentralization of the blockchain. However, this approach further reduces the probability of malicious nodes becoming primary nodes and improves the efficiency of the consensus algorithm. In this paper, the selection of the primary node in each consensus group is based on the comprehensive evaluation model, and the selected primary node is more consistent with the preferences of all nodes. The flowchart of primary node selection is shown in Figure 4. The specific steps are as follows:

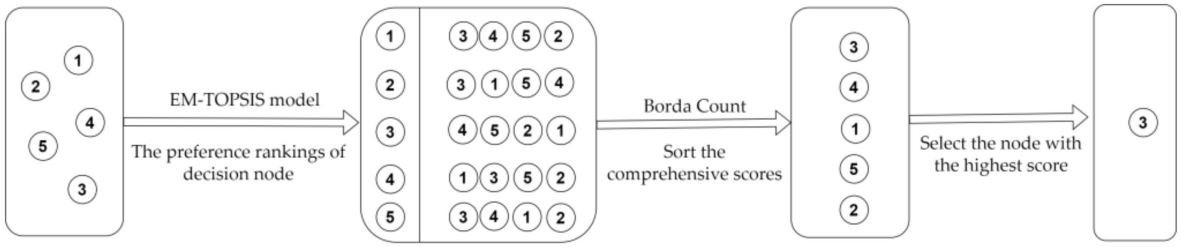


Figure 4. Primary node selection.

Step 1. In each consensus group, each node is used as a decision node to determine the index weight of other nodes by using the entropy weight method.

Step 2. The preference rankings of decision node k on other nodes are obtained according to the TOPSIS method.

Step 3. A Borda count is used to calculate the cumulative score of all decision nodes on each node, and obtain the final ranking result according to the comprehensive score.

Step 4. The node with the highest comprehensive score is set as the primary node in the group, if there are multiple nodes with the same highest scores, one is randomly selected as the primary node in the group.

Therefore, combining the comprehensive evaluation model with the consensus algorithm, and the performance and behavior of nodes as the basis for decision nodes to vote, each decision node not only needs to choose the best node, but also needs to sort all the nodes. The comprehensive score of each node is determined by all decision nodes. Therefore, the primary node selected by the comprehensive evaluation model will be safer and more reliable, thus ensuring the stability of the blockchain consensus system.

3.4. Consensus Process of the TB-PBFT Algorithm

The TB-PBFT algorithm is improved on the basis of the original PBFT algorithm. Our algorithm adopts the grouping structure, the client distributes the transaction to the primary node of each group, and the PBFT algorithm is implemented to reach local consensus in the group. Each primary node participates in global consensus on behalf of the group. Assuming that there are n nodes in the system, consensus nodes are divided into several groups, the number of groups is no less than three, and each group has m nodes. Our algorithm executes the PBFT algorithm in the group, so each group needs at least three nodes. When $n < 9$, our algorithm adopts the PBFT algorithm to complete consensus. The consensus process of the TB-PBFT algorithm is shown in Figure 5, and the detailed consensus process is as follows:

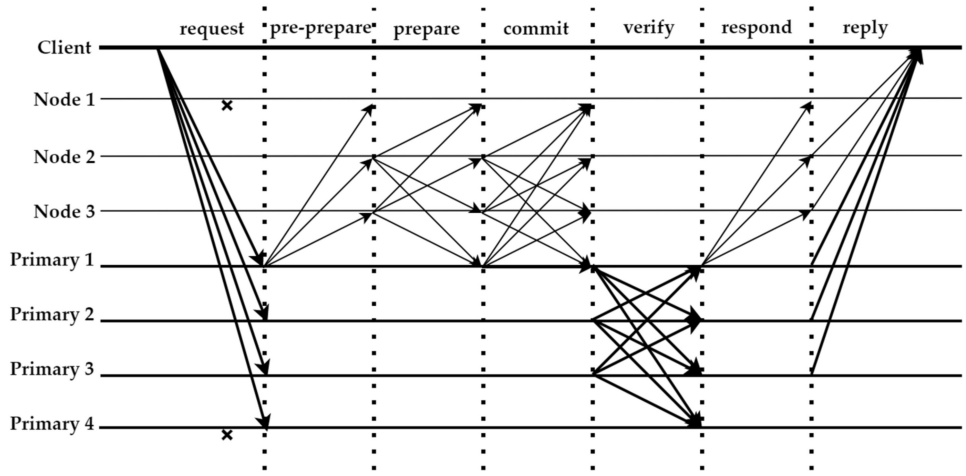


Figure 5. The consensus process of TB-PBFT algorithm.

Request: The client sends the *request* message M = <*request*, o, t, c> to the primary node of all the groups, where o is the specific operation of the request message M, t is the timestamp of the request message M, and c is the client.

Pre-prepare: After the primary node p receives the message M from the client, it multicasts the <<*pre-prepare*, v, n, $d(M)$, p>, M> message to other nodes in the group, where v is the view number, n is the sequence number, $d(M)$ is the digest of M, and p is the primary node number.

Prepare: When node i receives the *pre-prepare* message from the primary node p in the group, it verifies the message and multicasts the <*prepare*, v, n, $d(M)$, i> message to other nodes if the message is valid, where i is the node number.

Commit: When node i receives more than $2f_m+1$ the valid *prepare* message from other nodes in the group, it means that the node enters the commit phase. At this time, node i multicasts <*commit*, v, n, $d(M)$, i> messages to other nodes. Here, f_m is the maximum number of Byzantine nodes that can be tolerated in the group.

Verify: When node i receives more than $2f_m+1$ valid *commit* messages from other nodes in the group, it indicates that the group has reached a local consensus and waits for the response message from the primary node in the group, and primary node p will participate in the global consensus with other primary nodes on behalf of the group. Therefore, the committed primary node enters the verify phase by multicasting <*verify*, v, n, $d(M)$, p> messages to other primary nodes.

Respond: When the primary node p receives more than $2f_p+1$ valid *verify* messages, it indicates that the global consensus has been reached among the primary nodes. The primary node p will multicast <*respond*, v, n, $d(M)$, p> messages to other nodes in the group. Here, f_p represents the maximum number of Byzantine nodes in the global consensus.

Reply: When node i receives the *respond* message from the primary node p, it multicasts a <*reply*, v, t, r, i> message to the client if the message is valid. When the client receives more than $2f + 1$ *reply* messages from all consensus nodes, the complete consensus is reached and the transaction is added to the blockchain. Here, f represents the maximum number of Byzantine nodes in the system that can be tolerated.

4. Evaluation

In this section, we use MATLAB to analyze the performance of TB-PBFT and PBFT algorithms, and compare TB-PBFT algorithms with other BFT-type algorithms.

4.1. Communication Complexity Analysis

- The communication complexity of the PBFT algorithm:

Assuming the number of nodes of the PBFT algorithm is n, the consensus process of PBFT algorithm can be divided into five phases: The request phase, pre-prepare phase, prepare phase, commit phase, and reply phase. The communication complexity can be calculated to be $1 + (n-1) + (n-1)(n-1)+n(n-1)+n$, which is simplified to $2n^2 - n+1$. Therefore, we can obtain the communication complexity of the PBFT algorithm as follows:

$$C_1 = 2n^2 - n+1 \tag{18}$$

- The communication complexity of the TB-PBFT algorithm:

For convenience of description, it is assumed that n nodes are divided into n/m groups on average, and each group has m nodes. There are four stages of local consensus within the group: The pre-prepare phase, prepare phase, commit phase, and respond phase, and the communication complexity of local consensus is calculated as $((m-1) + (m-1)(m-1)+m(m-1) + (m-1))n/m$. There are three phases in the global consensus: The request phase, verify phase, and reply phase, and the communication

complexity of global consensus is calculated as $n/m+(n/m-1)n/m+n$. Therefore, the total communication complexity of the TB-PBFT algorithm is:

$$C_2 = \frac{n}{m}(m-1)(2m+1) + (\frac{n}{m})^2 + n \tag{19}$$

Therefore, the ratio of the communication complexity of the TB-PBFT algorithm to that of PBFT algorithm is $R = C_2/C_1$. We take m = 4, 10, and 16 to conduct simulation experiments when the total number of network nodes is less than 1000 and compare the ratio of the communication complexity of the TB-PBFT algorithm to the PBFT algorithm. The experimental results are shown in Figure 6. It can be seen from the figure that regardless of the value of m, compared with the PBFT algorithm, the communication complexity of the TB-PBFT algorithm is extremely low.

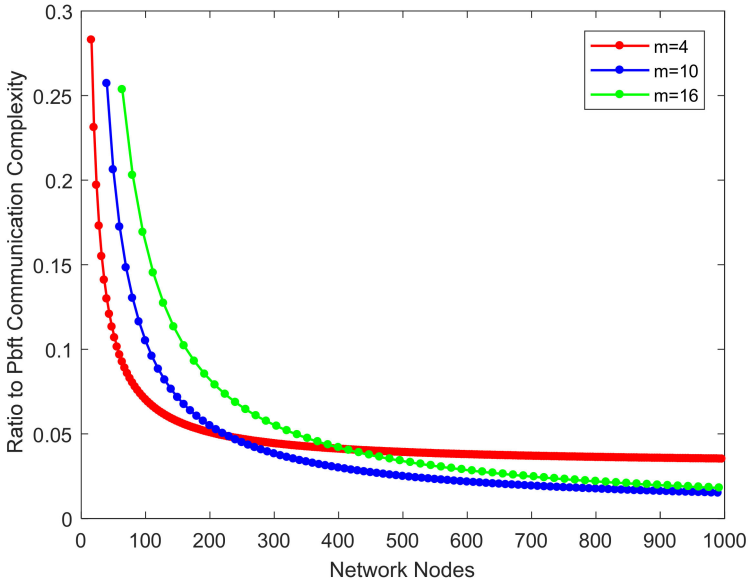


Figure 6. Ratio of the communication complexity of the TB-PBFT to the PBFT.

4.2. Fault Tolerance Analysis

In this part, we analyze and compare the fault tolerance of the TB-PBFT algorithm and PBFT algorithm, and verify the effectiveness of our comprehensive evaluation model.

4.2.1. The Maximum Number of Byzantine Fault Tolerances Analysis

The TB-PBFT algorithm improves the network topology of the original PBFT algorithm to the grouped blockchain architecture. Similarly, for convenience of description, it is assumed that n nodes are divided into n/m groups on average, and each group has m nodes. To calculate the maximum number of Byzantine fault tolerances that the system can tolerate, we assume that the Byzantine nodes are divided into optimal locations. We know that the maximum number of Byzantine groups that the system can tolerate is $B_1 = (n/m-1)/3$ (where the Byzantine group means that the primary node is a Byzantine node or the number of Byzantine nodes is more than $(m-1)/3$ in the group), and when all nodes are Byzantine nodes in the Byzantine group, the maximum number of Byzantine nodes is $m(n/m-1)/3$. However, in other groups, the primary node must be honest, and the maximum number of Byzantine nodes that can be tolerated is $(m-1)/3$ in each group, and the total number of Byzantine nodes that can be tolerated is $B_2 = (n/m-(n/m-1)/3)(m-1)/3$. Therefore, the maximum number of Byzantine nodes that the TB-PBFT algorithm can tolerate is:

$$B = B_1 + B_2 = \frac{5mn - 2n - 2m^2 - m}{9m} \tag{20}$$

We take m = 4, 10, and 16 to conduct simulation experiments when the total number of network nodes is less than 500 and compare the maximum number of Byzantine nodes of the TB-PBFT algorithm to the PBFT algorithm for different network nodes. The experimental results are shown in Figure 7. The figure shows that the fault tolerance of the TB-PBFT algorithm is better than that of the original PBFT.

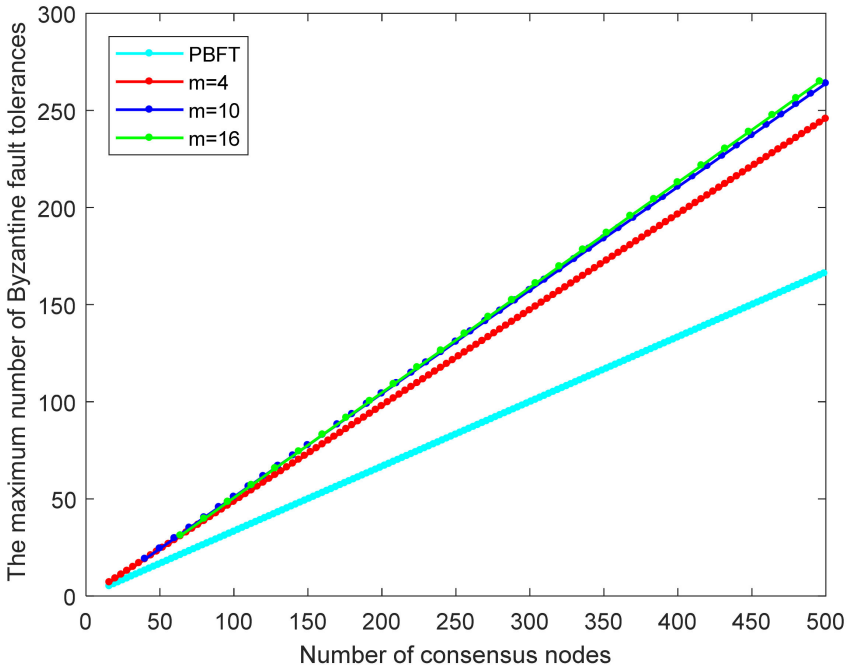


Figure 7. The maximum number of Byzantine fault tolerances of TB-PBFT and PBFT.

In addition, the TB-PBFT algorithm proposed a comprehensive evaluation model to select the excellent primary nodes. The primary node is selected by combining the scores of all the nodes in the group. The probability that the primary node is the Byzantine node is very small, so the comprehensive evaluation model can effectively improve the fault tolerance and create conditions for the system to reach the maximum number of Byzantine nodes under certain circumstances.

4.2.2. Consensus Success Rate Analysis

In order to verify the effect of the comprehensive evaluation model on the consensus system, we reflect the behavior of honest nodes and malicious nodes through evaluation indicators. Honest nodes have normal consensus behavior and vote fairly on the behavior and performance of nodes as voters, while malicious nodes will not only produce behavior that affects the consensus of the system, but also conduct malicious scoring and voting on each node to affect the final voting results. However, the selection of the primary node is determined by the common preferences of all nodes through the comprehensive evaluation model. For this reason, we verify the validity of our model through simulation experiments. In our model, each node is not only an evaluation object but also a voter (decision node), it is assumed that the evaluation matrix of each decision node is X_1, X_2, $\cdots$, X_K, respectively, and the number of consensus nodes n is 16 and 40, respectively, which is divided into 4 groups on average, so the number of nodes m in each group is 4 and 10, respectively. We conduct comparative experiments by using the comprehensive evaluation model and not using the comprehensive evaluation model. The experimental results are shown in Figure 8. It can be seen from Figure 8a,b that the comprehensive evaluation model can improve the consensus success rate of our algorithm and reduce the impact of malicious nodes when the total number of nodes n is 16 and 40, respectively.

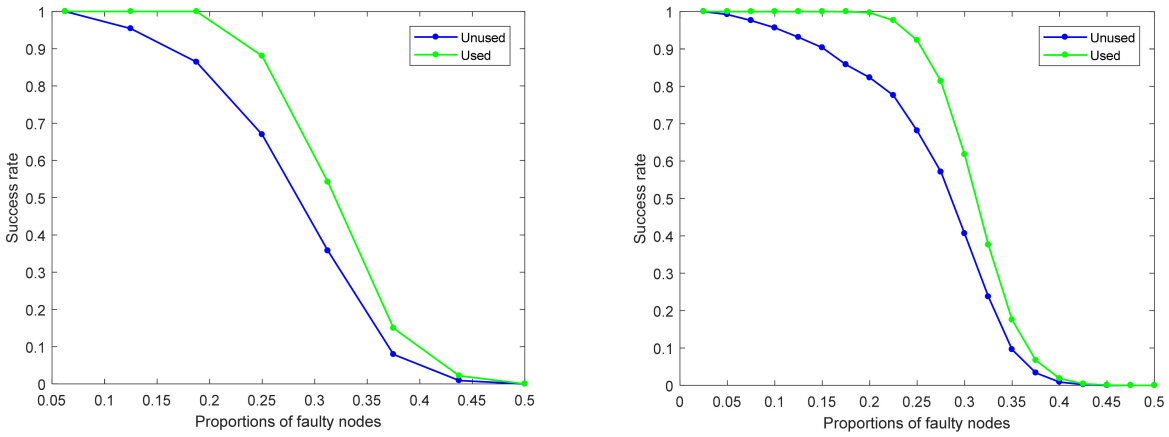


Figure 8. The influence of comprehensive evaluation model on TB-PBFT algorithm, (**a**) m = 4, n = 16, (**b**) m = 10, n = 40.

4.3. Scalability and Dynamics

As we know, the PBFT algorithm has poor scalability and nodes are not free to join and exit the blockchain system. However, our algorithm adopts a grouping strategy based on the UAV cluster, which can isolate faulty nodes or malicious nodes in local areas. When the nodes have the behavior of disturbing communication in the formation, the UAV clusters communicate with each other to replace the malicious nodes, and the original nodes will be re-formed with the new node in the formation. At the same time, when nodes need to be expanded to perform other tasks, new nodes can form a new formation and can join the current cluster after system review. Therefore, the TB-PBFT algorithm is more in line with the actual requirements of the blockchain and can adapt to large-scale application scenarios.

4.4. Comparisons of TB-PBFT with Other BFT-Type Consensus Algorithms

We compare the performance of the TB-PBFT algorithm with other BFT-type algorithms, as shown in Table 1. In terms of Byzantine fault tolerance, the BFT-type algorithms in the chart all have Byzantine fault tolerances of $(n-1)/3$. Our algorithm also has $(m-1)/3$ Byzantine fault tolerances in each group. However, whether each primary node is a Byzantine node will greatly affect the fault tolerance of the TB-PBFT algorithm. In the worst case, when more than one third of the primary nodes are Byzantine nodes, our algorithm has low fault tolerance. However, the primary node is selected based on the comprehensive evaluation model. Therefore, as shown in the previous results, the maximum fault tolerance of the TB-PBFT algorithm is $(5mn - 2n - 2m^2 - m)/9m$. The communication complexity of the PBFT algorithm is $O(n^2)$, and the asynchronous HoneyBadgerBFT [27] protocol has significant performance improvement in other aspects, but its communication complexity of $O(n^2 + n^3 \log^n)$ is higher. Other algorithms such as Tendermint, Zyzzyva, and HotStuff use different mechanisms or models to significantly reduce the communication complexity. Our algorithm has lower communication complexity than the PBFT algorithm and HoneyBadgerBFT algorithm, but slightly higher complexity than other BFT-type algorithms. Our algorithm, like the Tendermint and HotStuff algorithms, has excellent node scalability and can support a large number of nodes. Compared with other BFT-type algorithms, the TB-PBFT algorithm has a higher degree of decentralization, and the consensus approach of multi-primary nodes solves the problem of the heavy communication burden of a single primary node due to large transactions. At the same time, our algorithm also reduces the probability that the primary node is the Byzantine node, so the probability of view change is low, and additional communication consumption is further reduced. Therefore, the TB-PBFT algorithm is suitable for most scenarios of consortium blockchain.

Table 1. Comparisons of TB-PBFT with other BFT-type consensus algorithms.

	PBFT	Tendermint	HoneyBadgerBFT	Zyzzyva	HotStuff	TB-PBFT
Maximum Byzantine fault tolerance	$\frac{n-1}{3}$	$\frac{n-1}{3}$	$\frac{n-1}{3}$	$\frac{n-1}{3}$	$\frac{n-1}{3}$	$\frac{5mn-2n-2m^2-m}{9m}$
View change probability	High	High	High	High	High	Low
Communication complexity	$O(n^2)$	$O(n^2)$	$O(n^2+n^3\log^n)$	$O(n)$	$O(n)$	$O((\frac{n}{m})^2+mn)$
Scalability	Low	High	Low	Low	High	High
Degree of decentralization	Low	Low	Low	Low	Low	High

5. Conclusions

In this paper, we propose a TB-PBFT algorithm based on a comprehensive evaluation model for consortium blockchain. First, to address the large-scale distributed blockchain system, this paper is based on the formation strategy of UAV clusters. Nodes are divided into several groups and communication between nodes is limited locally. Then, to meet the requirements of real scenarios, the primary node is selected according to the performance of the node and combined with the comprehensive preferences of all nodes, which is approved by most of the nodes. Therefore, it can reduce the probability that malicious nodes are selected as the primary node to some extent and ensure the reliability of the primary node. Finally, theoretical analysis and simulation results show that the TB-PBFT algorithm can significantly reduce the communication complexity, enhance the scalability of nodes, and reduce the probability of view change. However, in the case that most of the primary nodes are malicious nodes, the fault tolerance performance of the system is greatly reduced. In today's information age, which is highly demanding for network security, how to design a consensus algorithm with high fault tolerance in the complex and severe blockchain environment needs to be considered in the future.

Author Contributions: Conceptualization, W.J.; methodology, W.J.; formal analysis, W.J., X.W., and M.S.; writing—original draft preparation, W.J.; writing—review and editing, Z.J. and J.Q.; visualization, W.J.; project administration, Z.J. and J.Q.; funding acquisition, Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Major Project of Xinjiang Uygur Autonomous Region (No. 2020A03001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. White Paper, 2008. Available online: http://www.bitcoin.org/bitcoin.pdf (accessed on 31 October 2008).
2. Michael, M.; Mills, S. The missing links in the chains mutual distributed ledger (aka Blockchain) standards. *Soc. Sci. Electron. Publ.* **2016**, *8*, 11–15.
3. Diffie, W.; Hellman, M. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*; IEEE Transactions on Information Theory; Association for Computing Machinery: New York, NY, USA, 2006; Volume 22, pp. 644–654.
4. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the IEEE International Congress on Big Data (Big Data Congress), Honolulu, HI, USA, 25-30 June 2017; pp. 557–564.
5. Gilbert, S.; Lynch, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News* **2002**, *33*, 51–59. [CrossRef]
6. Sunny, K.; Scott, N. PPcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. Available online: https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf (accessed on 28 December 2021).
7. Larimer, D. Delegated proof-of-stake (dpos). *Bitshare Whitepaper* **2014**, *81*, 85.
8. Lamport, L.; Shostak, R.; Pease, M. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [CrossRef]

9. Castro, M.; Liskov, B. Practical byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999; Volume 99, pp. 173–186.
10. Yin, M.; Malkhi, D.; Reiter, M.K. HotStuff: BFT consensus with linearity and responsiveness. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, Toronto, ON, Canada, 29 July–2 August 2019; pp. 347–356.
11. Kotla, R.; Alvisi, L.; Dahlin, M.; Clement, A.; Wong, E. Zyzzyva: Speculative byzantine fault tolerance. In Proceedings of the Twenty-First ACM SIGOPS Symposium on Operating Systems Principles, Stevenson, WA, USA, 14–17 October 2007; Volume 41, pp. 45–58.
12. Wang, Q.; Yu, J.; Peng, Z. Security analysis on dBFT protocal of NEO. In *International Confernece on Financial Cryptography and Data Security*; Springer: Cham, Switzerland, 2020; pp. 20–31.
13. Kwon, J. Tendermint: Consensus without Mining [EB/OL]. Available online: https://tendermint.com/static/docs/tendermint (accessed on 28 April 2022).
14. Gao, S.; Yu, T.; Zhu, J.; Cai, W. T-PBFT: An eigentrust-based practical byzantine fault tolerance consensus algorithm. *China Commun.* **2019**, *16*, 111–123. [CrossRef]
15. Gilad, Y.; Hemo, R.; Micali, S. Algorand: Scaling byzantine agreements for cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, 28 October 2017; pp. 51–68.
16. Micali, S.; Rabin, M.; Vadhan, S. Verifiable random functions. In Proceedings of the Symposium on Foundations of Computer Science IEEE Computer Society, Denver, CO, USA, 17–19 October 1999; pp. 120–130.
17. Veronese, G.S.; Correia, M.; Bessani, A.N.; Lung, L.C.; Verissimo, P. Efficient byzantine fault-tolerance. *IEEE Trans. Comput.* **2013**, *62*, 16–30. [CrossRef]
18. Feng, L.B.; Zhang, H.; Chen, Y. Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain. *Appl. Sci.* **2018**, *8*, 1919. [CrossRef]
19. Li, W.; Feng, C.; Zhang, L.; Xu, H.; Cao, B.; Imran, M.A. A scalable multi-layer PBFT consensus for blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1146–1160. [CrossRef]
20. Ni, X.; Zeng, S.; Han, X.; Yuan, Y.; Wang, F.Y. Organizational management using software-defined robots based on smart contracts. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 274–279.
21. Shuai, W.; Yong, Y.; Wang, X.; Li, J.; Qin, R.; Wang, F.Y. An overview of smart contract: Architecture, applications, and future trends. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 108–113.
22. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
23. Tzeng, G.H.; Huang, J.J. *Multiple Attribute Decision Making: Methods and Applications*; CRC Press: Boca Raton, FL, USA, 2011.
24. Michael, D. The borda count and agenda manipulation. *Soc. Choice Welf.* **1998**, *15*, 289–296.
25. Qiu, W.H. *Management Decision Science and Application of Entropy*; Machinery Industry Press: Beijing, China, 2002.
26. Gardas, B.B.; Heidari, A.; Navimipour, N.J.; Unal, M. A Fuzzy-Based Method for Objects Selection in Blockchain-Enabled Edge-IoT Platforms Using a Hybrid Multi-Criteria Decision-Making Model. *Appl. Sci.* **2022**, *12*, 8906. [CrossRef]
27. Reilly, B. Social choice in the south seas: Electoral innovation and the borda count in the pacific island countries. *Int. Political Sci. Rev.* **2002**, *23*, 355–372. [CrossRef]
28. Fraenkel, J.; Grofman, B. The borda count and its real-world alternatives: Comparing scoring rules in Nauru and Slovenia. *Aust. J. Political Sci.* **2014**, *49*, 186–205. [CrossRef]
29. Kuzmin, A.; Znak, E. Blockchain-base structures for a secure and operate network of semi-autonomous Unmanned Aerial Vehicles. In Proceedings of the 2018 IEEE in Ternational Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, 31 July–2 August 2018; pp. 32–37.
30. Mariottini, G.L.; Morbidi, F.; Prattichizzo, D.; Valk, N.V.; Michael, N.; Pappas, G.; Daniilidis, K. Vision-Based Localization for Leader–Follower Formation Control. *IEEE Trans. Robot.* **2009**, *25*, 1431–1438. [CrossRef]
31. Justh, E.W.; Krishnaprasad, P.S. *A Simple Control Law for UAV Formation Flying*; University of Maryland: College Park, MD, USA, 2002.
32. Swan, M. Blockchain thinking: The brain as a decentralized autonomous corporation. *IEEE Technol. Soc. Mag.* **2015**, *34*, 41–52. [CrossRef]
33. Tapscott, D. How blockchain will change organizations. *Mit Sloan Manag. Rev.* **2017**, *58*, 10–13.
34. Zhang, J.; Tang, Z.; Deng, B.; Liu, S.; Xiang, Y. A Case Study of Grassroots Water Conservancy Services Evaluation and Obstacle Factors Diagnosis Based on Gray Correlation-TOPSIS Model in Hunan Province, China. *Int. J. Environ. Res. Public Health* **2023**, *20*, 174. [CrossRef] [PubMed]
35. Mohammed, M.A. Benchmarking Methodology for Selection of Optimal COVID-19 Diagnostic Model Based on Entropy and TOPSIS Methods. *IEEE Access* **2020**, *8*, 99115–99131. [CrossRef]
36. Khater, M.M.A. Nonlinear Biological Population Model; Computational and Numerical Investigations. *Chaos Solitons Fractals* **2022**, *162*, 112388. [CrossRef]
37. Khater, M.M.A. Abundant Accurate Solitonic Water and Ionic Liquid Wave Structures of the Nanoparticle Hybrid System. *Comput. Appl. Math.* **2022**, *41*, 177. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.