RSU Cluster Deployment and Collaboration Storage of IoV Based Blockchain

Chen Chen 1 and Shi Quan 2,*

1 School of Information Science and Technology, Nantong University, Nantong 226019, China
2 School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China
* Correspondence: sq@ntu.edu.cn

Abstract: The Internet of Vehicles is increasingly becoming an indispensable platform to make interactions among vehicles, humans, and roadside infrastructures, and it is continuing to evolve with improvements on its application scenarios and service robustness. Academia and industry have also been concerned with the security issue of Internet of Vehicles (IoV). The Blockchain technology provides a new solution to improve security of the IoV, and it has drawn increased attention. Blockchain technology deals with the network model, identity authentication, trust management, and access control. However, there are insufficient studies on strategy of nodes deployment in the existing BloV (Blockchain-based IoV). Based on the principle of partition, this paper studies the Blockchain-based IoV model in which the vehicle network system consists of vehicles and Road Side Unit (RSU) nodes. The Blockchain network is formed by the RSU nodes. By optimizing the LEACH algorithm, we partition RSUs and select cluster heads in BloV, which has good scalability while maintaining a reasonable scale of Blockchain. Simulation results indicated that the improved-LEACH algorithm (I-LEACH) is more effective than the LEACH, DEEC, and the improved-DEEC algorithms (I-DEEC) with respect to the network life cycle and data transmission. Additionally, in order to reduce dependence of the storage resources from BloV, we designed an intra-cluster collaborative storage scheme by adopting three algorithms for comparison—the genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and the model quantum genetic algorithm (QGA). It revealed that the intra-cluster collaborative storage scheme is effective to ease the bottleneck of storage space demanded in BloV, and verified by the simulation experiments.

Keywords: Internet of Vehicles (IoV); Blockchain; LEACH algorithm; distributed collaborative storage; genetic algorithm (GA); particle swarm optimization algorithm (PSOA); quantum genetic algorithm (QGA)

1. Introduction

With the rapid development of digital networking and intelligence in industries, such as connected vehicles and intelligent transportation, the Internet of Vehicles (IoV) has become an essential part of global infrastructure construction. Blockchain was first mentioned in 2008 by Satoshi Nakamoto in a paper titled “Bitcoin: A Peer-to-Peer Electronic Cash System” [1]. In the more than ten years of its development, financial, medical, and other industries [2] have conducted in-depth research and diversified its application. Relevant researchers have successively proposed the combination of Blockchain and IoV technology to ensure safety of IoV [3]. However, the integration of IoV and Blockchain has some challenges to meet.

First, the Internet of Vehicles (IoV) is a super-scale and high-mobility network with colossal number of nodes and complicated categories, including various vehicle nodes and roadside devices. Vehicle nodes, in particular, have high mobility and their moving
paths are predictable. However, Blockchain belongs to a P2P network, which has the characteristics of a stable network structure and equivalent nodes. In the literature, the work focuses on dividing the nodes of IoV, constructing the network model, and including Blockchain features. The network model was proposed by integrating IoV and Blockchain technology to improve the scalability and robustness of Blockchain-based IoV.

Second, the Internet of Vehicles (IoV) is desired to be an efficient, low-latency, and robust network. However, the current Blockchain network nodes are distributed worldwide, and the encryption algorithm, consensus algorithm, and smart contract adopted by the network increase delay of data transmission, which is a challenge to meet the low-latency requirement. The network delay further increases with the expansion of the Blockchain network scale. There were investigations on efficient signature algorithms, low-latency consensus algorithms, message authentication, and identity management. However, the fundamental problem remains. Hence, an efficient network model is required to integrate IoV with Blockchain technology to achieve high transmission efficiency and low latency while ensuring network scalability, robustness, and security.

The last, The Internet of vehicles (IoV), consisting of huge number of nodes, generates massive amounts of data. How to securely transmit and store data to ensure its integrity and privacy has always been a primary concern. After integrating Blockchain technology, data security can be guaranteed using encrypted lines, traceability, and non-tampering lines. On the other hand, a strategy to handle data storage more efficiently has always been the subject of research, and this problem is intensified by introduction of Blockchain.

This paper investigates the network model aiming for integration IoV and Blockchain and seeks the solution to reduce the storage dependence of Blockchain nodes based on this model. The main contributions of this paper are:

1. A Blockchain-IoV network model based on RSU partition is proposed. By comprehensively considering the inter-cluster distance, intra-cluster distance, and network coverage, the model splits the Blockchain-based IoV into RSU partitions, manages the scale of the Blockchain, and improves the performance of consensus, storage, and transaction processing.

2. According to the above-mentioned model, we attempt to solve the data storage optimization problem of Blockchain-IoV. A mathematical model for optimized data storage from various nodes was proposed. The nodes store the data of BloV in the local area to solve the storage bottleneck problem of Blockchain in IoV application scenarios.

3. This study adopted three algorithms, including a Genetic algorithm (GA), Particle swarm optimization (PSO), and Quantum genetic algorithm (QGA), to solve the collaborative storage optimization problem. The optimal allocation scheme of collaborative storage in the area is obtained by integrating query cost and average storage occupancy rate. Finally, the computational complexity and convergence of the three algorithms are compared by simulation experiments.

2. Related Works

2.1. The System Model of BloV

The system model of the IoV-based on Blockchain (BloV) can be divided by the number of Blockchain. In [4], Gao L. et al. proposed a hierarchical resource management framework composed of a Blockchain service layer, infrastructure layer, and network layer. In addition, a monitoring system was designed to allocate system resources and improve system flexibility efficiently. Yahaya A. S. et al. [5] proposed a dual-Blockchain-based vehicle system to optimize data storage and secure communication in the Internet of Vehicles (IoV) networks and build incentive and reputation mechanisms to reduce distrust behaviors in the IoV nodes. In [6], the authors constructed a scenario of integrating edge computing and Blockchain technology into the Internet of vehicles. In this paper, RSUs are considered as ES (Edge Server), and an approximate algorithm for
random deployment of ES was applied which satisfied a criterion that the Coverage of ES to vehicle nodes ensures the reachability of messages. The simulation results showed that the two key factors determining the network connectivity are the distance of the vehicle nodes and the number of ES. However, with expansion of the scale of IoV, it is necessary to study the cooperation between ES for communication, computing, and storage to improve the utilization and scalability of ES nodes.

We established the BloV system model and proposed an optimized deployment algorithm in a cluster based BloV (CBl0V). First, the BloV system model based on the partition is studied, then, the partitioned network is established by the improved-LEACH algorithm. Additionally, a storage optimization scheme based on RSU cooperation is proposed, and the heuristic algorithm was used to solve the storage optimization problem.

2.2. RSU Cluster of IoV

Aissaoui R et al. [7] stated that clustering divides the network into different node groups called clusters. Each cluster has a member which acts as the cluster head (CH), allowing communication between cluster members and other clusters. In general, the clustering algorithm is implemented in main stages: (1) Cluster building, which includes cluster head selection and cluster formation; (2) Cluster maintenance, the purpose is to maintain the stability of clusters and recover the faulty links in the communication within or outside clusters. Clustering has several advantages: scalability advantages related to splitting the network into clusters, easier administration, and reduced coordination of messages exchanged among all nodes. We focus on RSU clustering in this paper. Senouci O. et al. [8] proposed a heuristic clustering algorithm (HCAR), which is named as HCAR-based-RSU, which used graph theory concepts characterized by node degree and adjacency matrix. They addressed distributed technologies to maintain the cluster and improve its stability. In [9], Senouci O. et al. established a new distributed clustering algorithm for IoV-based-dominant set (DS), termed DCA-DS, introducing new parameters and using greedy strategies, a new method based on a simple heuristic is developed.

Wang C. et al. [10] realized clustering by deploying RSUs at the roadside by approximating the road traffic, taking traffic flows into account at different periods, and grouping traffic locations with similar patterns into a class. The cluster’s center with the most significant local impact is considered an RSU location at a particular time interval. Different parameters were analyzed, e.g., speed and density, by considering all cluster centers at various times, and deployment strategy of RSUs is determined. This approach used the affinity propagation (AP) algorithm to form clusters as the AP had a lower complexity and higher accuracy, which is more suitable for networks in extensive deployment scale. However, the utilization rate of RSUs was low, resulting in a potential waste of resources due to unpredictable traffic patterns. Hence, some traffic prediction models and real-time transmission scheduling of RSU resources are desired to improve the resource utilization of the entire network. An efficient ERASCA core selection algorithm is designed in [11,12] by comprehensively considering battery capacity and position (maximum connectivity) for cluster head selection. However, the legitimacy and security of nodes were not investigated [13]. In [14], BET is introduced to detect and discard selfish and malicious receivers. However, in order to ensure convergence of Blockchain, it is necessary to consider how RSUs can collaborate to solve the technical bottlenecks, such as network latency and storage of Blockchain. We compared the proposed algorithm with the main algorithms for clustering, as shown in Table 1.
Table 1. Comparison of Clustering Algorithms.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Method</th>
<th>Strengths/Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>[8]</td>
<td>A heuristic clustering algorithm based on RSU</td>
<td>Considering only the transmission distance and the vehicle location, a heuristic algorithm is introduced</td>
</tr>
<tr>
<td>[9]</td>
<td>Distributed clustering algorithm for Internet of Vehicles based on DS</td>
<td>a greedy algorithm</td>
</tr>
<tr>
<td>[10]</td>
<td>Clustering based on traffic location grouping</td>
<td>The need to predict the traffic flow pattern increases the resource consumption</td>
</tr>
<tr>
<td>[11,12]</td>
<td>Based on battery capacity and position (maximum connectivity)</td>
<td>The mobility of nodes is considered, but security is not considered</td>
</tr>
<tr>
<td>This paper</td>
<td>Blockchain-based RSU partition model</td>
<td>Considering the inter-cluster distance, intra-cluster distance and network coverage, an optimization model was established to solve the partition problem, and the blockchain was introduced to ensure security</td>
</tr>
</tbody>
</table>

2.3. Distributed Storage Technology of Blockchain

Blockchain is a distributed, shared transaction ledger that can reduce financial network conflicts caused by intermediate interactions by adopting different portfolio infrastructures. With the attractive characteristics, Blockchain has broad prospect in many new applications beyond cryptocurrencies, such as medicine, supply chain management, global trade, and government services [15]. However, Blockchain works on the premise that each peer stores the entire transaction ledger as a hash chain, even if it makes no sense to non-transacting peers. Therefore, a single node incurs high storage costs [16]. Most scenarios of IoV are real-time and mobile. All nodes in IoV are connected in a complex and unstable wireless network environment, generating and exchanging vast traffic data. Therefore, when the Blockchain is too long, it is impossible or inefficient to store the complete Blockchain in each terminal device, which imposes a technical challenge and becomes a bottleneck for applying Blockchain to IoV.

In [17], Xu Z. et al. proposed a consistent cell-based storage scheme. They assumed the devices stored the entire Blockchain within a consensus cell collaboratively and were connected over a wired network. Thus, any node can check for any transaction by querying all other nodes within the same CU, restoring the entire Blockchain with almost an error-free transport. However, their scheme does not reach the optimal allocation without considering the wireless communication scenario. Li J et al. [18] designed a Blockchain-based distributed cloud storage security architecture to optimize storage capacity and network delay to solve the allocation problem. Users can divide their files into encrypted data blocks and upload them randomly to peer-to-peer network nodes, which provide free storage capacity. Third-party cloud storage solved the Blockchain storage problem but increases network delay and brings security issues, such as privacy exposure. Raman R. K. et al. [19] developed secure encoding theorems by combining key sharing, encryption, and distributed storage—reducing required storage capacity to a fraction of the initial load. Coding schemes still rely heavily on securely distributing transaction data between peers. As a result, this scheme’s recovery and maintenance costs are higher than traditional systems. We compared the proposed Blockchain-based storage optimization scheme with others, as shown in Table 2.
Table 2. Comparison of storage optimization schemes based on Blockchain.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Method</th>
<th>Strengths/Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>[17]</td>
<td>Collaborative storage within the consensus unit</td>
<td>The wireless scenario is not considered.</td>
</tr>
<tr>
<td>[18]</td>
<td>Distributed cloud storage scheme</td>
<td>The introduction of a third party increases the delay of network data transmission and increases the risk of privacy leakage.</td>
</tr>
<tr>
<td>[19]</td>
<td>Coding-based distributed storage</td>
<td>The decoding recovery and maintenance of data increase the consumption.</td>
</tr>
<tr>
<td>This paper</td>
<td>Cooperative storage in RSU based on Blockchain</td>
<td>Considering the wired and wireless network environment, the cooperative sharing in the area ensures the security and privacy of data.</td>
</tr>
</tbody>
</table>

In addition, since the data stored by individual nodes varies with dynamic distributed storage schemes, the blocks must communicate to each other in a point-to-point manner rather than broadcast, as in traditional Blockchain systems. This process undoubtedly increases the cost of communication in the network. We propose an intra-cluster collaborative storage scheme to solve the excessive demand from the Blockchain expansion whereas reduce network communication costs.

3. The Systematic Models

In order to apply Blockchain to IoV, the past work focused on the network model, identity authentication, trust management, and access control. In which structural model was established, but lack of information on deployment details of Blockchain nodes, including the number and location of nodes and the specific role of RSUs. In this section, the systematic models for deploying RSU clusters and developing interregional collaborative strategy to store the local data.

3.1. The System Model of CBloV

This subsection mainly discusses the system model based on IoV, as shown in Figure 1. Vehicles and RSU nodes are the indispensable parts of the IoV system and the basis for V2X communication. Therefore, controlling and securing the vehicle and RSU nodes is essential. Therefore, we integrate IoV with Blockchain to improve the security of data sharing, identity authentication, and node trust management in IoV. However, with the increasing number of vehicles and RSUs nodes, the system scale keeps expanding, leading to a large-scale Blockchain.

Figure 1. The system model of CBloV.

Partition is a prime instrument to deal with large-scale Blockchains. The CBloV system model was proposed to make partitioning analysis. We clustered RSU nodes
according to their locations, selected the appropriate cluster head, then used it to provide Blockchain services. The main advantage of CBlOv is that it can effectively control the size of the Blockchain and yields good scalability for the network without compromising system performance. In addition, the collaborative modes between clusters in the proposed model required data transmission only between local clusters hence a low latency and achieved improvements on storage space utilization and overall throughput.

We formed the CBlOv model with three layers:

1. **BIoV device layer**: consists of vehicles with On Board Units (OBU), and each cluster of vehicles is controlled by an RSU within its communication range. Vehicles form different clusters according to their driving directions and geographical locations. When a vehicle joins a cluster, RSU must control and complete vehicle safety authentication. It also needs to control the vehicle to create, join, exit the cluster, and maintain the vehicle cluster list. When a vehicle is moved from one RSU to the next RSU within a cluster, only the RSU-maintained vehicle cluster list is required for updating.

2. **RSU Blockchain layer**: To improve the scalability of BIoV, we designed the RSU partition algorithm to split the RSU nodes and select the cluster head. The cluster head is the FN function node of the RSU Blockchain layer, which is responsible for the network routing in the partition and provides the vehicle trust management function. Ordinary RSUs are UN nodes that can generate transaction data and regional consensus and allow devices to use regional Blockchain services and provide the function of storing blocks. The RSUs within a partition constitutes sharing the same distributed ledger. Information about data transactions is stored on the Blockchain to enable transparency Blockchain. The FN node RSU similarly shares the distributed ledger of the entire RSU Blockchain layer. When a vehicle moves from one cluster to another, it requests the new RSU to join the cluster, and the normal RSU authenticates the vehicle to the cluster head. After the cluster head succeeds, it initiates a response allowing it to join the cluster.

3. **Blockchain service layer**: It mainly realizes consensus, transaction management, block management, and security trust management, among partitions.

### 3.2. Mathematical Model of RSU Blockchain Layer

In order to characterize the RSU Blockchain layer with a mathematical model, the following assumed were made:

1. RSUs had the same computing and storage resources. We considered the power consumption in the communication process and the multipath attenuation channel in the communication process between RSUs.

2. The UN in each zone between default clusters generates transaction data, and the FN (cluster head node) decides whether to create blocks and then allocates block storage. The Blocks can be assigned to multiple nodes simultaneously in our allocation problem.

3. The Blockchain’s transfer rate of block data does not change. This study only considers power consumption in the transmission process.

Based on the assumptions, we first address RSU partitioning, then describe the intra-area collaborative storage scheme.

### 3.2.1. Mathematical Model of Cluster

To intuitively describe the mathematical model of the RSU cluster, we carry out the parameter definition in Table 3.
Table 3. Parameter description of cluster head selection in the mathematical model.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{dd}$</td>
<td>Total energy loss from distance between cluster heads</td>
</tr>
<tr>
<td>$E_{ll}$</td>
<td>Loss of distance in cluster</td>
</tr>
<tr>
<td>$m$</td>
<td>Total number of nodes in a cluster</td>
</tr>
<tr>
<td>CH</td>
<td>The total number of cluster head</td>
</tr>
<tr>
<td>Coverage</td>
<td>Network coverage of cluster heads</td>
</tr>
<tr>
<td>$E_{Ts}(k,d)$</td>
<td>The energy consumption of the RSU nodes in the cluster</td>
</tr>
<tr>
<td>$\epsilon_{fs}$</td>
<td>Communication parameters for multipath fading</td>
</tr>
<tr>
<td>$\epsilon_{mp}$</td>
<td>Communication parameters in free space</td>
</tr>
<tr>
<td>$E_{\text{Round}}$</td>
<td>The total energy consumed by the network in a round</td>
</tr>
</tbody>
</table>

The mathematical model in Figure 2 comprehensively considers the typical intra-cluster distance (the distance between nodes in a cluster and cluster-head nodes), inter-cluster distance (the distance between cluster-head nodes), and the number of cluster-head nodes. We treat it as an optimization problem whose objective function represents the minimum transmission distance. Moreover, we considered the influence of the number of cluster heads in the model.

![RSU clustering scheme](image)

**Figure 2.** RSU clustering scheme.

**Definition 1.** If we consider the transmission energy consumption and transmission distance as a linear relationship, then the sum of the distance between the cluster head and the node in the cluster is shown in Figure 3, that is, the distance loss $E_{ll}$ [20] within the cluster:

$$E_{ll} = \sum_{j=1}^{K} \sum_{i=1}^{m_j} l_{ij}$$  \hspace{1cm} (1)

where $K$ represents the number of RSU cluster heads, $m_j$ represents the number of $m$ nodes contained in the $j$-th cluster head, and $l_{ij}$ represents the distance from the $i$-th node to the $j$-th cluster head.
Figure 3. Intra-cluster storage query scheme.

Then, we get the first objective function:

$$\text{Min } f_1 = E_{II}$$  \hspace{1cm} (2)

**Definition 2.** The total energy loss of distance between cluster heads is $E_{dd}$:

$$E_{dd} = \sum_{k,j=1}^{K} L_{jk}$$  \hspace{1cm} (3)

where $L_{k,j}$ denotes the distance from the $k$-th cluster head to the $j$-th cluster head. When $k = j$, $L_{k,j} = 0$.

Thus, we give the second objective function [20]:

$$\text{Min } f_2 = \omega_1 (E_{dd}) + \omega_2 \left(\frac{CH_i}{m}\right)$$  \hspace{1cm} (4)

Here, $CH_i$ denotes the number of cluster heads in each round. $\omega_1$, $\omega_2$ are predefined constant weights, and $\omega_1 + \omega_2 = 1$.

Then, we define the network coverage as $\frac{CH_i}{m}$.

**Definition 3.** Energy consumption of RSU nodes in a cluster: The energy consumed by RSU nodes when sending/receiving $K$-bit data to neighboring nodes in the distance stage [21]:

$$E_{TX}(k,d) = \begin{cases} E_{elec} * d + \varepsilon_{fs} * k * l^2, & d < d_0 \\ E_{elec} * d + \varepsilon_{mp} * k * l^4, & d \geq d_0 \end{cases}$$  \hspace{1cm} (5)

where $E_{elec}$ represents the energy required by RSU nodes in the partition to process a unit bit of data, $\varepsilon_{fs}$ is the multipath fading communication parameter, and $\varepsilon_{mp}$ denotes the communication parameters in free space. We can compute the threshold $d_0$:

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$$  \hspace{1cm} (6)

**Definition 4.** The total energy consumed by the network in a round, assuming that $M$ nodes distribute in the $M \times M$ area and each ordinary node sends one-bit data to the cluster head

$$E_{\text{round}} = 2mE_{elec} + mE_{DA} + \varepsilon_{fs} * CH_i * L^4 + m\varepsilon_{fs} * l^2$$  \hspace{1cm} (7)

where $E_{DA}$ means data aggregation to consume the energy in cluster heads. $L$ is the average distance between cluster heads, and $l$ is the average distance between cluster heads and members in a partition.
Through the above definitions of the two objective functions (2) and (4), we established the mathematical model of the RSU partitioning scheme in the Blockchain-based IoV model.

3.2.2. Mathematical Model for Intra-Cluster Collaborative Storage

Since the Blockchain stores the entire transaction ledger in a hash chain, a single node incurs significantly increased storage and network communication costs as the size of the network nodes increases. The storage resources and peer performance of nodes are limited in practice. Therefore, based on the BioV model of the RSU partition, we designed a collaborative storage optimization scheme in the RSU partition.

This section introduced the mathematical model [22] of Intra-cluster Collaborative Storage and related concepts. We explained the collaborative scheme and analyzed its feasibility to achieve optimal communication costs for queries.

Following the method from [17], a cluster node is connected to the cluster head through wireless links, and the cluster head stores the block index and the bulk hash. The query process is illustrated in Figure 3. If \( N_1 \) wants to send a query in the block \( b_3 \), (1) \( N_1 \) needs to send a query request to the cluster head (if the cluster head is not online, it directly broadcasts the query request to other nodes). (2) After receiving the query request from \( N_1 \), the cluster head sends \( N_3 \) and \( N_5 \) nodes responsible for storing \( b_3 \) and the block hash to \( N_1 \). (3) After receiving feedback from the cluster head, \( N_1 \) sends a query request to \( N_3 \) or \( N_5 \). (4) \( N_1 \) verifies the integrity and validity of \( b_3 \) by checking the block copy returned by \( N_3 \) or \( N_5 \) against the block hash returned by the cluster head at (1).

We set \( V = \{v_1, v_2, v_3, ..., v_m\} \) of nodes that store the block of set \( BC = \{b_1, b_2, b_3, ..., b_n\} \).

**Definition 5.** Let \( b_i \) represent the \( i \)-th block with a size of \( s_i, i \in \{1, ..., n\} \), \( n_j \) \( j \in \{1, ..., m\} \) means the \( j \)-th in-cluster node, and \( R(b_i) \) is the set of nodes assigned the \( i \)-th block \((0 \leq i < n) \). \( l_j \) shows the storage limit of the node. The size of Blockchain block data in the entire partition \( EBS = \sum_{i=1}^{n} s_i \). Thus, in a partition, \( \sum_{j=1}^{m} l_j \geq EBS \). We use \( R(v_j) \) to indicate that node \( v_j \) stores a series of block sets, and \( \bigcup_{j=1}^{m} R(v_j) = BC \). Then, we can have a piece of \( b_i \) distribution set \( D(b_i) = \{j|b_i \in R(v_j)\} \).

**Definition 6.** an \( m \times n \)-dimensional matrix \( C \), \( C_{jk} \) expresses the communication cost generated by node \( n_j \) inquiring block from node \( n_k \) in the cluster.

Figure 4 shows the framework of a complete cluster storage. There are five intra-cluster nodes in this cluster, and the cluster head node is responsible for managing the index of the storage block on the node and storing the bulk hash. As an example, we took \( Node_1 \) \( (N_1) \), and it stores blocks \( b_4, b_5, \) and \( b_6 \), and it needs accessing \( b_2, b_2, \) and \( b_3 \). Therefore, when \( N_1 \) wants to access data in \( b_1 \), it must get it from either \( N_2 \) or \( N_4 \), which incurs communication costs. Since the cost of block data transmission is far greater than the cost of the cluster-head node querying the index of the storage block from the cluster-head node and storing the bulk hash, we only consider the cost of data transmission between nodes after the query. To describe the mathematical model, we define specified parameters in Table 4.
Sustainability 2022, 14, 16152

Figure 4. Intra-cluster collaborative storage framework.

Table 4. Parameters of Intra-cluster Collaborative Storage model.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_i )</td>
<td>The ( i )-th block</td>
</tr>
<tr>
<td>( s_i )</td>
<td>The size of block ( b_i )</td>
</tr>
<tr>
<td>( n )</td>
<td>The number of blocks</td>
</tr>
<tr>
<td>( m )</td>
<td>Total number of nodes in a cluster</td>
</tr>
<tr>
<td>( v_j )</td>
<td>The ( j )-th intra-zone node</td>
</tr>
<tr>
<td>( x_{ij} )</td>
<td>( b_i ) whether to assign value to ( N_j )</td>
</tr>
<tr>
<td>( D_{jm} )</td>
<td>Distance from ( j )-th to ( m )-th</td>
</tr>
<tr>
<td>( F_i = { f_i } )</td>
<td>The node ( v_j ) queries the access probability of ( b_i ) within a period of time</td>
</tr>
<tr>
<td>( R({v_j}) )</td>
<td>The node ( b_i ) stores a set of blocks</td>
</tr>
<tr>
<td>( D({b_i}) )</td>
<td>The collection of distribution for the block ( b_i )</td>
</tr>
<tr>
<td>( l_j )</td>
<td>The storage limit of node ( v_j )</td>
</tr>
</tbody>
</table>

We aim to assign all blocks in a partition to nodes in the same partition, and the total query cost is minimized, which is an allocation optimization problem.

**Definition 7.** An \( n \times m \)-dimensional 0-1 matrix \( X = \{ x_{ij} \}_{1 \leq i \leq n, 1 \leq j \leq m} \) represents the variable of the storage allocation scheme, which is the variable of our optimization problem.

\[
x_{ij} = \begin{cases} 
0, & b_i \text{ not in } N_j \\
1, & \text{otherwise}
\end{cases} \quad (8)
\]

**Definition 8.** The node \( v_j \) in the partition contains \( p_j (p_j > 0, j \in [1, m]) \) frequently accessed set \( P_j \), which indicates the node \( v_j \) queries the access probability of assigned \( b_i \) within a time interval. Thus, the query cost \( [22] \) from the node \( v_j \) to block \( b_i \):

\[
E_i^j = \min \{ c_{jk} \} \times s_i \times p_j \quad k \in D(b_i) \quad (9)
\]
The total query cost can express as:

\[
Cost = \sum_{j=1}^{m} \sum_{i=1}^{n} E_i^j
\]  

(10)

To sum up, we obtain the mathematical model: \( b_i \in BC \) allocate to \( v_j \in V \) to obtain a block allocation instance \( I \):

(1) The nodes in the cluster within the partition must meet the requirement of storing at least complete block data of the Blockchain:

\[
\bigcup_{v_j \in V} R(v_j) \geq BC
\]  

(11)

(2) The data stored in a node cannot exceed its limit:

\[
\sum_{b_j \in R(v_j)} S_i \leq l_j \quad (\forall v_j \in V)
\]  

(12)

(3) The objective is to minimize the total query cost within the partition:

\[
\sum_{i=1}^{n} \sum_{j \in F_i} \min\{c_{jk}\} \times s_i \times p_j
\]  

(13)

Finally, we analyze the feasibility of the collaborative storage allocation scheme. Compared with the traditional Blockchain technology storage framework, the storage pattern in the solution has changed, and it only requires a verification mechanism to ensure that each node obtains a valid block from other nodes. For mainstream Blockchains, blocks can divide into two parts, as shown in Figure 5: the block header and the block body. The core information of the block was stored, which included timestamp, target hash, previous hash (pre-hash), and Merkel hash [18]. The block body stored detailed transaction data generated over a period and required more storage than the header. Our model assumed that all cluster head nodes must contain all blockheads because Merkel-root hashing is included in the blockheads, and it is necessary to be used to verify if the transaction data is valid in a block. In addition, the size of block header is negligible compared to that of a block. Many ordinary nodes have enough space to store the block body regarding hashing, so our scheme is feasible.

**Figure 5.** The dominant block structure of Blockchain.

### 4. Solution

#### 4.1. The Solution of the RSU Cluster

LEACH (Low energy adaptive clustering hierarchy) algorithm [21] is an adaptive clustering topology algorithm. Its running process is periodic, and the concept of “round” is defined to realize the periodicity. Each cycle will establish the phase of cluster establishment and stable data communication.

First, in the cluster establishment phase, neighboring nodes dynamically form clusters and randomly generate cluster heads.
Second, in the data communication phase, the nodes in the cluster send the data to the cluster head, and the cluster head performs data fusion and sends the results to the sink node. Since the cluster head needs to complete the data fusion and communicate with the sink node, the energy consumption is high. The LEACH algorithm can ensure that each node acts as the cluster head with equal probability so that the nodes in the network consume relatively balanced energy.

The procedure for selecting a cluster head in LEACH is as follows: a node generates a random number between 0 and 1. If this number is less than the threshold $T(n)$, it announces that it is the cluster head. In each round, if a node has already been elected as cluster head, then $T(n)$ is set to 0. The node will not be re-elected as a cluster head. Nodes will become cluster heads with probability $T(n)$, which have not been elected as cluster heads. As the number of selected nodes increases, the threshold $T(n)$ of the remaining nodes to be elected cluster head increases, and the probability of a node generating a random number less than $T(n)$ increases. So, the probability of a node being elected cluster head increases. When only one node is left unelected, $T(n) = 1$, which means that this node must be elected. We describe $T(n)$ as follows [21]:

$$T(n) = \begin{cases} \frac{P}{1 - P \cdot (r \mod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (14)$$

$P$ shows the percentage of cluster heads to the total number of nodes. We usually set a fixed value, such as 0.05. The number of selection rounds is $r$. We use $(r \mod 1/P)$ for the number of nodes elected cluster heads in this cycle. $G$ is the nodes that have not become cluster heads in the last $1/P$ round. We improved the LEACH algorithm as follows:

First, in the LEACH algorithm, $P$ is the percentage of the number of cluster heads in the number of all nodes, which is generally set to a fixed value. In the improved algorithm, we defined Coverage as the percentage of the number of cluster heads in the number of all nodes. Allocation of Weight coefficient is performed to optimize the number of cluster heads.

Second, the MAC layer mechanism based on TDMA/CDMA is adopted in the LEACH algorithm to reduce intra-cluster and inter-cluster conflicts. However, the communication distance between cluster heads was not included. Therefore, we considered the distance between intra-cluster and inter-cluster in the improved algorithms to optimize the cluster head selection, as shown in (4).

The pseudo-code of the improved-LEACH algorithm (I-LEACH) is shown in Algorithm 1, and is described as follows: (1) Initialization, the algorithm generates several nodes in a specific area through a random network generation function, and these nodes are regarded as ordinary nodes; (2) Partition and cluster head selection stage: In this stage, the algorithm adopts the idea of the heuristic algorithm. Firstly, the cluster head node is randomly selected, and then the intra-cluster distance matrix is used to obtain $f_i$ in (2) if the node is an ordinary node. If it is the cluster head node, the inter-cluster distance matrix $f_i$ is used through the preset weight values $w_1$ and $w_2$, the minimum value of the objective function (6) is obtained by comparison, and the network Coverage is calculated at the same time. (3) Energy consumption phase: we assume that the RSU node needs to consume energy $E_{TX/RX}$ when sending and receiving $k$-bits data to the adjacent node at the distance of this phase. After a node deceased due to energy consumption, the partition and cluster head will be re-elected according to step (2).
Algorithm 1: the improved-LEACH algorithm.

Input: \( n, w_1, w_2 = 1 - w_1, E_0 \)

Output: Coverage, Min F, CH

1. Initializes parameters: \( 100 \times 100 \) randomly produces 100 nodes \( n = 100, w_1 = 0.8 \)

2. District Election phase:
   
   for \( 1 \leq i \leq n \) do
   
   Initialize: All nodes are normal nodes and cluster head node is randomly selected
   
   if the nodes are ordinary nodes, and the intra-cluster distance matrix is used to obtain \( f_1 \)
   
   if the node is the cluster head node, and the weight parameter of the distance matrix between cluster and Coverage is used to obtain \( f_2 \)
   
   Jude Min F by the (4)

3. Data transfer phase
   
   for \( 1 \leq i \leq n \) do
   
   the data is sent/received and the energy consumption \( E_r \) is calculated according the (7)
   
   if \( E_0 - E_r = 0 \) into the next round “District Election phase”

4.2. Intra-Cluster Collaborative Storage Scheme

According to Section 3.2.2, we gave an abstract mathematical model of intra-cluster collaborative storage for BiOv. This section will adopt GA, PSO, and QGA to make comparison.

First, we write it in the form of 0-1 integer programming, and its objective function and constraint conditions are as follows:

\[
\min Z = \sum_{i=1}^{n} \sum_{j \in F_i} \{C_{jk}\} * S_i * p_j * x_{ij}
\]  \hspace{1cm} (15)

\[
\min F = \sum_{i,j} x_{ij} / n
\]  \hspace{1cm} (16)

s. t. \( 1 \leq \sum_{j} x_{ij} \leq m \) \hspace{1cm} (17)

\[
\sum_{b_i \in \mathcal{R}(v_j)} s_i \leq l_j
\]  \hspace{1cm} (18)

\[
\sum_{i,j} x_{ij} > n
\]  \hspace{1cm} (19)

(15) seeks the minimum value of the query cost of the allocation scheme. We use (16) to indicate usage of block storage. A single block has at least one node storage in the constraint conditions like (17). The constraint (18) means that a node stores at least one block copy and the storage upper limit of each node is \( l_j \). It is noted that block storage requires redundancy, as shown in (19).

It solves 0-1 integer programming problem with three methods: the precise algorithm [23] (i.e., dynamic programming, recursive method, backtracking, branch, and bound method, etc.), the approximate algorithm [24] (i.e., the greedy algorithm and Lagrange algorithm, etc.), and intelligent optimization algorithm [25] (i.e., the simulated annealing
algorithm, genetic algorithm, the genetic annealing evolutionary algorithm, ant colony algorithm, etc.). By using precise algorithms, we can obtain the optimal solution. However, computational and temporal complexity can grow exponentially with the number of variables, i.e., nodes and blocks in a cluster. Therefore, it is more suitable for “small-scale” networks, which means fewer nodes and blocks. The approximate algorithms can effectively obtain the approximate optimal solution with lower precision than the exact algorithms. As a more widely used algorithm, intelligent optimization algorithms can rapidly reach a high-precision approximate solution.

4.2.1. Intra-Cluster Collaborative Storage-Based GA (ICSGA)

John Holland proposed the theory of genetic algorithms in the 1960s. This algorithm simulates the principle of “better chance of survival of the fittest” proposed by Charles Darwin and has attracted extensive attention from combinatorial optimization researchers. People use GAs to solve different NP-hard problems [26].

The first step is to transform the collaborative area storage into an n × m-dimensional 0-1 matrix. All constraints must be met for any viable chromosome, as mentioned in (17)–(19); also in addition, we will remove the failed chromosome.

Second, the performance of the genetic algorithm depends mainly on the design of the crossover operator. The standard two-parent binary single-point crossover operator makes it easy to violate the constraints. There are ways to repair failed chromosomes, but they are time-consuming. As shown in Figure 6, we randomly select a segmentation point and divide the parents into two parts. We have p genes before the split point. Progeny A will inherit the pre-P gene from parent 1 and the rest of the n-p gene from parent 2. The other offspring are acquired by switching parental roles. In Algorithm 2, this paper gives the pseudo-code of Intra-cluster Collaborative Storage based GA (ICSGA).

![Figure 6](image_url) The standard two-parent binary single-point crossover operator.

**Algorithm 2:** Intra-cluster Cooperative Storage algorithm based GA.

<table>
<thead>
<tr>
<th>Input: m, n, N, gamma, M, Pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: Xp, LC1, LC2, LC3, LC4</td>
</tr>
</tbody>
</table>

1. Initializes the 0-1 matrix of n*m of decision variable X
2. generate the initial population randomly, checked the constraint conditions and discarded the unsatisfied constraints
3. Enter the iterative process
4. a single point crossover with two parents
5. select replication, and check whether the new individuals meet the constraints, eliminate the new individuals that do not meet the constraints
6. the variation probability Pm carries on the variation
7. Update Z, F and Fitness
8. Output optimal results
Finally, with the two objective functions, this paper uses the moderate function defined by the normalization of the weight gamma:

\[
\text{Fitness}(x) = \sum_{i=1}^{k} \sum_{j=1}^{q} \min \{ \text{sign} (f_i(x) - f_i(x_j)), 0 \} \cdot \gamma_i
\]  

(20)

where \( k = 2 \), \( f_1 = Z = \sum_{i,j} x_{ij} E_i^j \), \( f_2 = F = \frac{\sum_{i,j} x_{ij}}{n} \), \( \gamma_i = (0.8, 0.2) \).

4.2.2. Intra-Cluster Collaborative Storage-Based PSO (ICSPSOA)

The particle swarm optimization (PSO) algorithm is a general heuristic exploration method used to solve allocation problems \[27\], similar to GA. The PSO algorithm can perform practical search tasks through feedback and memory. This algorithm simulates the flight paths of birds and flies towards their best historical positions, which are the best global historical positions shared by all particles through multidimensional solution space. In traditional particle swarm optimization (PSO), the position vector and velocity vector values are real numbers, which cannot be applied to our collaborative storage algorithm. The allocation of blocks is a discrete two-dimensional matrix. Therefore, we can refer to the DPSO described in \[28\].

The first stage defines the collaborative storage scheme as an \( n \times m \) dimensional 0-1 matrix based on a particle swarm. Convert an \( n \times m \) dimensional matrix to an array containing \( n \times m \) 0-1 element. The \( P_{\text{best}} \) and \( G_{\text{best}} \) denote the optimized collaborative storage scheme.

Second, the velocity vector \( v_i \) is defined as the probability of taking one binary bit and the value of velocity map to the interval \([0, 1]\). The method of mapping is generally adopted known as a sigmoid function.

\[
s(v_{id}) = \frac{1}{1 + \exp(-v_{id})}
\]  

(21)

Thus, we can write the updated formula of velocity and position as follows:

\[
v_{id} = \omega \cdot v_{id} + c_1 \cdot \text{rand()} \cdot (p_{\text{best}} - x_{id}) + c_2 \cdot \text{rand()} \cdot (g_{\text{best}} - x_{id})
\]  

(22)

\[
x_{id} = \begin{cases} 1 & \text{if } \text{rand()} \leq s(v_{id}) \\ 0 & \text{otherwise} \end{cases}
\]  

(23)

where \( \omega \) is the inertia weight, \( c_1, c_2 \) mean the learning factor.

Finally, as shown in Algorithm 3, we use the same moderate function as Section 4.2.1 to implement the algorithm and its pseudo-code.

---

**Algorithm 3: Intra-cluster Cooperative Storage algorithm based PSO.**

Input: \( m, n, M, w_{\text{min}} = 0.1, w_{\text{max}} = 0.6, c_1 = 2, c_2 = 2 \)

Output: \( g_{\text{best}}, \text{Min Fitness} \)

1. Initializes \( m, n, M, w_{\text{min}}, w_{\text{max}}, c_1, c_2, \text{position, } p_{\text{best}}, p_{\text{best}}, p_{\text{best_fit}} \)

2. Enter the iterative process

   3.1 define \( w = w_{\text{max}} + (((w_{\text{min}} - w_{\text{max}}) \times (k - 1)) / (K - 1)) \);

   3.2 calculate Fitness() and the position that did not meet the constraint conditions was eliminated

   if Fitness(i) < f(P best(i)), P_best(i) = position (i)

   if Fitness(i) < f(G best(i)), G_best(i) = position (i)

   3.3 update the \( v_{id} \) and \( x_{id} \) with (21), (22), (23)

3. Output min Fitness and G_best
4.2.3. Intra-Cluster Collaborative Storage-Based QGA (ICSQGA)

In quantum computing, quantum states are used as the primary information unit. The NP problem in classical computing can be solved by quantum union computation, which utilizes the properties of superposition, entanglement, and interference of quantum states. In 1996, Narayanan et al. [29] proposed the concept of a quantum genetic algorithm (QGA) by combining quantum theory with the evolutionary algorithm for the first time. Han et al. [30] proposed a genetic quantum algorithm (GQA) in 2000 and then extended it to the quantum evolutionary algorithm (QEA) to solve combinatorial optimization problems. In this section, we applied the quantum genetic algorithm to solve the intra-area collaborative storage optimization model problem. In the genetic algorithm in Section 4.2.1, we chose the dual-parent single-point crossover operator, mutation, and other methods, which are prone to the phenomenon of many iterations, slow convergence speed, and easy to fall into local optimal. In this section, the quantum state vector expression is introduced into genetic coding, and quantum logic gates realize the evolution of chromosomes.

First, we defined the observed chromosomes N and k as the number of qubits, the dimension of the solution space, and converted the quantum form into the binary form of the problem solution.

Second, we defined the quantum logic gate function. θ is the phase of the qubit:

\[
\begin{bmatrix}
\cos(\theta_{ij}) & -\sin(\theta_{ij}) \\
\sin(\theta_{ij}) & \cos(\theta_{ij})
\end{bmatrix}
\]

Finally, we implemented the algorithm using the same modest function in Section 4.2.1, whose pseudo-code is shown in Algorithm 4.

Algorithm 4: Intra-cluster Cooperative Storage algorithm based QGA.

Input: n, N, M, gamma

Output: Xp, Min Fitness

1. Initializes |0> and |1> of n * m decision in chromosome

2. Enter the iterative process (the end condition is the number of iterations)

2.1 Observe chromosomes and convert quantum states into binary problem solutions

2.2 According to the constraint function, repair chromosomes

2.3 Evaluate the fitness function of the problem solution and select the best one from the contemporary population

2.4 According to the quantum gate update, generation of the next generation of quantum state chromosomes

2.5 Contemporary optimal and maintain optimal comparison, retain one of the best

3. Output optimal results

5. Simulation

5.1. Analysis of Simulation Results of RSU Cluster

We chose two experiments to test the final results for the partitioned cluster head selection scheme.

First, we used MATLAB software to implement the I-LEACH Algorithm through five polling rounds to simulate the optimal cluster head selection. We used a random network function to generate 100 nodes in a 100 m × 100 m area. Then, we adjusted the value of \(\omega_1 (\omega_1 \geq 0.6)\) to determine the weight optimization value between the transmission distance factor and the number of cluster heads factor (the algorithm considers the
transmission distance as a critical index to be minimized). Figures 7 and 8 show the process of five rounds of full node selection when $\omega_1 = 0.6$ and $\omega_1 = 0.8$, respectively.

**Figure 7.** The process of polling for full node selection five times when $\omega_1 = 0.6$. (a–e) represents the result of the five-rotation trainings

**Figure 8.** The process of polling for full node selection five times when $\omega_1 = 0.8$. (a–e) represents the result of the five-rotation trainings

Figure 9 compared the results between the cases using $\text{Min } f_1$ and $\text{Min } f_2$ and the number of CH cluster heads and Coverage for different values of $\omega_1$. We can see that the Coverage values are the same when $\omega_1 = 0.7$ and $\omega_1 = 0.8$, but the values of $\text{Min } f_1$ and $\text{Min } f_2$ are more optimal when $\omega_1 = 0.8$. Therefore, in the cluster head selection model, when
our objective function $\omega_1$ is set to 0.8, it can obtain the optimal cluster head selection scheme.

**Figure 9.** The values of $\omega_1$ and $\min f_1$, $\min f_2$, CH, and Coverage. (a) indicates the relationship between $\omega_1$ and the number of full nodes, (b) is the relationship between $\omega_1$ and $\min f_1$ and $\min f_2$, (c) shows the relationship between $\omega_1$ and Coverage.

Second, we have carried out four algorithms in the network lifetime and data transmission comparison experiments: LEACH, DEEC, I-DEEC [31], and the I-LEACH algorithm. We assumed a randomly generated network of 100 RSUs, conducted 5000 rounds, and sent 4000 bits of data to get the final experimental data. In Figure 10, Leach represents the original algorithm, and the I-LEACH algorithm expresses the improved algorithm. In Figure 10a, the x-axis represents the number of polling times, and the y-axis represents the number of dead nodes that have run out of energy. From the figure, the improved algorithm performs better in network lifetime than the original algorithm, mainly because the distance between cluster heads and the network coverage of cluster heads are taken into account in the I-LEACH algorithm, which makes the network lifetime longer. In Figure 10b, the x-axis represents the number of polling, and the y-axis represents the amount of data transmitted by the cluster head node. As a result, the I-LEACH algorithm is more effective than the original algorithm in data transmission. In the I-LEACH algorithm, the network coverage ratio of cluster heads is considered, so the data transmission amount of cluster heads is further optimized. Then, we changed the number of nodes and compared the changes of the four algorithms in the network life cycle and data transmission through simulation, as shown in Figure 11. It can be seen from the figure that with the increase in the number of nodes, there is little difference between the I-LEACH algorithm and the I-DEEC algorithm in the network life cycle, as shown in Figure 11a. However, the data transmission performance is the best among the four algorithms, as shown in Figure 11b.
5.2. Analysis of Simulation Results of Intra-Cluster Collaborative Storage Solution

We mainly compared and analyzed three intra-area collaborative storage solution simulation algorithms.

First, we used Algorithm 2 to solve the collaborative storage scheme in the small-scale network area, as shown in Figure 12. We allocated six blocks among the five nodes, namely $6 \times 5$ moments and $N = 50, M = 20$. Figure 12a illustrates the convergence curve of sub-target $Z$, Figure 12b shows the convergence curve of sub-target $F$, and Figure 12c denotes the convergence curve of the average moderate function. Figure 12d indicates the convergence curve of the optimal moderate function.

Then, we expand the number of nodes and blocks and take the matrix $100 \times 50$, $N = 200, M = 20$ for analysis, as shown in Figure 13. Similarly, Figure 13a represents the convergence curve of sub-objective $Z$. Figure 13b shows the convergence curve of sub-objective $F$. Figure 13c illustrates the convergence curve of the mean moderate function. Figure 13d denotes the convergence curve of the optimal reasonable function. Finally, we obtained the optimized storage scheme after 80 iterations.
Figure 12. $6 \times 5$ matrix storage allocation genetic algorithm iteration. (a) illustrates the convergence curve of sub-target $Z$, (b) shows the convergence curve of sub-target $F$, and (c) denotes the convergence curve of the average moderation function. (d) indicates the convergence curve of the optimal moderate function.
In order to compare the performance of the three optimization algorithms, we used the genetic algorithm, particle swarm optimization algorithm, and quantum genetic collaborative storage optimization algorithm to compare the computational complexity and convergence of 100 × 50 blocks and node dimensions. (1) Convergence comparison of the three algorithms is shown in Figure 14. The convergence of the genetic algorithm using a two-parent crossover operator is better than that of the particle swarm algorithm. The convergence of collaborative memory optimization algorithms based on quantum genetics is optimal. Due to quantum parallel computation and logic gate update, the algorithm is better than selection, crossover, or mutation in genetic algorithm, and the convergence performance improves. (2) The complexity of the three algorithms is shown in Table 5. ICSQGA has low complexity in algorithms. That is because the quantum algorithm stores $N$ data in $\log 2N$ qubits, then simultaneously computes the values of $N$ functions $P$, slightly increasing the probability that the result will be optimal through quantum operations. The mathematical calculation can prove that after repeating the
above process \((\pi\sqrt{N/4})\) times, the probability of optimal data occurrence will reach the maximum [32].

Figure 14. The comparison of ICSGA and ICSPSOA in convergence. (a) means convergence comparison of ICSGA and ICSPSOA, (b) means convergence curves of ICSQGA.

Table 5. The comparison of ICSGA, ICSPSOA, and ICSQGA in computational complexity.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computation Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSGA</td>
<td>(O(m^N + MN + n))</td>
</tr>
<tr>
<td>ICSPSOA</td>
<td>(O(nM + MN^{m-n} + Mm^nn^N))</td>
</tr>
<tr>
<td>ICSQGA</td>
<td>(O(\sqrt{N}))</td>
</tr>
</tbody>
</table>

6. Summary and Prospect

Integration of IoV with Blockchain technology is a promising approach to address security concern of IoV networks. Additionally, it also brings problems, such as increased transmission delay and demand for large storage spaces. In order to deal with these problems comprehensively, after examining the characteristics of IoV nodes, this study proposed a BloV model with partitioning on RSUs. By considering the factors including the inter-cluster distance, intra-cluster distance, and network coverage, the RSU is partitioned, the cluster head is elected, and the Blockchain network is composed of cluster heads. All partitions are formed with a consensus. The proposed method controls the scale of the Blockchain and improves the performance of consensus, storage, and transaction processing. Compared with the conventional algorithms, the RSU partitioning algorithm has a higher network lifetime and volume. In addition, since the Blockchain stores the entire transaction ledger in the form of a hash chain, a single node potentially incurs significantly increased storage and network communication costs. Therefore, based on the BloV model with partitioning on RSUs, this work proposed a collaborative storage optimization scheme in the RSU partition. First, the mathematical model of collaborative storage optimization is established. Then, we applied the Genetic algorithm, Particle swarm optimization algorithm, and Quantum genetic algorithm to address the mentioned issues. Finally, the computational complexity and convergence of three kinds of short hair are compared by simulation experiments. Unfortunately, this paper only conducts theoretical analysis through simulation experiments and does not perform functional
implementation on Blockchain nodes. In the future, we will consider building Blockchain nodes to realize intra-region collaborative storage optimization schemes and introducing third parties, such as cloud service nodes, to conduct related research on storage cost and network communication budget.

**Author Contributions:** Writing—original draft, C.C.; Writing—review & editing, S.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This is a part research accomplishment of the following projects: (1) “Graduate Research innovation Project of Jiangsu Province, China (KYCX21_3087)”, which is supported by Education Department of Jiangsu Province, China; (2) “The National Natural Science Foundation of China (61771265)”, which is supported by National Natural Science Foundation of China; and (3) “The Key Science and Technology Foundation of Nantong (MS22021034),” which is supported by The Science and Technology Bureau of Nantong.

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

**References**


