

Wireless Communications and Mobile Computing

Allocation Mechanism Design and Application for Competitive Resources

Lead Guest Editor: Junwu Zhu

Guest Editors: Ting Tan and Chunpeng Ge





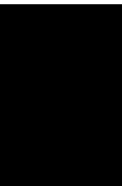
Allocation Mechanism Design and Application for Competitive Resources

Wireless Communications and Mobile Computing

Allocation Mechanism Design and Application for Competitive Resources

Lead Guest Editor: Junwu Zhu


Guest Editors: Ting Tan and Chunpeng Ge





Copyright © 2021 Hindawi Limited. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Zhipeng Cai , USA

Associate Editors

Ke Guan , China
Jaime Lloret , Spain
Maode Ma , Singapore

Academic Editors

Muhammad Inam Abbasi, Malaysia
Ghufran Ahmed , Pakistan
Hamza Mohammed Ridha Al-Khafaji ,
Iraq
Abdullah Alamoodi , Malaysia
Marica Amadeo, Italy
Sandhya Aneja, USA
Mohd Dilshad Ansari, India
Eva Antonino-Daviu , Spain
Mehmet Emin Aydin, United Kingdom
Parameshchhari B. D. , India
Kalapaveen Bagadi , India
Ashish Bagwari , India
Dr. Abdul Basit , Pakistan
Alessandro Bazzi , Italy
Zdenek Becvar , Czech Republic
Nabil Benamar , Morocco
Olivier Berder, France
Petros S. Bithas, Greece
Dario Bruneo , Italy
Jun Cai, Canada
Xuesong Cai, Denmark
Gerardo Canfora , Italy
Rolando Carrasco, United Kingdom
Vicente Casares-Giner , Spain
Brijesh Chaurasia, India
Lin Chen , France
Xianfu Chen , Finland
Hui Cheng , United Kingdom
Hsin-Hung Cho, Taiwan
Ernestina Cianca , Italy
Marta Cimitile , Italy
Riccardo Colella , Italy
Mario Collotta , Italy
Massimo Condoluci , Sweden
Antonino Crivello , Italy
Antonio De Domenico , France
Floriano De Rango , Italy

Antonio De la Oliva , Spain
Margot Deruyck, Belgium
Liang Dong , USA
Praveen Kumar Donta, Austria
Zhuojun Duan, USA
Mohammed El-Hajjar , United Kingdom
Oscar Esparza , Spain
Maria Fazio , Italy
Mauro Femminella , Italy
Manuel Fernandez-Veiga , Spain
Gianluigi Ferrari , Italy
Luca Foschini , Italy
Alexandros G. Fragkiadakis , Greece
Ivan Ganchev , Bulgaria
Óscar García, Spain
Manuel García Sánchez , Spain
L. J. García Villalba , Spain
Miguel Garcia-Pineda , Spain
Piedad Garrido , Spain
Michele Girolami, Italy
Mariusz Glabowski , Poland
Carles Gomez , Spain
Antonio Guerrieri , Italy
Barbara Guidi , Italy
Rami Hamdi, Qatar
Tao Han, USA
Sherief Hashima , Egypt
Mahmoud Hassaballah , Egypt
Yejun He , China
Yixin He, China
Andrej Hrovat , Slovenia
Chunqiang Hu , China
Xuexian Hu , China
Zhenghua Huang , China
Xiaohong Jiang , Japan
Vicente Julian , Spain
Rajesh Kaluri , India
Dimitrios Katsaros, Greece
Muhammad Asghar Khan, Pakistan
Rahim Khan , Pakistan
Ahmed Khattab, Egypt
Hasan Ali Khattak, Pakistan
Mario Kolberg , United Kingdom
Meet Kumari, India
Wen-Cheng Lai , Taiwan

Jose M. Lanza-Gutierrez, Spain
Pavlos I. Lazaridis , United Kingdom
Kim-Hung Le , Vietnam
Tuan Anh Le , United Kingdom
Xianfu Lei, China
Jianfeng Li , China
Xiangxue Li , China
Yaguang Lin , China
Zhi Lin , China
Liu Liu , China
Mingqian Liu , China
Zhi Liu, Japan
Miguel López-Benítez , United Kingdom
Chuanwen Luo , China
Lu Lv, China
Basem M. ElHalawany , Egypt
Imadeldin Mahgoub , USA
Rajesh Manoharan , India
Davide Mattera , Italy
Michael McGuire , Canada
Weizhi Meng , Denmark
Klaus Moessner , United Kingdom
Simone Morosi , Italy
Amrit Mukherjee, Czech Republic
Shahid Mumtaz , Portugal
Giovanni Nardini , Italy
Tuan M. Nguyen , Vietnam
Petros Nicolitidis , Greece
Rajendran Parthiban , Malaysia
Giovanni Pau , Italy
Matteo Petracca , Italy
Marco Picone , Italy
Daniele Pinchera , Italy
Giuseppe Piro , Italy
Javier Prieto , Spain
Umair Rafique, Finland
Maheswar Rajagopal , India
Sujan Rajbhandari , United Kingdom
Rajib Rana, Australia
Luca Reggiani , Italy
Daniel G. Reina , Spain
Bo Rong , Canada
Mangal Sain , Republic of Korea
Praneet Saurabh , India



Hans Schotten, Germany
Patrick Seeling , USA
Muhammad Shafiq , China
Zaffar Ahmed Shaikh , Pakistan
Vishal Sharma , United Kingdom
Kaize Shi , Australia
Chakchai So-In, Thailand
Enrique Stevens-Navarro , Mexico
Sangeetha Subbaraj , India
Tien-Wen Sung, Taiwan
Suhua Tang , Japan
Pan Tang , China
Pierre-Martin Tardif , Canada
Sreenath Reddy Thummaluru, India
Tran Trung Duy , Vietnam
Fan-Hsun Tseng, Taiwan
S Velliangiri , India
Quoc-Tuan Vien , United Kingdom
Enrico M. Vitucci , Italy
Shaohua Wan , China
Dawei Wang, China
Huaqun Wang , China
Pengfei Wang , China
Dapeng Wu , China
Huaming Wu , China
Ding Xu , China
YAN YAO , China
Jie Yang, USA
Long Yang , China
Qiang Ye , Canada
Changyan Yi , China
Ya-Ju Yu , Taiwan
Marat V. Yuldashev , Finland
Sherali Zeadally, USA
Hong-Hai Zhang, USA
Jiliang Zhang, China
Lei Zhang, Spain
Wence Zhang , China
Yushu Zhang, China
Kechen Zheng, China
Fuhui Zhou , USA
Meiling Zhu, United Kingdom
Zhengyu Zhu , China

Contents



Data Transmission Evaluation and Allocation Mechanism of the Optimal Routing Path: An Asynchronous Advantage Actor-Critic (A3C) Approach

Yahui Ding , Jianli Guo , Xu Li , Xiujuan Shi , and Peng Yu 
Research Article (21 pages), Article ID 6685722, Volume 2021 (2021)


A Random Label and Lightweight Hash-Based Security Authentication Mechanism for a UAV Swarm

Feng Hu , Hongyan Qian , and Liangjun Liu
Research Article (17 pages), Article ID 6653883, Volume 2021 (2021)



Blockchain-Based Trust Auction for Dynamic Virtual Machine Provisioning and Allocation in Clouds

Hao Xu , Weifeng Liu, and Xu Liu 
Research Article (10 pages), Article ID 6639107, Volume 2021 (2021)




Coordinated Control of Distributed Traffic Signal Based on Multiagent Cooperative Game

Zhenghua Zhang , Jin Qian, Chongxin Fang, Guoshu Liu, and Quan Su
Research Article (13 pages), Article ID 6693636, Volume 2021 (2021)

Path Planning of Mobile Robot Based on Improved Multiobjective Genetic Algorithm

Kairong Li , Qianqian Hu , and Jinpeng Liu
Research Article (12 pages), Article ID 8836615, Volume 2021 (2021)



Hierarchical Annotation Event Extraction Method in Multiple Scenarios

Shi Wang , Zhujun Wang , Yi Jiang , and Huayu Wang
Research Article (9 pages), Article ID 8899852, Volume 2021 (2021)




A Survey of k Nearest Neighbor Algorithms for Solving the Class Imbalanced Problem

Bo Sun  and Haiyan Chen 
Review Article (12 pages), Article ID 5520990, Volume 2021 (2021)

A Novel Pricing Mechanism for User Coalition in Blockchain

Yi Jiang , Xu Liu , and Jun Dai
Research Article (12 pages), Article ID 8885179, Volume 2020 (2020)

Efficient Planning and Solving Algorithm of S-Shape Acceleration and Deceleration

Zhijie Li , Ligang Cai , and Zhifeng Liu 
Research Article (14 pages), Article ID 8884678, Volume 2020 (2020)

Research Article

Data Transmission Evaluation and Allocation Mechanism of the Optimal Routing Path: An Asynchronous Advantage Actor-Critic (A3C) Approach

Yahui Ding ^{1,2}, Jianli Guo ², Xu Li ², Xiujuan Shi ², and Peng Yu ^{1,2}

¹State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

²Science and Technology on Communication Network Laboratory, The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China

Correspondence should be addressed to Jianli Guo; 15232132720@163.com

Received 16 November 2020; Accepted 29 July 2021; Published 3 September 2021

Academic Editor: Chunpeng Ge

Copyright © 2021 Yahui Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The delay tolerant networks (DTN), which have special features, differ from the traditional networks and always encounter frequent disruptions in the process of transmission. In order to transmit data in DTN, lots of routing algorithms have been proposed, like “Minimum Expected Delay,” “Earliest Delivery,” and “Epidemic,” but all the above algorithms have not taken into account the buffer management and memory usage. With the development of intelligent algorithms, Deep Reinforcement Learning (DRL) algorithm can better adapt to the above network transmission. In this paper, we firstly build optimal models based on different scenarios so as to jointly consider the behaviors and the buffer of the communication nodes, aiming to ameliorate the process of the data transmission; then, we applied the Deep Q-learning Network (DQN) and Advantage Actor-Critic (A3C) approaches in different scenarios, intending to obtain end-to-end optimal paths of services and improve the transmission performance. In the end, we compared algorithms over different parameters and find that the models build in different scenarios can achieve 30% end-to-end delay decline and 80% throughput improvement, which show that our algorithms applied in are effective and the results are reliable.

1. Introduction

Delay tolerant network (DTN) which has high delay and lower delivery rate is a newly developing network framework, aiming to realize the interconnection and asynchronous data stable transmission in hybrid environment. DTN has a wide range of application, like the sensor networks and the mobile networks, which have obtained the attention and deep research of the academic and industries.

Although DTN can be applied in many challenged scenarios, the reliability cannot be guaranteed for the characteristics of discontinuity and randomness. So many scholars have proposed plenty of routing algorithms based on “carry-store-forward” to improve the quality of transmission. The algorithms can be classified into two types of strategies; the first kind of algorithms gets better delivery rate through message

copies, like the “Epidemic” algorithm forwards the data in the manner of flooding, but too many copies of the message occupy much memory and increase the overhead of network. The second kind of algorithms forwards the data through classifying, like “First Contact” chooses end-to-end paths randomly to forward data and takes no account of the priori data; “Minimum Expected Delay” uses Dijkstra algorithm to find the path if minimum delay, but which only considers the limited prior knowledge not necessarily global optimal. Although the above algorithms can provide great convenience for us, they also increase the risk if the security of an SDN (Software Defined Network (SDN)) network is compromised. So a new authentication scheme called the hidden pattern (THP) was proposed, which combines graphics password and digital challenge value to prevent multiple types of authentication attacks at the same time [1].

DTN can complete the delivery of the message in the complex environments of frequent interruption just because the nodes can store messages. But the above routing algorithms have not considered the memory management, so it is important to determine the optimal end-to-end paths considering the effective management and usage of node capacity.

So in our paper, we have did research on the DTN and forming three different scenarios when the communication links break down firstly, and then, we applied the DQN algorithm and A3C algorithm in our proposed optimal models; finally, we compared algorithms over different parameters and find that the models build in different scenarios can achieve 30% end-to-end delay decline and 80% throughput improvement.

The main innovation of this paper lies in the following:

- (i) We have researched on various scenes and different actions of the nodes, build optimal models in different scenarios
- (ii) We adopt the DQN algorithm and A3C algorithm in the optimal models, with the aim of optimizing the throughput of the service data
- (iii) We compare the DRL algorithm with other DTN routing algorithms over different parameters

The composition of this paper is as follows. Section 1 outlines the characters of DTN and the related works. The optimal models which built over different scenarios can be found in Section 2.

Section 3 states the procedure and structure of the DQN algorithm and A3C algorithm. Section 4 gives the simulation topology and parameters. Section 5 shows the performances of the algorithms over different simulation parameters and gives the analysis results. At last, Section 6 states the conclusions and future improvements.

2. The Outline of DTN

For the existence of the Bundle Layer in DTN, it can implement store-and-forward message switching and the service of custody transfer. These two functions are described in details below.

When forwarding messages from the source node to the destination node in TCP/IP network, the messages can query route to find the path in the relay nodes and cannot be stored permanently, for this network has continuous connection to complete the transmission. But in DTN networks, the nodes can store the messages for a period of time and move by carrying the messages when meet the appropriate node to forward the message in the manner of message bundle like the Figure 1 shown.

After the nodes sending the messages to the next rely node in the form of bundles, if the node has not received the receipt confirmation information from the next node, then the node will choose an appropriate time to forward the bundle again. As Figure 2 shown, the relay node will return a receipt to the previous node, and when the relay node forwarding messages to the next hop, the next relay

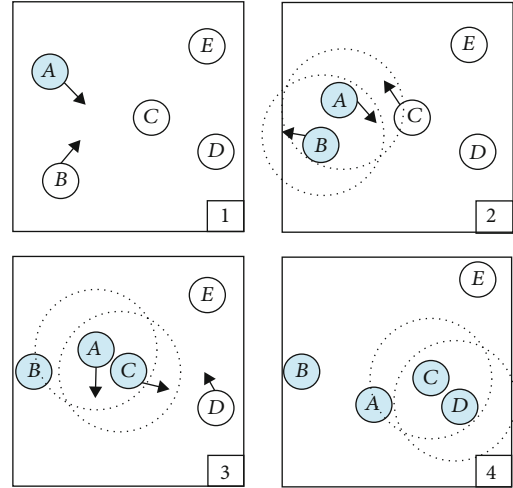


FIGURE 1: Carry-store-forward.

node will also send a receipt to the relay node; this procedure is carried on until the destination node receive the bundle [2]. The purpose of the custody transfer is to increase the reliability of the data transmission; only the node receives the receipt from the next hop, or the message is overdue, or the memory is full; the node deletes the message.

Ensuring DTN completes the service data transmission is important, so the scholars have studied and improved the routing algorithm in specific scenarios and proposed many routing algorithms [3].

In view of whether the infrastructure is required in the process of data forwarding, the DTN routing algorithms are composed of infrastructure-aided algorithms and non-infrastructure-aided algorithms as Figure 3 shown.

For the above problems, some recent studies [4–6] have proposed efficient cooperative caching schemes, in which data is cached at proper nodes or router nodes with limited sizes. But these papers need long time and large memory to broadcast the services. In [7], a joint optimization framework about caching, computation, and security for delay-tolerant data in M2M communication networks was proposed and adopted deep Q-network (DQN) in the model. In [8], a shortest weighted path-finding problem is formulated to identify the optimal route for secure data delivery between the source–destination pair, which can be solved by employing the Dijkstra’s or Bellman–Ford algorithm.

In [9], this paper uses the reliability of travel time as the weight of path selection, and solving by Dijkstra algorithm can reflect the actual vehicle path selection more accurately. This method is a beneficial improvement to the problem of static path selection. A dynamic routing algorithm based on energy-efficient relay selection (RS), referred to as DRA-EERS, is proposed in [10] to adapt to the higher dynamics in time-varying software-defined wireless sensor networks. In [11], a solution to the data advertising problem that is based upon random linear network coding was provided; the simulation results show that the proposed approach is both highly scalable and can significantly decrease the time for advertisement message delivery. A routing architecture and algorithm based on deep neural networks was proposed in [12], which can help routers

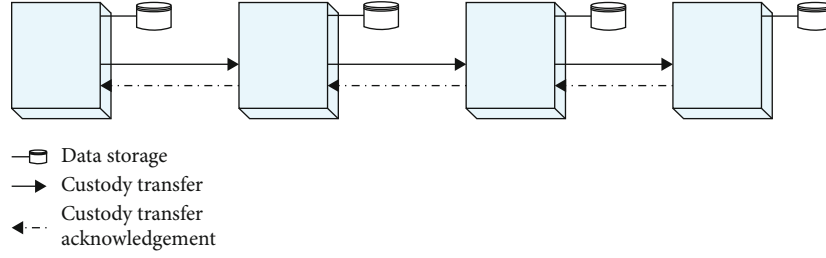


FIGURE 2: Custody transfer.

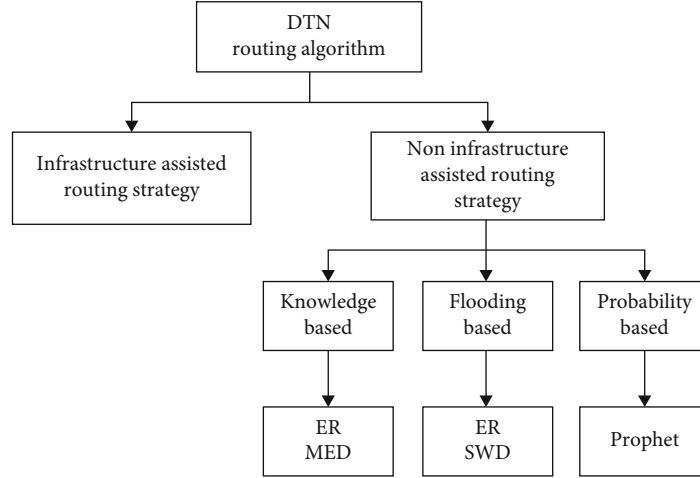


FIGURE 3: DTN routing algorithms.

make packet forwarding decisions based on the current conditions of its surroundings. A limited copy algorithm MPWLC based on service probability was provided in [13]; not only is the number of copies limited but the storage resources of the satellite are also taken into account to ensure reliable data transmission. The simulation results show that the proposed algorithm can effectively improve the efficiency of the network and ensure the reliable data transmission. In [14], a mathematical framework for DTN is introduced and suggested and applies it to a space network that is simulated using an orbital analysis toolkit. In [15], the problem of autonomously avoiding memory overflows in a delay tolerant node was considered, and reinforcement learning was proposed to automate buffer management given that this paper can easily measure the relative rates of data coming in and out of the DTN node.

3. Scenarios and System Model

Definition 1 (connected directed graph). We use $G = (V, E)$ to denote the connected graph if

- (i) G is a directed graph
- (ii) If the connections exist between node $v_i \in V$ and node $v_j \in V$, there will have $e_{i,j} \in E$

The connected directed graph can be seen in Figure 4, the communication nodes responsible for forwarding the messages, the schedule nodes responsible for scheduling of

service data, the connections among the nodes are affected by the actual environments.

Assume in our graph that there exist N nodes and M links and K services; the communication nodes and schedule nodes can assemble together in $V = \{v_i, i = 1, 2, \dots, N\}$, and the broadband links and narrow-band links can assemble together in $E = \{e_j, j = 1, 2, \dots, M\}$; the service data $S = \{s_k, k = 1, 2, \dots, n\}$ transmitting from the initial node v_s to the end node v_d and the end-to-end paths are expressed by $P = \{p_k, k = 1, 2, \dots, n\} = \{V_{pk}, E_{pk}\}$, and the time slots are expressed by $T = \{t, t = 1, 2, \dots, n\}$.

Due to the exceptional application environments of DQN, which always have long transmission delay and uncertain end-to-end paths, we have researched on the “carry-store-forward” and “custody transfer” mechanism and build optimal models in the following scenarios.

During service data communication, assume all service data fragments start with the shortest end-to-end path p_k . Each source node of service can send s_η fragments altogether but can only delivery one fragment $f_k(t)$ in time slot t and ascertain the transmission situations of the fragments which has been delivered before simultaneously. $I_{fk}(t)$ indicates the connection status from the node to the next-hop node in time slot t .

$$I_{fk}(t) = \begin{cases} 1, & \text{the connection is interrupted,} \\ 0, & \text{the connection is connected.} \end{cases} \quad (1)$$

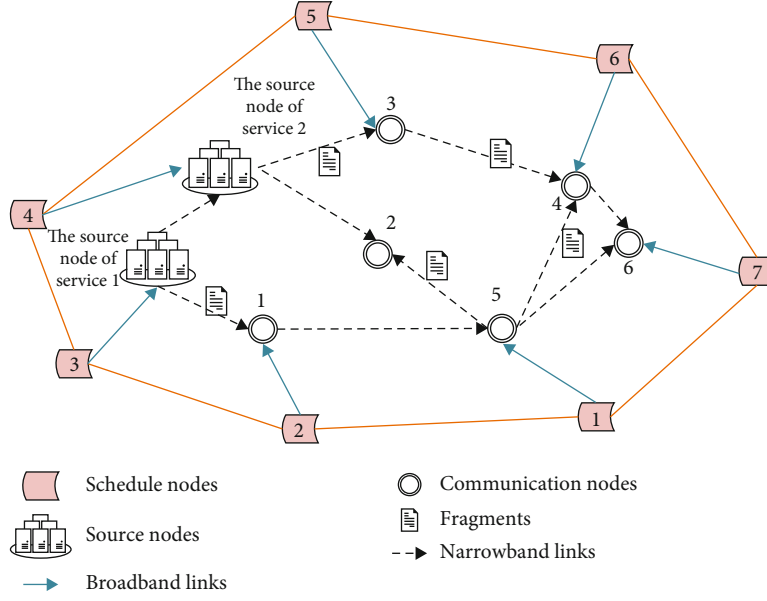


FIGURE 4: Connected directed graph.

Definition 2 (only consider cache scenario). We define a scenario only consider cache as Figure 5 shown, if

- (i) All fragments choose the shortest paths when sending from the source node
- (ii) If the fragments encounter interruption, the fragments only choose to store at the interrupted nodes and wait the nodes return to normal

In this scenario, we believe that every fragment $f_k(t)$ just chooses to store at the interrupted nodes, but at the same time, the cache data will increase the node cache processing delay $\alpha_f(t)$ and the link interrupt waiting delay $\beta_f(t)$; otherwise, there are no communication links interrupted ($I_{fk}(t) = 0$); the fragments need to consider the delivery delay on the link γ_e . This process is iterated until all the fragments reached the destination and calculate the throughput. Sup-

pose the total delay used to complete the service transmission is λ_1 :

$$\lambda_1 = \sum_{k=1}^K \left(\sum_{l=1}^{\eta} ((m_l - 1) \cdot \gamma_e) + \sum_{t=1}^T (I_{fk}(t) \cdot (\alpha_f(t) + \beta_f(t))) \right). \quad (2)$$

m_k is assumed to be the interrupted node in the shortest path p_k , and when all the services accomplish the transmission, the total length of fragments that reached the terminal nodes is ω_1 , and the throughput is ε_1 :

$$\omega_1 = \sum_{k=1}^K (\eta f_k(t)), \quad (3)$$

So the optimal model is:

$$\max \varepsilon_1 = \sum_{k=1}^K \frac{\omega_{1\{v_1\},\{e_1\}}}{\lambda_{1\{v_1\},\{e_1\}}} \left\{ \begin{array}{l} s.t. \forall v_i \in V_{p_k}, D_{v_i} = (I_{f^k}(t) * (\alpha_{f^k}(t) + \beta_{f^k}(t))), \quad (1) \\ \forall v_i \in V_{p_k}, D_{e_j} = \gamma_{e_j}, \quad (2) \\ \forall v_i \in V_{p_s}, \forall e_j \in E_{p_s}, D_R^{\eta} = \sum_{l=1}^{\eta} \left(\sum_{i=1}^m D_{v_i} + \sum_{j=1}^{m-1} D_{e_j} \right), \quad (3) \\ \forall v_i \in V_{p_s}, B \geq B_R^k, \quad (4) \\ \forall v_i \in V_{p_s}, \sum_{n=1}^k \sum_{t=1}^T I_{f^k}(t) f^k(t). \quad (5) \end{array} \right. \quad (4)$$

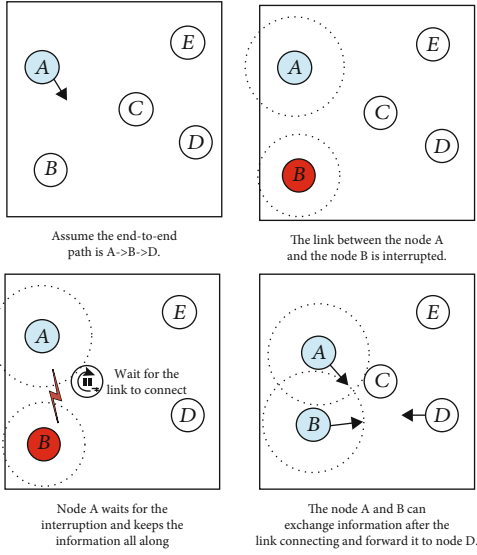


FIGURE 5: Only consider cache scenario.

Here, D_{v_i} denotes the total delay when the fragments store in the nodes in (1), D_{e_j} denotes the transmission delay in the path in (2), (3) states that the sum of processing delay D_{v_i} and transmission delay D_{e_j} cannot exceed the value bound s_R^k , (4) states that the bandwidth should be enough for the data transmission, and (5) denotes the maximum caching space of every node, so the total cache fragments cannot exceed the target value.

Definition 3 (only consider detour scenario). We define a scenario only considered choosing the detour path as Figure 6 shown, if

- (i) All fragments choose the shortest paths when sending from the source node
- (ii) If the fragments encounter interruption, the fragments only choose other available paths which have more identical nodes with the initial shortest path, so the fragments choose not to store and wait the nodes return to normal

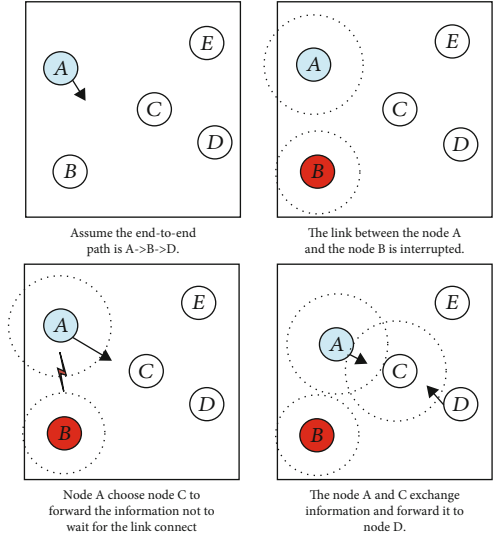


FIGURE 6: Only consider detour scenario.

In this scenario, we believe that every fragment $f_k(t)$ just chooses other detour paths p_s in interrupted nodes ($I_{f_k}(t) = 1$); the shortest path is p_k ; otherwise, the links are connected ($I_{f_k}(t) = 0$); the fragment just transmitted along the initial shortest path and only takes into account the delivery delay γ_{e_j} . The source nodes need to continuously pay attention to the transmission till all services accomplish data delivery. We assume that the fragments run into interruption in the u_k -th node; the transmission delay of every service in the alternate path is $Ds(f_R^k(t))$. Assume the total delay of all services is λ_2 .

$$\lambda = \sum_{k=1}^K \sum_{l=1}^{\eta} \sum_{j=1}^m \left(I_{f^k}(t) (j_k - 1) \gamma_{e_j} + D_R^d(f^k(t)) \right), \quad (5)$$

when all the services accomplish the transmission, the total length of fragments that reached the terminal nodes is ω_2 as formula (3), and the throughput is ε_2 , so the optimal model is

$$\max \varepsilon_2 = \sum_{k=1}^K \left(\frac{\omega_{2\{v_i, \{e_j\}\}}}{\lambda_{2\{v_i, \{e_j\}\}}} \right) \begin{cases} s.t. \forall v_i \in V_{p_s}, \forall e_j \in E_{p_s}, D_R^s = \sum_{i=1}^u \left(1 - I_{f^k}(t) \right) \gamma_{e_i}, & (1) \\ \forall v_i \in V_{p_s}, \forall e_j \in E_{p_s}, \forall v_i \in V_{p_d}, \forall e_j \in E_{p_d}, D_R^s + D_R^d \leq s_R^k, & (2) \\ \forall v_i \in V_{p_s}, B \geq B_R^s, & (3) \\ \forall v_i \in V_{p_d}, B \geq B_R^d. & (4) \end{cases} \quad (6)$$

Here, (1) states the transmission delay D_R^k in the initial shortest path, D_R^s denotes the transmission delay of every service in the alternate path, (2) states that the sum of transmission of the entire path cannot be exceed the target value s_R^k , and (3) and (4) state that the bandwidth of the entire path should be enough for the data transmission.

Definition 4 (comprehensive scenario). We define a scenario comprehensive the end-to-end path as Figure 7 shown, if

- (i) All fragments choose the shortest paths when sending from the source node
- (ii) If the fragments encounter interruption, the fragments jointly consider choosing other available paths which have more identical nodes with the initial shortest path or storing at the interrupted nodes, at last choose a path has the minimum end-to-end delay after comparing the above circumstances

In this scenario, we believe that every fragment $f_k(t)$ takes both the storage and the detour paths into account; $U_{f(t)}$ indicates the choice of the fragment. If the fragments chooses to wait at the nodes ($U_{f(t)} = 1$), which generates the waiting and transmitting delay λ_1 , which denotes by $\lambda_1 = \sum_{k=1}^K \sum_{j=1}^{\eta} (m_j - 1) \cdot \gamma_e + \sum_{j=1}^{m^k} (I_{f(t)})(\alpha_{f(t)} + \beta_{f(t)})$; if the fragment chooses a detour path ($U_{f_k}(t) = 0$) and tries to ensure that the available paths have more identical nodes with the initial shortest path and the total delivery delay λ_2 is $\lambda_2 = \sum_{k=1}^K (\sum_{l=1}^{\eta} (\sum_{j=1}^m (I_{f(t)}(j_k - 1) \cdot \gamma_{e_j} + D_R^s f(t))))$, the fragment will compare the above delays and choose the path that has minimum delay. The source nodes need to continuously pay attention to the transmission till all services

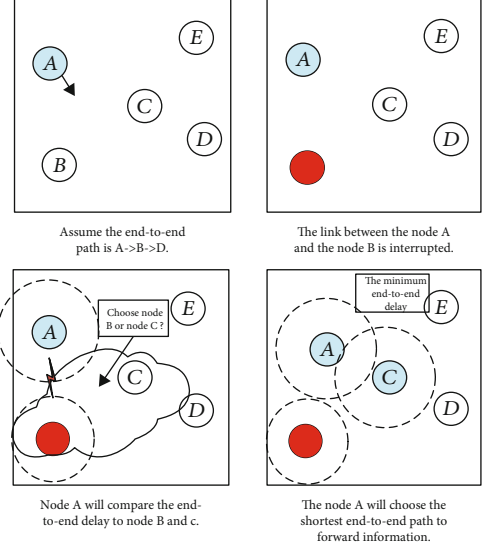


FIGURE 7: Comprehensive scenario.

accomplish data delivery. Assume the total delay of all services is λ_3 .

$$\lambda_3 = \sum_{n=1}^k \sum_{l=1}^{\eta} J_{v_i} \left(I_{f^k}(t) T_1 + (1 - I_{f^k}(t)) T_2 + (1 - J_{v_i})(m_k - 1) \gamma_e \right), \quad (7)$$

when all the services accomplish the transmission, the total length of fragments that reached the terminal nodes is as formula (3), so the optimal model is as follows:

$$\max \varepsilon_3 = \sum_{k=1}^K \frac{\omega_{3\{v_i\}, \{e_j\}}}{\lambda_{3\{v_i\}, \{e_j\}}} \left\{ \begin{array}{l} s.t. \forall v_i \in V_p, D_{v_i} = \alpha_{f^k}(t) + \beta_{f^k}(t), \quad (1) \\ \forall e_j \in E_{p_s}, D_{e_j} = \sum_{u=1}^u \gamma_e, \quad (2) \\ \forall v_i \in V_{p_s}, \forall e_j \in E_{p_s}, D_R^{\eta} = \sum_{l=1}^{\eta} \left(\sum_{i=1}^m D_{v_i} + \sum_{j=1}^{m-1} D_{e_j} \right) \leq s_R^k, \quad (3) \\ \forall v_i \in V_{p_s}, \forall e_j \in E_{p_s}, \forall v_i \in V_{p_d}, \forall e_j \in E_{p_d}, D_R^s + D_R^d \leq s_R^k, \quad (4) \\ \forall v_i \in V_{p_s}, B \geq B_R^s, \quad (5) \\ \forall v_i \in V_{p_d}, B \geq B_R^d, \quad (6) \\ \forall v_i \in V_{p_s}, \sum_{n=1}^k \sum_{l=1}^{\eta} U_{f^k}(t) \cdot f_k(t) \leq C_{v_i}. \quad (7) \end{array} \right. \quad (8)$$

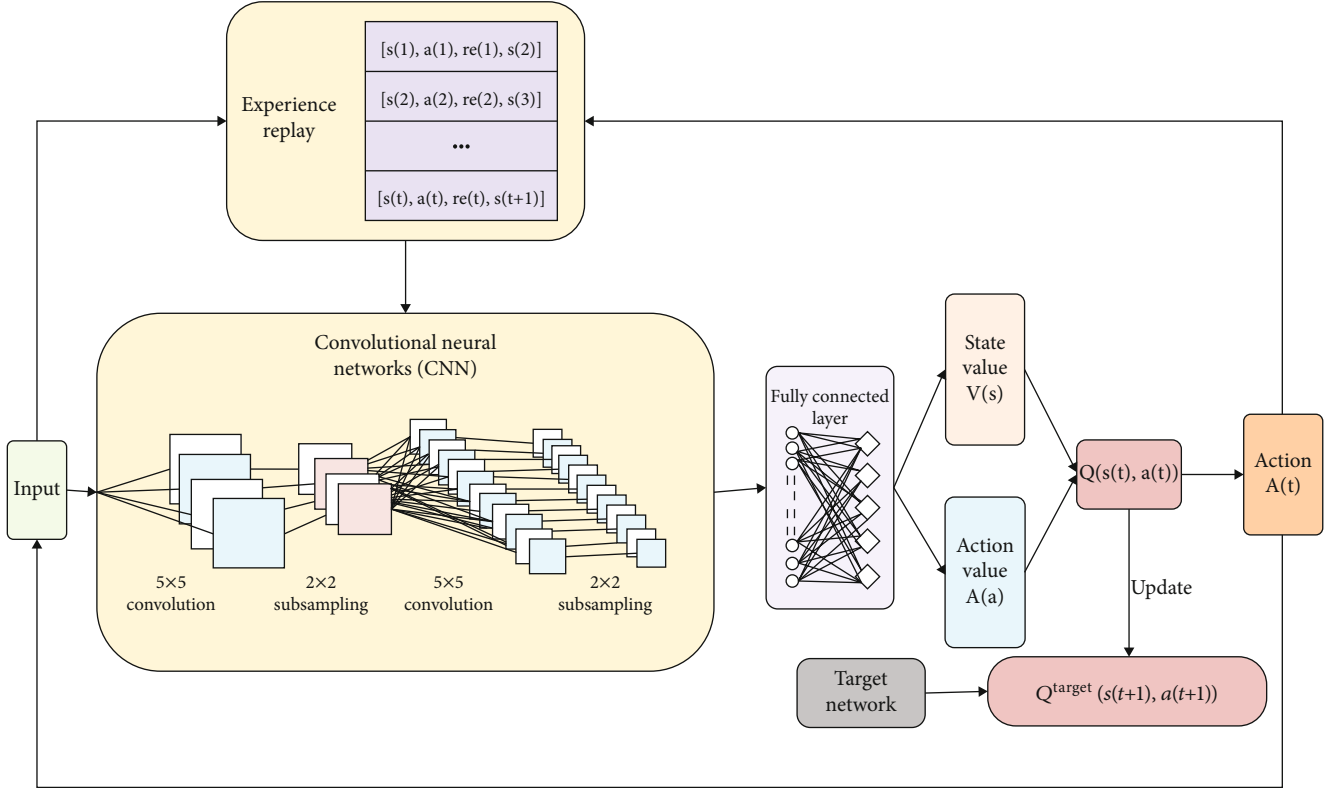


FIGURE 8: The work process of DQN algorithm.

Input: The initial locations of the service nodes

Output: The optimal path and throughput of every service

1 Initialize the topology of all the nodes and the start and end nodes of every service;

2 Initialize thread step counter $t \leftarrow 1$;

3 **while** $T \leq T_{\max}$ **do**

4 Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$;

5 Synchronize thread-specific parameters $\theta^1 = \theta$ and $t_{start} = t$;

6 Get state s_t , that is the start node of every service;

7 **while** terminal s_t not is the end node of every service or $t - t_{start} = t_{\max}$ **do**

8 Perform a_t that is the next-hop according to policy $\pi(a_t | s_t; \theta^1)$;

9 If all the constraints in models are satisfied, in consideration of the sequence of the fragments stay at the interrupted node and according to the choice of every fragment, like continue to store at the node or choose other detour paths, which can build different scenarios and have various the next-hop s_{t+1} and reward r_t ;

10 $t \leftarrow t + 1$;

11 $T \leftarrow T + 1$;

12 **end**

13 Set

$$R = \begin{cases} \mathbf{f}_0 & \text{for terminal } s_t \\ V(s_t, \theta_v^1) & \text{for non-terminal } s_t \text{ or Bootstrap from last state} \end{cases}$$

14 **for** $i = t - 1$ to t_{start} **do**

15 $R \leftarrow r_i + \gamma R$;

16 Accumulate gradients wrt θ^1 : $d\theta \leftarrow d\theta + R - V(s_t; \theta_v) \delta_{\theta^1} \log \pi(A_i | s_t; \theta)$;

17 Accumulate gradients wrt θ_v^1 : $d\theta_v \leftarrow d\theta_v + \partial(R - V(s_t; \theta_v)^2) / \partial \theta$;

18 **end**

19 Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$;

20 **end**

ALGORITHM 1: Data transmission evaluation of the optimal routing path with A3C algorithm.

Input: The initial locations of the service nodes
Output: The optimal path and throughput of every service

- 1 Initialize replay memory D to capacity N ;
- 2 Initialize action-value function Q with random weights θ ;
- 3 Initialize target action-value function \hat{Q} with weights $\theta^- = \theta$;
- 4 **forepisode** = 1 to N **do**
- 5 Initialize the topology of all the nodes;
- 6 Get the initial state X_k of all nodes and the distance between nodes;
- 7 Set sequence $s_1 \leftarrow e_1$, preprocess $\varphi_1 \leftarrow \varphi(s_1)$;
- 8 **fort** = 1 to T **do**
- 9 Select a random action a_t for every node i with probability ϵ otherwise select
 $a_t = \max_a Q(\varphi(st), a; Q)$;
- 10 Execute action a_t in emulator and observe reward r_t ;
- 11 If all the constraints in models are satisfied, in consideration of the sequence of the fragments stay at the interrupted node and according to the choice of every fragment, like continue to store at the node or choose other detour paths, which can build different scenarios and have various the next-hop, Set $st + 1 \leftarrow (st, at, et + 1)$ and preprocess $\varphi_{t+1} = \varphi(st + 1)$, otherwise go back to step 7; 13 Store transition $(\varphi_t, a_t, r_t, \varphi_{t+1})$ in D ;
- 12 Sample random mini batch of transitions $(\varphi_t, a_t, r_t, \varphi_{t+1})$ from D ;
- 13 Set

fr_j	stops at step $j + 1$
$y_j =$	
$r_j + \gamma \max_a \hat{Q}(\varphi_{j+1}, a; \theta^-)$	Otherwise
- 14 Perform a gradient descent step on $y_j - Q(\varphi_j, a_j; \theta)^2$ with respect to the network parameters θ ;
- 15 Every C step do reset $\hat{Q} = Q$;
- 16 **end**
- 17 **end**

ALGORITHM 2: Data transmission evaluation of the optimal routing path with DQN algorithm.

Here, D_{v_i} denotes the total delay when the fragments store in the nodes in (1), D_{e_j} denotes the transmission delay in the path in (2), D^s indicates the transmission delay of the alternate path in (3), the sum of transmission of the entire path cannot exceed the target value s_R^k , (4) states that the sum of processing delay D_{v_i} and transmission delay D_{e_j} cannot exceed the value bound s_R^k , (5) and (6) state that the bandwidth of the entire path should be enough for the data transmission and $B^k = \min B_{v_i}, \forall v_i \in V_{p_k}$, and in (7), C_{v_i} denotes the maximum caching space of every node, so the total cache fragments cannot exceed the target value.

4. DRL Algorithm Procedure and Structure

Deep Reinforcement Learning (DRL) is a special and environment-friendly machine learning method, which uses the environment feedback as input and is the learning from environment state to behavior mapping; RL can maximize the cumulative return of system behavior from the environment, mainly consisting of agents and the external environment. But traditional reinforcement learning has bottlenecks; it uses table to save every state and the Q value of every action based on state [16]. And deep Q-learning network (DQN) can adopt neural networks to solve the above problems and make the state and action as the input of neural network; it can obtain the Q value through the neural network not the table, reducing the memory con-

sumption. In DQN, they use experience replay to avoid correlation of data samples, but every time the agent interacts with the environments needs huge memory and computing power, and experience replay can only generate data by the old policy. So, A3C uses multithread of CPU to realize the parallel actor learner for multiagent instance; each thread corresponds to different exploration strategies. This parallelization can be used to decorrelate data and replace experience replay to save storage cost.

4.1. DQN Algorithm. Deep Learning has been proved to be a powerful tool to solve nonconvex and high complexity problems and has been widely used in many ways. Reinforcement learning pays more attention to the maximal rewards over a long period time which obtained by interacting with environment and carry out the optimal action. The deep Q-learning adopts deep neural network to develop an action plan and behave well when deal with dynamic time-varying environments. So, DTN provides a promising technology for the data transmission in delay tolerant network.

With regard to the reinforcement learning, it interacts with the environment through the agent, which can inspect the environment and obtain the states (t) and then take action $a(t)$ based on the state at the time slot t . Next, the external environment observes the action taken by agent and deliveries the latest state $s(t + 1)$ and the reward $r(t)$ to the agent. The above process is aimed at finding the maximal reward value by the optimal policy π^* . DQN uses neural networks to approximate the value function $Q(s, a)$, and

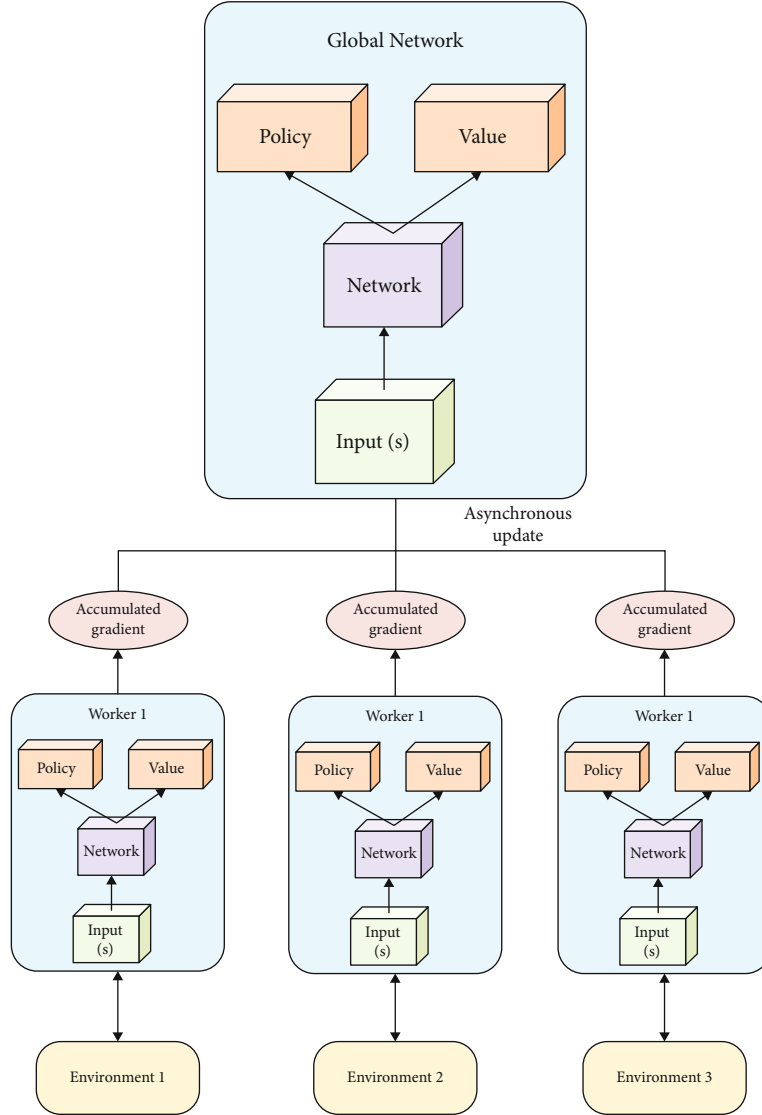


FIGURE 9: The work process of A3C algorithm.

the reward value $Q^*(s, a)$ can be obtained according to the following:

$$Q^*(s, a) = E^* \left[\sum_{s \in S} r(s, a, s^1) + \gamma \max Q^*(s^1, a^1) \right]. \quad (9)$$

In which the reward is computed based on the state s and the action a , γ represents the discount factor which means the future impact on the present calculation, and $E^*[\bullet]$ denotes the expectation function. Hence, DQN chooses the action which can get the Q value to the maximum.

The Q value updated at every step in DQN as follows:

$$Q(s, a) = Q(s, a) + \sigma (r_{t+1} + \gamma \max Q(s^1, a) - Q(s, a)). \quad (10)$$

In which σ denotes the leaning rate and should be in the range of $[0,1]$, with the increase of the learning rate, the

influence of the past on the present is getting smaller and smaller. The process of the DQN is shown in Figure 8.

In DQN, $\langle A, S, R, P \rangle$ is a typical Quaternions [17], in which Action A indicates the action set taken by agent, State S indicates the state set observed from the environment, Reward R indicates the set of rewards value, and P denotes the deep learn mode in probability state space of agent learning. Based on the above typical Quaternions, the specific definition of DQN is shown as follows:

- (1) *State Space*. $S = (X_1, X_2, \dots, X_K)$, in which it is a vector and denotes the location of the source nodes of the k -th service; the vector has k dimensions, and only one dimension is 1
- (2) *Action Space*. $A = (A_1, A_2, \dots, A_K)$, in which it is a vector and denotes the nodes of the k -th service nodes can be connected; the vector has k dimensions, and only one dimension is 1

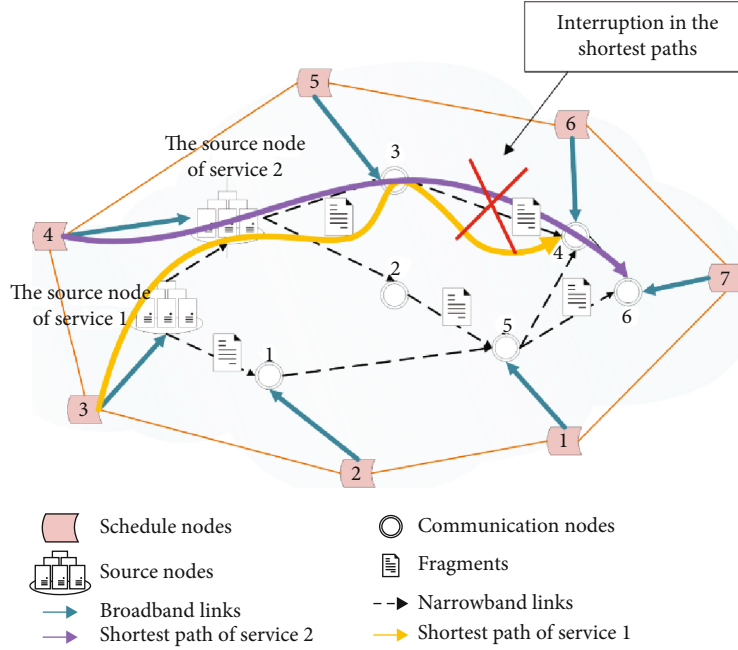


FIGURE 10: Network topology.

- (3) *System Reward*. After each time slot t , the system will get the immediate reward $r(t)$ based on different taken action $a(t)$. In our paper, we denote the reward as the cost of the current node to other nodes; the reward $r(t)$ is smaller if the distance between the nodes is longer; otherwise, reward is bigger; $r(t)$ is shown as follows:

$$r(t) = \begin{cases} \text{distance}(i, j) + c, & \text{Distance is bigger} \\ -100, & \text{No Connection} \\ \text{distance}(i, j) - c, & \text{Distance is smaller} \end{cases} \quad (11)$$

With above analysis, the DQN procedures for the optimal path and throughput are shown in Algorithm 2.

4.2. A3C Algorithm. A3C uses the method of multithreading; at the same time, it interacts with the environment in multiple threads, and each thread summarizes the learning results to global net as shown. In addition, each thread regularly takes back the results of common learning from global net to guide the learning interaction between the thread and the environment. Through this method, A3C avoids the problem of strong correlation of experience playback and achieves asynchronous and concurrent learning model.

The A3C algorithm is based on the actor-critic consists of the value function $V(s_t, \theta_v)$ and policy function $\pi(a | s_t; \theta)$, which will not use the traditional Monte Carlo to update the parameters until the end of the scenario episode but uses temporal-difference learning to update parameters in each steps. About the actor-critic, it has two networks; one is the actor network, which is responsible for choosing the

TABLE 1: The parameters of simulation.

Parameters	Symbol	Value
Time slot	t	0.1 s
Fragment size	f	1400 bits
Maximum storage space of node	C	10 bits
Velocity of propagation	v_i	$3 \cdot 10^4$ bits
Waiting delay	β_{e_i}	ats
Cache rate	α_{e_i}	700 bps
Bandwidth	B	6000 bps
Learning rate for DQN	σ_D	0.001
Discount factor for DQN	γ_D	0.09
Memory pool capacity-DQN	N	2000
Minibatch-DQN	b	320
Execute steps-DQN	T	5000
Global maximum iterations-A3C	Nt_{mt}	1000
Maximum iterations in thread-A3C	Nt_{st}	10
Discount factor for A3C	γ_A	0.9
Learning rate for A3C-actor	σ_a	0.001
Learning rate for A3C-critic	σ_c	0.001
Entropy factor	ρ	0.001

actions on account of the policy $\pi(a | s_t; \theta)$; the critic network is responsible for evaluating each action from the actor network. And after the actor network obtains the scores of the action, it will optimize the policy to get the maximal reward over the algorithm executions. The critic network use the following formula to calculate the score of the

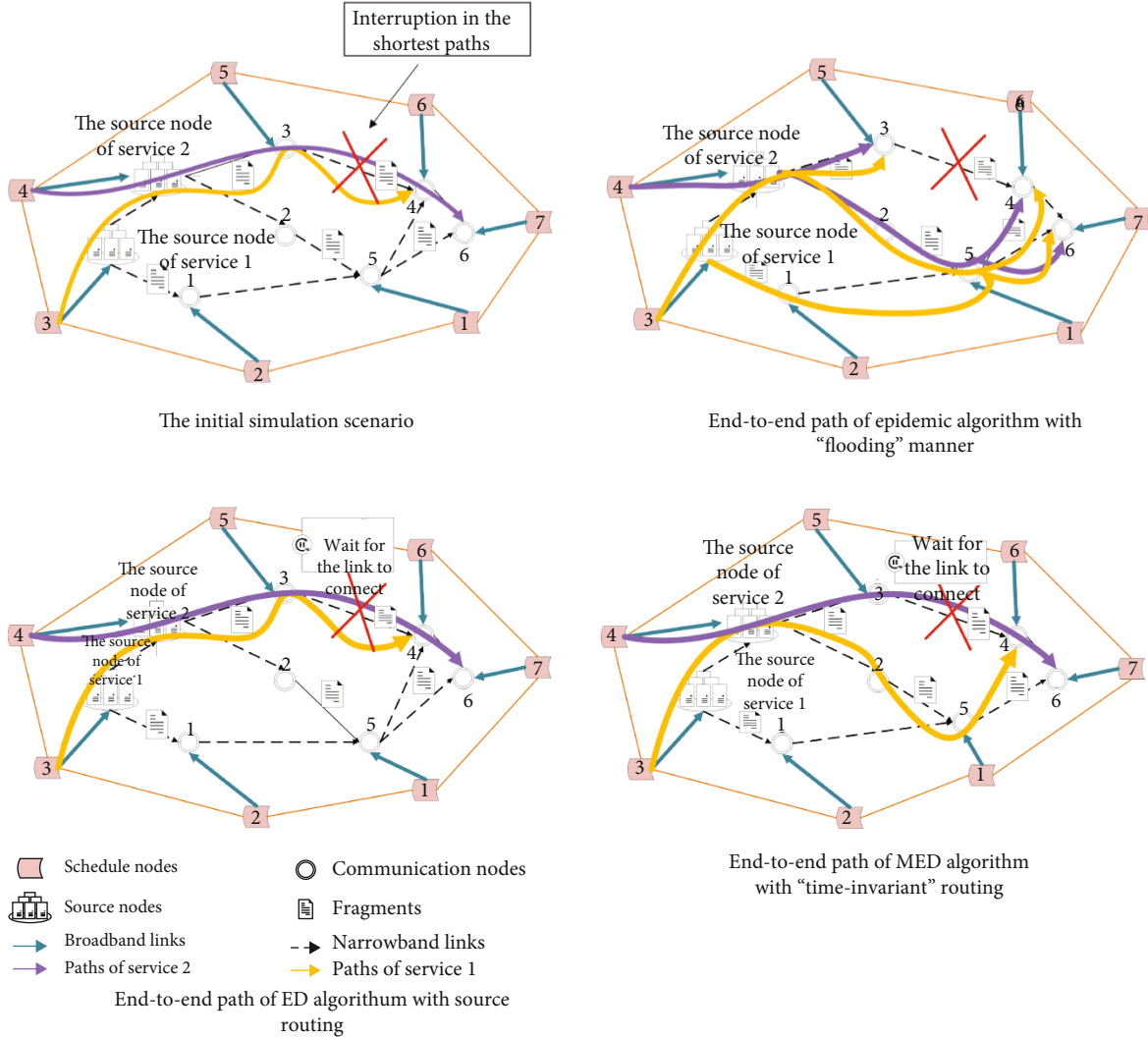


FIGURE 11: Process of the above algorithms.

actions

$$6\theta_v = \frac{\partial R - V(s_t; \theta_v)^2}{\partial \theta}, \quad (12)$$

in which, the R denotes the reward of taking the action a . Through calculating the gradient of the formula (4), the actor network can update the parameter θ ; the gradient can be seen from the following:

$$6\theta_v = R - V(s_t; \theta_v) 6\theta \log \pi(A_i | s_t; \theta). \quad (13)$$

In A3C, it has defined a new function which called advantage function $A(s, a)$ that can be shown as

$$A(s, a) = R - V(st; \theta). \quad (14)$$

It expresses that if the action chosen is better than the average, then the advantage function is positive; otherwise, it is negative.

Figure 9 shows the process of A3C; it has one global network which includes the functions of actor network and critical network. And it also has n workers; each worker has the same network structure as the global neural network, and each worker will interact with the environment independently to get experience data. These workers do not interfere with each other and run independently.

Each worker and the environment interact with a certain amount of data, then calculate the gradient of the neural network loss function in its own thread; but these gradients do not update the neural network in its own thread but update the global neural network. In other words, n workers will independently use the accumulated gradient to update the common part of the neural network model parameters. Every once in a while, the thread will update the parameters of its own neural network to the parameters of the common neural network and then guide the subsequent environment interaction.

The specific description of A3C algorithm is presented in Algorithm 1.

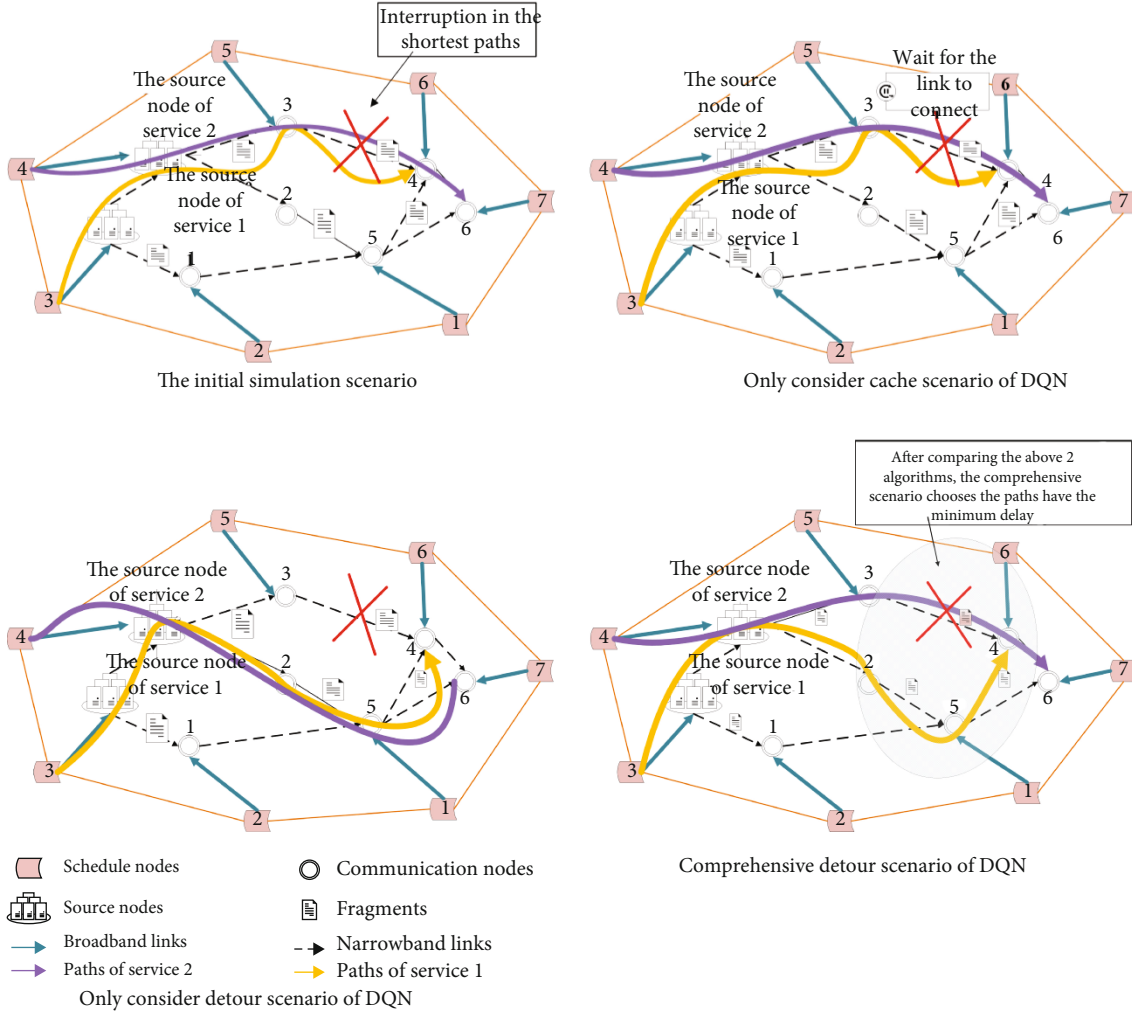


FIGURE 12: Process of the DQN.

5. Simulation and Analysis

5.1. Simulation Parameters. The simulation scenario is shown in Figure 10, and we assume to send two services in total, and the services are send in the different source nodes, and every service will send hundreds of data segments; the topology not only has broadband links but also has narrowband links, so the two services have different shortest paths, so when the service segments encounter disruption in the process of service data transmission, they can choose to cache in the interrupted nodes or choose other detour paths to arrive the target nodes. The purple and the yellow links denote the service’s end-to-end path, respectively; the red links denote the interruption in the shortest end-to-end path of every service.

In the process of simulation, we assume the transmission rate of service data is same in every source node, and the number of data segment cache disunified in every node. And the transmission rate of service segments is the constant value ($3 * 10^4$ bps).

In the above topology, the source nodes may send many fragments, which can result in block at the inter-

rupted node for too many fragments store at the node, making DQN algorithm consumes more time to analyze the queue problem in the node. If the simulation topology is more complicated, the DQN algorithm needs more time to train and ensure the end-to-end path of services. So in the simulation process of the DQN algorithm, we assume the DQN network consists of three layers of neural network, and the every layer has a certain number of neurons, and the learning rate of DQN is 0.001; the discount factor to calculate the reward is 0.09 [18], and the size of memory pool which used to execute experience replay is 2000. Other specific parameters can get in Table 1; based on the above settings, the DQN algorithm can find the optimal path of services.

For the execution process of the A3C algorithm, the simulation environment is the same in every worker which is the independent kernel in computer, and we set the learning rate of actor, and critic is 0.001; the discount factor to calculate the reward is 0.09; other specific parameters can get in Table 1. Based on the parameters and the special structure of the algorithm, it has higher execution speed and performance than DQN algorithm.

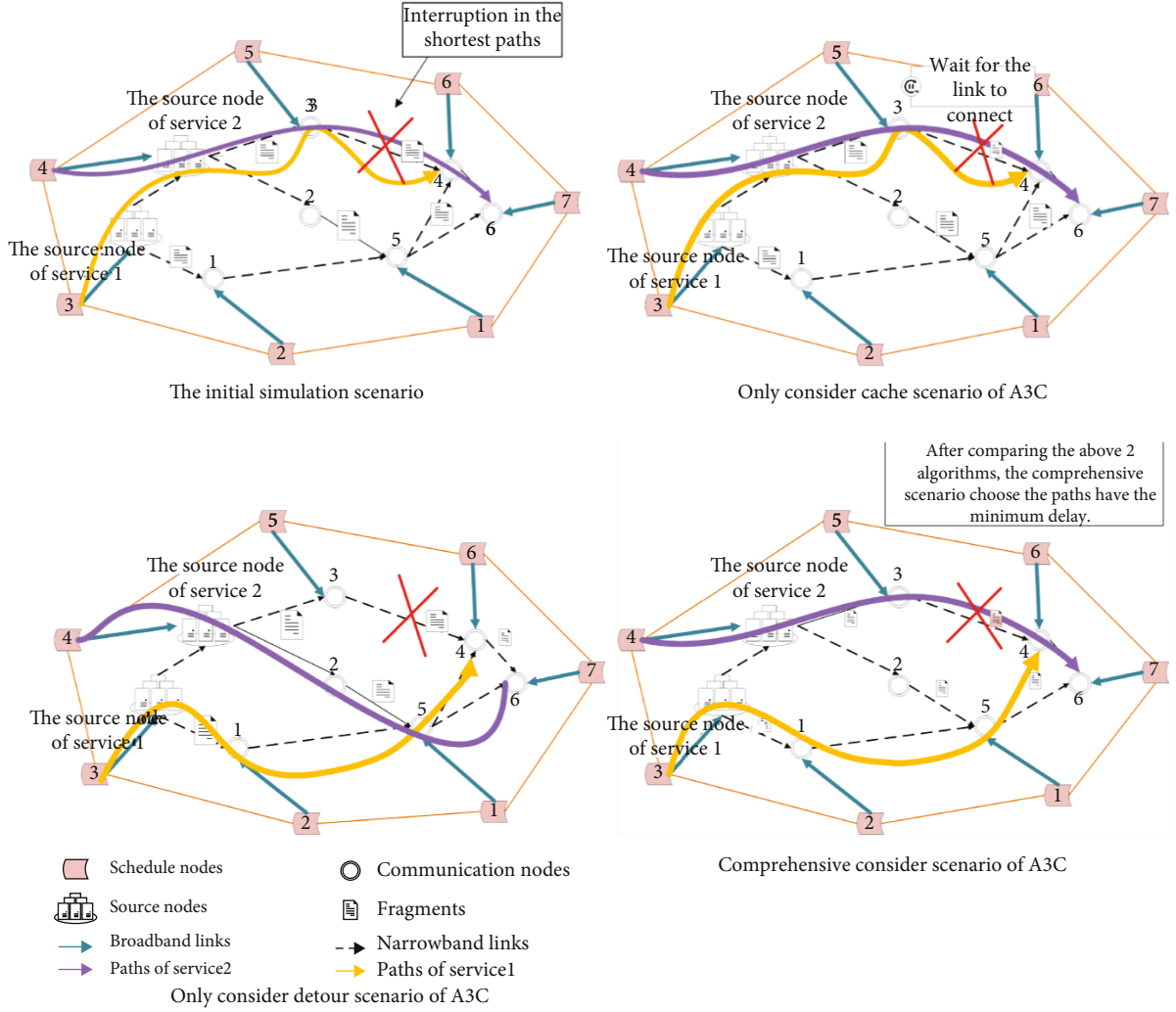


FIGURE 13: Process of the A3C.

We will compare the performance of the algorithm from the following aspects, and the parameters of simulation are shown in Table 1.

The delivery rate can be expressed as the ratio of the number of fragments reached the destination node and the number of fragments send from the source nodes, as the following shows:

$$D = \frac{\sum_{k=1}^K Ld_k}{Ls}, \quad (15)$$

in which Ld_k denotes the number of fragments that reach the destination node of service and Ls denotes the total number of fragments that send from the source nodes.

The end-to-end delay can be represented as delay from the time of the source nodes start to send fragments to the time of the last fragment reach the destination, as the following shows:

$$De = t_d - t_s, \quad (16)$$

in which, t_d is the time of the last fragment reach the

destination, and t_s is the time of the service data start to transmit.

The throughput of service can be expressed as the number of service data successfully transmit from the source node to the destination node per unit time, as the following shows:

$$Th = \frac{\sum_{k=1}^K Ld_i}{De}, \quad (17)$$

in which, $\sum_{k=1}^K Ld_i$ is the total service data reach the destination node and De is the end-to-end delay.

The equilibrium of nodes is the number of average service carrying of every node, as the following shows:

$$N = \frac{\sum_{i=1}^V}{N_i p}, \quad (18)$$

in which, N_i is the number of service carrying of the i -th node, and P is the total number of nodes in the topology.

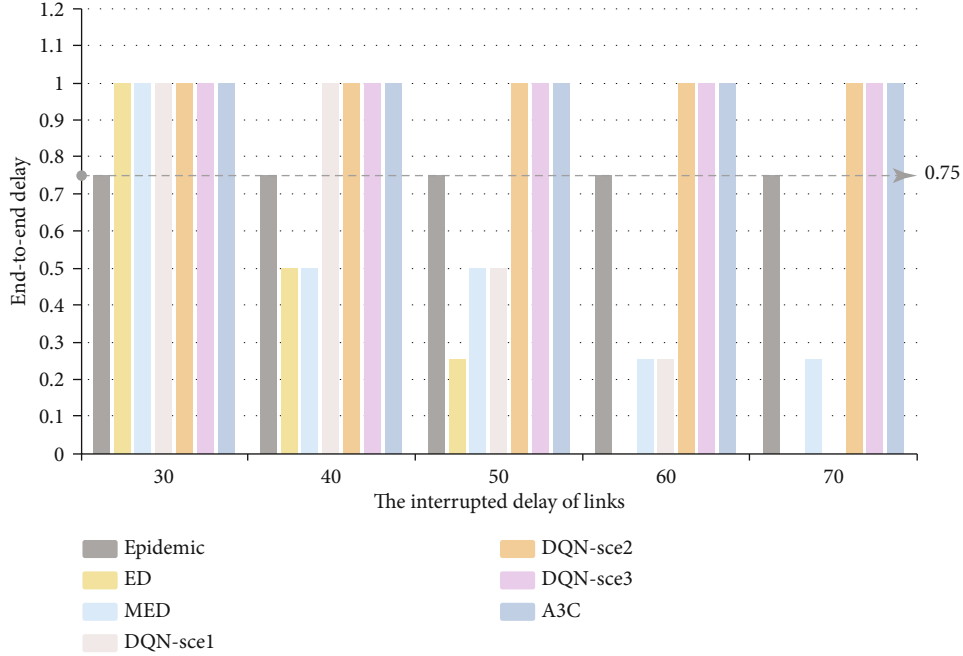


FIGURE 14: The delivery rate.

TABLE 2: The delivery rate.

Delivery rate	Algorithms						
	Epidemic	ED	MED	DQN-sce1	DQN-sce2	DQN-sce3	A3C
Interrupted delay							
30	0.75	1	1	1	1	1	1
40	0.75	0.5	0.5	1	1	1	1
50	0.75	0.25	0.5	0.5	1	1	1
60	0.75	0	0.25	0.25	1	1	1
70	0.75	0	0.25	0	1	1	1

The equilibrium of links is the number of average service carrying of every link, as the following shows:

$$L = \frac{\sum_{i=1}^E L_i}{L_i Q}, \tag{19}$$

in which, L_i is the number of service carrying of the i -th link, and Q is the total number of links in the topology.

6. Simulation Results

6.1. Simulation Algorithms. The ‘‘Epidemic’’ algorithm belongs to the spread routing; it forwards the data in the manner of flooding; that is to say, all nodes encountered will ‘‘infect’’ the message [19], so there will have lots of copies of the message in the network and occupy much memory and increase the overhead of network.

But in the circumstance of sufficient network resources, the Epidemic algorithm will show faster delivery rate and be the alternative algorithm in this circumstance. If there

are too many messages to transmit, some messages may be discarded, resulting in higher packet loss rate.

The ED algorithm has not taken the queue problem into consideration, and routing path is determined when the source nodes send the service data, so the ED algorithm belongs to the source routing. But when there have more fragments and queuing, it will affect the computation of weight of the ED algorithm; thus, making the computation of the source route emerges errors and cannot produce the optimal path.

The MED algorithm has taken the transmission delay, the propagation delay, and the average waiting delay into consideration; the goal of the algorithm is to find the path of minimum delay, and the path adopted is identical when the source nodes and destination nodes are same. After ensuring the path of source routing, even there has better choice, this algorithm will not change the routing choices [20], so it is just the optimal path over the limited prior knowledge and not necessarily global optimal, so the MED algorithm belongs to ‘‘time-invariant’’ algorithm.

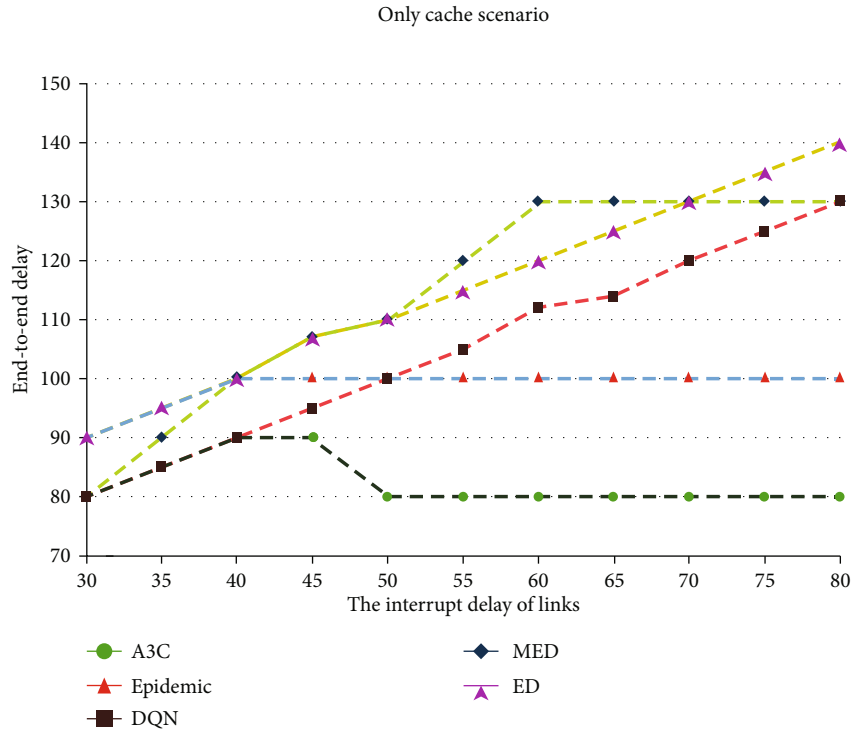


FIGURE 15: The end-to-end delay of service 1.

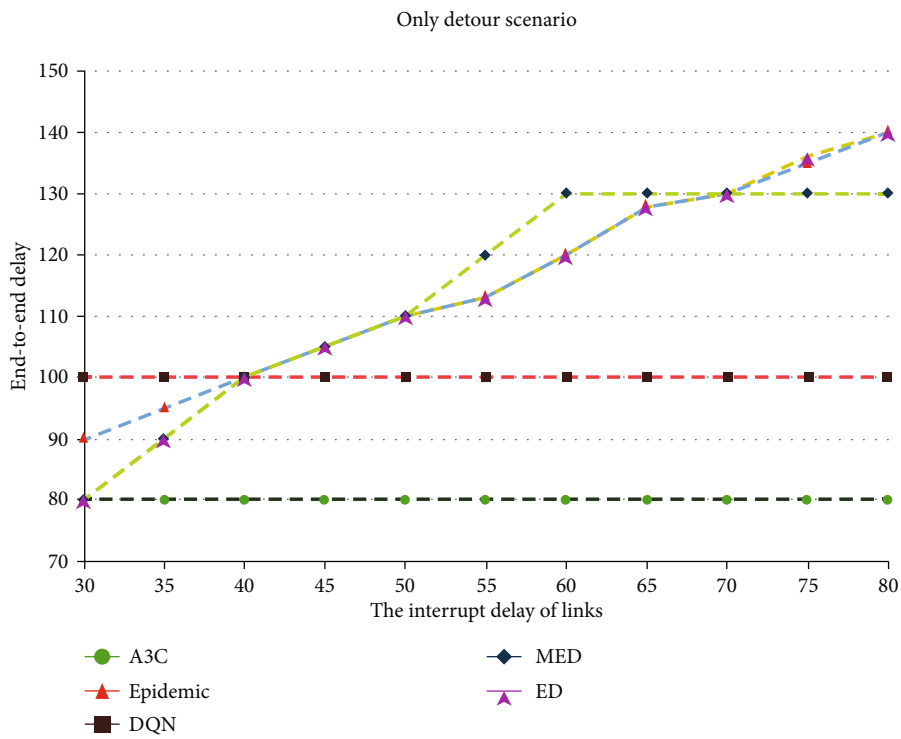


FIGURE 16: The end-to-end delay of service 1.

Deep Learning has been proved to be a powerful tool to solve nonconvex and high complexity problems and has been widely used in many ways. The deep Q-

learning adopts deep neural network to develop an action plan and behaves well when deal with dynamic time-varying environments.

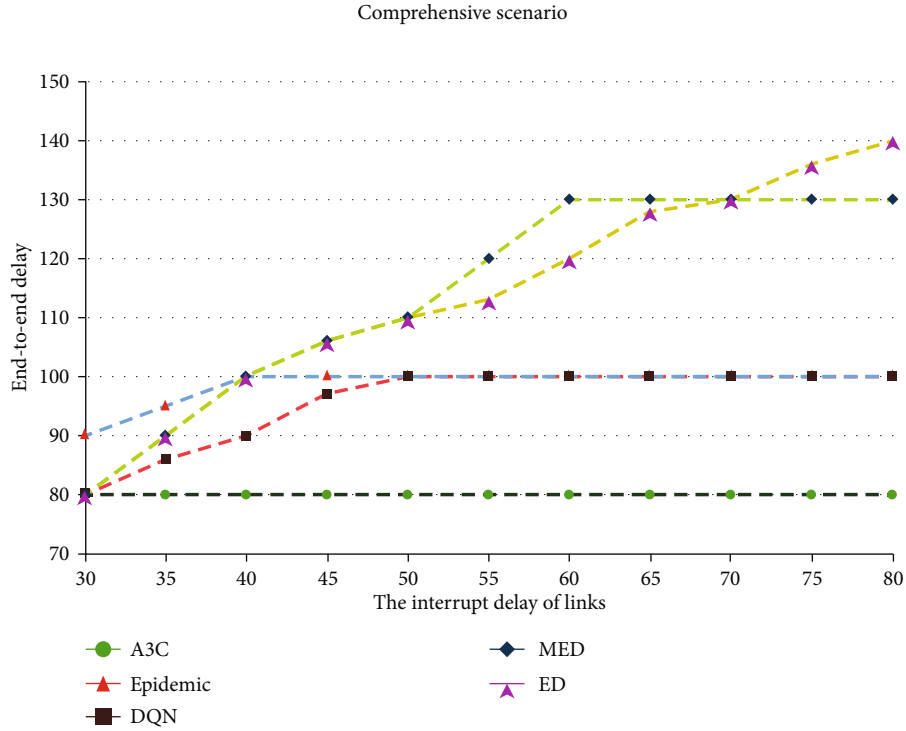


FIGURE 17: The end-to-end delay of service 1.

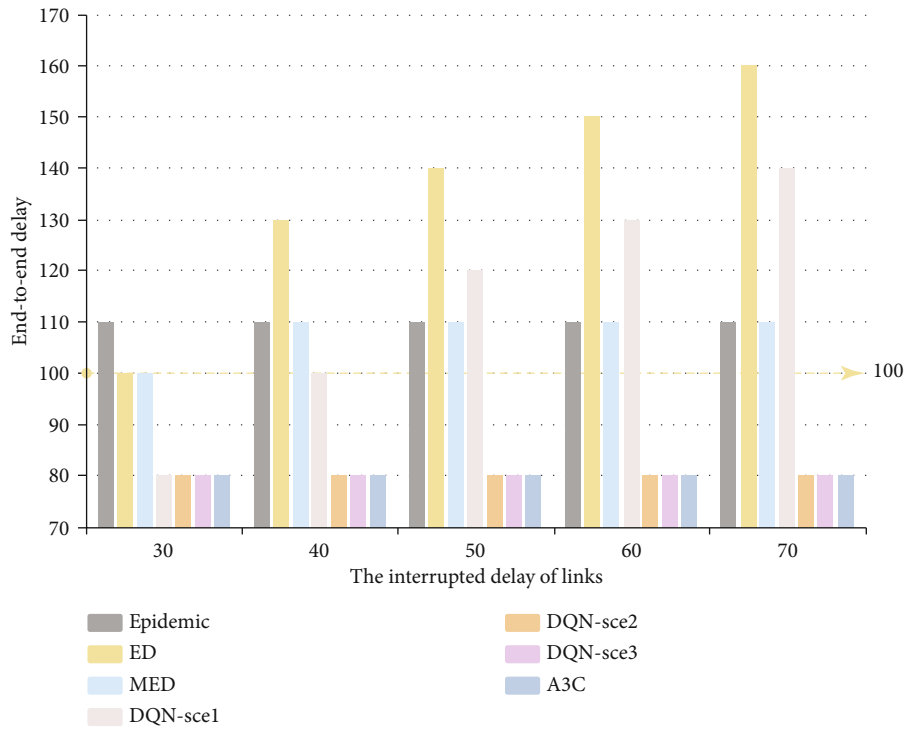


FIGURE 18: The end-to-end delay of service 2.

A3C uses the method of multithreading; at the same time, it interacts with the environment in multiple threads, and each thread summarizes the learning results to global

net. In addition, each thread regularly takes back the results of common learning from global net to guide the learning interaction between the thread and the environment.

TABLE 3: The end-to-end delay of service 2.

End-to-end delay	Algorithms						
	Epidemic	ED	MED	DQN-sce1	DQN-sce2	DQN-sce3	A3C
Interrupted delay							
30	110	100	100	80	80	80	80
40	110	130	110	100	80	80	80
50	110	140	110	120	80	80	80
60	110	150	110	120	80	80	80
70	110	160	110	140	80	80	80

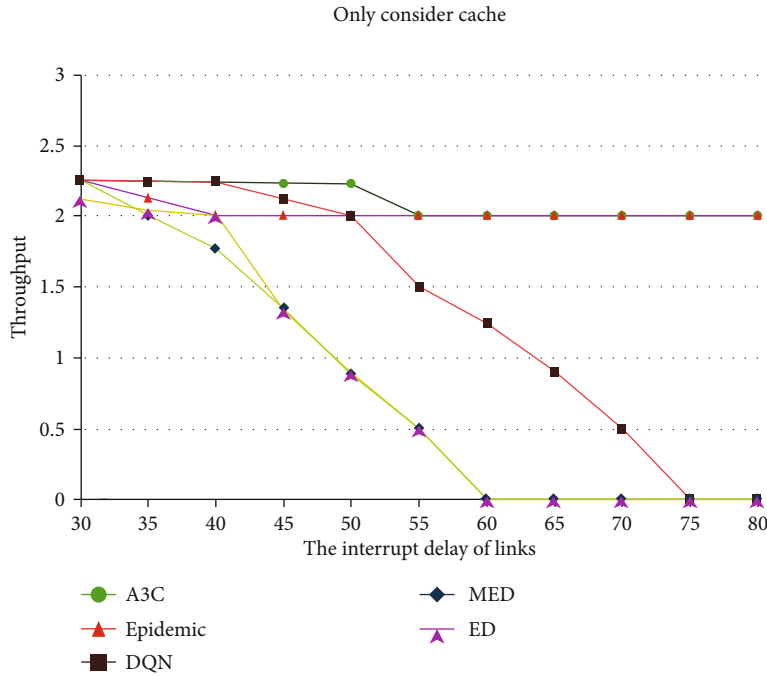


FIGURE 19: The throughput of service 1.

6.2. Simulation Results

6.2.1. *Comparison of End-to-End Paths.* The end-to-end paths of different algorithms can be seen in Figures 11–13; owing to the different operating mechanism of algorithms, they choose different paths when encountering the interrupted nodes. We can see that only the DQN algorithm and A3C algorithm can change the route when facing different scenarios, and they often choose the paths that have the minimum end-to-end paths. In the comprehensive scenario, the DQN algorithm and A3C algorithm have compared the end-to-end delay of the above scenarios and have the optimal path in the three scenarios.

6.2.2. *Comparison of Delivery Rate.* In this paper, we broadcast 2 services in this topology and record delivery rate based on different link break delay over different algorithms and assume the minimum delivery rate is 0.75. It is shown in Figure 14 that as the link break delay increases, the delivery rate is decrease over majority of algorithms, but in the only consider detour scenario and the comprehensive scenario of DQN, the delivery rate is remain unchanged and is the

maximum, because in these scenarios, the source nodes choose the detour path and not affected by the interrupted links, so increasing the delivery rate. Through the result of all algorithms shown in Table 2, we found that the DQN algorithm has higher delivery rate in the majority of circumstances, which shows the delivery rate improvement of our proposed models and the DQN algorithm. But the A3C algorithm has the highest delivery rate in every scenario, because the A3C has many subthreads which can find the optimal paths in a very short time. The A3C algorithm and DQN algorithm can satisfy the constrained delivery rate in most cases.

6.2.3. *Comparison of End-to-End Delay.* In DTN network, due to the particularity of data connection, there may be open circuit between data connections, which makes data have to be stored in the node waiting for the link to be connected again. However, the storage space at the node is limited. When some greedy algorithms are adopted, multiple data storage may cause the use of nodes pace, so that when the following data arrives, it will cause data loss.

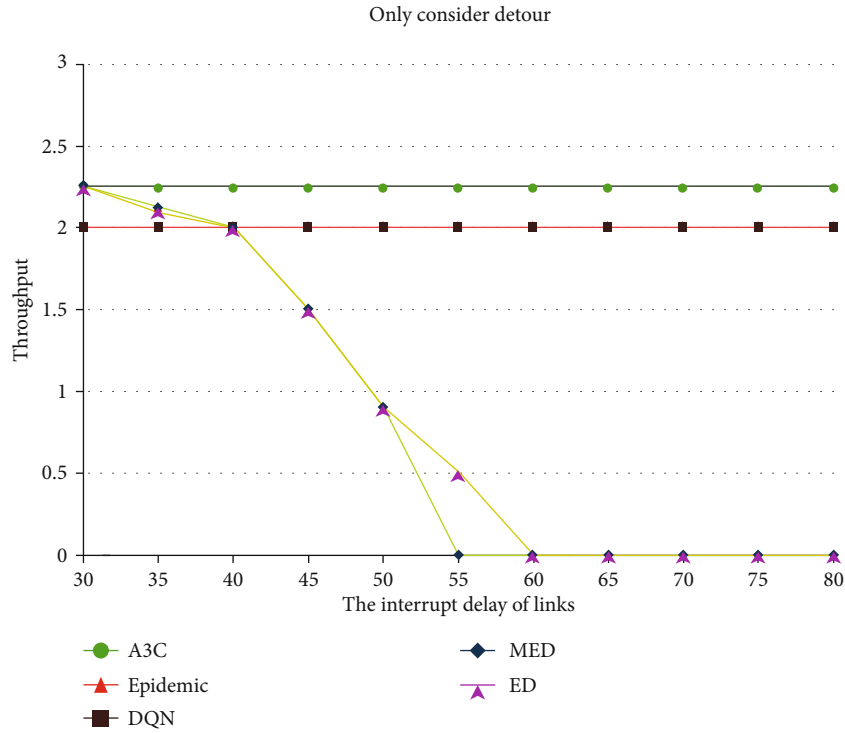


FIGURE 20: The throughput of service 1.

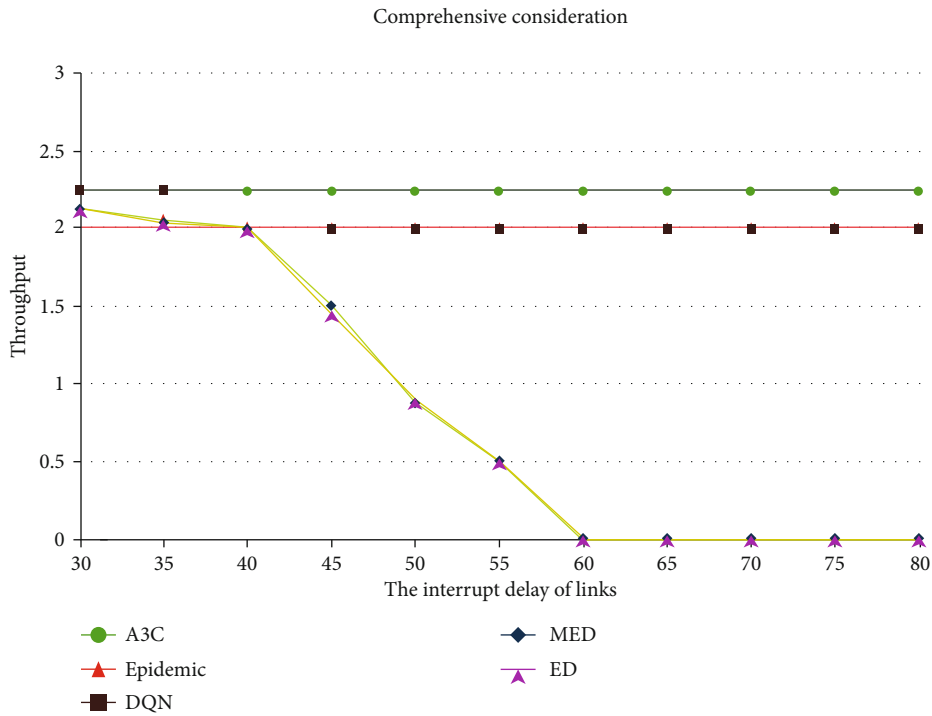


FIGURE 21: The throughput of service 1.

The waiting delay at the node is reflected in the total end-to-end delay of data transmission, including not only the transmission delay on the link but also the waiting delay at the node. When using epidemic, ED, and other algorithms, due to the particularity of the algorithm, it will copy multiple copies of data to be transmitted in the network, so

compared with DQN and other reinforcement learning algorithms, it will increase the waiting delay at the node.

Because the transmission of multiple copies of data will cause congestion at the node and when the transmission continues, it will increase the queuing delay. For the transmission of data, intelligent algorithms such as DQN will

TABLE 4: The throughput of service in the scenario of comprehensive.

Throughput	Algorithms				
	Epidemic	ED	MED	DQN	A3C
Interrupted delay					
30	2.1	2.1	2.25	2.25	2.25
40	2	2	2.25	2.25	2.25
50	2	0.9	0.9	2	2.25
60	2	0	0	2	2.25
70	2	0	0	2	2.25
80	2	0	0	2	2.25

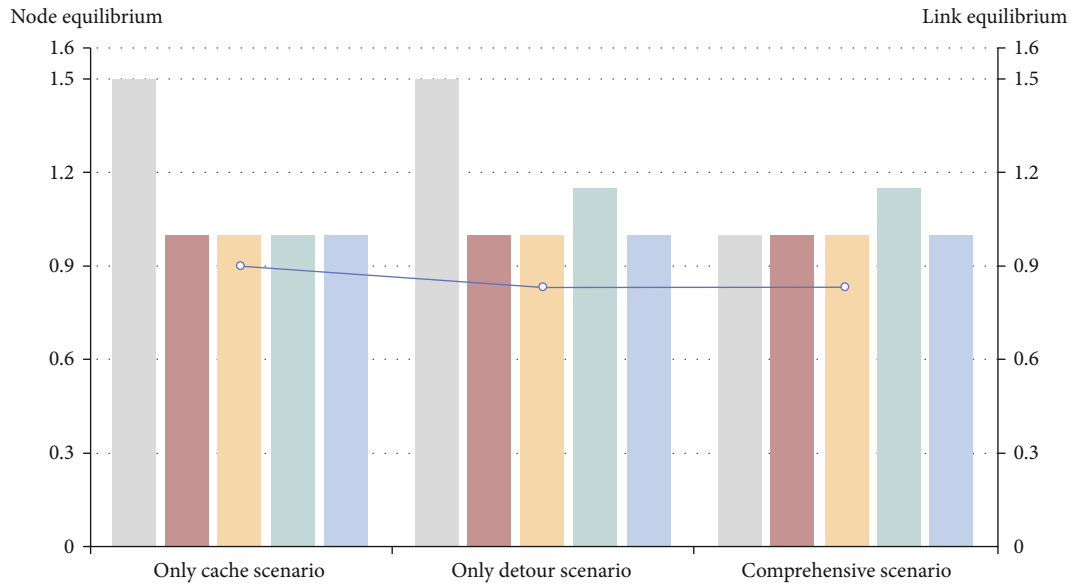


FIGURE 22: The node and link equilibrium of services.

not transmit multiple copies of the same data but take the reward in the algorithm as the guidance, minimize the end-to-end delay in the network, and reduce the occurrence of congestion at the node, so the end-to-end delay of each algorithm is shown in the figure below.

The total transmission delay of the service 1 and service 2 is shown in Figures 15–18; it can be seen that in the three scenarios, the minimum transmission delay is 100 ms, and the A3C algorithm has the minimum transmission delay, and DQN algorithm has lower delay, but in the scenario 1, the total transmission delay of service 1 and service 2 is increased as the interrupted delay of link is increasing, and the specific data are shown in Table 3, because in this scenario when the fragments encounter the interruption, they choose to store at the node, so the delay keeps on rising. Because in our paper, we think the capacity of node is available, so over the Epidemic algorithm, the fragments can arrive the destination node smoothly, and the total transmission delay of the Epidemic is not too high. But over the ED algorithm, it has the highest transmission delay; for the ED algorithm is the source routing algorithm, it determines the transmission path when the source nodes send the fragments, so when the fragments encounter the interruption, they will not change the path and result the high delay.

6.2.4. Comparison of Throughput. The throughput of service is shown in Figures 19–21; it can be seen that in the three scenarios, the A3C algorithm has the maximum throughput, which is better than the DQN algorithm; in the consider cache scenario, the throughput is decrease, for in this the scenario the transmission delay is a little higher and the nodes that reach the destination node are not a lot. The specific throughput data of all algorithms are shown in Table 4, and the throughput of ED algorithm and MED algorithm of service 1 and service 2 is also decrease, because the delay is increase as the interrupted delay of link increase. But in the only consider detour and comprehensive consider scenarios, there has the maximum throughput, which can be proved that our models and the adopted algorithm have improved the transmission.

6.2.5. Comparison of Node Equilibrium and Link Equilibrium. The node equilibrium of services can be seen from Figure 22; we can see the Epidemic algorithm has the maximum node equilibrium; for in the process of the algorithm, it forwards the fragments over the flooding manner, which means when the node comes into the communication scope of other nodes, if the node found that other nodes

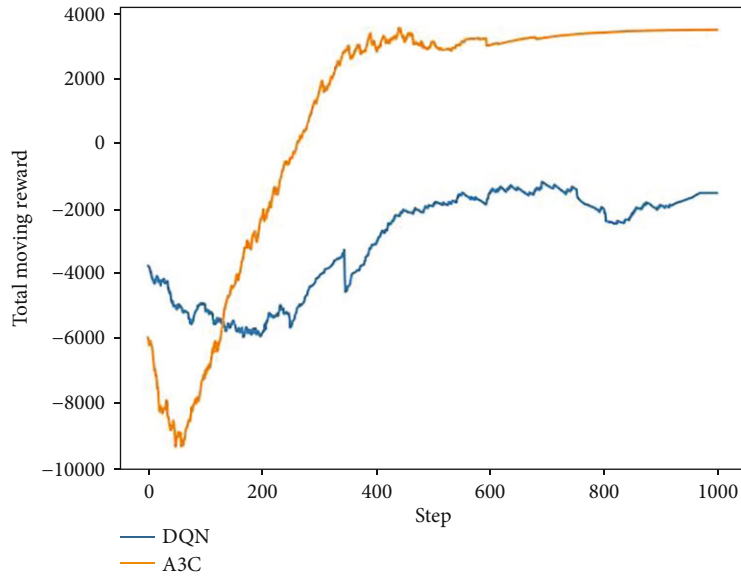


FIGURE 23: The node and link equilibrium of services.

have not the fragment, it will send the fragment to the other nodes. So it can lead to many copies of fragments in the work, so every node may store every fragment of every service, so the equilibrium is the highest. The node equilibrium of the ED algorithm and the MED algorithm is a little lower, and average node equilibrium of DQN is a little higher, which has to be improved. And for the link equilibrium, the Epidemic algorithm has the maximum link equilibrium; the cause of the result is the same to the node equilibrium, because it forward too many duplicates in the network. The average link equilibrium of DQN is a little higher but not very large as the node equilibrium, so it also has to be improved, so the A3C algorithm has improved the node and link equilibrium, which has lower equilibrium compared to the DQN algorithm.

6.2.6. Comparison of Total Reward and Loss. From Figure 23, we can see that the A3C algorithm has higher reward and can converge very quickly. At first, the value of reward is random jitter, because the exact value cannot be obtained in a short time. And the reward of A3C can get close to its top value with 400 episodes, but DQN needs about 700 episodes. The results prove that our models and algorithms adopted can reach an optimal value and converge.

7. Conclusions

In this paper, we have proposed optimal models based on different scenarios consist of the only consider cache scenario, the only consider detour scenario, and comprehensive consider scenario; the models are intend to jointly consider the behavior and the buffer of the nodes to improve the performance of the data transmission. Owing to different choices of the nodes, there will form three scenarios, and we adopted the DQN algorithm to solve the complex non-linear optimization problem and to get the optimal solutions, which consist of lower end-to-end delay, higher

throughput, and better data delivery guarantees. The results of simulation show that compared to other algorithms like Epidemic, ED algorithm, and MED algorithm, the DQN algorithm we adopted has better performance improvement.

As future work, we are going to improve the optimal models and decrease the overhead of nodes and links, expecting the application of DQN can be further studied in the delay tolerant networks.

Data Availability

The data is available from the following link: https://github.com/MorvanZhou/Reinforcement-learning-with-tensorflow/tree/master/contents/10_A3C.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded open fund project of the Science and Technology on Communication Networks Laboratory (Grant No. SXX19641X073).

References

- [1] L. Fang, Y. Li, X. Yun et al., "THP: a novel authentication scheme to prevent multiple attacks in SDN-based IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5745–5759, 2020.
- [2] W. Zhang, J. Xiong, L. Gui, B. Liu, M. Qiu, and Z. Shi, "On popular services pushing and distributed caching in converged overlay networks," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, Valencia, Spain, June 2018.
- [3] B. Fajardo, K. Yasumoto, N. Shibata, W. Sun, and M. Ito, "DTN-based data aggregation for timely information

- collection in disaster areas,” in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 333–340, Barcelona, Spain, October 2012.
- [4] Y. Wu and Z. Li, “Queueing analysis for delay/disruption tolerant networks with random link interruptions,” in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 94–99, Chengdu, China, December 2016.
- [5] Z. Yang, R. Wang, Q. Yu et al., “Analytical characterization of Licklider transmission protocol (LTP) in cislunar communications,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2019–2031, 2014.
- [6] H. Wu, Y. Li, J. Jiao, B. Cao, and Q. Zhang, “LTP asynchronous accelerated retransmission strategy for deep space communications,” in *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 99–104, Aachen, Germany, September 2016.
- [7] R. Lent, “Analysis of the block delivery time of the Licklider transmission protocol,” *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 518–526, 2019.
- [8] Y. Xu, J. Liu, Y. Shen, J. Liu, X. Jiang, and T. Taleb, “Incentive jamming-based secure routing in decentralized Internet of Things,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 3000–3013, 2021.
- [9] L. Zhi, X. Zhou, and J. Zhao, “Vehicle routing for dynamic road network based on travel time reliability,” *IEEE Access*, vol. 8, pp. 190596–190604, 2020.
- [10] Z. Ding, L. Shen, H. Chen, F. Yan, and N. Ansari, “Energy-efficient relay-selection-based dynamic routing algorithm for IoT-oriented software-defined WSNs,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9050–9065, 2020.
- [11] N. Bezirgiannidis and V. Tsaoussidis, “Packet size and DTN transport service: evaluation on a DTN testbed,” in *International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 1198–1205, Moscow, Russia, October 2018.
- [12] H. Lu, F. Jiang, J. Wu, and C. W. Chen, “Performance improvement in DTNs by packet size optimization,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 2987–3000, 2015.
- [13] G. Araniti, N. Bezirgiannidis, E. Birrane et al., “Contact graph routing in DTN space networks: overview, enhancements and performance,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.
- [14] J. Segui, E. Jennings, and S. C. Burleigh, “Enhancing contact graph routing for delay tolerant space networking,” in *Proceedings of the Global Communications Conference, GLOBECOM*, pp. 1–6, Houston, TX, USA, December 2011.
- [15] N. Bezirgiannidis, C. Caini, D. D. P. Montenero, M. Ruggieri, and V. Tsaoussidis, “Contact graph routing enhancements for delay tolerant space communications,” in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, pp. 17–23, Berlin, Germany, December 2018.
- [16] Lindgren, A. Doria, and O. Schelen, “Probabilistic Routing in Intermittently Connected Networks,” in *Service Assurance with Partial and Intermittent Resources*, pp. 239–254, Springer, 2014.
- [17] P. Maitreyi and M. S. Rao, “Design of binary spray and wait protocol for intermittently connected mobile networks,” in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–3, Las Vegas, NV, USA, January 2017.
- [18] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Max-prop: routing for vehicle-based disruption-tolerant networks,” in *25TH IEEE International Conference on Computer Communications*, pp. 1–11, Barcelona, Spain, January 2016.
- [19] R. Lent, “A cognitive network controller based on spiking neurons,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, May 2018.
- [20] R. Lent, D. Brooks, and G. Clark, “Validating the cognitive network controller on NASA’s scan testbed,” in *2020 IEEE International Conference on Communications (ICC)*, pp. 554–559, Dublin, Ireland, May 2020.

Research Article

A Random Label and Lightweight Hash-Based Security Authentication Mechanism for a UAV Swarm

Feng Hu , Hongyan Qian , and Liangjun Liu

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Correspondence should be addressed to Hongyan Qian; qhy98@nuaa.edu.cn

Received 20 November 2020; Accepted 26 May 2021; Published 1 July 2021

Academic Editor: Junwu Zhu

Copyright © 2021 Feng Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, the application of a UAV swarm is becoming more and more widespread in the military field, and more and more attention is paid to the security of mission resource allocation. However, the relay node forwarding in the wireless transmission process brings greater risks to data leakage, and the computing power and energy of the UAV consumption is limited, so a lighter solution is required. This paper proposes a mechanism for the safe allocation of UAV swarm mission resources based on random labels. Each task has a random label to solve the problem of database security and wireless transmission security in the process of UAV task assignment. Furthermore, a lightweight stream cipher encryption scheme is illustrated to ensure the security of the UAV database. The irreversible hash function SHA-256 and the lightweight foam structure hash function SPONGENT-128 are used to generate random labels and then allocate task resources. In the case of energy consumption, it reduces the possibility of the enemy successfully obtaining private data. The simulation results show that the scheme has good performance in terms of security and has better performance than existing methods in terms of throughput and delay, without increasing too much energy consumption.

1. Introduction

UAV networks are currently attracting more and more attention. The coordination and collaboration between multiple UAVs and ground base stations have established UAV swarm systems, which are easy to deploy, low in purchase and maintenance costs, and large in coverage and battery capacity. Small, therefore limited flight time but high maneuverability, so it is a practical choice for civilian and military applications, especially in the military field [1], can be used for environmental and natural disaster monitoring, border patrols, emergency assistance, search and rescue missions, and cargo delivery. At the same time, the UAV swarm can also be deployed as an aerial base station. A leading UAV serves as a mobile control station for command and information gathering tasks. The surrounding member UAVs execute commands. Such a team has the ability to execute ordinary equipment, but cannot complete the task [2]. As the number of UAVs increases,

UAV networks can provide collaboration strategies, but there are still challenges in airborne networks in terms of information sharing and information security in airborne networks. Fast-moving nodes will produce constantly changing topologies in the network, leading to frequent disconnections and interruptions in the network; in an airborne network, each node is regarded as an autonomous embedded system with local- and network-level resource constraints. The system, as the autonomous system develops through interaction with the environment, also becomes vulnerable to malicious control by competitors [3]. The application of the UAV swarm in the military field is more and more popular as the demands grow more and more. In particular in terms of missions on collecting sensitive information or confidential tasks, the security and integrity of mission resource assignment on UAV during a flight are still a problem needed to be resolved [4, 5]. Not only does the content of the mission resources need to be protected but also the transmission data.

Therefore, a security strategy is bound to be needed, which can protect the integrity of its embedded system and allow it to be verified to make the system trustworthy. Although there are many researches on the safe communication and information sharing of UAVs, most of them do not take into account the effective limited computing power of UAVs, or the high latency cannot obtain better efficiency in the field of military applications. The flying mobile ad hoc network is a complex system. There is much room for improvement from the physical and network layers. Therefore, it is necessary to design and implement a large-scale UAV swarm security certification model to effectively protect the database security and wireless transmission security during the mission of the UAV. In this paper, we attach labels to every certain mission resource, and we propose the stream cipher-based database security approach to protect the content of the mission in the beginning and the two-hash-function-based security model for the process of transmission to prevent the system from intercepting or eavesdropping. The UAV swarm can be divided into several task units during the execution of tasks; each unit contains the following:

- (i) A leader UAV plays the role of commanding, collecting, and distributing messages. It is equivalent to a mobile ground station, with stronger computing capabilities, and also acts as the core of the entire team
- (ii) Several surrounding member UAVs as the relay nodes are responsible for receiving the leader's messages and forwarding data packets, keep in touch with the leader, and are verified by the leader every certain period
- (iii) The peripheral UAV followers around the relay nodes are responsible for receiving messages from the relay nodes and are verified by the relay nodes in a higher frequency

In the process of performing the mission, the swarm will inevitably encounter problems such as accidental damage or being attacked by malicious UAVs during the flight and eavesdropping. Therefore, the safety of the leader of the UAV is imperative. If the leader is compromised, then the entire system will fall short. Therefore, in order to protect the security of the leader, we adopt the RC4 lightweight encryption scheme to make sure that the leader's database is no longer exposed to the enemy's line of sight, and the followers around the leader cannot be trusted fully; in order to make their status always clearly informed by the system, we use two hash functions to protect the transmission data privacy, and label verification is done every certain period to ensure the reliability of the follower's identity. This paper takes the safety and energy consumption limits of UAV swarms into account and uses SHA-256 and lightweight hash function—SPONGENT-128—to protect the security of the system to the greatest extent, while reducing the probability of malicious attacks and information acquisition.

We have done a lot of research and found that most of the existing UAV security solutions cannot take the balance between task efficiency and energy consumption into consideration. Some related experts and scholars have proposed

some encryption protocols, such as LCAP protocol [6] and hash lock [7]. However, these protocols have hidden security risks of being attacked in the network. For these problems, the idea of a random label was utilized to build a security authentication model for a large-scale UAV swarm in our previous work [8], where a security authentication mechanism based on a random label was proposed, and the processing logic of data packets in the security authentication process of the UAV swarm was explained. Different from [8], we propose and implement a UAV swarm security authentication mechanism based on the random label and lightweight hash in this paper, so that the information sent by the leader can be encrypted by the RC4 encryption scheme before being distributed, which protects the database of the system. After the member UAVs receive the information, they will decrypt it firstly and then generate random labels corresponding to SHA-256 (hash-1) or SPONGENT-128 (hash-2) to verify their identity. Since the whole system contains mostly UAVs with calculation limitation, we try to adapt lightweight hashing schemes. And the verification between the systems is completed by irreversible hashing correspondingly to generate random labels that are difficult to decipher. Moreover, in order to prevent the enemy from using the same encryption scheme to detect the decryption method, the leader changes the custom key every other period of time. In this way, even if the enemy intercepts this information, there is no way to know the decryption scheme, which greatly reduces the probability of the UAV swarm being infiltrated without increasing the consumption of a lot of energy. Experimental results show that the model enables the entire UAV system to achieve better performance in terms of ensuring robustness and task completion efficiency.

The main contents of our work are as follows:

- (i) This paper introduces the random labels, and uses a certain label to generate and verify uncertain label to ensure the security of UAV swarm system
- (ii) A security model of large-scale UAV swarms based on random labels is proposed. A lightweight encryption scheme is used to protect the security of the database. For UAVs with complex computing capabilities, an improved hash function is used for label generation and verification. For UAVs with weak computing power, a lightweight hash function generation and verification is proposed to protect the safety of data transmission
- (iii) The safety analysis of the proposed scheme is carried out, and the analysis results show that the model proposed in this paper has good safety performance
- (iv) A simulation experiment was carried out on the proposed safety model, and its performance was compared with the existing mechanism extensively. The simulation results show that the proposed scheme can achieve better performance in terms of task efficiency and privacy protection

The rest of the paper is organized as follows. The related works are illustrated in Section 2. The preliminaries and

system model are presented in Section 3. Section 4 describes the design details of our proposed secure mechanism for mission resource allocation and the safety analysis of the system. Simulation results and analysis are illustrated in Section 5. Finally, important conclusion is drawn in Section 6.

2. Related Works

The main security threats faced by UAVs include [9] wireless signal hijacking and jamming, sensor network attacks, and GPS spoofing [5].

Since the communication of the UAV is open, it is easy for the enemy to attack and interfere with the wireless signal, and this can directly affect the normal operation of the UAV. For this kind of attack, [5] proposed the eCLSC-TKEM communication protocol based on the key encapsulation mechanism; [10] constructed a UAV monitoring system to determine whether the UAV has been infiltrated and is not under someone else's control; [11] proposed two private communication methods that can be used in a mobile ad hoc network for UAV flight. The security of the wireless sensor network composed of a UAV swarm also has great vulnerabilities, such as database security and the security of the wireless transmission process, which are easily intercepted, cracked, analyzed, and attacked by the enemy. There are some solutions for UAV communication security in the existing literature, which are mainly implemented through physical layer security, secure routing, and security algorithms.

Physical layer security has studied several technologies to realize the positive privacy rate of the UAV communication system, such as artificial noise [12], power control, k -anonymity algorithm for UAV location security [13], flexible location privacy protection [5], and the construction of protected areas [14]. Secure routing enables FANET to transmit information in a more secure routing, such as that based on a geographic location algorithm [15, 16], based on a multipath planning algorithm [17, 18], based on a swarm secure routing protocol algorithm [19], based on a security cryptographic algorithm [18, 20], and based on an intelligent algorithm [21]. The security algorithm encrypts data and authenticates the identity of the UAV to achieve more security of FANET. For example, the ChaCha20 encryption algorithm and Poly1305 for encryption authentication have become popular alternative methods in the industry to perform advanced encryption (AE) [22].

In addition, the attacker can also forge GPS signals to deceive the UAV's GPS, causing the UAV's navigation system or position coordinates to point to an offset position [23, 24], causing the UAV to fly to the wrong target location, affecting the efficiency of task completion. However, most of the existing work focuses on UAV-to-ground systems in a two-dimensional space, and there is only one communication model from the ground to the air. The security of the UAV-UAV (A2A) system in a three-dimensional (3D) space will be more complicated, because the receiver or eavesdropper works in all directions. At present, the three-dimensional security model has not been well studied and understood, leaving an open question. At the same time, because of the

limited computing power of UAVs, most of the existing security algorithms have high latency and high computational complexity. This paper requires a lighter and faster security strategy to ensure the safety of the UAV swarm system.

3. Preliminaries and System Models

3.1. System Model. After taking off from the ground station, the UAV swarm may be disconnected. At this time, the swarm is divided into several small mission units, and each mission unit has a leader equipped with higher ability to carry out complicated calculation and analysis. The leader collects the information sent by the following UAVs, distributes new packets, and acts as the mobile ground station. As shown in Figure 1, a circle of UAVs around the leader is relay nodes (leader surrounding); they are responsible for collecting the information sent by the leader and forwarding the data packet to the peripheral follower. The relay nodes directly communicate with the leader and directly undergo the verification by the leader, but the relay node needs to verify the identity of the more peripheral members. Therefore, we allow the leader UAV to have higher computing power. They are more capable of processing more complex computing logic than the member UAVs that follow. But the computing power of the relay nodes and peripheral UAV is limited, and in order to save energy consumption, we require a lighter solution to support the more frequent authentication process [25, 26].

First of all, in order to prevent the leader from attracting firepower due to system centralization and to ensure the security and credibility of the leader's database, the certain label issued by the leader needs to be simply symmetrically encrypted through RC4 to play the role of query protection, and the key of the symmetrically encrypted plan usually only need to be XORed once to get the original text without causing excessive energy consumption; secondly, in order to ensure data security during the wireless transmission of the entire system, similar to the work [27, 28], in addition to the certain label (clabel) used to track the conditions of the information packet, under this model, we introduce an uncertain label (ulabel) to allow the UAV to prove its identity [26].

When the relay node receives the data packet sent by the leader, it needs to decrypt and get the certain label first and then use the improved SHA-256 scheme to generate the corresponding uncertain label (ulabel) and send it back to the leader every certain period to verify the identity. At the same time, the data packet is forwarded to the next hop, which is the peripheral follower UAV. After the peripheral UAV receives the data packet, it decrypts the clabels first, then uses the lightweight SPONGENT-128 scheme to generate an uncertain label (ulabel), and then transmits it back to the relay nodes for identification. In the verification process, since the hash is irreversible, the verifier needs to use the same scheme to generate a random label for result comparison. If they are consistent, the verification is passed; otherwise, the system triggers an alarm.

When the UAV swarm takes off from the ground control station, it disconnects from the ground station. At this time, the leading UAV will assign clabel to each member UAV for real-time tracking and identity verification. Because the

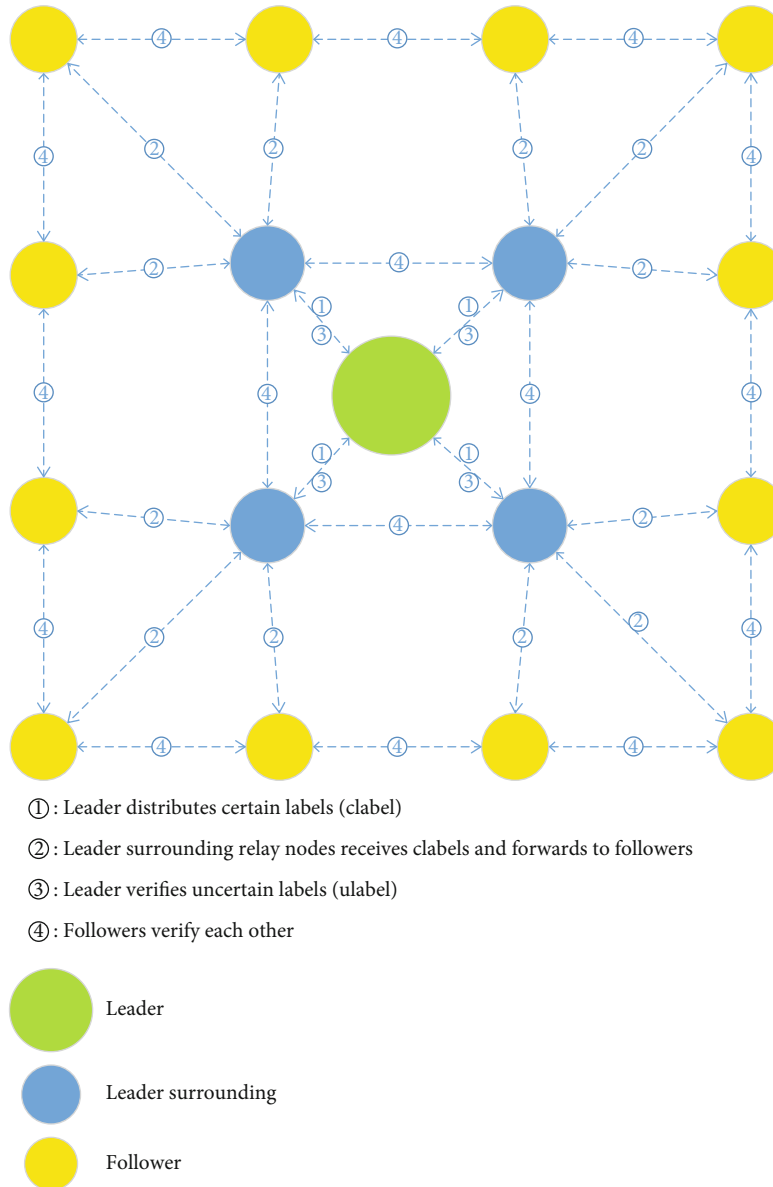


FIGURE 1: System model.

safety of the leading UAV is of utmost importance, in order to ensure the safety of the clabel issued, the leading UAV first processes the clabel with a lightweight stream cipher encryption scheme and then forwards it. In this way, even if the enemy breaks through the leading UAV, they cannot know the accurate clabel, which reduces the probability of the enemy attacking the leading UAV to a certain extent. Meanwhile, the lightweight encryption scheme will not bring a lot of energy to the system consumption.

3.2. Stream Cipher. A stream cipher is a common encryption algorithm based on XOR operation, which only operates one byte at a time. Stream cipher encryption is a widely used algorithm in lightweight encryption algorithms. It has fast speed and is convenient for hardware to realize along with the advantages of less memory and less error propagation. One

of the most representative algorithms is the Rivest Cipher (RC4) algorithm [29]. Taking into account the energy consumption of UAVs, we use this lightweight encryption algorithm to protect the safety of the clabel, thereby protecting the safety of the leading UAV.

3.3. SHA-256. SHA (Secure Hash Algorithm) is a series of cryptographic hash functions designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST), including SHA-1, SHA-224, and SHA—variants such as SHA-256, SHA-384, and SHA-512. It is mainly applicable to the Digital Signature Algorithm (DSA) defined in the Digital Signature Standard (DSS). Because the security of SHA-1 has been seriously questioned because it has been cracked, but the security of the SHA-2 series can still resist most attacks, considering the impact of the length

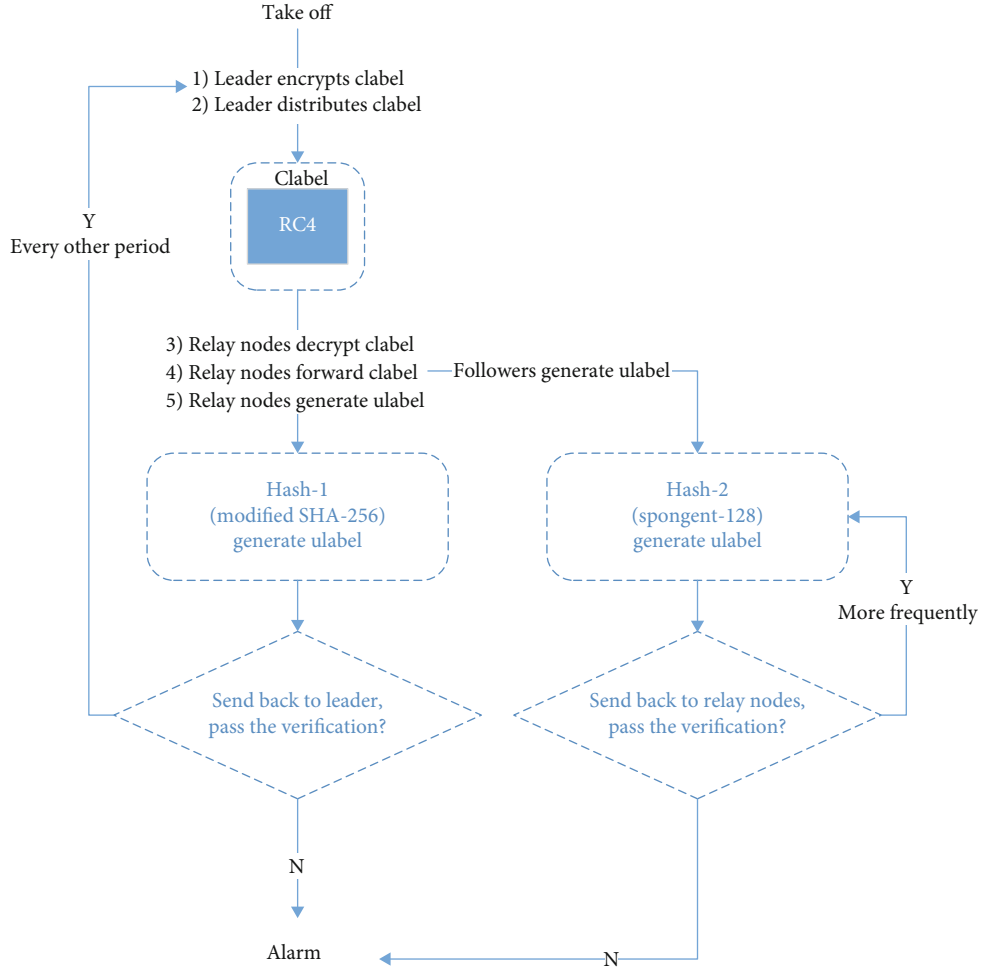


FIGURE 2: Mechanism workflow.

of the hash value and the number of calculation cycles on energy consumption and UAV, we use the SHA-256 hash algorithm for processing.

3.4. SPONGENT-128. In the existing Internet of Things (IOT) applications, a hash function with a collision resistance of 280 is acceptable to reduce overhead to the greatest extent in a safe environment. According to the different construction methods used in the function design, the design principles of the lightweight hash functions that have been publicly published can be divided into three categories: (1) based on permutation functions, (2) based on block ciphers, and (3) based on mathematical difficulties. We use a lightweight PRESENT-like block cipher permutation function SPONGENT-128 [30] to solve the ulabel generation by high-frequency verification between the followers.

The domain extension structure used by SPONGENT-128 is the sponge structure, which can compromise encryption speed, memory requirements, and security by adjusting parameters [31]; the internal transformation uses an 8-round block cipher transformation. Through the three stages of initialization, absorption, and squeezing, five different output length example applications can be realized through different combinations of rate and capacity. The sponge structure is mainly

composed of three components: (1) filling function P , (2) memory state S , and (3) transfer function F . Among them, the memory state S has a total of b bits, including two parts, the data rate r and the capacity c , and has an output of n bits ($n = c$). We set the parameters $r = 8$, $c = 128$, $n = 128$, and $b = 128$.

4. Random Label and Lightweight Hash-Based Security Authentication Mechanism Design

The previous section proposes several important models and definitions in the security encryption model of UAV swarms based on random labels. This section discusses their specific design schemes and implementations based on these definitions.

Considering that each UAV is an agent with computing and decision-making capabilities, the UAV can be considered the result of multiagent calculation and collaboration. After takeoff, the leading UAV acts as a mobile ground station and needs to perform tasks such as information collection and command, and safety is particularly important. In the entire process of performing the task, the task of safety certification mainly includes two aspects: one is the label generation and verification between the relay node and the peripheral member UAV and the other is the member UAV and the leader UAV label generation and verification. In order to ensure

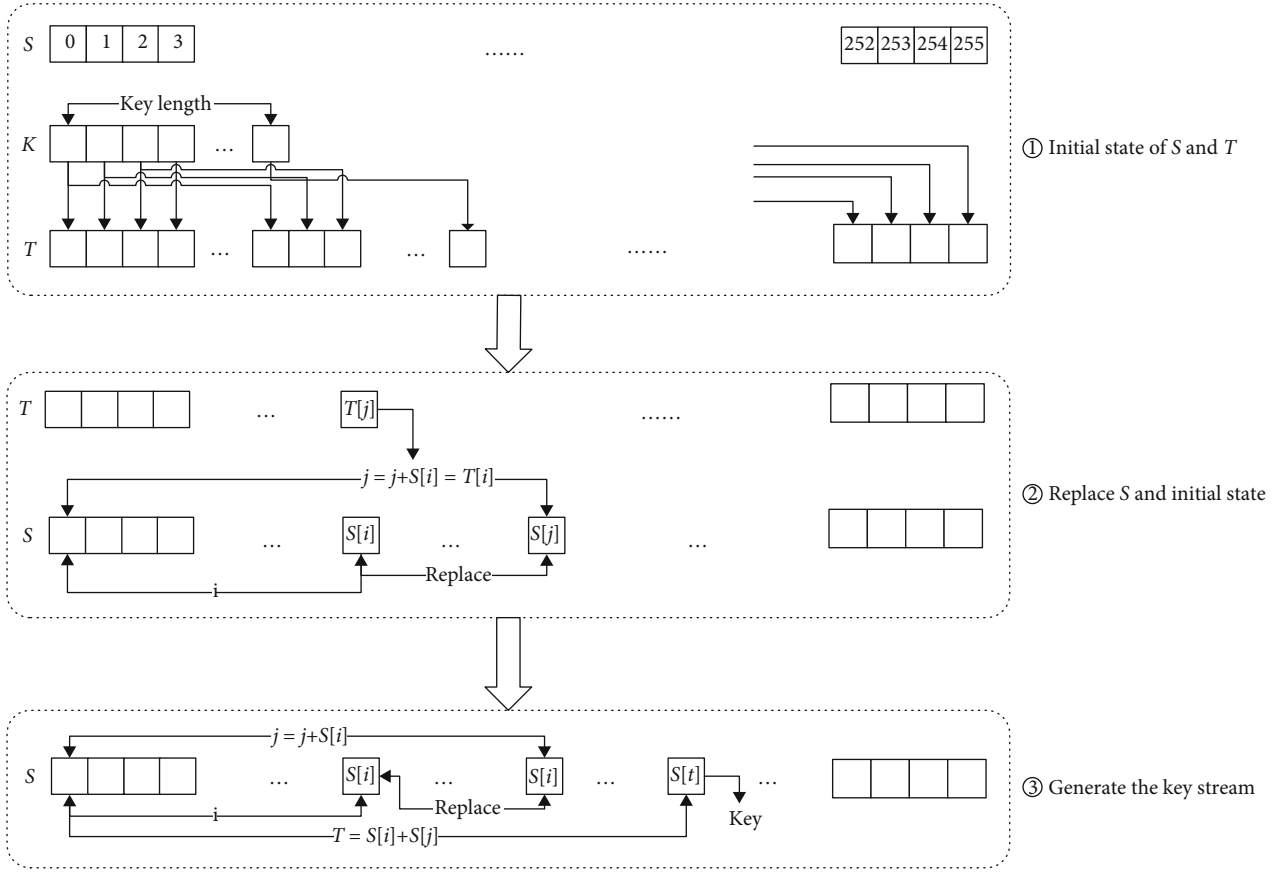


FIGURE 3: The process of RC4 encryption.

the safety of the leading UAV and to comply with the UAV's energy consumption limits and computing capabilities, two solutions have been specially introduced to ensure the feasibility of this model. The flow of this mechanism is discussed in Figure 2.

4.1. Encrypt the Certain Labels. In this part, we elaborate on the label processing logic of the UAV swarm system under the security encryption model based on random labels. After taking off, the leading UAV will distribute a label to followers in order to ensure the integrity of the surrounding UAVs and track their location. However, once the label is intercepted by the enemy or the leading UAV is once attacked, the entire system falls short. Besides, the label runs through the whole process of completing the task of the entire system. Therefore, in order to reduce the probability of the enemy attacking the leading UAV and also reduce the probability of the enemy intercepting the certain label, we firstly perform a lightweight encrypt algorithm to enhance the security of UAV data packets before issuing. Figure 3 is the detailed process of generating an encrypted label.

4.1.1. Generate Key Stream Seed 1. Given a state vector S , initialize it and assign 0, 1, 2, ..., 254, 255 to each byte in ascending order.

4.1.2. Enter the Initial Key. The initial key is defined by ourselves as any combination of less than 256 bytes and then

filled in circularly until 256 bytes are filled, and then, the final result obtained is defined as a vector T to generate seed 2.

4.1.3. Disrupt Initial Seed 1. Perform a replacement operation on the state vector S , starting from the 0th byte and executing 256 times to ensure that each byte is processed.

4.1.4. Generate Key Stream and Encrypt. After the leader encrypts the label, it starts distributing. After each member UAV receives it, they start decryption, because the lightweight encryption scheme we use is symmetric encryption, and the original text is the same after being XORed twice with the same key. Thus, knowing the key means knowing the way to the original text. Of course, the key will be changed regularly so that malicious ones will not be able to detect the key. Finally, the original text is obtained, so the encrypted label UAV only needs to use the key stream to XOR one more time to obtain the original text and then perform subsequent operations. The relay nodes forward the identification label to the follower UAVs because they are relatively trustworthy. The complete process of generating LDPC-based multiuser superimposed information ciphertext is shown in Algorithm 1.

4.2. Between Leader and Leader Surrounding Relay Nodes. We focus on the modified SHA-256-based algorithm adapted to label generation and verification between the leader and relay nodes in this part.

```

1: Generate key stream seed 1:
2: Given a state vector S, initialize it, and assign 0, 1, 2, ..., 254, 255 to each byte in ascending order;
3: for  $i = 0; i < 256; i++$  do
4:    $S[i] = i;$ 
5: end for
6: Enter the initial key:
7: The initial key is defined as any combination of less than 256 bytes;
8: Fill in circularly until 256 bytes are filled;
9: The final result is defined as a vector T, which is used to generate seed 2;
10: Disrupt the initial seed 1:
11: Perform a replacement operation on the state vector S, starting from the 0th byte, and executing 256 times to ensure that each byte is processed;
12:  $j = 0;$ 
13: for  $i = 0; i < 256; i++$  do
14:    $j = (j + S[i] + T[i]) \bmod 256;$ 
15:    $\text{swap}(S[i], S[j]);$ 
16: end for
17: Generate key stream and encrypt:
18:  $i = 0, j = 0;$ 
19: while  $\text{clabel.length} \neq 0$  do
20:    $i = (i + 1) \bmod 256;$ 
21:    $j = (j + S[i]) \bmod 256;$ 
22:    $\text{swap}(S[i], S[j]);$ 
23:    $t = (S[i], S[j]) \bmod 256;$ 
24:    $k = S[t];$ 
25:    $\text{clabel}[] = \text{clabel}[] \oplus k;$ 
26: end while

```

ALGORITHM 1: The encryption algorithm of RC4.

In order to prevent the enemy from using the exhaustive method to crack our encryption method, we made a certain degree of improvement to SHA-256—intercept a piece of data in the final message digest H_n and discard it, and then, fill the discarded data with a random function, and the whole process would not change the number of bits after SHA-256 encryption. The difficulty of cracking continues to increase without causing excessive energy consumption.

Under this model, the UAV data packet is enhanced through two main processes of label generation and label verification. The generation and verification of the label are highly coalesced, and only the labeled data packets need to be subjected to label verification. By setting the unused bit area to zero by default, we can distinguish marked data packets from unmarked data packets, so as to determine which data packets need to be verified. Data packets that do not require verification indicate that the UAV does not carry label when it departs from the ground station, and the system can directly trigger an alarm. Below is the detailed process of label generation and verification.

4.2.1. Label Generation. To generate a ulabel for the swarm boils down to the hashing process. In order to simplify the configuration, we can use the same hash parameter on the UAV. We let Pkt^{clabel} represent the data packet header with clabel and $\text{Sample}(Pkt^{\text{clabel}})$ the bit marked in it, and the length of which is represented by a 64-bit value. The ulabel to be generated by the UAV is formulated as follows:

$$\text{ulabel} = \text{Hash}\left(\text{Sample}\left(Pkt^{\text{clabel}}\right)\right), \quad (1)$$

where $\text{Hash}(\cdot)$ is the adopted SHA-256 algorithm. We adapt the SHA-256 algorithm to generate the ulabel for the packet header of UAV. First, we preprocess the $\text{Sample}(Pkt^{\text{clabel}})$, which is to add the needed information after the message to be hashed; second, we generate the message list and use logical functions to calculate the message digest. It takes four steps to completely generate a ulabel: (a) initiating constants, (b) adding filling bit and length value, (c) calculating the message digest (the ulabel), and (d) modification.

4.2.2. Initiating Constants. In this algorithm, we use 8 initial hash values and 64 hash constants. The 8 initial hash values forming the initial mapping value H_0 are shown in Figure 4.

These initial values are the first 32 bits of the decimal part of the square root of the first 8 prime numbers (2, 3, 5, 7, 11, 13, 17, 19) in natural numbers. For example, the decimal part of $\sqrt{2}$ is approximately 0.414213562373095048, and

$$0.414213562373095048 \approx 6 \times 16^{-1} + a \times 16^{-2} + 0 \times 16^{-3} + \dots \quad (2)$$

So, the first 32 bits of the decimal of $\sqrt{2}$ is 0x66a09e667, that is, the h_0 . The 64 hash constant values are shown in Figure 5.

1	h1	:=	0×6a09e667
2	h2	:=	0×bb67ae85
3	h3	:=	0×3c6ef372
4	h4	:=	0×a54ff53a
5	h5	:=	0×510e527f
6	h6	:=	0×9b05688c
7	h7	:=	0×1f83d9ab
8	h8	:=	0×5be0cd19

FIGURE 4: The hash initial value of SHA-256.

Similar to the 8 initial hash values, these constants are derived from the first 32 bits of the decimal part of the cube root of the first 64 prime numbers (2, 3, 5, 7, 11,13,17,19,23,29 ...) in natural numbers.

4.2.3. Add Filling Bit and Length Value. Add filling bits to the end of the $\text{Sample}(Pkt^{\text{clabel}})$ so that the remainder of the $\text{Sample}(Pkt^{\text{clabel}})$ length is 448 after modulating 512. The first additional bit is 1 and then 0, until the length of 512 is sufficient and the remainder is 448. Then, we get $\text{Sample}(Pkt^{\text{clabel}})_1$. After that, we append $\text{Sample}(Pkt^{\text{clabel}})$ with $\text{Sample}(Pkt^{\text{clabel}})_1$, which gives us the message.

4.2.4. Calculating the Message Digest (ulabel). Since the minimum calculation unit of the algorithm is a 32-bit word, we break the message into sixteen 32-bit big-endian words W_0, W_1, \dots, W_{15} and other 48 words $W_{16}, W_{17}, \dots, W_{63}$, which are derived from the following iterative formula, respectively:

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-2}) + W_{t-16}. \quad (3)$$

For each iteration, as shown in Figure 6, the 8 letters ABCDEFGH are updated according to certain rules, and the dark blue squares are 6 nonlinear logic functions. For the convenience of expression and calculation, we will express these words as x, y, z , and the functions are shown below.

$$\begin{aligned} \text{Ch}(x, y, z) &= (x \wedge y) \oplus (x \wedge z), \\ \text{Ma}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), \\ \sum_0(x) &= S^2(x) \oplus S^{13}(x) \oplus S^{22}(x), \\ \sum_1(x) &= S^6(x) \oplus S^{11}(x) \oplus S^{25}(x), \\ \sigma_0(x) &= S^7(x) \oplus S^{18}(x) \oplus S^3(x), \\ \sigma_1(x) &= S^{17}(x) \oplus S^{19}(x) \oplus S^{10}(x). \end{aligned} \quad (4)$$

The red blocks stand for the operation of mod 2^{32} addition, which means when the adding result is greater than 2^{32} ,

1	428a2f98	71374491	b5c0fbcf	e9b5dba5
2	3956c25b	59f111f1	923f82a4	abl5ced5
3	d807aa98	12835b01	243185be	550c7dc3
4	72be5d74	80deb1fe	9bdc06a7	c19bf174
5	e49b69c1	efbe4786	0fc19dc6	240ca1cc
6	2de92c6f	4a7484aa	5cb0a9dc	76f988da
7	983e5152	a831c66d	b00327c8	bf597fc7
8	c6e00bf3	d5a79147	06ca6351	14292967
9	27b70a85	2e1b2138	4d2c6dfc	53380d13
10	650a7354	766a0abb	81c2c92e	92722c85
11	a2bfe8a1	a81a664b	c24b8b70	c76c51a3
12	d192e819	d6990624	f40e3585	106aa070
13	19a4c116	1e376c08	2748774c	34b0bcb5
14	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
15	748f82ee	78a5636f	84c87814	8cc70208
16	90befffa	a4506ceb	bef9a3f7	c67178f2

FIGURE 5: The hash constant value of SHA-256.

you must divide it by 2^{32} and find the remainder. The initial values of ABCDEFGH are $H_{i-1}(0), H_{i-1}(1), H_{i-1}(2), \dots, H_{i-1}(7)$. For each iteration, let Kt denote the key over the t -th iteration, corresponding to the 64 constants we mentioned before, and Wt is the t -th word this block generates. The original message is cut into fixed-length 512-bit blocks; for each block, 64 words are generated, and the eight letters ABCDEFGH are cyclically encrypted by repeatedly running the cycle n times. The eight characters generated in the last loop are the hash string Hn corresponding to the n th block, which is the last 256-bit message digest. The above algorithm of the iteration process is presented in Algorithm 2.

4.2.5. Modification

- (1) Use the interception function to intercept the encrypted ciphertext (Hn), and intercept the number ($0 < \text{number} < 32$) digits from the beginning number ($0 < \text{beginnumber} < 32$) position to obtain the password A , where $A = \text{left}(\text{SHA} - 256(\text{password}), \text{beginnumber} - 1)$
- (2) Use the intercept function to intercept the value B of the number digits of the encrypted plaintext, where $B = \text{right}(\text{SHA} - 256(\text{clabel}), \text{SHA} - 256 - \text{digit} - (\text{beginnumber} + \text{number} - 1))$

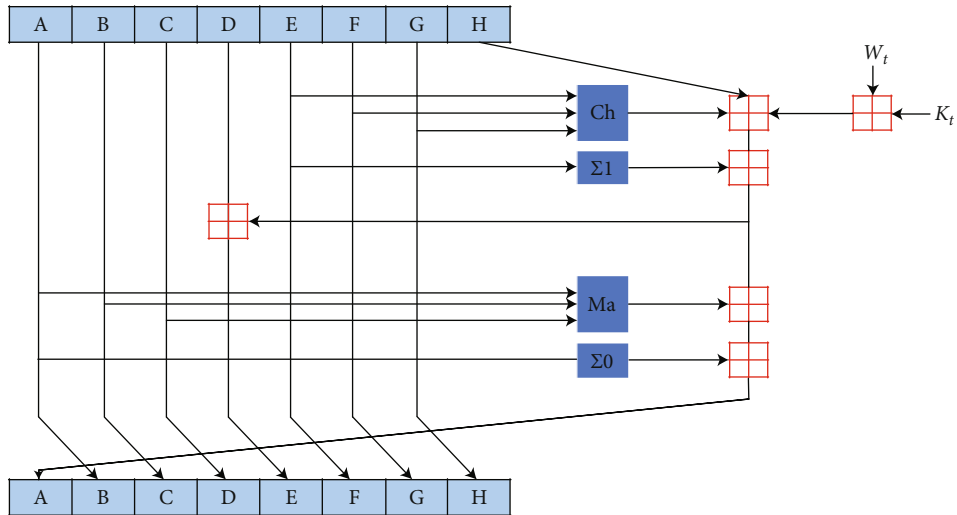


FIGURE 6: The iteration process of SHA-256.

```

1: for  $i$  from 0 to 63 do
2:   Logical function operation:
3:    $S_0 = (A \text{ rightrotate } 2) \wedge (A \text{ rightrotate } 13) \wedge (A \text{ rightrotate } 22)$ ;
4:    $Ma = (A \text{ and } B) \wedge (A \text{ and } C) \wedge (B \text{ and } C)$ ;
5:    $t_2 = S_0 + Ma$ ;
6:    $S_1 = (E \text{ rightrotate } 6) \wedge (E \text{ rightrotate } 11) \wedge (E \text{ rightrotate } 25)$ ;
7:    $Ch = (E \text{ and } F) \wedge (\text{not } E) \wedge G$ ;
8:    $t_1 = h + S_1 + Ch + K[i] + W[i]$ ;
9:   Update each word:
10:   $H = G$ ;
11:   $G = F$ ;
12:   $F = E$ ;
13:   $E = D + t_1$ ;
14:   $D = C$ ;
15:   $C = B$ ;
16:   $B = A$ ;
17:   $A = t_1 + t_2$ ;
18:  Add the hash output of the message block to the existing hash output:
19:   $h_0 = h_0 + A$ ;
20:   $h_1 = h_1 + B$ ;
21:   $h_2 = h_2 + C$ ;
22:   $h_3 = h_3 + D$ ;
23:   $h_4 = h_4 + E$ ;
24:   $h_5 = h_5 + F$ ;
25:   $h_6 = h_6 + G$ ;
26:   $h_7 = h_7 + H$ ;
27:  Output the final hash value (big-endian):
28:   $\text{digest} = \text{hash} = h_0 \text{ append } h_1 \text{ append } h_2 \text{ append } h_3 \text{ append } h_4 \text{ append } h_5 \text{ append } h_6 \text{ append } h_7$ ;
29: end for
30: return  $\text{digest}$ ;

```

ALGORITHM 2: The iterative algorithm of SHA-256.

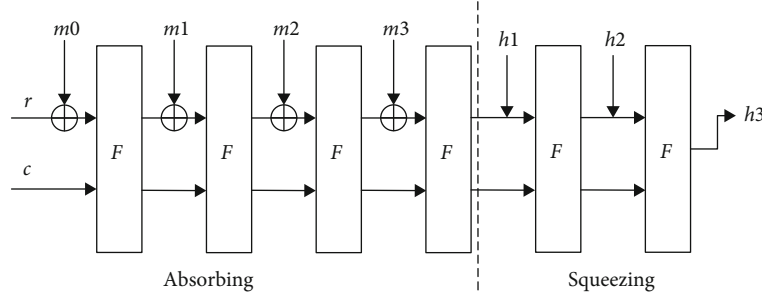


FIGURE 7: Absorbing and squeezing phases of spongy structure.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	E	D	B	0	2	1	4	F	7	A	8	5	9	C	3	6

FIGURE 8: The state of S-box.

- (3) Use the random function $\text{gen}_{\text{key}}(\text{number})$ to fill in the value of the intercepted number; the converted password value is $\text{encrypt}_{\text{password}} = A \& \text{get}_{\text{key}}(\text{number}) \& B$

From the above, we get the final message digest (ulabel). This process randomly hashes the input to the output, which makes reverse engineering using input-output pairs difficult.

4.3. Label Verification. The authentication between the leader and relay nodes will seem successful as long as the following two conditions are met: first, the data packet comes from the correct front-hop node (UAV) and second, the data packet should carry the correct ulabel generated by the UAV.

- (1) The leader UAV verifies whether the clabel of the member UAV is correct, and if correct, it indicates that the data packet comes from the correct previous hop node
- (2) The leader UAV also generates an Hn according to the received clabel and then intercepts the first half from the beginning number to get A' , and the second half gets B' and then reads the A and B parts of the password from the database, and finally if $A = A'$ and $B = B'$, it is considered that the password entered by the user matches the password in the database, indicating that $\text{ulabel}' = \text{ulabel}$, and the verification is passed. Otherwise, the label verification fails and the UAV triggers an alarm

4.4. Among Followers. We focus on the lightweight SPONGENT-128-based algorithm adapted to label generation and verification among the followers in this part.

4.4.1. Label Generation. After the follower UAV receives the clabel encrypted by the leader, it first decrypts the RC4 to obtain the original clabel. Because the lightweight encryption scheme we use is symmetric encryption, the original text is obtained by XORing the plaintext with the same key twice,

so the encrypted clabel UAV only needs to use the keystream to XOR once again to get the original text.

The encryption process of the hash function using the sponge structure can be finished through three stages [30, 32]:

Initialization Phase. The b -bit memory state S is initialized to all 0s, and the plaintext message clabel is filled with the filling function P . As shown in Figure 7, the specific method is to first add 1 after the message, and then, add a sufficient number of 0s, so that the number of bits of the message after filling is exactly r an integer multiple (for example, when the message has 57 bits and $r = 8$, the filled content should be "1000000").

Absorbing Phase. The filled message will be divided into r message blocks (m_i). The message block is XORed to the rightmost r -bit position of state S , which forms part of the hash output. Each time the r -bit message is XORed with the first r -bit of S , then the entire S is converted by F to obtain the new state and then repeat the above steps (absorption), until all the filled messages are absorbed.

F conversion: $F_2^b \rightarrow F_2^b$. It is an R -round block cipher conversion of the input b -bit state. $s\text{BoxLayer}_b$ and $p\text{Layer}_b$ describe how the state evolves. $l\text{Counter}_b(i)$ is the state of the LFSR at time i that depends on b . It generates an integer constant for round i and adds it to the rightmost bit of the state. $l\text{Counter}_b(i)'$ is the value of $l\text{Counter}_b(i)$, whose bit order is reversed and added to the leftmost bit of the state. Generally speaking, $s\text{BoxLayer}_b$ means to use a 4-bit to 4-bit S-box: $F_2^4 \rightarrow F_2^4$, distributed and operated $b/4$ times. S-box (substitution-box) is the basic structure of the symmetric key algorithm [33] to perform substitution calculation. The state of S-box is shown in Figure 8.

Besides, $p\text{Layer}_b$ means to move j in S state to position $P_b(j)$:

$$P_b(j) = \begin{cases} \frac{j \cdot b}{4 \bmod b - 1}, & \text{if } j \in \{0, \dots, b-2\}, \\ b-1, & \text{if } j = b-1. \end{cases} \quad (5)$$

TABLE 1: The ability of SPONGENT-128 against differential attack.

Rounds	ASN	Maximum difference probability
5	10	2^{-22}
10	29	2^{-68}

Counter***b*** is one of the three $\log_2 R$ bit linear feedback shift registers (LFSR). The initial value assigned to the register is called the “seed”, because the operation of LFSR is deterministic, so the data flow generated by the register is completely determined by the current or previous state of the register. Moreover, since the state of the register is limited, it will definitely end up in a repeated loop. However, through primitive polynomials, LFSR can generate seemingly random sequences with very long cycles. Shift registers are simple in structure and fast in operation. Most practical key stream generators are based on shift registers, and shift register theory has become the basis of modern stream cryptosystems. In this scheme, the LFSR is recorded every time the *S* state is used, and its final value is 1. Let ζ denote the unit root in the corresponding binary finite field. The SPONGENT-128 we use has the original trinomial $\zeta^7 + \zeta^6 + 1$ 7-bit LFSR, which is initialized to “1111010.”

Squeezing Phase. After all the messages are absorbed, the first r bits of *S* are output. If the number of digits to be output is $n > r$, then continue to perform *F* transformation on *S* and then output the first r bits of *S* and so on until the total number of digit output reaches n bit.

4.4.2. Label Verification. Because the followers are semi-trusted to us, we need to authenticate the follower at a higher frequency. This authentication is verified by the relay node according to SPONGENT-128 to verify the ulabel returned by the follower. If the result is the same as the returned one and if they are consistent, the follower is credible. If they are inconsistent, the relay node will trigger an alarm.

In summary, after taking off, the leader first performs RC4 encryption to the clabel and then distributed them to the relay nodes, and the relay nodes forward to the peripheral member UAV followers to ensure the security of the database; after the relay node decrypts the clabel, they adapt SHA-256 to generate ulabels and are verified by the leader every certain period to ensure the safety of relay nodes during the flight; after receiving the forwarded clabels, followers adapt SPONGENT-128 to generate ulabels to finish the verification cycle faster, and the ulabel sent back to the relay node for verification indicates that its identity guarantees the safety of the peripheral member UAVs; the security of the information in the wireless transmission process is protected through two hash schemes, and there is no too much additional energy consumption.

Therefore, our proposed scheme enables the inside of the UAV swarm to use labels that may be unknown to the attacker to mark the data packet header. In addition, the labeling strategy is synchronized between the leading UAV and the following UAV for data packet verification. If a malicious UAV attacker manipulates a follower UAV, it must forge the correct label to avoid verification.

4.5. Safety Analysis. This part analyzes the security model proposed in this paper from mainstream attack methods such as traditional attacks and differential attacks.

4.5.1. Security Assumptions. Cryptography has defined three security assumptions about hash functions. If a hash function satisfies the following three assumptions, such a hash function can be considered safe [34]:

- (i) The original image having stable assumption: choose output value h arbitrarily, and make $H(M) = h$ by finding a message string M which is infeasible
- (ii) The second preimage firm assumption: given a message string M , find another message string Z to make $H(M) = H(Z)$ which is infeasible
- (iii) Assumption of collision stability: it is infeasible to find any two message strings M and Z such that $H(M) = H(Z)$.

4.5.2. Traditional Attack. From the perspective of attack principles, traditional attacks do not use the structure of the hash function and any weak algebraic properties and are only affected by the length of the hash value. Therefore, the most effective way to resist traditional attacks is that the hash value must have sufficient length. The SHA-256 used in this paper uses a 256-bit hash value, and its magnitude is $2^{128} \approx 3.4 * 10^{38}$, and it takes several million years to find a collision [7].

According to the literature [30], for the verification between the relay node and the UAV, it can be seen that the domain extension structure of the sponge structure has good safety performance in a large-scale and lightweight environment. Therefore, this paper adopts the domain extension structure of SPONGENT-128 verifying the integrity between UAVs and analyzing the cost of mainstream attack types to break this scheme:

- (1) The range of costs to be paid for a collision attack is

$$\min \{2^{n/2}, 2^{c/2}\}. \quad (6)$$

- (2) The range of costs to be paid for the preimage attack is

$$\min \left\{ 2^n, 2^c, \max \left\{ 2^{(n-R)/2}, 2^{c/2} \right\} \right\}. \quad (7)$$

- (3) The range of costs to be paid for the second preimage attack is

$$\min \{2^n, 2^{c/2}\}. \quad (8)$$

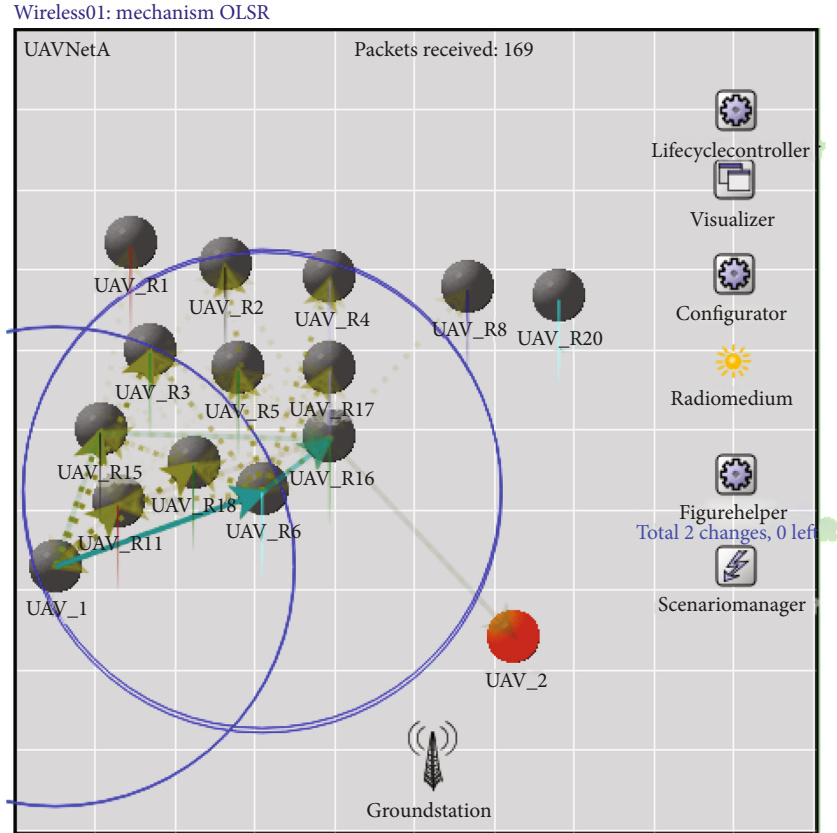


FIGURE 9: OMNeT++ simulation interface.

Set the main parameters in the sponge structure: output digits $n = 128$, memory state digits $b = 136$, capacity $c = 128$, data rate $r = 8$, and output digits in the extrusion stage $R = 8$, so the solution in this paper resists traditional attacks. The capabilities are as follows:

- (i) Collision attack resistance: 264
- (ii) Preimage attack resistance: 2120
- (iii) The second preimage attack resistance: 264

4.5.3. Differential Attack. Differential attack is currently one of the most effective methods to decipher the iterative hash function. The basic method is to use the impact of the input difference of the plaintext on the output difference and use the high probability of inheritance or elimination of the difference to produce the final same output [35]. The safety of a hash function ultimately depends on whether the overall collision of the function can be found. Because the SHA-256 algorithm has an iterative structure, according to the avalanche effect of the iterative algorithm, as the number of rounds increases, the corresponding overall collision complexity will be sharply ascending; this makes it very difficult to find the overall collision. Until now, the existing attacks have not been able to find a SHA-256 overall collision. Through the analysis of the Chabaud-Joux attack on SHA-256, a partial collision of SHA-256 was found, with a complexity of 266 [7], but an overall collision of SHA-256 could

not be found, so the SHA-256 algorithm can also resist existing differential attacks.

For a hash function with a sponge structure, its ability to resist differential attacks is directly determined by its internal transformation. In order to find the upper bound of the internal transformation's ability to resist differential attacks, the general method used is to find the number of minimum differential active S-boxes (ASN) in the encryption process. For the SPONGENT-128 used in this paper, the experimental results of the proponent of the scheme cited in this paper [30] are shown in Table 1.

5. Simulation Experiment and Result Analysis

In view of the fact that there is no platform that supports UAV swarm simulation, we tested and analyzed a variety of popular simulation platforms and chose OMNeT++ as the UAV flight simulation test environment. OMNeT++ is a modular, component-based C++ simulation library and framework, mainly used to build network simulators. OMNeT++ can be used for free in noncommercial simulations such as academic institutions and teaching. OMNeT++ itself is a simulation framework, and there is no model for network protocols such as IP or HTTP. There are several external frameworks for the main computer network simulation models. The most commonly used one is INET, which provides various network protocols and technologies of various models, such as IPv6 and BGP.

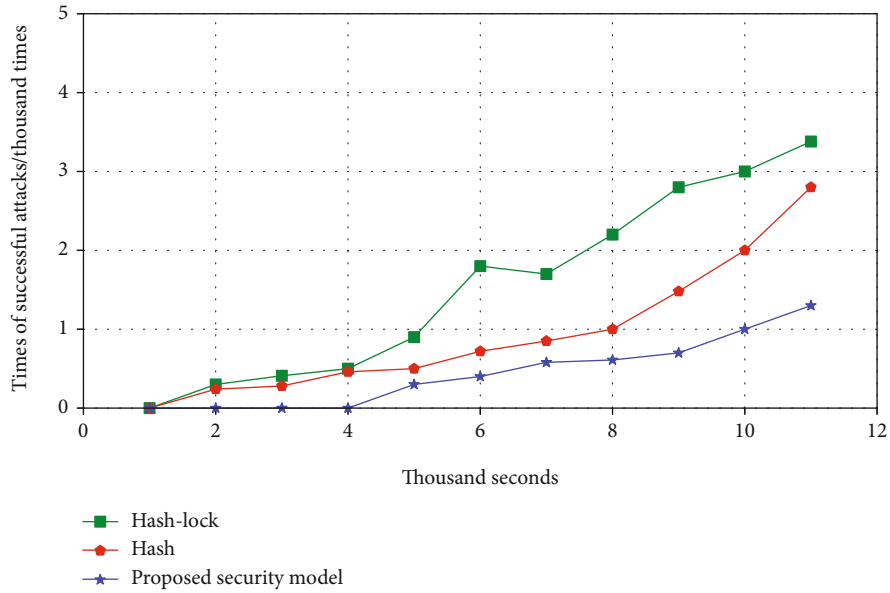


FIGURE 10: The times of successful attacks.

The UAV simulation test platform is composed of three main modules: communication module, computing module, and mobile module. The communication module is based on the INET framework, in which the physical layer uses the radio model and the medium model, the data link layer uses the ad hoc-based 802.11 protocol, the routing protocol uses the OLSR (Optimized Link State Routing) protocol, and the transmission protocol uses the UDP protocol. As shown in Figure 9, the calculation module performs related calculations and the operation of the two hash algorithms based on the information obtained by various sensors and provides different functions for the cluster nodes. For example, each node simulates the equipped geographic information sensor by reading its coordinate location information. The mobile module is based on the mobile model of INET and has been redesigned and implemented according to the UAV formation requirements. This module can simulate the waypoint flight mode of the UAV.

We compare the security model based on the hash chain [36] and the security model based on the hash-lock protocol [37] with the scheme proposed in this paper in terms of throughput, delay, and security to verify the scheme proposed in this paper on security and performance. In order to better evaluate and compare the security of these protocols, an attack program is added here to simulate an attacker's attack on the protocol. This paper simulates several common network attacks. We call the generation and authentication functions in the file/hash.cc. OMNeT++ processes the arriving data packets by calling the generation function in ProcessPacket(.) and forwards the processed data packets by calling the verification function in PacketCallback(.). The following is an analysis of the simulation results.

5.1. Statistics of the Number of Successful Attacks. With 1000 seconds as a time node, the number of attacks of the three models at each time node is counted, as shown in Figure 10. It can be seen that the security model based on the hash-lock

and hash chain is almost linearly distributed, basically every unit of time can be successfully attacked, and the security scheme based on the hash-lock protocol is based on the security of the hash chain after 4 s. After 8 s, the number of successful attacks on the scheme has increased at a faster rate, indicating that as long as the attack continues for a long time, these two schemes have great security risks.

The encryption scheme proposed in this paper, with the increase in time, has basically remained stable in the probability of successfully resisting common attacks. Compared with the other two schemes, the curve trend is relatively flat and there is no obvious growth trend. Therefore, the ability to resist common attacks always maintains advantages and has good security. Moreover, in the first four seconds of the simulation, the scheme proposed in this paper can completely resist the attack behavior, and it is quite effective for the UAV cluster with high real-time and mobility to resist the invasion of attackers.

5.2. Throughput and Delay. We compare the delay and throughput of the solution that does not use any security solution, the security model based on the hash chain, the security model based on hash-lock, and the solution that uses the security model of this paper.

As shown in Figure 11, for the throughput, it can be seen that when the number of packet lengths is small (64, 128 bytes), the security scheme has a greater impact on throughput, because the label length accounts for a larger proportion of the packet length with a security label. So it will reduce the throughput of the system, but when the packet length is longer (256, 512, and 1024), the throughput of the four schemes is not much different, so the security scheme is more suitable for data packets containing more data. The application scenario is the scenario with a large packet length, and the UAV will carry a large amount of data. Therefore, the security scheme proposed in this paper does not have much impact on the throughput while ensuring the security of the

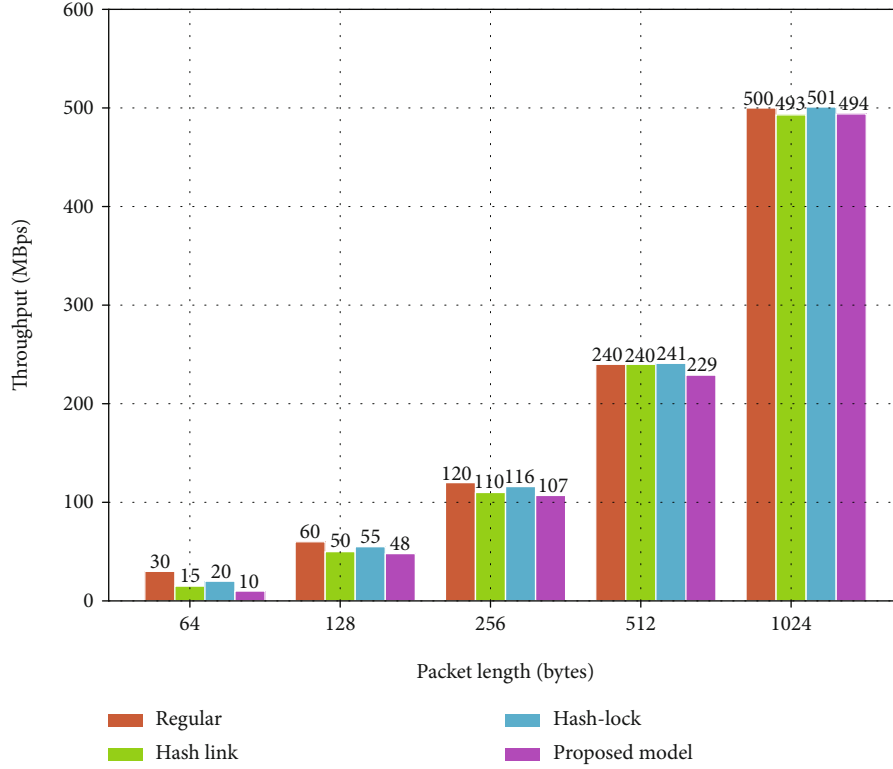


FIGURE 11: Throughput.

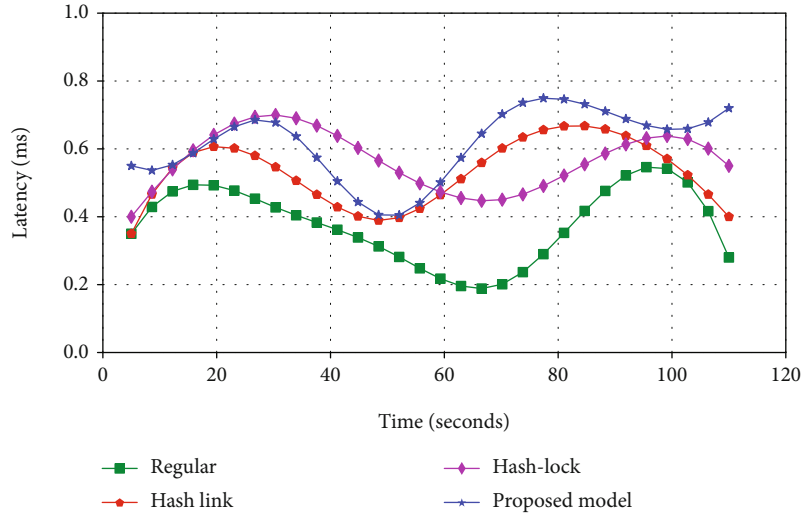


FIGURE 12: Delay.

system. As shown in Figure 12, for the delay, in order to facilitate observation and analysis, we ignored some inevitable jitter and focused on mainstream delay. As shown in the figure, the average delay of the security model based on the hash chain and hash-lock is higher than that of the solution proposed in this paper, and the delay of the security model proposed in this paper is not much different from that of the normal mode, only an increase of about 0.15 ms. From this aspect, it also illustrates the feasibility and low latency of the program.

5.3. *Feasibility.* While improving the security and robustness of the system, we also examine the distribution of uncertain labels, which significantly reduces the possibility of an attacker accurately predicting uncertain labels through random guessing. Because if the distribution is biased towards certain labels, the attacker is likely to use these labels to improve the prediction success rate. From Figure 13, we can see that the uncertain labels generated by the UAV have approximately normal distribution, which well limits the range of the attacker’s random inference.

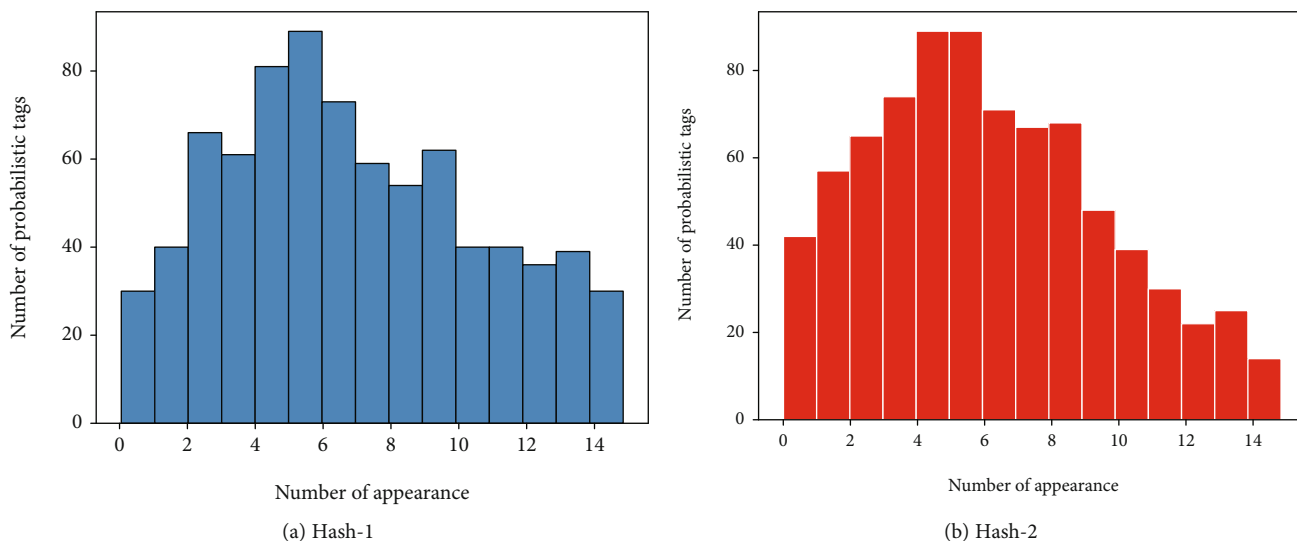


FIGURE 13: The distribution of ulabel generated by hash-1 and hash-2.

We used a 10-digit uncertain label and considered 5000 other packets in the same stream. We expect the average number of labels to appear as $5000/210 = 4.9$. When this expected value is reached and the number of occurrences is 4, the above two distributions reach their peaks. The above results show that the model proposed in this paper will not cause the attacker to continue to speculate on our label, nor will it issue wrong labels. After all, whenever it detects a packet with an incorrect label, it will alert the system. In addition, constantly refreshing the custom key also helps invalidate the attacker's inference.

6. Conclusions

Based on the previous work, our mechanism greatly reduces statistical inference. The SHA-256 algorithm and SPONGENT-128 are used to generate the ulabel of the UAV message header, which ensures that the system does not increase a large amount of energy consumption, while the safety is greatly improved. Since the current UAV supports thousands of [38] rule updates per second, we regularly update the UAV's custom key and the label authentication rules with the leader.

Finally, we proposed the situation where an attacker invaded the UAV swarm performing military missions and proposed a random labeling mechanism solution. This mechanism enables the UAV swarm system to mark packet headers that are difficult for attackers to infer and guess. The labeling strategy is to conduct swarm authentication through collaboration across the leader UAV, relay nodes, and member UAVs. If an attacker manipulates a damaged UAV to try to enter the system, it must forge the correct label to avoid verification, and our mechanism restricts the attacker to random guesses. In addition, this paper uses OMNeT++ to implement a security authentication mechanism based on random labels for large-scale UAV swarms. The experimental results show that, in terms of throughput, delay, feasibility, etc., the scheme proposed in this paper can achieve accurate inspection with reasonable cost and ensure the safety of the system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

This manuscript is an extension of [8], published in the 17th IEEE International Symposium on Parallel and Distributed Processing with Applications (IEEE ISPA 2019).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China, under Grant 2019YFB2102002; in part by the Key Research and Development Program of Jiangsu Province, under Grant BE2019012; and in part by the National Natural Science Foundation of China, under Grant 62001217.

References

- [1] C. Zhong, J. Yao, and J. Xu, "Secure UAV communication with cooperative jamming and trajectory control," *IEEE Communications Letters*, vol. 23, no. 2, pp. 286–289, 2019.
- [2] P. Perazzo, F. B. Sorbelli, M. Conti, G. Dini, and C. M. Pinotti, "Drone path planning for secure positioning and secure position verification," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2478–2493, 2017.
- [3] J. Zhang, T. Chen, S. Zhong et al., "Aeronautical ad-hoc networking for the Internet-above-the-clouds," *Proceedings of the IEEE*, vol. 107, no. 5, pp. 868–911, 2019.
- [4] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

- [5] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.
- [6] B. Song, J. Y. Hwang, and K. Shim, "Security improvement of an RFID security protocol of ISO/IEC WD 29167-6," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1375–1377, 2011.
- [7] D. Sun and Y. Mu, "Security of grouping-proof authentication protocol for distributed RFID systems," *IEEE Wireless Communications Letters*, vol. 7, no. 2, pp. 254–257, 2018.
- [8] L. Liu, H. Qian, and F. Hu, "Random label based security authentication mechanism for large-scale UAV swarm," in *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 229–235, 2019.
- [9] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1594–1606, 2018.
- [10] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, and D. Muller, "Unmanned aerial vehicle security using recursive parameter estimation," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 692–702, 2014.
- [11] R. Cabaniss, V. Kumar, and S. Madria, "Multi-party encryption (MPE): secure communications in delay tolerant networks," *Wireless Networks*, vol. 21, no. 4, pp. 1243–1258, 2015.
- [12] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management toward 5G radio access networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1656–1686, 2016.
- [13] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in VANET," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2034–2048, 2020.
- [14] Y. Yapici, İ. Güvenç, H. Dai, and A. Bhuyan, "Physical layer security for UAV swarm communications via protected zone," in *Resilience Week (RWS)*, vol. 1, pp. 174–177, IEEE, 2019.
- [15] Y. Zhou and L. Li, "A trust-aware and location-based secure routing protocol for WSN," *Applied Mechanics and Materials*, vol. 373–375, pp. 1931–1934, 2013.
- [16] E. Kline, A. Afanasyev, and P. Reiher, "Shield: DoS filtering using traffic deflecting," in *19th IEEE International Conference on Network Protocols*, pp. 37–42, 2011.
- [17] T. Hayajneh, R. Doomun, G. Al-Mashaqbeh, and B. J. Mohd, "An energy-efficient and security aware route selection protocol for wireless sensor networks," *Security and Communication Networks*, vol. 7, no. 11, p. 2038, 2020.
- [18] R. Sangeetha and M. Yuvaraju, "Secure energy-aware multipath routing protocol with transmission range adjustment for wireless sensor networks," in *IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–4, 2012.
- [19] J. J. Lotf, M. Hosseinzadeh, and R. M. Alguliev, "Hierarchical routing in wireless sensor networks: a survey," in *2010 2nd international conference on computer engineering and technology*, vol. 3, pp. V3650–V3654, IEEE, 2010.
- [20] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.
- [21] T. Hu and Y. Fei, "Qelar: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796–809, 2010.
- [22] N. Kabir and S. Kamal, "Secure mobile sensor data transfer using asymmetric cryptography algorithms," in *International Conference on Cyber Warfare and Security (ICWS)*, pp. 1–6, 2020.
- [23] A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, "GNSS spoofing detection in handheld receivers based on signal spatial correlation," in *IEEE/ION Position, Location and Navigation Symposium*, pp. 479–487, 2012.
- [24] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "Receiver autonomous spoofing detection: experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, vol. 1, pp. 124–130, 2009.
- [25] P. Gope and T. Hwang, "A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system," *Computers & Security*, vol. 55, pp. 271–280, 2015.
- [26] K. Bu, Y. Yang, Z. Guo, Y. Yang, X. Li, and S. Zhang, "Flow-Cloak: defeating middlebox-bypass attacks in software-defined networking," in *IEEE INFOCOM 2018- IEEE Conference on Computer Communications*, pp. 396–404, 2018.
- [27] Z. A. Qazi, R. Miao, C. C. Tu, V. Sekar, L. Chiang, and M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," in *Acm Sigcomm Conference on Sigcomm*, pp. 27–38, 2013.
- [28] E. Khan, M. W. El-Kharashi, F. Gebali, and M. Abd-El-Barr, "Design and performance analysis of a unified, reconfigurable HMAC-Hash unit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 12, pp. 2683–2695, 2007.
- [29] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen message attack *," *Discrete Algorithms and Complexity*, vol. 17, no. 2, pp. 287–310, 1987.
- [30] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede, "SPONGENT: A lightweight hash function," *Lecture Notes in Computer Science*, vol. 6917, pp. 312–325, 2011.
- [31] P. Megha Mukundan, S. Manayankath, C. Srinivasan, and M. Sethumadhavan, "Hash-One: a lightweight cryptographic hash function," *IET Information Security*, vol. 10, no. 5, pp. 225–231, 2016.
- [32] P. Li, L. Han, X. Tao et al., "Hashing nets for hashing: a quantized deep learning to hash framework for remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7331–7345, 2020.
- [33] N. Meghanathan, B. K. Kaushik, and D. Nagamalai, "Advances in networks and communications," *Communications in Computer & Information science*, vol. 132, 2012.
- [34] M. Wang and Y. Li, "Hash function with variable output length," in *2015 International Conference on Network and Information Systems for Computers*, pp. 190–193, 2015.
- [35] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, 1991.
- [36] S. Chang, Y. Park, and B. B. Ashok Babu, "Fast IP hopping randomization to secure hop-by-hop access in SDN," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 308–320, 2019.

- [37] O. Changqing, J. Wu, L. Zhengyan, and H. Shengye, "An enhanced security authentication protocol based on hash-lock for low-cost RFID," in *2008 2nd International Conference on Anti-counterfeiting, Security and Identification*, pp. 416–419, 2008.
- [38] G. Lombardi, E. Medvet, and A. Bartoli, "A language for UAV traffic rules in an urban environment and decentralized scenario," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1139–1143, 2017.

Research Article

Blockchain-Based Trust Auction for Dynamic Virtual Machine Provisioning and Allocation in Clouds

Hao Xu ¹, Weifeng Liu,¹ and Xu Liu ^{1,2}

¹School of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225127, China

²School of Business, Victoria University, Melbourne VIC 3011, Australia

Correspondence should be addressed to Xu Liu; sherryliu08@foxmail.com

Received 16 November 2020; Revised 21 April 2021; Accepted 18 May 2021; Published 16 June 2021

Academic Editor: Chunpeng Ge

Copyright © 2021 Hao Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud computing uses virtualization technology to provide users with different types of resources in the form of services. The third party plays a crucial role in coordinating cloud market between cloud providers and users. As for providing services or trading, the extra broker fees are required for the middleman because the third party facilitates transactions. Moreover, there is no guarantee that the third party is trusted, which can lead to information leakage, data tampering, and unfair trading. Blockchain technology is an emerging technology that can store and communicate data between entities that unnecessarily trust each other. To resolve the problems, this paper presents the blockchain-based trust and fair system and develops the smart contract of auction and transaction. The prototype system is implemented based on the Hyperledger Fabric. The experimental results prove the feasibility of the scheme.

1. Introduction

Cloud computing is a popular paradigm of offering services over the Internet [1]. With the development of Internet technology and virtualization technology, more and more enterprises and individuals outsource their workloads to cloud providers. Cloud computing services are generally provided in three types: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [2]. In cloud computing environment, cloud providers cooperate to form a huge abstract, virtualized, and dynamically expandable resource pool to provide cloud service and resources to users. A user acquires and releases resources by requesting and returning virtual machines in the cloud. To sell the VM instances to users, cloud providers can employ auction-based models. An example of implemented auction method in Cloud computing is the spot market introduced by Amazon [3].

Current cloud resource auction integrates the network technology into the bidding system. The desirable preconditions for managing cloud resource auction are “trust” and

“fair.” The main entities included in cloud resource auction are cloud providers, users, and the third party (i.e., auctioneer) as shown in Figure 1. The cloud provider is an organization that offers computing resources for use on payment. A user is a person or an organization that purchases cloud service. The third party is a middleman that provides a platform to make the product available in the market. Most of the third-party platforms are centralized middlemen which can lead to a host of trust and fair issues. First, the centralized cloud computing auction system may suffer the single node attack and has a higher risk of data tampering and privacy leakage [4]. Secondly, the third-party platform completely controls over the bidding process. Users have no way to ensure that the middleman never leaks their bidding information. Even if the auction platform has security and trust issues, it still charges a large fee for service as a middleman between cloud providers and users. In addition, many existing researches and models on cloud resource auction cannot consider the transaction process after the auction. There are multiple unfair problems associated with the transaction process. For example, malicious cloud providers may not

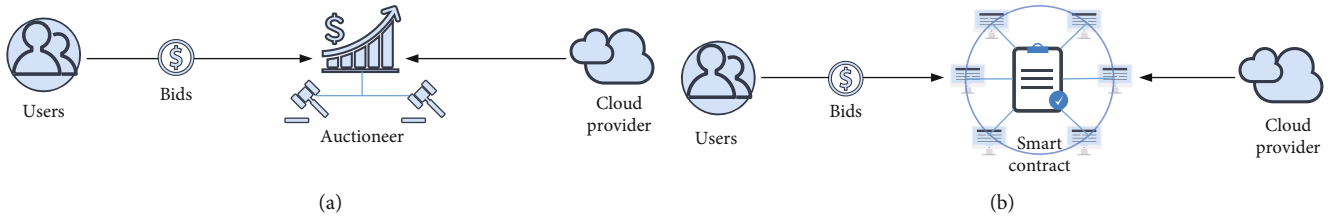


FIGURE 1: (a) Current centralized model of cloud resource auction. (b) Proposed architecture using smart contract which acts as a third-party platform.

provide services or unmatched resources after charging service fees, and malicious users may not pay service fees when obtaining resources.

Blockchain, as an emerging distributed ledger technology [5], has attracted more attention from various fields, including supply chain [6], IoT [7], medical [8], and other areas. Since data storage is based on a distribution architecture and data management is based on a peer-to-peer network, data records are immutable, verifiable, and traceable [9]. Not only blockchain provides a reliable system to store data but also it improves bidding information security and immutability. So blockchain can facilitate transactions between entities that unnecessarily trust each other.

Blockchain technology has the great potential to address the challenge of conventional cloud platform. References [10–12] mainly aim at trust problem in the cloud platform and design a credible service transaction method combined with blockchain. In [13], a decentralized framework is proposed to address issues such as trust and data security. A blockchain-based cloud resource allocation framework is proposed in [14]. Conspicuous feature, of this model, is the timed commitment scheme, state mechanism, and ladder payment to guarantee auction fairness and trade fairness.

In this paper, we combine blockchain and smart contract technology to propose a blockchain-based cloud resource auction scheme to ensure the security of bidding data and the fairness of cloud services. The scheme designs a data encryption transmission module, which uses symmetric encryption technology to encrypt bid data to protect user privacy and prevent malicious users from leaking data. Since the blockchain is a peer-to-peer access structure, the points in the structure can be trusted with each other. Consequently, the centralized third-party platform can be removed to reduce the transaction cost. Use smart contract technology to design the auction contract and transaction contract. Some rules are written in smart contracts to achieve fair trading of cloud resources. Users must pay transaction fees in advance to obtain VM instances, and cloud providers must provide cloud resources in order to receive transaction fees. Compared with the traditional cloud resource auction system, this scheme can realize the safe auction and fair transaction of cloud resources, protect the bidding privacy of users, and resist internal malicious users.

The structure of this paper is as follows. Section 2 introduces the background, system architecture and detailed implementations. Section 3 gives the results of the proposed architecture. The last section concludes the paper.

2. Background

2.1. Related Work. One helpful approach for the cloud computing resource supply is the mechanisms for auctions. Several authors have studied such mechanisms in different fields such as economics and computer science. In the context of cloud computing resource allocation, Jain et al. [15] proposed an efficient truthful-in-expectation mechanism for resource allocation in clouds. Zama and Grosu [16] designed truthful approximation mechanisms for the auction-based allocation of VM instances in clouds. Due to the characteristics of resources in cloud computing, more and more researchers introduce the quality of service parameter and nonprice attributes into cloud resource auction system to solve allocation problems and transaction fraud problems. In [17], the authors introduced a multiattribute auction framework and used evaluation functions to publish cloud providers for fraudulent behavior. The current works explore auctions of introducing features for the cloud resource auction system. However, none of the noticed solutions focus on the honest behavior of the auctioneer, nor in providing reliable records about transactions between the cloud provider and users.

In such a direction, to combine blockchain technology with the cloud resource auction system has become a new trend, because its decentralization and data tampering prevention might provide solutions to the shortcomings of current systems. Few works have been using blockchain to solve the cloud resource auction system issues. In one of them, Chen et al. [18] designed a blockchain-based architecture for a bidding system. Franco et al. [19] proposed a blockchain-based reserve auction for infrastructure supply in a virtual network. Also, An et al. [20] proposed a blockchain-based reserve auction for data transactions to solve the problem of third-party brokers. Ch et al. [21] presented a blockchain technology solution, using pentatope-based elliptic curve cryptography and SHA to improve the security and privacy device data. Thirumalai et al. [22] applied the knapsack method to encrypt ENPKES keys to enrich high security in cloud systems.

2.2. Blockchain. Blockchain originated from Bitcoin. As the underlying application technology of Bitcoin, it has received widespread attention as Bitcoin became famous. As a decentralized distributed ledger technology, blockchain integrates technologies such as distributed data storage, encryption algorithms, and consensus mechanisms. Blockchain eliminates third-party authoritative centers [23]. It can enhance

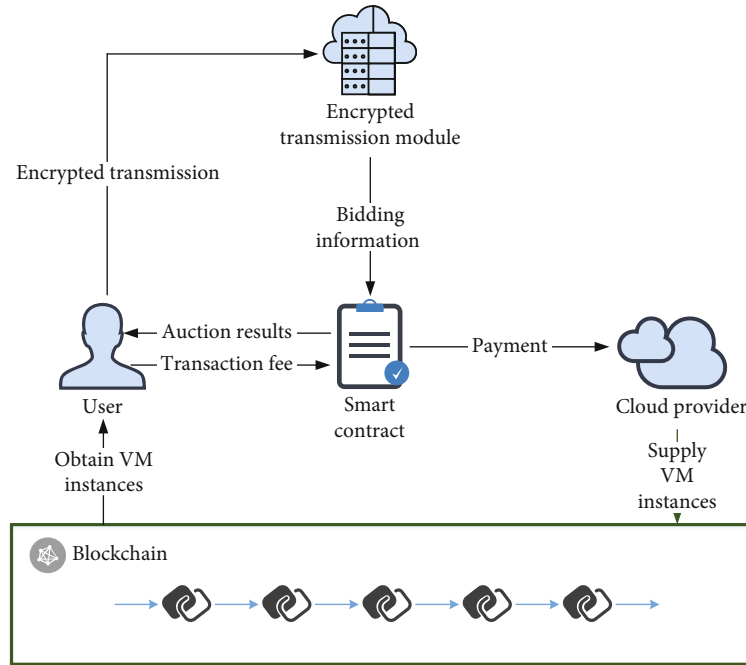


FIGURE 2: Architecture model.

the trust between participants without relying on the third-party trusted authority, because all consensus nodes participate in the process of maintaining blocks on the chain and the transaction record are stored on a peer-to-peer network to ensure that the data cannot be tampered with and any node can audit the data in the block. In addition, due to the irreversibility of the hash function and the security advantages of the consensus mechanism, the transparency, security, and immutability of transaction data are guaranteed.

Since the blockchain is public and transparent, when it comes to private data, it needs to be encrypted. This paper mainly uses *Diffie-Hellman key exchange algorithm* [24] and *symmetric encryption algorithm* [25]. The characteristic of the *DH algorithm* is that both parties can exchange keys securely in an insecure channel. In the *DH algorithm*, the sender and receiver use their own private key and the other party's public key to generate a shared key. In the *symmetric encryption algorithm*, the sender uses the symmetric encryption key to encrypt the data, and the receiver uses the key to decrypt the data to obtain the data, which is characterized by the same encryption and decryption keys.

In the blockchain 2.0 stage, the concept of smart contract [26] is introduced. Developers can create various applications through smart contracts. In this paper, the smart contracts replace the third-party platform in the existing system and perform resource allocation and transaction management according to predefined rules.

2.3. Proposed Model. This paper focuses on IaaS and is built under the situation that the cloud provider offers different types of resources in the form of VM instances. The blockchain-based cloud computing resource auction model is shown in Figure 2, including user, cloud provider, certifi-

cate authority, data encryption transmission module, and smart contract module.

The main function of the user is to obtain the VM instances. Each user has his own public and private key PK_{user} , SK_{user} , and address $Addr_{user}$. In the process of applying for obtaining the VM instances, the user can obtain the resources only after being authorized by smart contract. The main function of the cloud provider is to supply the VM instances. The cloud provider has his own public and private key PK_{cp} , SK_{cp} , and address $Addr_{cp}$.

The certification authority is a role in the system, which is mainly responsible for verifying the identity of users who want to join the system and then sending digital certificates to users. Only users who pass identity verification can join the system, which improves the security of the system to a certain extent.

The data encryption transmission module encrypts each user's bid information and uploads it to the smart contract module to ensure that the bid information will not be leaked. The smart contract module realizes the security and fairness of the whole auction process and solves the problems such as auction centralization, bidding information disclosure, and dishonest transaction. The module is mainly composed of two smart contracts: auction contract and transaction contract. The auction contract (AC) executes allocation algorithm and payment function, which is aimed at social welfare maximization to protect the interests of users. Use the transaction contract (TC) to determine the applicant's transaction information before the transaction occurs, freeze the prestored transaction fee when the transaction occurs, and send the transaction fee to the cloud provider after the transaction is completed and the applicant receives the VM instances. Use smart contract to replace trusted third parties

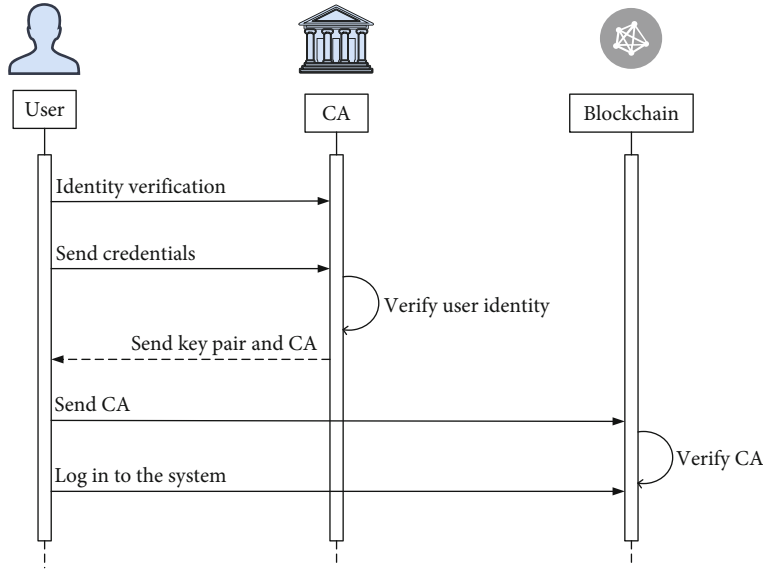


FIGURE 3: User registration process.

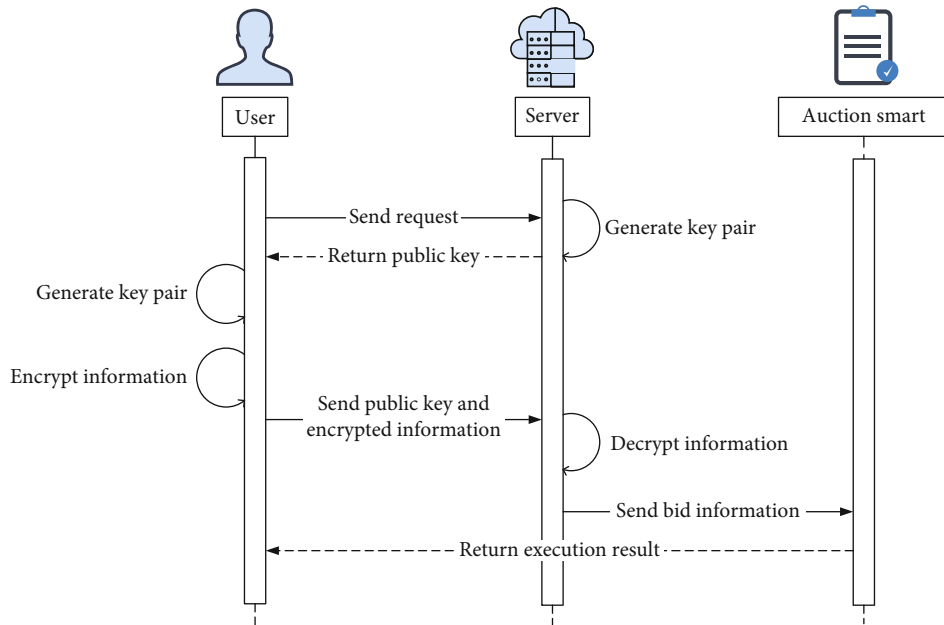


FIGURE 4: Timing diagram of encrypted transmission module.

to realize transaction payment management, effectively preventing users from not paying and sellers not providing resources, and ensuring fair transactions.

2.4. Blockchain-Based Cloud Computing Resource Auction Steps

2.4.1. User Registration. The process of identity authentication and registration is shown in Figure 3.

Step 1. The user needs to send an identity verification application to the certification authority (CA) before logging in to the alliance chain network. After the verification is passed, the applicant will obtain the key pair and a digital certificate.

Step 2. The user sends his registration and digital certificate to the system after obtaining the digital certificate and key pair.

Step 3. The node verifies the digital certificate of the new user and confirms the identity with the CA. After identity is approved, the user will be added to the system.

2.4.2. User Submission. The process of submitting bid data is shown in Figure 4.

Step 1. The user sends a request to the server to generate the public and private key on the server side. The server side returns the production key to the user.

Input: server public key PK_{ser} , client public key KEY_{cli} , encrypted bidding information $CInfo_{bid}$, actual submission time t_A , deadline T_s , user A address $Addr_A$

Output: complete submission

- 1 $KEY_{cli} = client.getSecret(PK_{ser})$
- 2 $CInfo_{bid} = Enc(Info_{bid}, KEY_{cli})$
- 3 $Client\ node(PK_{cli}, CInfo_{bid}, t_A, Addr_A) \rightarrow Server\ node$
- 4 $KEY_{ser} = server.getSecert(PK_{ser})$
- 5 $PInfo_{bid} = Dec(CInfo_{bid}, KEY_{ser})$
- 6 **if** $t_A < T_s$ && $PInfo_{bid} = true$ **then**
- 7 $Server\ node(PInfo_{bid}, Addr_A) \rightarrow AC$
- 8 **return** "Bid information submission successful"
- 9 **else**
- 11 **return** "Bid information submission failure"
- 12 **end if**

ALGORITHM 1: Encrypted submission.

Input: Users bidding information : $Info_{bid} = \{\theta_1, \theta_2, \dots, \theta_n\}$, $\theta_n = (\beta_n, b_n)$
 Users address information: $Addr_U = \{Addr_1, Addr_2, \dots, Addr_n\}$.
 Resources capacities: $C = \{C_1, C_2, \dots, C_M\}$

Output: $Addr^*, x^*, P$

- 1 {Collect}
- 2 **for all** $i \in U$ **do**
- 3 collect user bidding information and address from user i
- 4 {Allocation}
- 5 $(Addr_i, x_i) \leftarrow G\text{-VMPAC-II-ALLOC}(\theta, C)$
- 6 $Addr^* \leftarrow Addr^* \cup Addr_i$
- 7 $x^* \leftarrow x^* \cup x_i$
- 8 {Payment}
- 9 $P_i \leftarrow G\text{-VMPAC-II-PAY}(\theta, C)$
- 10 $P \leftarrow P \cup P_i$
- 11 **return** $Addr^*, x^*, P$

ALGORITHM 2: The function of AC.

Step 2. The user executes *DH algorithm* to generate public and private key pair and generate symmetric key according to the public key of the server, which is used to encrypt the transaction information.

Step 3. The user sends his public key and encrypted information to the server.

Step 4. The server generates a symmetric key according to the user's public key and decrypts the encrypted information by using the symmetric key.

Step 5. Send the bid information to the auction contract, and return the execution result to the user.

Algorithm 1 describes the process of the data encrypted submission. First of all, user A requests and obtains the public key of the server. Lines 1 to 2 indicate that user A generates the symmetric key KEY_{cli} according to the public key of the server and encrypts the bid information $Info_{bid}$ with KEY_{cli} to generate cipher text $CInfo_{bid}$. Line 3 shows that user A sends the public key of the client PK_{cli} and encrypted

bid information $Info_{bid}$ to the blockchain server. Lines 4 to 5 show that the blockchain server generates the same symmetric key KEY_{ser} ($KEY_{cli} = KEY_{ser}$) and decrypts the $CInfo_{bid}$ using KEY_{ser} to obtain the details of the bid. Lines 6 to 11 show that the submission time limits and verifies whether the bid information has been tampered with. Finally, the server sends bid information to the AC.

2.4.3. Auction. This section presents the function of the auction contract. Mahyar et al. [27] proposed truthful greedy mechanism for VM provisioning and allocation in clouds. The mechanism is mainly divided into two parts: allocation and payment. The function of auction smart is described in Algorithm 2. Lines 1 to 3 show that AC collects the bidding information and address from the users. Lines 4 to 7 show the optimal allocation of solving cloud resource allocation by calling the allocation algorithm G-VMPAC-II-ALLOC [27]. Lines 8 to 10 show that the payment of user i is calculated by calling G-VMPAC-II-PAY [27]. Finally, AC returns three output parameters: $Addr^*$, the address set of winner; x^* , the optimal allocation of cloud resources to the users; and P the payment.

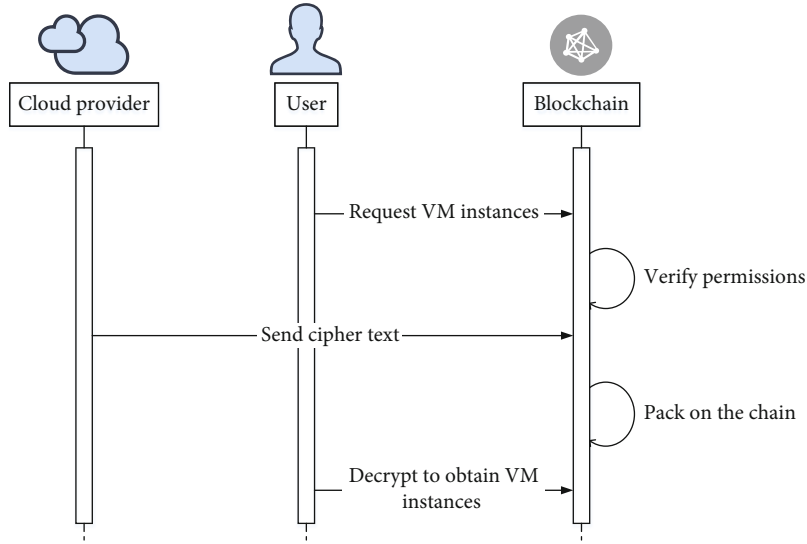


FIGURE 5: Resource supply process.

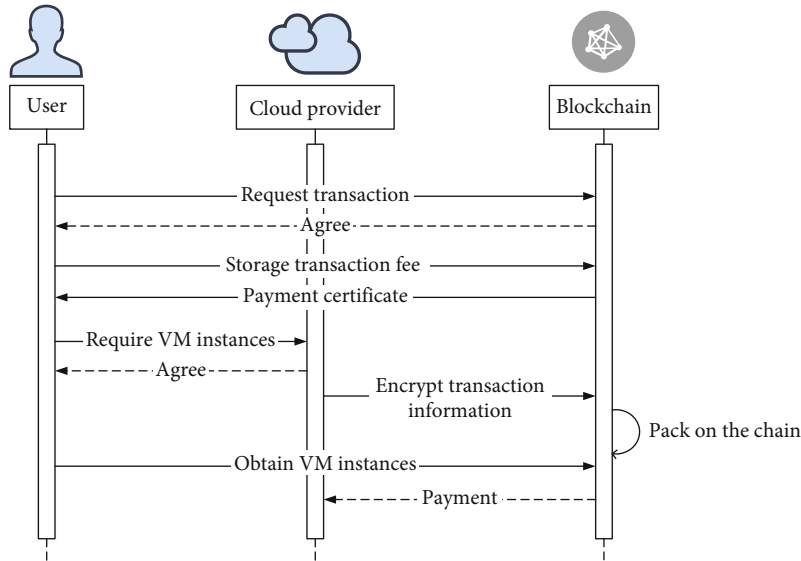


FIGURE 6: Pay service fee process.

2.4.4. *Transaction Management.* After the user receives the auction result, the winner can apply for resources. The TC needs to call the AC to obtain the address and the payment of the winner user to verify the applicant’s transaction information. Transaction management is divided into two parts, including resource supply and payment management. The specific steps are shown in Figures 5 and 6.

(1) *Resource Supply.*

Step 1. Applicant A sends the request information; TC first confirms his payment certificate.

Step 2. If applicant A has payment certificate, the VM instance information will be encrypted with the applicant’s public key and signs it with the cloud provider’s private

key. The cloud provider sends encrypted information and signatures to the network as transaction data.

Step 3. The node verifies the signature and packages the verified transaction data into blocks and waits for the consensus to be on the chain.

Step 4. After the block is on the chain, applicant A can obtain the VM instance information from the block. Applicant A verifies the signature and uses his private key to decrypt the transaction data to obtain the VM instance information.(2) *Payment Management.*

Step 1. Applicant A sends a request to TC for prestored payment. After TC receives the transaction fee, it will generate a

Input: Applicant A address $Addr_A$, applicant public and private keys (PK_A, SK_A) , the CFP public and private keys (PK_{cfp}, SK_{cfp}) , VM instances permission information $Info_{VM}$, block ID_{block} .

Output: Applicant A obtains virtual machine permission information

- 1 **function** $ResourceSupply(Addr_A, PK_A, Info_{VM}, SK_{cfp})$
- 2 $x_i \leftarrow matchAC(Addr_A, Addr^*)$
- 3 **if** $(x_i = true)$ **then**
- 4 $CInfo_{VM} = enc(Info_{VM}, PK_A)$
- 5 **return** $CInfo_{VM}$
- 6 $sign_{cfp} \leftarrow Sign(hash(CInfo_{VM}), SK_{cfp})$
- 7 $block \leftarrow Add(CInfo_{VM}, sign_{cfp})$
- 8 **end function**
- 9 **function** $GetInfo(ID_{block}, SK_A, PK_{cfp})$
- 10 $sign_{cfp}, CInfo_{VM} \leftarrow GetBlockInfo(ID_{block})$
- 11 **validate** $(sign_{cfp}, PK_{cfp})$
- 12 $Info_{VM} \leftarrow dec(CInfo_{VM}, SK_A)$

ALGORITHM 3: Resource supply.

Input: Applicant A address $Addr_A$, actual stored payments s_A , actual stored time t_A , deadline T_p , auction contract AC, CFP address $Addr_{cfp}$

Output: Complete transaction fee storage and transfer

- 1 User node $(s_A, t_A, Addr_A) \rightarrow TC$
- 2 **function** $TC(P, Addr, T_p)$
- 3 $P_A \leftarrow matchAC(Addr_A, P)$
- 4 **if** $t_A < T_p \& \& s_A = P_A$ **then**
- 5 **return** o_{pay}
- 6 **else if** $t_A < T_p \& \& s_A < P_A$
- 7 **return** "Insufficient transaction fee"
- 8 **else if** $t_A > T_p$
- 9 **return** "Storage failure"
- 10 **end function**
- 11 **function** $Pay(s_A, Addr_{cfp})$
- 12 $s_A \rightarrow Addr_{cfp}$
- 13 **end function**

ALGORITHM 4: Payment management.

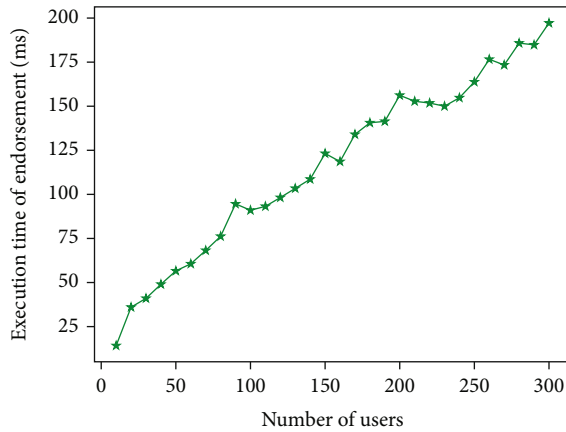


FIGURE 7: Execution time of auction in endorsing peer.

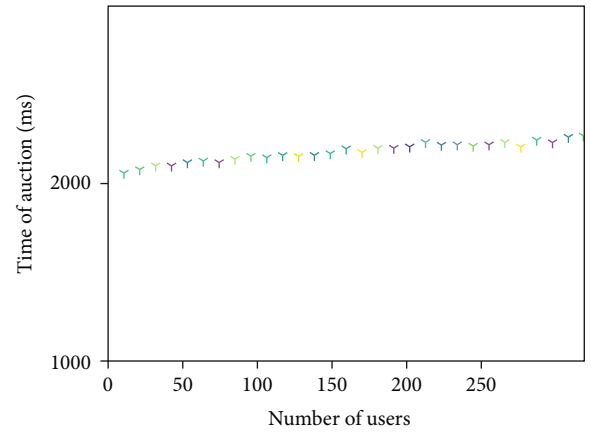


FIGURE 8: Time of auction phase in fabric.

TABLE 1: This work was compared with the models in the literature review.

Items	Models	Privacy protection	Decentralization for auction	Transaction management
[13]	Centralized	×	×	×
[14]	Centralized	×	×	√
[4]	Blockchain-based	×	√	×
This work	Blockchain-based	√	√	√

payment certificate o_{pay} and send it to applicant A, indicating that the prestored transaction fee has been completed.

Step 2. Applicant A sends the payment certificate, public key, and related information to the cloud provider.

Step 3. The cloud provider confirms the transaction and provide VM instances to applicant A.

Step 4. TC pays the service fee to the cloud provider after it receives the confirmation information that applicant A has obtained the virtual machine instance.

Algorithm 3 describes the process of the cloud provider providing resources to the applicant. Lines 2 to 5 show whether applicant A is the winner. If he is the winner, the cloud provider encrypts the resource information with his public key. Lines 6 to 9 show that the cloud provider publishes resource information on the blockchain. Lines 10 to 12 show that the applicant achieves VM instance permission information.

Algorithm 4 describes the payment management. Lines 2 to 10 show whether applicant A has stored enough money within a limited time. Lines 11 to 13 show that TC sends the transaction fee to the cloud provider.

3. Experiment

The simulation development of the system runs on a PC and uses the Ubuntu operating system. This solution is based on the open-source framework of Hyperledger Fabric v1.1 version. The system is simulated and developed through a cloud server. The development software uses Visual Studio Code and Remote Development plug-ins. The database Couchdb is used, and the system chain code is developed using node.js.

3.1. Performance Testing. The auction phase includes reading users' data sets, executing the auction allocation algorithm, and uploading the auction results to the chain. Endorsement node (endorser) simulates the execution of the allocation algorithm by calling the chain code; ordering service (orderer) sorts transaction requests and creates the block and peer updates ledgers after verification. In order to reflect the variation of auction time with the number of users, this performance test mainly studies the execution time of auction in endorsers and auction transaction time in fabric network. The system test results of the above two items are shown in Figures 7 and 8.

Figure 7 shows that the execution time of the allocation algorithm is affected by the increase in the number of users.

As the number of users varies from 10 to 300, the execution time sees an upsurge from about 14 ms to 197 ms. By testing the auction execution time in the endorser, it is found that the increase in the number of users has a positive impact on the execution time. Figure 8 mainly tests the time of auction transactions in the fabric network. After many tests, each auction transaction time of the system is close to 2000 ms, which can meet the basic business needs.

Through the analysis of test results, the system has passed the functional test and performance test, completed the expected design goal, and verified the feasibility and effectiveness of the system.

3.2. Comparison. We compared the performance of our system with mainstream system. The result is presented in Table 1. In the table, the systems are compared on the three criteria:

- (i) Privacy protection: in cloud auction, bidding prices and cloud resource demands are crucial for privacy. This system tackled privacy leakage through the data encryption module. This approach encrypts private information to ensure that no plain text appears during transmission, which improves data security
- (ii) Decentralized for auction: it was already well known that decentralized systems, in which the smart contract acts as the third-party platform, were much more secure than centralized systems. Furthermore, to cut down the broker fees between users and cloud providers, the system should be based on the blockchain auction system
- (iii) Transaction management: in the cloud market, both auction security and fair transactions are required. This proposed system eliminates malicious behavior in cloud resource transactions in which each user prestores transaction fees. This approach improves fairness, especially when the user refuses to pay or the cloud provider provides substandard cloud resources

4. Conclusions

This paper first introduces the difficulties and challenges of the current cloud computing resource auction system. Then, through the thorough analysis of the major issues of the system, we propose a cloud computing resource auction system based on blockchain for data security and transaction fairness. To ensure the privacy and integrity of bidding data, we design an encrypted transmission module, which uses

key exchange protocol and symmetric encryption technology to encrypt and transmit bidding data. In the transaction process, asymmetric encryption technology is used to encrypt the permission information of VM instance permission information to protect the rights and interests of users. Realize transaction fairness without a trusted third party through AC and TC. Transaction fairness realizes that users cannot receive VM instances without paying the corresponding service fees, and the CFP will certainly obtain the corresponding service fee after providing VM instances.

This paper is a preliminary exploration of applying blockchain technology to the cloud computing resource auction field. There are still some shortcomings, including the following: (1) the honest bidding of users and the fair distribution of the system are mainly affected by the auction mechanism. Therefore, we should optimize the auction mechanism to improve the fairness of distribution and pricing; (2) this paper only discusses the situation where one cloud service provider provides only four resources. We can explore multiple cloud auction systems and study more versatile fairness solutions to adapt to multiple market environments.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by the Key Research Projects in Education Informatization in Jiangsu Province under Grant 20180012, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18_2366, in part by the Yangzhou Science and Technology Bureau under Grant YZ2018209 and Grant YZ2019133, in part by the Yangzhou University Jiangdu High-End Equipment Engineering Technology Research Institute Open Project under Grant YDJD201707, and in part by the Open Project in the State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, under Grant 1907.

References

- [1] G. Vinu Prasad, A. S. Prasad, and S. Rao, "A combinatorial auction mechanism for multiple resource procurement in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 904–914, 2018.
- [2] Z. Li M. Li et al., "A hierarchical cloud pricing system," in *2013 IEEE Ninth World Congress on Services*, Santa Clara, CA, USA, June–July 2013.
- [3] "Amazon EC2 Instance Types," <http://aws.amazon.com/ec2/instance-types/>.
- [4] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization," *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2018.
- [5] Y. Zhang, X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, "Blockchain-based trust mechanism for IoT-based smart manufacturing system," *IEEE Transaction on Computational Social System*, vol. 6, no. 6, 2019.
- [6] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, Kunming, China, June 2016.
- [7] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.
- [8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, pp. 25–30, Vienna, Austria, August 2016.
- [9] T. M. Fernandez-Carames P. Fraga-Lamas et al., "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.
- [10] R. Li, T. Chen, P. Lou, J. Yan, and J. Hu, "Trust mechanism of cloud manufacturing service platform based on blockchain," in *Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, August 2019.
- [11] Q. Wang, C. Liu, and B. Zhou, "Trusted transaction method of manufacturing services based on blockchain," *Computer Integrated Manufacturing Systems*, vol. 25, pp. 3247–3257, 2019.
- [12] A. V. Barenji, Z. Li, and W. M. Wang, "Blockchain cloud manufacturing: shop floor and machine level," in *Smart Sys-Tech 2018; European Conference on Smart Objects, Systems and Technologies*, Munich, Germany, June 2018.
- [13] Y. Y. Xu and Y. Wang, "Research on blockchain in cloud manufacturing resource allocation," *Journal of Frontiers of Computer Science and Technology*, vol. 8, 2021.
- [14] Z. Chen, W. Ding, Y. Xu, M. Tian, and H. Zhong, "Fair auctioning and trading framework for cloud virtual machines based on blockchain," *Computer Communications*, vol. 171, pp. 89–98, 2021.
- [15] N. Jain, I. Menache, J. Naor, and J. Yaniv, "A truthful mechanism for value-based scheduling in cloud computing," *Thryory of Computing Systems*, vol. 54, pp. 178–189, 2011.
- [16] S. Zaman and D. Grosu, "A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds," *IEEE Transaction on Cloud Computing*, vol. 1, no. 2, pp. 129–141, 2013.
- [17] G. Baranwal and D. P. Vidyarthi, "A truthful and fair multi-attribute combinatorial reverse auction for resource procurement in cloud computing," *IEEE Transaction on Service Computing*, vol. 12, no. 6, pp. 851–864, 2019.
- [18] Y. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, Japan, April 2018.
- [19] M. F. Franco, L. Z. Granville, L. Z. Granville, and B. Stiller, "BRAIN: blockchain-based reverse auction for infrastructure supply in virtual network functions-as-a-service," in *2019 IFIP*

- Networking Conference (IFIP Networking)*, Warsaw, Poland, May 2019.
- [20] B. An A. Liu et al., “Truthful crowdsensed data trading based on reverse auction and blockchain,” in *Database Systems for Advanced Applications. DASFAA 2019. Lecture Notes in Computer Science*, vol. 11446, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds., Springer, Cham, 2019.
 - [21] R. Ch, G. Srivastava, T. R. Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, “Security and privacy of UAV data using blockchain technology,” *Journal of Information Security and Applications*, vol. 55, article 102670, 2020.
 - [22] C. Thirumalai, S. Mohan, and G. Srivastava, “An efficient public key secure scheme for cloud and IoT security,” *Computer Communications*, vol. 150, pp. 634–643, 2020.
 - [23] Q. Lin, H. Wang, X. Pei, and J. Wang, “Food safety traceability system based on blockchain and EPCIS,” *IEEE Access*, vol. 7, pp. 20698–20707, 2019.
 - [24] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transaction on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
 - [25] G. J. Simmons, “Symmetric and asymmetric encryption,” *ACM Computing Surveys*, vol. 11, no. 4, pp. 305–330, 1979.
 - [26] S. Omohundro, “Cryptocurrencies, smart contracts, and artificial intelligence,” *AI Matters*, vol. 1, no. 2, pp. 19–21, 2014.
 - [27] M. M. Nejad, L. Mashayekhy, and D. Grosu, “Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds,” *IEEE Transaction on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, 2015.

Research Article

Coordinated Control of Distributed Traffic Signal Based on Multiagent Cooperative Game

Zhengkua Zhang¹, Jin Qian,¹ Chongxin Fang,¹ Guoshu Liu,² and Quan Su²

¹College of Information Engineering, Yangzhou University, Yangzhou, China

²Yangzhou Guomai Communication Development Co. LTD., Yangzhou, China

Correspondence should be addressed to Zhengkua Zhang; zhangzh@yzu.edu.cn

Received 7 November 2020; Revised 9 December 2020; Accepted 19 May 2021; Published 2 June 2021

Academic Editor: Zhipeng Cai

Copyright © 2021 Zhengkua Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the adaptive traffic signal control (ATSC), reinforcement learning (RL) is a frontier research hotspot, combined with deep neural networks to further enhance its learning ability. The distributed multiagent RL (MARL) can avoid this kind of problem by observing some areas of each local RL in the complex plane traffic area. However, due to the limited communication capabilities between each agent, the environment becomes partially visible. This paper proposes multiagent reinforcement learning based on cooperative game (CG-MARL) to design the intersection as an agent structure. The method considers not only the communication and coordination between agents but also the game between agents. Each agent observes its own area to learn the RL strategy and value function, then concentrates the Q function from different agents through a hybrid network, and finally forms its own final Q function in the entire large-scale transportation network. The results show that the proposed method is superior to the traditional control method.

1. Introduction

With the continuous growth of modern urban road traffic volume and road network density, traffic congestion has become a global common problem. In a region, with the passage of time, the congestion of intersection may gradually spread to several intersections in the surrounding area or even all intersections in the whole area. However, due to the limitation of urban space, it is difficult to realize the connection by expanding the road [1]. Therefore, on the premise of not changing the road network structure, the traffic signal adaptive control strategy can improve the traffic efficiency of the intersection and effectively reduce the emission pollution caused by vehicle starting and braking.

1.1. Related Work

1.1.1. Nonreinforcement Learning Method. The traditional traffic signal uses a fixed phase time control method [2], that is, the time and sequence of the traffic light switching at each intersection are set to a fixed mode. Although this method is

simple and easy to implement, it is easy to cause a situation where one phase is unblocked and another phase is congested, which is not conducive to alleviating traffic conditions. Later, with the development of wireless sensors, the control method began to become intelligent. Heuristic signal control methods appeared; the signal control scheme can be formulated according to the traffic information at the intersection [3], such as Jiao et al. [4] switched phase sequence to achieve congested phase priority traffic. In actual application scenarios, this method greatly reduces the traffic congestion problem compared to the fixed mode control method. But its control range is limited to a single isolated intersection. It not only has no coordination mechanism with adjacent intersections but also is not suitable for complex traffic network and dynamic traffic flow. In recent years, the algorithm of traffic signal control is developing towards intelligence, for example, swarm intelligence algorithms (particle swarm optimization (PSO), ant colony optimization (ACO), etc.) [5], fuzzy logic reasoning (FLR) [6], and artificial neural network (ANN) [7]. Although the above methods can solve the optimization problem of traffic signals, most of

the methods are limited to relatively stable traffic conditions and cannot complete real-time strategy updates in actual complex and changeable traffic conditions.

1.1.2. Single-Agent Reinforcement Learning. Recently, the advantages of RL application in transportation are gradually reflected. RL has strong adaptive control capabilities as a branch of machine learning (ML). As one of the classic RL algorithms, Q-learning [8] was first used in traffic signal control algorithms [9]. It belongs to model-free, and it usually learns the optimal strategy directly through continuous attempts. In the beginning, Q-learning showed excellent ability in single intersection signal control [10–14]; then, it is combined with the multiagent model to apply to multi-intersection signal control. But as the number of intersections increases, the state of the intersection will increase, and the global/joint action space of the agent will increase exponentially. This brings the challenge of vector dimensionality explosion. In order to avoid or solve this problem, many scholars have done research. In [15], feedforward neural network and Q-learning value function to approximate. Kernel methods are introduced in [16] to continuously update the feature vector to adjust the Q function. In [17], radial basis function, the A-CATs controller, acts as the Q function approximator. Although these methods can solve the problem of dimensional explosion, they are limited to simple traffic environment, until Mnih et al. [18] proposed the deep Q network (DQN) algorithm, which uses a neural network to approximate the Q value of all possible behaviours in each state. It is one of the widely used deep reinforcement learning (DRL) algorithms. This algorithm also has a very wide range of applications in the field of intelligent transportation [19–22]. Li et al. [19] use DQN pairs to take the sampled traffic state as input and the maximum intersection throughput as output. A cooperative DQN with Q value transmission (QT-DQN) is proposed in [20], which discretized the traffic information and then entered the state encoding. The method has combined with the state of neighboring agents to search for the optimal control strategy. However, no matter how powerful a neural network is in actual use, without an excellent coordination mechanism, it will be difficult to cope with the large-scale, complex, and changeable traffic environment.

1.1.3. Multiagent Reinforcement Learning. The MARL framework is generally used in the coordinated control of multiple intersections. In the MARL framework, each agent transmits information to each other and cooperates with each other to obtain the overall optimal strategy. The max-plus algorithm is used in [23] to implement an explicit coordination mechanism in the agent's reward function so that the DQN of a single agent can be applied to the multiagent system (MAS). In [24], there is direction of traffic flow that each agent should emphasize during the learning process and improve the conflict between adjacent intersections in the max-plus algorithm, thereby enhancing the coordination between agents. Although the above algorithm can find the Nash equilibrium strategy of multiagent reinforcement learning to a certain extent, it cannot allow the agent to adjust and optimize its

own strategy according to the opponent's strategy. The algorithm can only find the Nash equilibrium strategy of the random game. In addition, when further extended to actual traffic systems with more intersections, this method will quickly become difficult to calculate. For large-scale control tasks through DRL, the work in [25] considers the application of policy gradient methods to control traffic signal timing. The author describes the problem as a continuous control problem and applies the depth deterministic strategy gradient (DDPG) to the centralized control of traffic signals in the entire transportation system. However, in their experiments, this simple centralized method only achieved slightly better performance than ordinary Q-learning on the traffic grid at six intersections.

1.2. Contributions. Many studies have realized the optimization methods to solve the traffic control problems by offline searching control parameters according to historical traffic patterns. They have obvious limitations when the traffic flow changes significantly in a short period of time. In order to solve the related problems, this study is aimed at using agent-based framework and emerging intelligent control technology to achieve traffic adaptive control. A computational framework for reinforcement learning of cooperative game multiagent (CG-MARL) is proposed. The main contributions can be summarized as follows:

- (1) The paper adopts an algorithm to make the monotonicity of the joint action value function and each local value function the same. Taking the maximum action for the local value function is to maximize the joint action value function
- (2) Our work verifies the proposed algorithm from three evaluation indexes: average vehicle flow speed, average vehicle delay time, and average emission of CO_x

1.3. Organization. The other parts of this article are arranged as follows. Section 2 mainly introduces the multiagent system model. Section 3 elaborates on the algorithm implementation. Section 4 is the experimental results and evaluates the performance of the proposed method. Lastly, we make a conclusion in Section 5.

2. System Model

We aim to design a multiagent learning framework with cooperative learning. The framework can make full use of the state information of intersections and the influence of adjacent intersections. The multi-intersection traffic network in the region is modeled as a multi-intersection agent system. Each agent controls an intersection through a deep learning network. In this way, the method can optimize the overall traffic signal plan for regional traffic scenarios and balance the congestion traffic at each intersection.

In a multiagent control system, each agent improves its strategy by interacting with the environment to obtain reward values, as to obtain the optimal strategy in the environment. This section mainly describes the multiagent traffic signal control framework, as shown in Figure 1. This

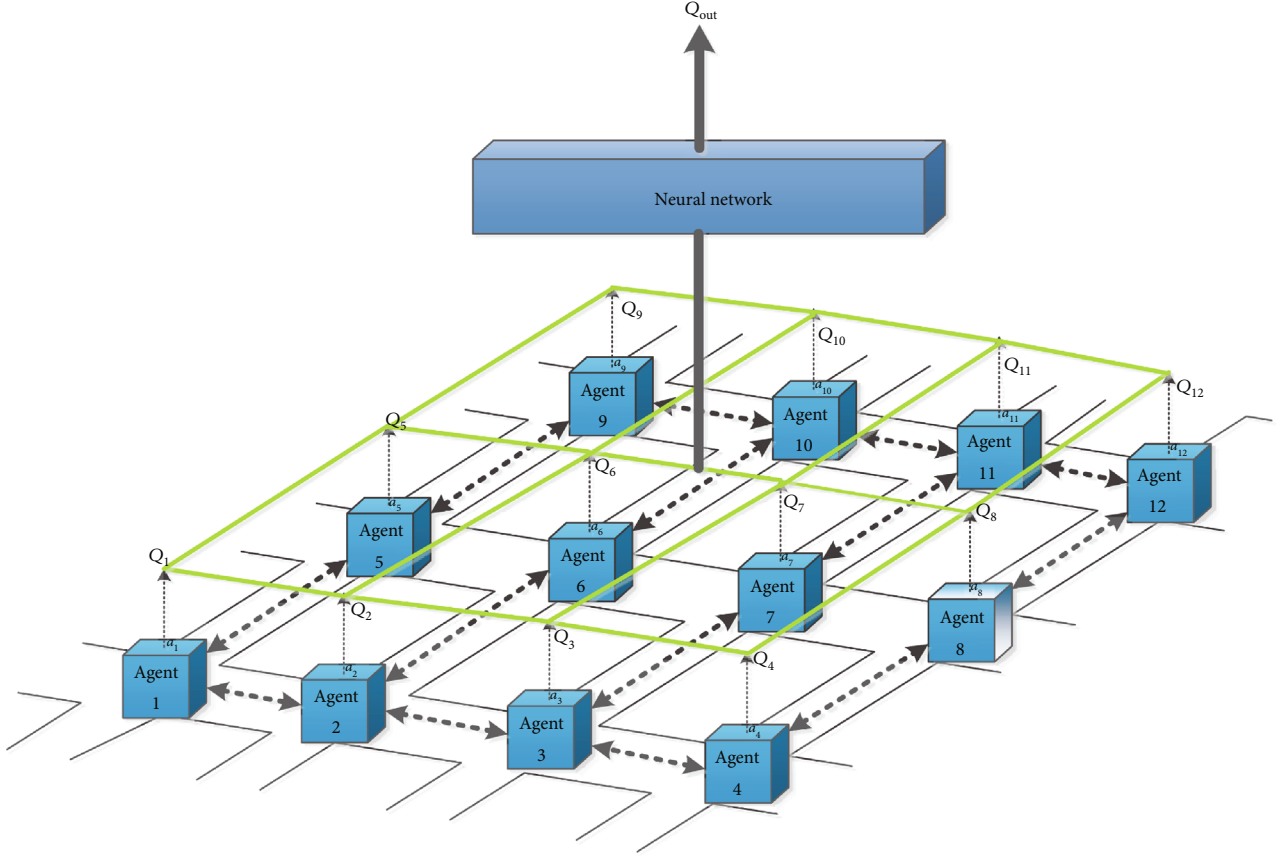


FIGURE 1: Multiagent reinforcement learning model.

framework consists of three parts. The bottom layer is the perception layer, that is, the agent perceives local traffic state information and makes corresponding decision actions to generate the Q function value; the middle layer is the Q network layer, which is the Q value generated by each agent merging; the highest layer is the coordination layer, which judges the pros and cons of all Q function values and coordinates and controls all agents in the lowest layer. In this framework, each intersection is equivalent to an agent structure, the intersection agent can perceive the surrounding traffic state information, and each agent makes decisions in discrete time intervals based on the perceived traffic state information. This framework is a distributed structure. Each agent can communicate with its neighbors to publish its current state information, ensuring that agents can coordinate with each other, as to achieve the goal of stabilizing the entire system.

3. An Algorithm for Multi-Intersection Signal Control

This part first introduces the related theoretical basis of reinforcement learning and then proposes a multiagent multi-intersection signal control reinforcement model and a collaborative reinforcement learning algorithm.

3.1. Reinforcement Learning. RL is a branch of machine learning, which is realized through interaction with the envi-

ronment. This is a kind of goal-oriented learning. The learner does not know which behaviour is the best at first, and this is determined from the consequences of his actions. The basic model of RL, Markov Decision Process (MDP), provides a mathematical framework for modelling decision scenarios, which can be defined by 6 tuples:

$$\langle S, A, P_{ss'}^a, R_{ss'}^a, \gamma, V | s, s' \in S, a \in A \rangle, \quad (1)$$

where S represents a set of states the agent and s, s' belong to the state at a certain time and the next time. A represents the execution of the agent when it transitions from one state to another. $P_{ss'}^a$ is the transition probability, which is the probability that the agent performs a certain behaviour a and transfers from one state s to another state s' . $R_{ss'}^a$ is the probability that the agent will perform a certain behaviour a and transfer from one state s to another state s' . γ is the discount factor, which controls the importance of instant rewards and future rewards. Since there is no final state in the continuous task, its return function can be redefined. The sum of the returns is infinity, which introduces a discount factor. The reward function is defined as

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (2)$$

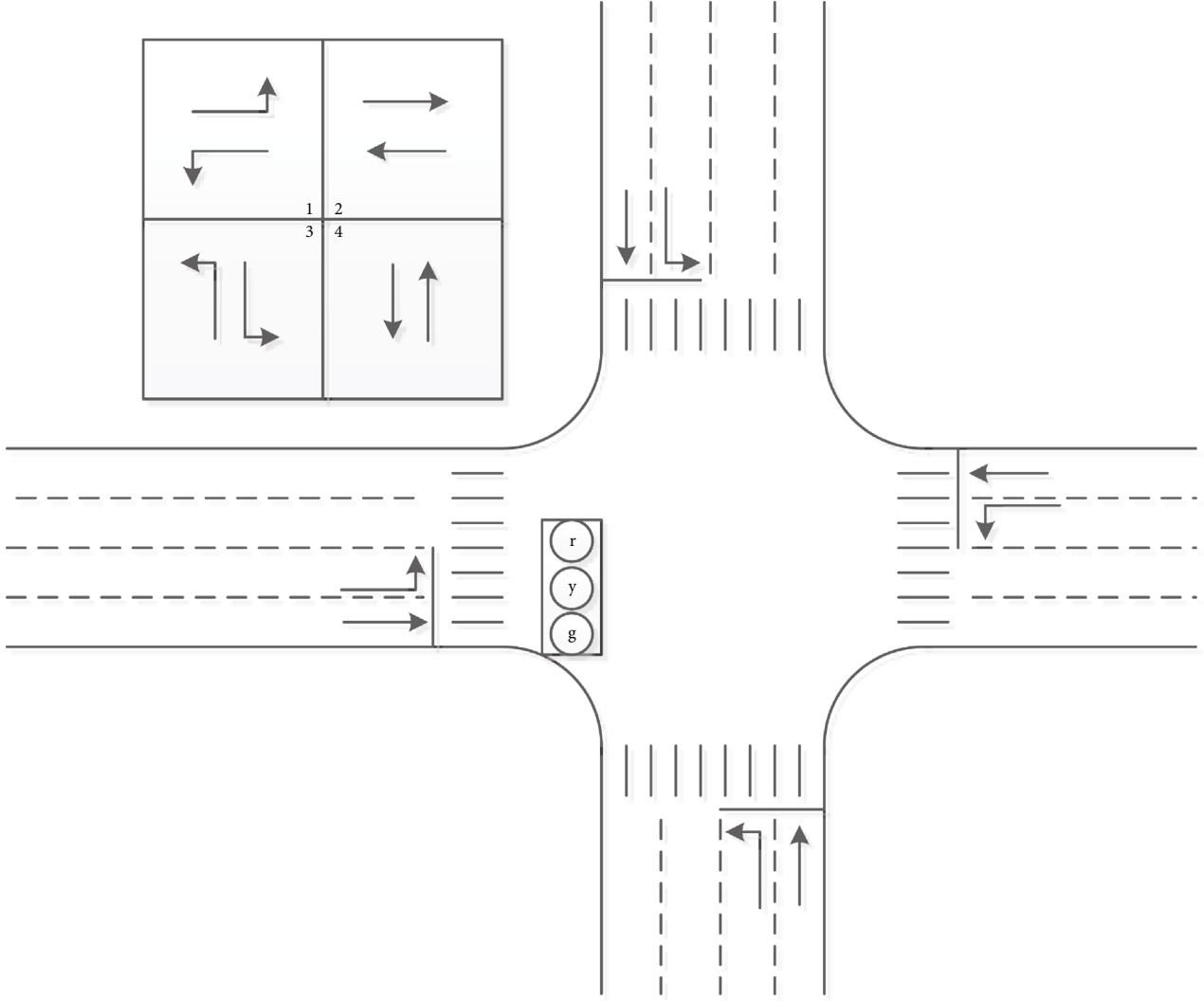


FIGURE 2: Schematic diagram of intersection.

where R_t is the sum of all the rewards and r_t is the reward value at each step.

MDP is essentially a probability model, which only depends on the current state to predict the next state, but has nothing to do with the previous state. Another way to think of this is that the future action state is independent of the past state and the selected action. However, it is obviously unrealistic to calculate the future discount reward for each state. Therefore, the state behaviour value function, also known as Q function, is introduced to indicate the optimal degree of agent following the specific behaviour of policy π in a certain state. The expression is

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right]. \quad (3)$$

Q-learning is one of the classical algorithms in RL. It is a time difference algorithm where the state behaviour value

pair is considered, that is, the action of behaviour a is executed in states. The expression is

$$Q(s, a) = Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right). \quad (4)$$

However, the traditional Q-learning algorithm is only suitable for an environment with a finite number of states and a finite number of behaviours. Later, with the development of artificial neural networks, deep reinforcement learning overcomes this point, which is to approximate the Q function with a certain parameter: $Q(s, a; \theta) \approx Q^*(s, a)$. Use a weight for the neural network to approximate the values of all possible behaviours in each state, approximate the Q function as a function approximator, and minimize the error through gradient descent, such as DQN which uses CNN network and DRQN which uses RNN network.

The agent's environment is stable in single-agent reinforcement learning. On the contrary, the environment is

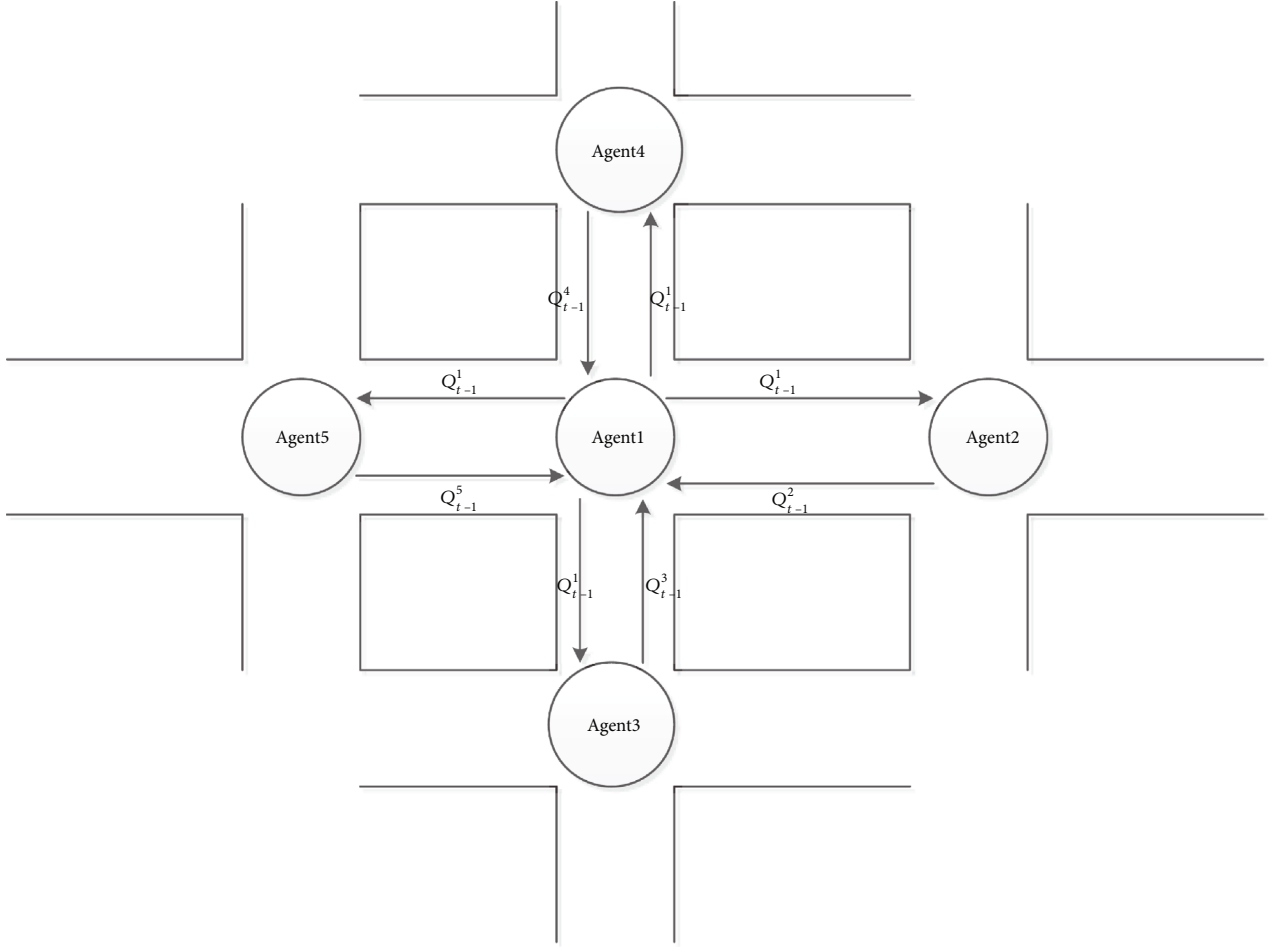


FIGURE 3: The structure of the CG-MARL.

complex and dynamic in multiagent reinforcement learning, which brings great difficulties to the learning process. Multiagent reinforcement learning is a random game that combines the Nash strategy of the stage game in each state to become an agent's strategy in a dynamic environment. It continuously interacts with the environment to update the Q value function (game reward) in the stage game of each state.

For a random game, it can be written as $(n, S, A_1, \dots, A_n, Tr, \gamma, R_1, \dots, R_n)$, where n represents the number of agents, S represents the state space, A_n represents the action space of the n -th agent, $Tr : S \times A_1 \times \dots \times A_n \times S \rightarrow [0, 1]$ represents the state transition probability, $R_i : S \times A_1 \times \dots \times A_n \times S \rightarrow R$ represents the return value obtained by the n -th agent under the current state and connected actions, and γ represents the cumulative reward discount coefficient. Random games are also Markovian. The next state and reward are only related to the current state and the current connection action.

A multiagent reinforcement learning process is to find the Nash equilibrium strategy for each state and then combine these strategies. $\pi_i : S \rightarrow A_i$ is an agent's strategy, which selects the best Nash strategy in each state. The optimal strategy of multiagent reinforcement learning (Nash

equilibrium strategy of random game) can be written as $(\pi_1^*, \dots, \pi_n^*)$, $\forall s \in S, i = 1, \dots, n$, satisfying

$$V_i(s, \pi_1^*, \dots, \pi_i^*, \dots, \pi_n^*) \geq V_i(s, \pi_1^*, \dots, \pi_i, \dots, \pi_n^*), \forall \pi_i \in \Pi_i, \quad (5)$$

where $V_i(s, \pi_1^*, \dots, \pi_i^*, \dots, \pi_n^*)$ is discount cumulative status value function. Abbreviate the above formula to $V_i^*(s)$. Use $Q_i^*(s, a_1, \dots, a_n)$ to represent the action state discount cumulative value function. In the stage game of each fixed state, Q_i^* is used as the reward of the game to solve the Nash equilibrium strategy. According to the Bellman formula in reinforcement learning, we can get

$$\begin{aligned} V_i^*(s) &= \sum_{a_1, \dots, a_n \in A_1 \times \dots \times A_n} Q_i^*(s, a_1, \dots, a_n) \pi_1^* \\ &\quad \cdot (s, a_1) \dots \pi_n^*(s, a) Q_i^*(s, a_1, \dots, a_n) \\ &= \sum_{s'} Tr(s, a_1, \dots, a_n, s') \left[R_i(s, a_1, \dots, a_n, s') + \gamma V_i^*(s') \right]. \end{aligned} \quad (6)$$

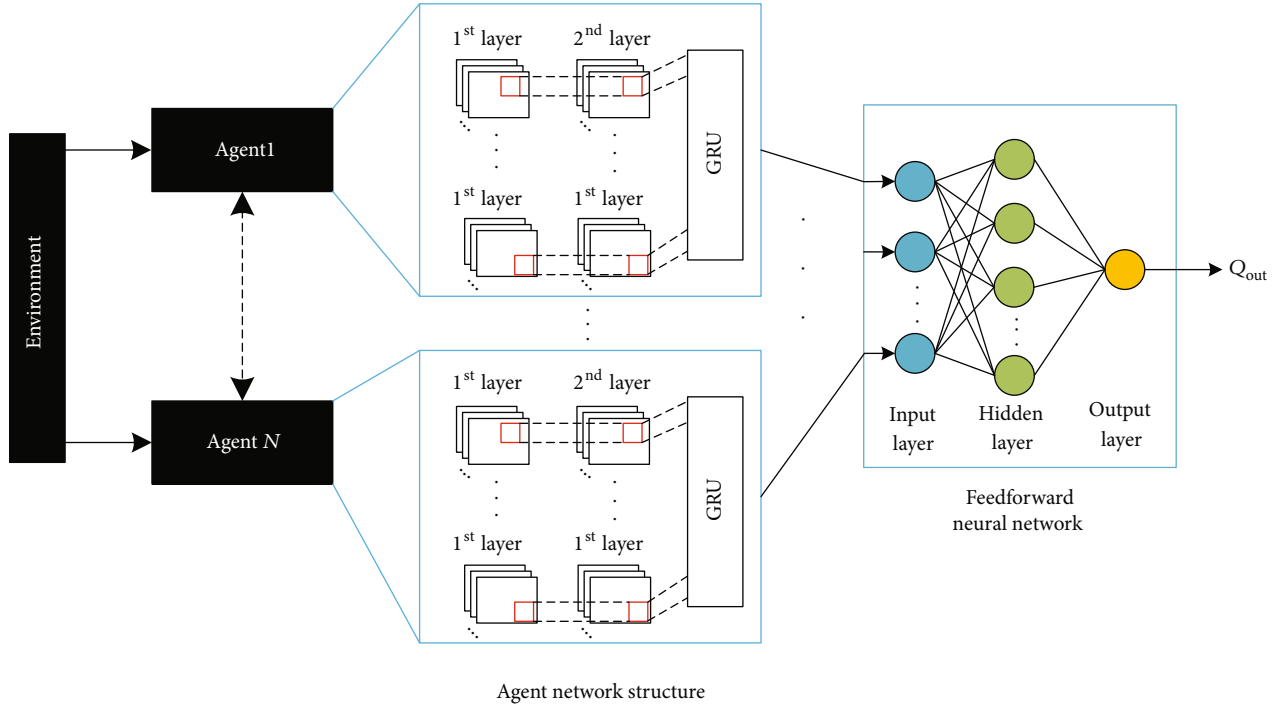


FIGURE 4: The network structure of CG-MARL.

```

Initialize GRUi with random weight  $\theta_i > 0$ 
Initialize feedforward neural network with random weight  $\vartheta_i > 0$ 
Initialize  $\varepsilon, t_d$ 
Local_len (hyperparameter: the training time for each episode)
Initialize  $Q_0^i = 0$  for episode = 1 to  $N$  do
  for  $t = 1$  to Local_len do
    Observe current intersection state  $s_t^i$ ;
    The agent randomly selects an action with probability  $\varepsilon$  and selects an action  $a_t^i = \arg \max_{a' \in A} Q_i^i(s_t^i, a'; \theta_i)$  with probability  $1 - \varepsilon$ ;
    Execute action  $a_t^i$  and observe all agent reward  $r_t^i$  and next state  $s_{t+1}^i$ ;
    Use GRU neural network update  $Q_i(s_t^i, a_t^i)$ ;
     $t = t + t_d$ .
  end
The global argmax performed on  $Q_{out}$  is the same as a single set of argmax operations performed on each agent.  $Q_{out}$  is updated as (8).
end

```

ALGORITHM 1: Cooperative game multiagent reinforcement learning for intersections signal control.

The Nash strategy of MARL can be rewritten as

$$\begin{aligned}
 & \sum_{a_1, \dots, a_n \in A_1 \times \dots \times A_n} Q_i^*(s, a_1, \dots, a_n) \pi_1^*(s, a_1) \dots \pi_n^*(s, a_n) Q_i^*(s, a_1, \dots, a_n) \\
 & \geq \sum_{a_1, \dots, a_n \in A_1 \times \dots \times A_n} Q_i^*(s, a_1, \dots, a_n) \pi_1^*(s, a_1) \dots \pi_i(s, a) Q_i^*(s, a_1, \dots, a_n).
 \end{aligned} \tag{7}$$

Random games can be classified according to the reward function of each agent. If the reward function of the agent is the same, it is called a fully cooperative game or a team game. If the reward function of the agent is reversed, it is called a perfect competition game or a zero-sum game. In order to solve

the random game, it is necessary to solve the stage game of each state, and the reward value of each stage game is $Q(s, \bullet)$.

3.2. Multiagent Reinforcement Learning Framework. This paper will use the basic theory of game reinforcement learning to build a framework for multiagent reinforcement learning [26]. The value function of each agent is integrated to obtain a joint action value function. $\tau = (\tau_1, \dots, \tau_n)$ denotes the joint action-observation history, where $\tau_i = (a_{i,0}, o_{i,1}, \dots, a_{i,t-1}, o_{i,t})$ is the action-observation history, $a = (a_1, \dots, a_n)$ is the joint action value function, $Q_i(\tau_i, a_i; \theta_i)$ is the local action value function of agent i , and the local value function only depends on the local observation of each agent. The

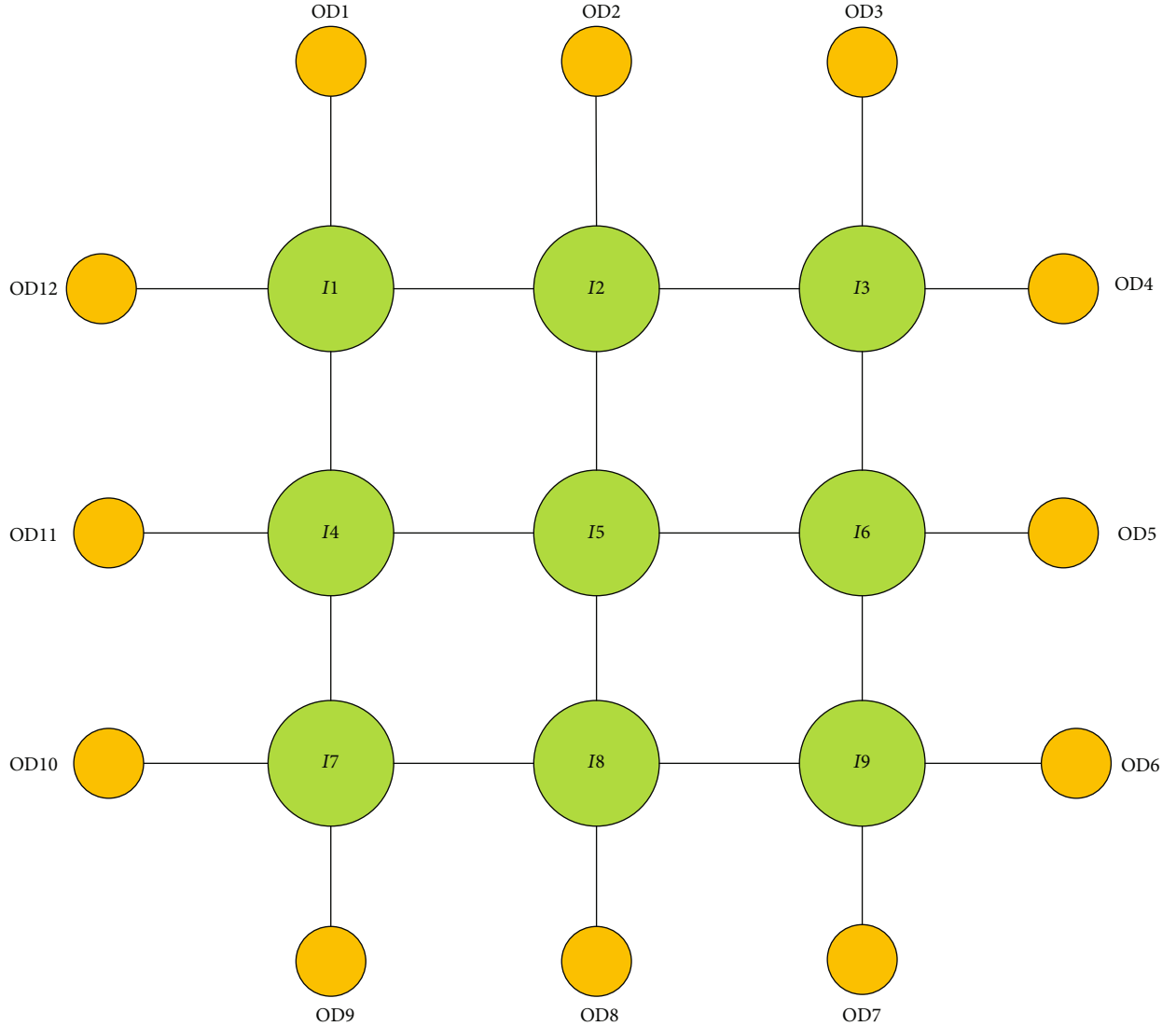


FIGURE 5: The structure of road network for the nine intersections.

joint action value function is equal to each local action value function, and its monotonicity is the same, as shown in the following formula:

$$\text{argmax} Q_{\text{out}}(\tau, u) = \begin{pmatrix} \text{argmax}_{u_1} Q_1(\tau_1, u_1) \\ \cdot \\ \cdot \\ \cdot \\ \text{argmax}_{u_n} Q_n(\tau_n, u_n) \end{pmatrix} \quad (8)$$

s.t. $\frac{\partial Q_{\text{out}}}{\partial Q_i} \geq 0, \quad \forall i \in \{1, 2, \dots, n\}.$

3.3. Model Building. In order to facilitate the description of signal control problems at multiple intersections, this article takes an intersection as an example. The structure of the

intersection is shown in Figure 2, which contains four phases, each with two lanes. The intersection signal lights control and adjust the phase sequence switching to ensure the orderly passage of vehicles.

3.3.1. State Space. Traffic signal control mainly depends on the vehicle information at the intersection, that is, the size of the vehicles queued at an intersection. In MAS, the joint state space increases exponentially. If the joint state space is designed according to the number of queues for each phase at each intersection, the result of dimensionality explosion will obviously occur, so the state space of each intersection needs to be simplified. In this article, in order to better describe the traffic state of the intersection, a model is established according to the number of vehicles in the queue and the maximum vehicle capacity of the intersection, where x is the state of the intersection, l_{que} is the number of vehicles in the queue, l_{max} is the maximum number of vehicles that

TABLE 1: OD flow matrix.

Input OD	Output OD												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1		229	255	186	136	57	400	308	176	61	61	178	2047
2	387		310	240	335	119	138	251	162	95	334	159	2530
3	276	109		333	218	256	208	218	114	59	98	219	2108
4	383	220	359		146	280	77	217	398	157	326	108	2671
5	303	262	173	65		271	219	247	233	145	182	199	2299
6	290	227	100	379	343		387	203	188	301	57	358	2833
7	131	77	205	281	267	94		75	264	161	244	186	1985
8	291	59	126	108	180	57	214		67	211	303	366	1982
9	174	232	250	271	389	86	118	370		221	256	69	2436
10	142	221	286	281	187	304	124	339	60		217	285	2446
11	227	346	144	398	289	394	283	140	279	345		179	3024
12	67	199	196	112	269	389	139	171	374	358	315		2589
Total	2671	2181	2404	2654	2759	2307	2307	2539	2315	2114	2393	2306	

can be queued, and φ is the queuing evaluation parameter (this article takes the value 0.5). Each intersection agent has four phases, and the combined state of the four phases is $X = [x^1, x^2, x^3, x^4]$, that is, the joint state space of the entire regional traffic can be expressed as $X = [X_1, X_2, \dots, X_n]$.

$$x = \begin{cases} 0, & \frac{l_{\text{que}}}{l_{\text{max}}} < \varphi, \\ 1, & \frac{l_{\text{que}}}{l_{\text{max}}} \geq \varphi. \end{cases} \quad (9)$$

3.3.2. Action Space. Action space means that the agent selects an action $a_i \in A_i$ after observing the state of intersection i at time step t . A_i refers to the collection of all actions of the agent and then executes the selected action. In this article, the possible action is the traffic signal phase configuration. The intersection as shown in Figure 1 can be set up with four different actions according to the four phases, which are turn left from east to west, go straight from east to west, turn left from north to south, and go straight north-south. The time for each execution action is a fixed minimum unit time interval with a length of τ . At time step $t + 1$, the agent observes the new state affected by the latest operation and selects the next operation. The agent can take the same action at time step $t + 1$ and t .

3.3.3. Reward. The reward function is to evaluate how well the actions performed by the agent interact with the environment affect the environment. The agent first observes the state of the external environment, selects, and executes a pre-set action. Then, the environment feedbacks the effect of the executed action on itself to the agent, so the reward function can give the agent scalar feedback information. The agent is looking for strategies in the direction of reward maximization. There are many reward mechanisms in traffic signal control, such as the cumulative delay time of the vehicle, the flow rate of the vehicle, and the throughput of the vehicle.

In this article, the change of queued vehicles is used as the reward function, which is defined as follows:

$$r_i(s_t^i, a_t^i) = l_t^i - l_{t+1}^i, \quad (10)$$

where l_t^i and l_{t+1}^i are the average queue length at time t and $t + 1$, respectively, and are the reward function at time t . It can be seen from the function expression that if the average long queue of vehicles at time $t + 1$ is less than that at time t , the function value is positive, which means that the currently executed action has a positive impact on the current traffic state.

3.4. Cooperative Game Multiagent Reinforcement Learning (CG-MARL) Algorithm. In this section, a multiagent distributed learning algorithm based on cooperative games is proposed to coordinate the signal control of multiple intersections. CG-MARL establishes an agent network model for all intersections in an area. Each agent has a DQN to maintain the control of each intersection and tries to find the optimal strategy solution in a dynamic environment. At the same time, it interacts with neighboring intersection agents to achieve the purpose of distributed and coordinated control of the entire area signal. The plane structure of CG-MARL is shown in Figure 3.

In CG-MARL, the Q value of the neighboring agent will be transferred to the local agent for policy learning, and then, the Q value of each agent's learning result will be uploaded to the hybrid network for further cooperative game solving so that the multiagent system can coordinate control of all intersections in the entire area. The action selection of an intersection not only considers its own Q value but also depends on the Q value of its neighbors. The overall system considers the Q value of all agents in the area. This cooperation mechanism is conducive to balancing the traffic flow between intersections and improving the overall performance of the regional transportation network.

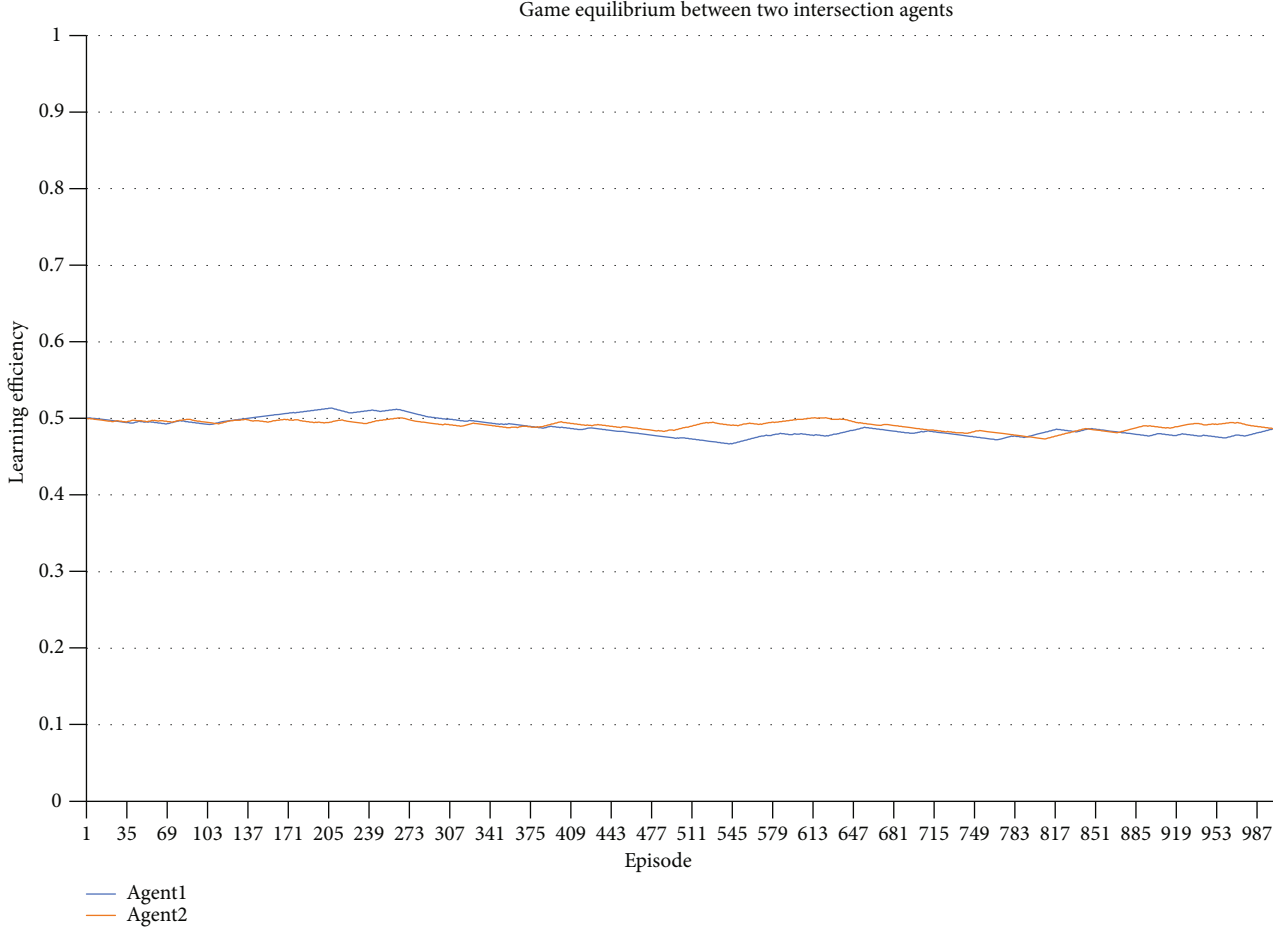


FIGURE 6: Game equilibrium between two intersection agents.

The input of the entire network is the discrete state coding of intersection traffic information, and the output is a vector formed by the estimated Q values of all actions in the observation state. GRU estimation can use gradient-based training algorithms to automatically extract features from the original traffic state of the intersection and approximate the Q function. Specifically, firstly, each agent has its own separate network representation value function. Through the deep Q network, the input at each time step is the current observation result of the environment and the result of the last time step operation. Secondly, the output of each agent is integrated into a hybrid network. The hybrid network is a feedforward neural network, which takes the output of each agent as input and monotonically mixes it to produce the output of the entire area network. It is worth noting that in order to emphasize the monotonicity constraint, the weight of the hybrid network must be restricted to nonnegative numbers, which allows the hybrid network to approximate any monotonic function arbitrarily. The structure of the entire network is shown in Figure 4.

The loss function of the entire network is

$$\text{Loss} = \sum_{i=1}^n \left[(y_i^{\text{out}} - Q_{\text{out}}(s, a; \theta))^2 \right]. \quad (11)$$

The pseudocode is shown in Algorithm 1. At each time step t , the state observed by agent i is input into the evaluation network. The agent i chooses an action performed by the α -greedy method according to the output Q value. The agent gets the reward and enters the next state. For each agent a , there is an agent network representing its individual value function. We denote the proxy network as RQN, and they receive the input of the current personal observation result and the last operation at each time step, as shown in Figure 4.

The parameter of GRU is updated by the stochastic gradient descent algorithm. A schematic diagram of the training process is also shown in Figure 4. Since the cooperative training agent hybrid network, the weight is generated by a separate super network. Each super network takes the state as input and generates the weight of one layer of the hybrid network. It is composed of a single linear layer, followed by an absolute activation function to ensure that the weight of the hybrid network is nonnegative.

Then, the output of the super network is a vector, which is shaped into a matrix of appropriate size. The bias is generated in the same way, but is not limited to nonnegative. The final deviation is generated by the nonlinear 2-layer super network. The state is used by the super network instead of being passed directly to the hybrid network, because Q_{out}

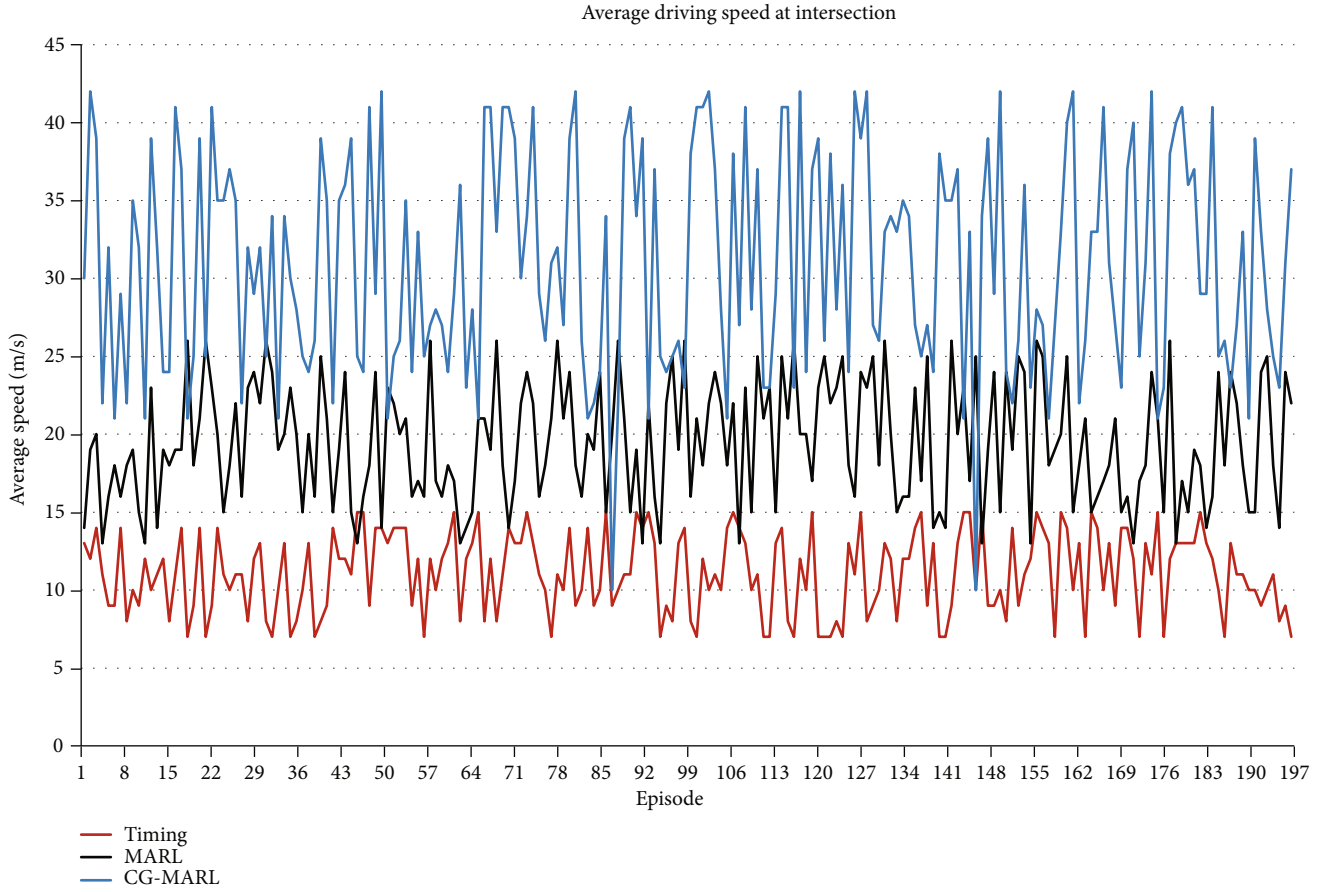


FIGURE 7: Average driving speed at intersection.

allows to rely on additional state information in a nonmonotonic way. Therefore, overconstraint passes certain function of s along with the single-agent value through the monotonic network. On the contrary, the use of the super network can adjust the weight of the monotonic network in any way, merging the complete state into the joint action value estimate.

4. Results and Discussion

In order to verify the feasibility and superiority of the proposed algorithm in multi-intersection signal control, the experiment mainly establishes a traffic model and compares it with other algorithms. All traffic flow models are established based on the cell transmission model (CTM), which uses finite differences to design an approximate method for the macroscopic road traffic flow model. It has a huge advantage in response to the traffic flow characteristics with large fluctuations. Three evaluation indicators are used in the experiment: unit travel speed, that is, the speed of the vehicle in the lane (km/s); vehicle delay per unit time, that is, the average delay time of the vehicle in the lane (veh/s); vehicle emission per unit time, that is, the average CO_x pollutants emitted by the vehicle in the lane.

This paper uses a 3×3 grid network as the experimental simulated road network environment. The specific structure is shown in Figure 5. The number I represents the intersec-

tion node, the number OD represents the input node of the network, the connection between each node is represented as a two-way driving lane, the length is 800 m, the capacity is 1800 veh/h, and the average length of the vehicle is 4 m.

In the experiment, the arrival situation of the traffic flow is random. Each vehicle arrives at an intersection and turns to the next intersection at random. The turning probability of each intersection is set to 0.3~0.7. In order to simulate low, medium, and high flow rates, the flow rate in each OD direction in Table 1 is adjusted with a ratio of 1 to 2, and the control effect of CG-MARL under different flow rates is analyzed.

Figure 6 shows the game balance result diagram of two intersection agents. In this figure, we can see that the two intersection agents gradually reach a stable state. It illustrates the game balance is reached.

Figure 7 shows the average speed of vehicles at all intersections in the simulation step. Note that the traveling speed of the vehicle in Figure 7 represents the average speed of the vehicle traveling at all intersections due to the switching of traffic lights. If the length of the green band at the intersection can be long enough, the speed of the passing vehicle will not decrease, and the speed will be relatively faster. The CG-MARL algorithm can improve speed of the vehicle. Because the nature of reinforcement learning can optimize control actions, more vehicles are not waiting at intersections, thereby increasing the speed of the entire process. If we

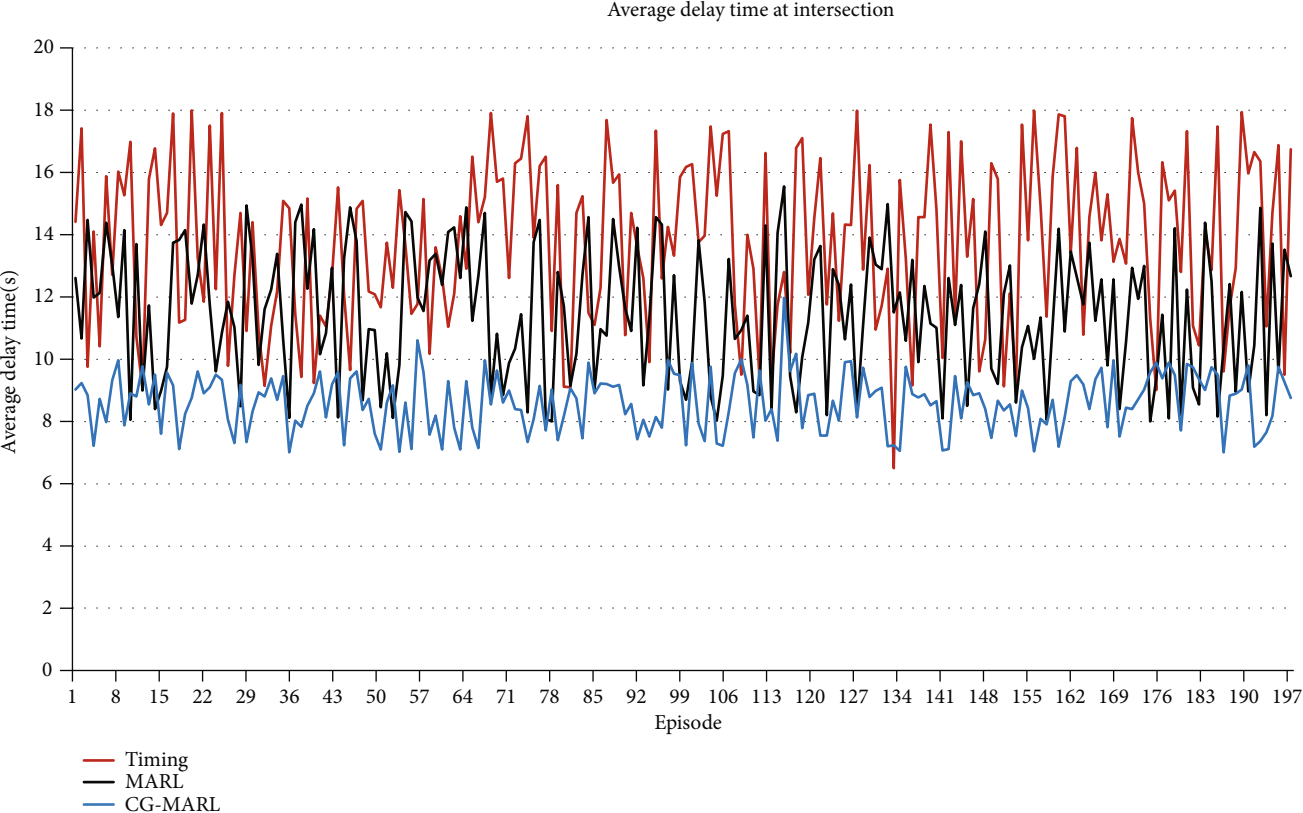


FIGURE 8: Average delay time at intersection.

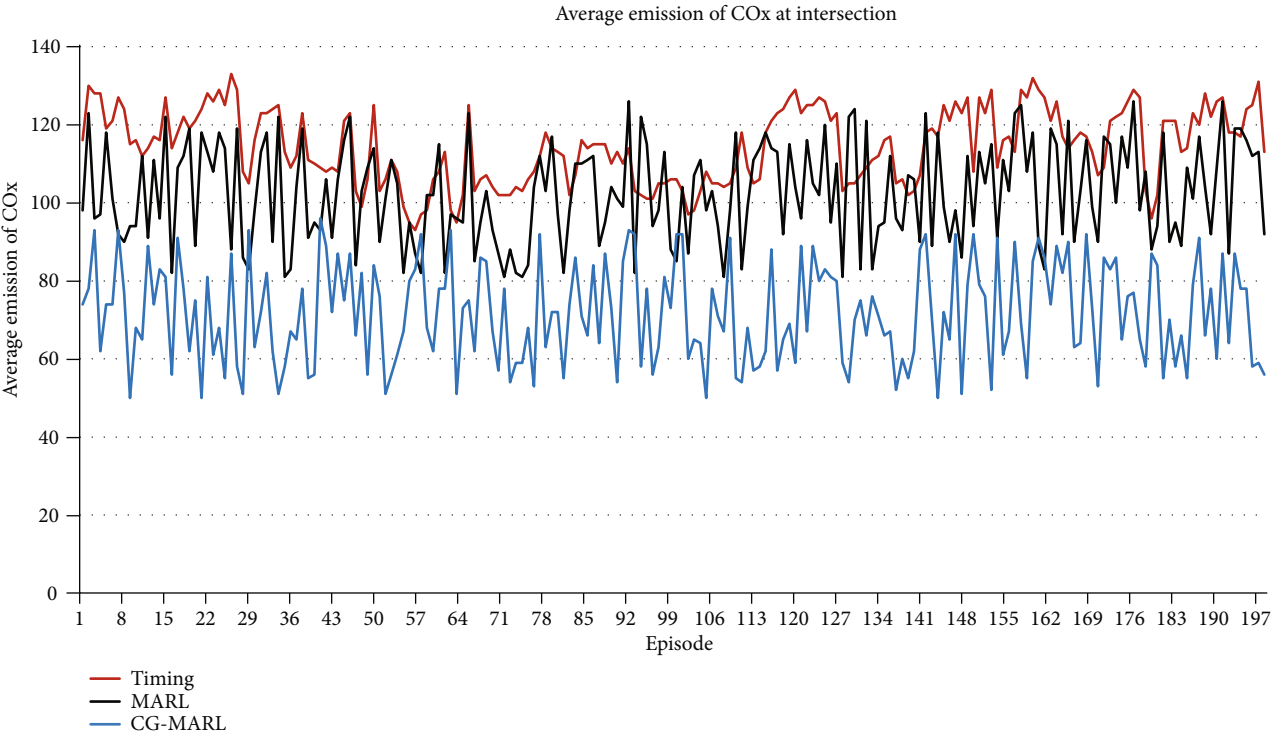


FIGURE 9: Average emission of COx at intersection.

compare the performance between the CG-MARL algorithm proposed in this paper and the traditional MARL algorithm, we can observe that our solution is better than MARL, and the average speed increased by 43.02%. The main reason is that our solution also considers that our solution exchanges information between multiple agents, which is very useful for realizing the optimization of the entire system. In the traditional MARL algorithm, the agent only obtains the information of its own environment and does not pay attention to the information interaction between the agents.

Figure 8 shows the comparative result of the average delay time. Likewise, our solution is superior to other solutions. In the results, the vehicle delay time is the most direct performance indicator, because the traffic light control algorithm should always reduce the vehicle delay time at the intersection. The average delay time is reduced by 26.59%. As the delay time is shortened, unnecessary parking and traffic congestion are reduced.

Figure 9 shows a comparison emission of CO_x. We can see that our solution is better than others. The result shows that when the delay time of vehicles at the intersection is reduced, the CO_x emission will also be relatively reduced, and the average CO_x emission is reduced by 32.18%. With the reduction of carbon dioxide emissions, it means that the waiting time for the red light at the intersection will also be reduced.

5. Conclusions

This paper proposes a cooperative game multiagent reinforcement learning (CG-MARL) for regional traffic control. CG-MARL can extract status information effectively at intersections and enable multiple intersections to control traffic conditions based on regional coordination signals. In the framework of learning, agents can exchange information with each other and then concentrate on learning to achieve the goal of regional coordination. We can train hierarchical architecture efficiently by using training models from simple tasks. Importantly, the proposed CG-MARL framework can be extended to have different intersection structures and numbers. The results show that the method proposed in this paper can greatly improve the operation efficiency of vehicles and reduce the delay time of vehicles and the emission of pollutants to a certain extent, respectively. At present, this algorithm is limited to traffic single objective optimization. In future work, the algorithm can be extended to multiobjective signal control and can be combined with traffic assignment algorithm.

Data Availability

The data are laboratory data. They are generated by simulation software.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] Y. S. Chang, Y. J. Lee, and S. S. B. Choi, "Is there more traffic congestion in larger cities? -scaling analysis of the 101 largest U.S. urban centers-," *Transport Policy*, vol. 59, pp. 54–63, 2017.
- [2] A. J. Miller, "Settings for fixed-cycle traffic signals," *Operational Research Quarterly*, vol. 14, no. 4, pp. 373–386, 1963.
- [3] B. De Schutter, "Optimal traffic light control for a single intersection," *European Journal of Control*, vol. 3, no. 10, pp. 2195–2199, 1999.
- [4] P. Jiao, T. Sun, D. Li, H. Guo, R. Li, and Z. Hou, "Real-time traffic signal control for intersections based on dynamic O–D estimation and multi-objective optimisation: combined model and algorithm," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 619–630, 2018.
- [5] J. García-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 823–839, 2013.
- [6] Y. Bi, X. Lu, Z. Sun, D. Srinivasan, and Z. Sun, "Optimal type-2 fuzzy system for arterial traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 3009–3027, 2018.
- [7] D. Srinivasan, M. C. Choy, and R. L. Chen, "Neural networks for real-time traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 261–272, 2006.
- [8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [9] C. A. Helm, W. Knoll, and J. N. Israelachvili, "Intelligent traffic light control," *Proceedings of the National Academy of sciences of the United States of America*, vol. 88, no. 18, pp. 8169–8173, 1991.
- [10] C. Jacob and B. Abdulhai, "Automated adaptive traffic corridor control using reinforcement learning: approach and case studies," *Transportation Research Record*, vol. 1, no. 19, pp. 1–8, 2006.
- [11] J. Chen and X. Ma, "Adaptive group-based signal control by reinforcement learning," *Transportation Research Procedia*, vol. 10, no. 15, pp. 207–216, 2015.
- [12] C. Wan and M. Hwang, "Value-based deep reinforcement learning for adaptive isolated intersection signal control," *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 1005–1010, 2018.
- [13] P. Ha-li and D. Ke, "An intersection signal control method based on deep reinforcement learning," in *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pp. 344–348, Changsha, China, 2017.
- [14] Y. Liu, L. Liu, and W. Chen, "Intelligent traffic light control using distributed multi-agent Q learning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, Yokohama, 2017.
- [15] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.
- [16] T. Chu, J. Wang, and J. Cao, "Kernel-based reinforcement learning for traffic signal control with adaptive feature selection," in *53rd IEEE Conference on Decision and Control*, pp. 1277–1282, Los Angeles, CA, 2014.
- [17] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

- [18] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transportation Research Part C: Emerging Technologies*, vol. 85, no. 10, pp. 732–752, 2017.
- [19] L. Li, Y. Lv, and F. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.
- [20] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control," *IEEE Access*, vol. 7, pp. 40797–40809, 2019.
- [21] I. Lamouik, A. Yahyaouy, and M. A. Sabri, "Smart multi-agent traffic coordinator for autonomous vehicles at intersections," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, Fez, Morocco, 2017.
- [22] T. Wu, P. Zhou, K. Liu et al., "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020.
- [23] J. C. Medina and R. F. Benekohal, "Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 596–601, Anchorage, AK, USA, 2012.
- [24] A. E. Ohazulike and T. Brands, "Multi-objective optimization of traffic externalities using tolls," in *2013 IEEE Congress on Evolutionary Computation*, pp. 2465–2472, 2013.
- [25] A. Agliari, C. H. Hommes, and N. Pecora, "Path dependent coordination of expectations in asset pricing experiments: a behavioral explanation," *Journal of Economic Behavior & Organization*, vol. 121, pp. 15–28, 2016.
- [26] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning," *International Conference of Machine Learning*, vol. 80, pp. 4295–4304, 2018.

Research Article

Path Planning of Mobile Robot Based on Improved Multiobjective Genetic Algorithm

Kairong Li ¹, Qianqian Hu ¹ and Jinpeng Liu²

¹School of Information Engineering, Yangzhou University, Yangzhou 225127, China

²The 29th Research Institute of China Electronics Technology Group Corporation (CETC-29), Chengdu 610036, China

Correspondence should be addressed to Kairong Li; krli@yzu.edu.cn and Qianqian Hu; 1357824557@qq.com

Received 12 August 2020; Revised 21 September 2020; Accepted 13 April 2021; Published 30 April 2021

Academic Editor: Alessandro Bazzi

Copyright © 2021 Kairong Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Path planning is the core technology of mobile robot decision-making and control and is also a research hotspot in the field of artificial intelligence. Aiming at the problems of slow response speed, long planning path, unsafe factors, and a large number of turns in the conventional path planning algorithm, an improved multiobjective genetic algorithm (IMGGA) is proposed to solve static global path planning in this paper. The algorithm uses a heuristic median insertion method to establish the initial population, which improves the feasibility of the initial path and generates a multiobjective fitness function based on three indicators: path length, path security, and path energy consumption, to ensure the quality of the planned path. Then, the selection, crossover, and mutation operators are designed by using the layered method, the single-point crossover method, and the eight-neighborhood-domain single-point mutation method, respectively. Finally, the delete operation is added, to further ensure the efficient operation of the mobile robot. Simulation experiments in the grid environment show that the algorithm can improve the defects of the traditional genetic algorithm (GA), such as slow convergence speed and easy to fall into local optimum. Compared with GA, the optimal path length obtained by planning is shortened by 17%.

1. Introduction

A mobile robot is a kind of machine device which can perform work automatically [1]. It can not only accept the command of human beings but also act according to the scheme made by artificial intelligence technology [2]. With the development of science and technology, mobile robot has attracted more and more attention. Its application scope is also greatly expanded; from industry, agriculture, medical, service, and other industries to national defense, space exploration, and other dangerous occasions, mobile robots have played an important role [3]. As a key technology for the mobile robot to realize autonomous motion, path planning is one of the most active research directions in the field of mobile robot [4, 5]. The path planning problem of the mobile robot is to make the robot autonomously find a noncollision optimal or suboptimal path from the given starting point to the target point according to certain index requirements (such as path length, path safety, and planning time) within the specified environment with obstacles [6]. According to the degree of

mastering the environment, path planning can be divided into global path planning in known environment and local path planning in unknown environment. According to whether the environmental information changes with time, path planning can be divided into static path planning and dynamic path planning. This paper mainly studies the known static path planning in global environment [7–9]. The general steps include environment modeling, path searching, and path optimization. The planning algorithm used in the path search is the core of the whole mobile robot path planning problem, and the selection of the algorithm determines the quality of the planning path [10]. At present, there are many effective solutions to the problem of mobile robot path planning, but with the continuous development of science and technology, the environment faced by path planning technology will become more complex [11], and the task requirements will become more stringent. This requires that the path planning algorithm should have the ability to quickly respond to the complex environment [12], less energy consumption, and be able to avoid all obstacles. This

is not a problem that can be solved by a single traditional algorithm [13]. Therefore, an improved multiobjective genetic algorithm (IMGGA) is proposed in this paper. The purpose is to improve the shortcomings of the traditional genetic algorithm by optimizing some genetic operations and quickly plan the shortest, collision-free, and less turning safe operation path of the mobile robot in a static grid environment based on multiple planning indexes, so as to solve the path planning problem with more complex environment and task requirements.

In short, the main contributions of this paper are as follows:

- (i) The heuristic median insertion method is designed to generate a feasible initial path and accelerate the algorithm convergence rate
- (ii) The multiobjective fitness function is proposed, which gives different measurement criteria and weights to each indicator according to the planning requirements, so as to ensure that the planned path has the shortest path length, better safety, and smoothness and achieve multiobjective optimization of the algorithm
- (iii) In order to maintain the population diversity in the later stage of the algorithm and avoid premature convergence, the selection operator is designed by the layered method, the crossover operator is designed by the single-point crossover method, and the mutation operator is designed by the eight-neighborhood-domain single-point mutation method
- (iv) The delete operation is added to remove the redundant nodes of the running path to obtain a more optimal path
- (v) On the premise of ensuring that each method is simple and effective, and the optimal path can be generated, the algorithm complexity is reduced and the running speed is accelerated

The rest of this paper is organized as follows. Section 2 reviews the related work on path planning techniques. Section 3 introduces the problem description and environment modeling of path planning. Section 4 describes the path planning method based on improved multiobjective genetic algorithm (IMGGA), including the operating environment model of the mobile robot, the establishment of the initial population, the generation of the multiobjective fitness function, the improved genetic operation, and the redundant point deletion of the optimal path. The simulation results and analysis are given in Section 5. Section 6 summarizes the paper and plans the direction of future work.

2. Related Work

In this section, the related work of the existing robot path planning problem is reviewed, and its advantages and disadvantages are discussed.

The commonly used path planning algorithms can be roughly divided into four categories: traditional algorithms, graphics methods, intelligent bionics algorithms, and other algorithms [14]. Among them, the genetic algorithm (GA) is widely used in mobile robot path planning research because of its good scalability, robustness, and fast random search ability [15]. However, the traditional genetic algorithm not only has slow convergence speed but also is easy to fall into premature convergence [16]. Therefore, many researchers have improved the genetic operator to improve the limitations of GA. Zhang et al. [17] have discussed an improved genetic algorithm based on visible space. The main ideas are the concept of visible space, matrix coding, and improved mutation operators. This method is applicable in both static and dynamic environments. However, the smoothness and security of the path are not considered, and the selection operation is very random, which may increase the running time of the algorithm. Lamini et al. [18] have proposed an improved same adjacency crossover operator to solve the robot path planning problem. Considering the variable length of the chromosome, the operator can generate a feasible path with better fitness value and avoid premature convergence. However, the initial population quality of this method still needs to be improved, and the path length does not reach the optimal.

Through the optimization of fitness function, the feasible path can be more in line with the actual requirements. Chen and Chen [19] have used prior knowledge to generate uninterrupted feasible paths and redefine the fitness function, thereby effectively optimizing the genetic algorithm and enabling the algorithm to complete evolution in a shorter time. However, the optimal path generated by this method still has redundant nodes, the operator optimality of the genetic algorithm needs to be improved, and the timeliness and accuracy of the algorithm also need to be enhanced. Cheng et al. [20] regarded path planning as a multiobjective optimization problem and evaluated the performance of the results based on four self-defined fitness objective functions, so as to solve the path planning problem of reconfigurable robot in a complex obstacle environment, but the efficiency of the algorithm still needs to be improved.

With the deepening of research, the effective combination of the path planning algorithm provides the possibility to solve the new cross science problems that a single planning algorithm cannot face. A parallel search algorithm combining the genetic algorithm and artificial potential field method has been proposed by Duan and Chen [21]. This method makes the potential field function jump out of the local minimum point by introducing the filling potential field, and then, the genetic algorithm is built on the improved artificial potential field model to optimize the global path. However, the planned path is too tortuous, and there are a large number of redundant sections. Yi et al. [22] have proposed an improved metaheuristics-adaptive genetic algorithm. The algorithm first uses the random Dijkstra algorithm to create the initial population and then replaces the conventional selection operator with an adaptive operator. Simulation experiments in 2D complex environments prove that the method can effectively avoid the local convergence problem

in path planning. However, this method only considers the length of the path in terms of fitness function, and the method of generating the initial path is too complicated, resulting in the algorithm running time is too long. Sun et al. [23] have proposed an improved adaptive genetic algorithm for robot path planning. Firstly, the idea of simulated annealing is introduced in the selection operation of the genetic algorithm. Then, the self-adjusting strategies of crossover and mutation operators are improved. Finally, the planning indexes such as safe driving speed and turning times are added into the fitness function. Simulation experiments show that the improved algorithm has good efficiency. However, the generation of the initial population needs to be improved, and the index measurement method is too complex, which leads to the increase of the running time of the algorithm.

It is worth noting that although existing studies have improved traditional genetic algorithms to some extent [24, 25], there is still no one method that can fully consider the selection of the initial population, the multiobjective fitness function, and the improvement of genetic operation. Moreover, most of the improved algorithms have high algorithm complexity and low operating efficiency. For this reason, this paper proposes an improved multiobjective genetic algorithm (IMGA), which solves the defects of the traditional genetic algorithm effectively by designing each link of the algorithm. The specific design details will be introduced in the rest of this paper.

3. Problem Description and Environment Modeling

In this section, we describe some constraints and environment modeling method for mobile robot path planning.

3.1. Path Planning Problem Description. The theoretical complexity of mobile robot path planning is exponentially times of its number of degrees of freedom [26]. Therefore, in the research process of path planning, some specific constraints and assumptions can simplify the solution of the problem and help researchers find out the optimal solution better. In view of this situation, this paper makes the following provisions:

- (1) The mobile robot is regarded as a particle without considering the size of the robot and some factors of kinematics
- (2) The mobile space of the robot is defined as a two-dimensional plane
- (3) All obstacles in the robot's moving space are static obstacles. The shape of the obstacles is not limited, and the size and position of the obstacles are known, and their height is not considered

By simplifying the shape, motion state, and moving space of the robot, the path planning problem is still an NP-hard problem.

3.2. Environment Modeling. Using the grid method to model the environment can simply and effectively represent the unknown running space into a clear and intuitive grid map [27]. Therefore, before designing a path planning algorithm for a mobile robot, this paper first uses the grid method to model the running environment of the robot.

Figure 1(a) is a two-dimensional map of the environment without grid processing, which is placed in a rectangular coordinate system. Then, the whole moving space is divided into a 10×10 grid diagram, as shown in Figure 1(b). Finally, the obstacles in Figure 1(b) are mapped into corresponding obstacle grids according to the principle that if obstacle is less than a grid, it should be covered with the grid and regarded as a complete obstacle grid. So then, a rasterized environment map of Figure 1(c) is generated. It is obvious that the setting of grid size and number has a great impact on the accuracy of the environmental map. The smaller the unit grid is, the more detailed and accurate the environmental map is.

In Figure 1(c), S, 2, 3, ..., 99 and G are the serial numbers of grid nodes, black is the obstacle grid, white is the free grid, S is the starting point of mobile robot operation, and G is the target point. Since the entire grid map is in a rectangular coordinate system, the serial number of each grid node N can correspond to a path point coordinate (x, y) . Assuming that the size of the grid map is $n \times n$, the corresponding relationship between the grid serial number and the grid coordinate is

$$\begin{cases} x = \text{mod}(N, n) + 1, \\ y = \text{int}\left(\frac{N}{n}\right) + 1. \end{cases} \quad (1)$$

3.3. Path Coding. Path coding is the premise of the genetic algorithm to solve the path planning problem [28]. Through a certain coding method, the feasible solution of the problem can be transformed from its solution space to the search space that the genetic algorithm can handle [29–31]. In order to shorten the encoding length as much as possible, the serial number of the grid node is used for encoding. Any chromosome in the genetic algorithm can find a feasible path from the starting point to the target point on the grid map, and each gene on the chromosome corresponds to the grid number on the feasible path. To ensure the possibility of the path and enhance the operation efficiency of the algorithm, this paper stipulates that the coding sequence number of any feasible path cannot contain the repeated sequence number and the corresponding sequence number of the obstacle grid. As shown in Figure 1, a feasible path can be expressed as S-41-70-G.

4. IMGA Algorithm and Analysis

In this section, we describe the improved multiobjective genetic algorithm (IMGA) to plan and select the optimal running path of the mobile robot in the grid static environment.

4.1. Population Initialization. The traditional genetic algorithm usually uses a random method to generate the initial population [32]. Although this method is short in time

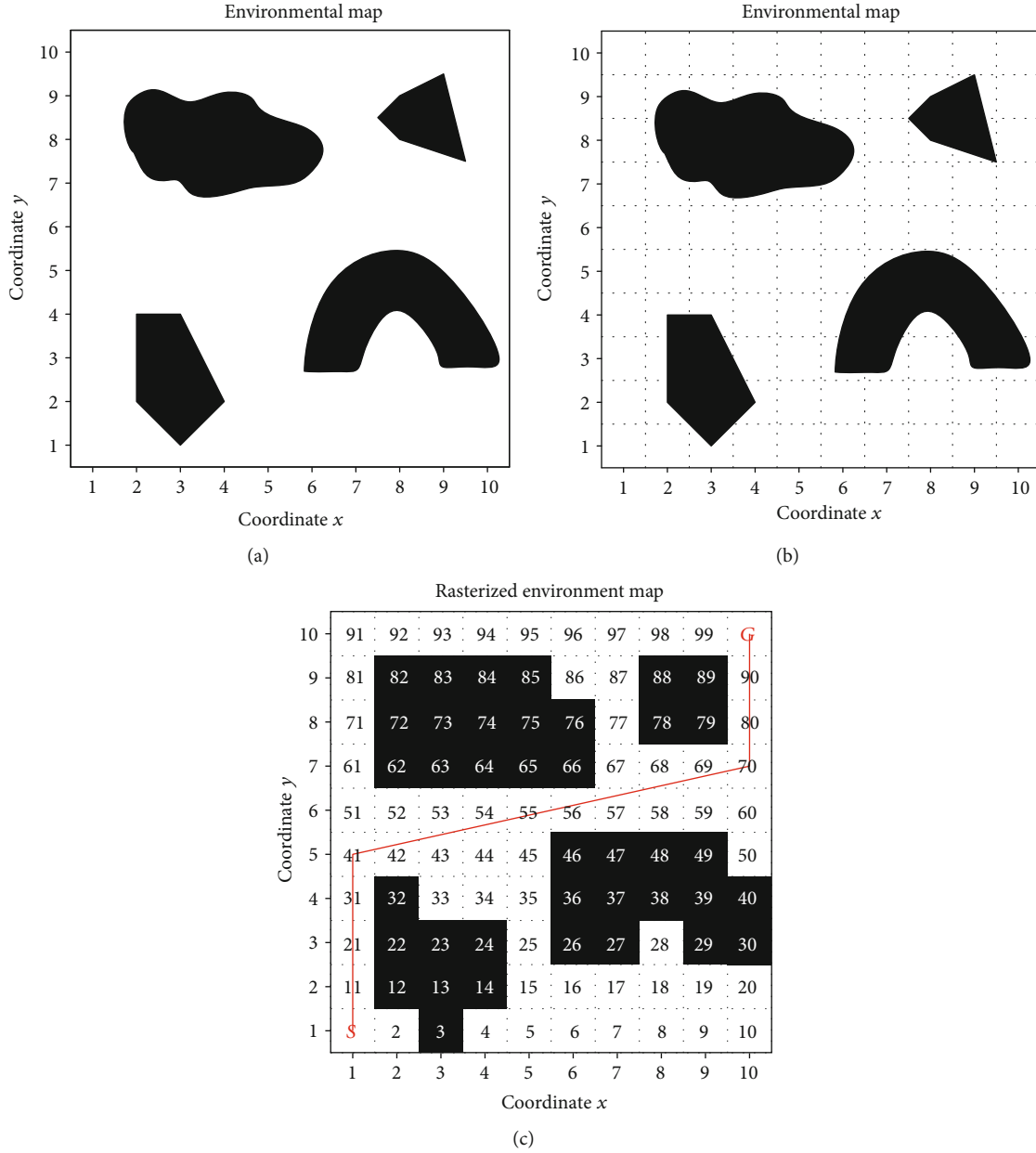


FIGURE 1: Mapping from real environment map to rasterized environment map. (a) Real environment map. (b) Real environment map with grid. (c) Rasterized environment map.

and easy to operate, the proportion of infeasible paths in the generated path is too large, which affects the convergence speed and operating efficiency of the entire algorithm [33]. Aiming at efficiently generating a higher quality initial population and improving the global performance of the algorithm, this paper proposes a heuristic median insertion method to establish the initial population. The specific process of this method is as follows:

- (1) Determine the size of the population M
- (2) Determine the size of the grid map $n * n$ (representing n rows and n columns), the starting point S of the mobile robot path planning (the starting point S

can be expressed as N_1), the target point G (the target point G can be expressed as N_{n*n}), and the number of obstacle grids f (the number of free grids is $n * n - f$).

- (3) Generate a path chromosome from the starting point S to the target point G , where the starting point S is always the first gene of the chromosome and the target point G is always the last gene of the chromosome
 - (i) Randomly generate a grid number N_i (not belonging to the starting point, ending point, or obstacle grid). At this time, the path of the robot can be expressed as $S - N_i - G$ (or $N_1 - N_i - N_{n*n}$)

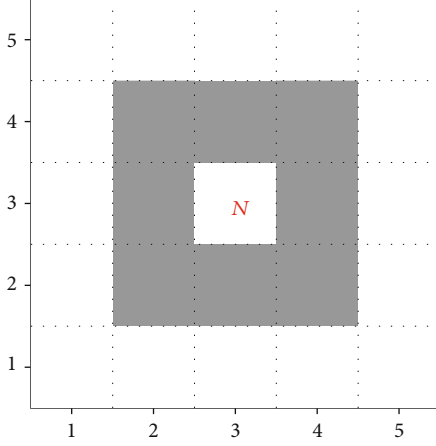


FIGURE 2: Eight neighborhood nodes.

- (ii) Determine whether the adjacent nodes in the path are continuous according to equation (2)

$$\Delta = \max \{ \text{abs}(x_{i+1} - x_i), \text{abs}(y_{i+1} - y_i) \}. \quad (2)$$

In the equation, (x_i, y_i) and (x_{i+1}, y_{i+1}) represent the rectangular coordinates of two adjacent path points N_i and N_{i+1} , respectively. If $\Delta = 1$, then N_i and N_{i+1} are continuous; otherwise, the two points are discontinuous. At this point, the next insertion point is selected through the median method to fill the discontinuous path. The specific calculation is shown in equation (3):

$$\begin{cases} x_i' = \text{int} \left[\frac{(x_i + x_{i+1})}{2} \right], \\ y_i' = \text{int} \left[\frac{(y_i + y_{i+1})}{2} \right], \\ N_i' = x_i' + n \cdot y_i'. \end{cases} \quad (3)$$

In the equation, x_i' and y_i' are the coordinates of the candidate grid, n is the number of rows and columns of the grid, and N_i' is the number of the candidate grid. If the calculated grid N_i' is a free grid, it is directly inserted between N_i and N_{i+1} ; otherwise, the free grid in the eight neighborhood nodes of N_i' is randomly selected as the newly inserted node. As shown in Figure 2, the gray area around the N node is the eight neighborhoods of the point. If there is no free grid in the eight neighborhoods of N_i' , it indicates that this operation is invalid, and the individual is directly discarded. Repeat the above insertion steps to generate a continuous feasible path.

- (iii) Repeat the above operations until an initial population with M nonrepeating chromosomes is generated

4.2. Fitness Function. After the initial population is established, the algorithm needs to establish a fitness function to

calculate the performance of each individual, so as to determine their pros and cons. Here, the fitness function is equivalent to the objective function of the problem [34]. In order to speed up the convergence of the genetic algorithm while ensuring low complexity of the algorithm, and to find the optimal path that can smoothly avoid obstacles and quickly reach the target point [35], a multiobjective fitness function based on path length, path safety, and path energy consumption is designed in this paper, which is specifically expressed as follows:

$$F(N) = \frac{1}{a \cdot L(N) + b \cdot S(N) + c \cdot E(N)}, \quad (4)$$

where $L(N)$ is a path length function, $S(N)$ is a path safety function, and $E(N)$ is a path energy consumption function. a , b , and c are the weights of the three indicators. The path length is the sum of Euclidean distances between all adjacent nodes, and the function definition is shown in equation (5).

$$L(N) = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (5)$$

where n is the number of path nodes, and the path safety function is shown in equation (6).

$$S(N) = \sum_{i=1}^{n-1} S_i, \quad (6)$$

where S_i is the security penalty value of the node i . Although it is stipulated in this paper that all available path nodes cannot be obstacle grids, the threat of obstacles to mobile robots still exists. The distance between the robot and the obstacle directly affects its safety. If the distance is too close, it may cause collision or affect the moving speed of the robot. However, it is not better to be as far away from the obstacle as possible. If the robot is too far away from the obstacle, the path length may be too long. Therefore, in this paper, the eight neighborhood domains of a path node are taken as the standard to measure the safe distance between the point and the surrounding obstacles. If there is no obstacle grid in the eight neighborhood domains of a path node, the point is the safe moving point. Otherwise, the penalty value is increased by 5. The path energy consumption function is shown in equation (7).

$$E(N) = \sum_{i=2}^n E_i(l_{i+1}, l_i), \quad (7)$$

where E_i is the energy consumption penalty value of the node i . The path energy consumption of a mobile robot mainly refers to the electric energy consumed by the robot in the course of driving. When the robot moves in a straight line, the electric energy consumed by the unit distance is the same. The most power-consuming process is the turning movement occurring when the robot changes the direction of movement. Therefore, the path energy consumption is

actually related to the number of turns and the angle of turning, and it is the least energy consumption only if it keeps moving towards the target point. As shown in Figure 3, if the robot turns at point i , then point i is taken as the origin, and the direction of the horizontal and vertical coordinates of point i is taken as the dividing line to divide the quadrant. When the robot turns and its forward direction is in the first quadrant, the energy penalty E_i is plus 1; when the robot's forward direction is in the second or fourth quadrant, the energy penalty E_i is plus 3; when the direction of the robot is in the third quadrant, plus 10. In this way, the number of turns of the robot can be reduced, and the smoothness of the entire planned path can be enhanced; it can also ensure that the robot always advances to the target point during the running process and does not make unnecessary deviation movements.

According to the construction of the fitness function, it can be known that the higher the function value, the better the path.

4.3. Genetic Operation

4.3.1. Selection Operation. In genetic operation, selection operation is the part that best embodies "survival of the fittest." In order to avoid the nondirectional error caused by the traditional roulette selection method [36], this paper adopts the layered method to design selection operator, which is described as follows: after initializing the population, a total of M individuals are generated, the fitness value of each individual is calculated according to the fitness function, and the fitness values are arranged in descending order. Then, the population can be divided into three classes, each part is a layer, namely, a layer has $M/3$ individuals. Copy the aliquot with the higher fitness value in the first layer in duplicate, make a copy with the middle fitness value in the second layer, and do not copy the last fitness with the lower fitness, in order to form the offspring population. And the number of population is still M (if M is not divisible by 3, the number of individuals in the last share is $M/3 + \text{mod}(M/3)$). After replicating the first two layers, $\text{mod}(M/3)$ individuals with high fitness value should be selected from the third layer to join the offspring population in order to ensure that the number of the offspring population is still M). This ensures that the best individuals can be passed on to the next generation while maintaining the diversity of the population. The selection method can be seen in Figure 4.

4.3.2. Crossover Operation. Crossover operation is the most important genetic operation in the whole algorithm, and its essence is genetic recombination of chromosomes. That is, by exchanging some of the genes of the parent chromosome, new progeny chromosomes are created. This paper adopts the method of single-point crossing. Randomly select one of the two parent individuals with the same sequence number (except the starting point S and the target point G) and then cross at this point. For example: parent individual M_1 : S-2-12-67-69-89-G; M_2 : S-11-22-67-78-80-G. If the same sequence number 67 is selected as the intersection point, the two offsprings generated are M_1' : S-2-12-67-78-80-G;

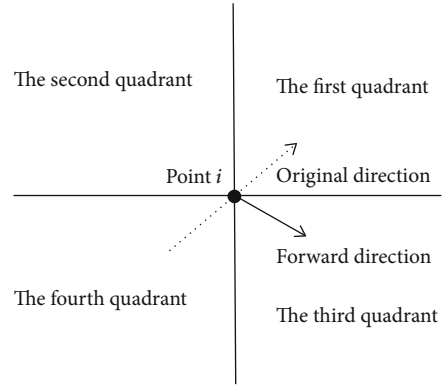


FIGURE 3: The quadrant diagram corresponding to the turning direction of the robot.

M_2' : S-11-22-67-69-89-G. If the same sequence number does not exist in two parent individuals, no crossover operation is performed.

4.3.3. Mutation Operation. The mutation operation is to mutate any gene on an individual's chromosome to generate a new chromosome to maintain the diversity of the population. However, traditional single-point mutation, multipoint mutation, and other processing methods are prone to produce infeasible paths, which will affect the operation efficiency of the algorithm. Therefore, this paper uses the method of eight-point neighborhood single-point mutation. First, randomly select a variation point N_i (except the starting point S and target point G) in the individual of the path to be mutated and then randomly determine a nonobstacle grid N_i' (except the adjacent nodes N_{i-1} and N_{i+1} in the path to be mutated) in the eight neighborhoods of the mutation point (as shown in Figure 2) to replace the original node. Then, according to the initial path generation method, N_{i-1} to N_i' and N_i' to N_{i+1} are connected into an uninterrupted path. If there is no free grid to choose from in the eight neighborhoods of the mutation points or a feasible uninterrupted path cannot be generated, this mutation fails. That is, jump out of this mutation operation and reselect the next individual to be mutated and its mutation point.

4.4. Termination Conditions. The termination condition is a standard to measure whether the genetic algorithm can terminate the operation. The termination condition given in this paper is as follows: the optimal fitness value for a given evolutionary algebraic threshold of 50 or 40 consecutive evolutionary populations remains unchanged or the algorithm runs longer than 5 minutes.

4.5. Delete Operation. For the case of redundant nodes in the path, a delete operation is added in this article. The main idea is if a node can be barrier-free connected to its front and rear nodes (nonadjacent path points), then the intermediate nodes between these two points are redundant. Delete these redundant nodes and connect these two points directly, so as to achieve the goal of avoiding unnecessary turning and reducing the path length. To avoid the reduction of path

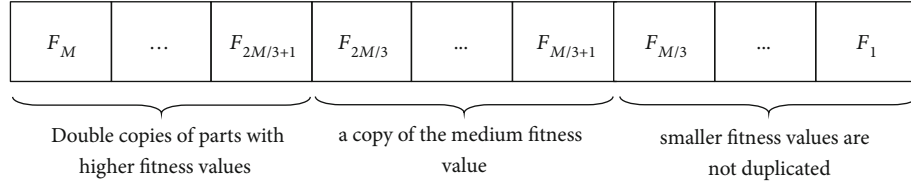


FIGURE 4: Fitness segmentation selection principle diagram.

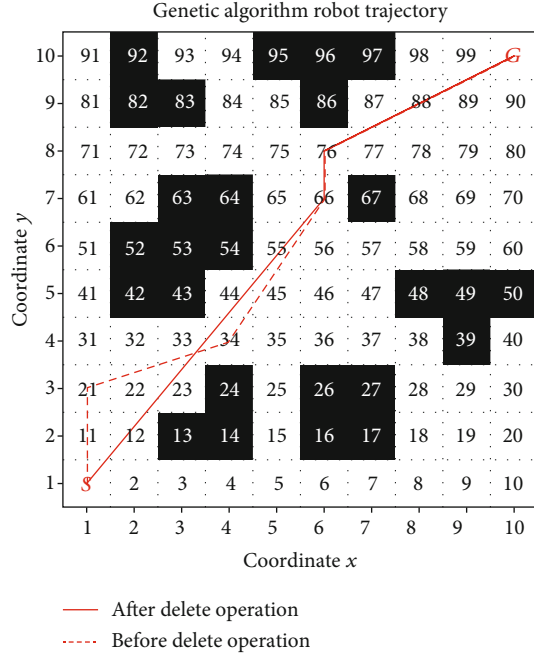


FIGURE 5: Path comparison before and after deletion.

Input: Grid map, Map; Start point, S; End point, G
Output: Shortest path point set, min_path

- 1: $t \leftarrow 0$;
- 2: Evaluate fitness_value of $P(t)$;
- 3: Best_fitness $\leftarrow \max(\text{fitness_value})$;
- 4: **while** not meet termination conditions **do**
- 5: Select operation to $P(t)$;
- 6: Crossover operation to $P(t)$;
- 7: Mutation operation to $P(t)$;
- 8: $P(t+1) = P(t)$;
- 9: $t = t + 1$;
- 10: Evaluate fitness_value of $P(t)$;
- 11: **if** Best_fitness < $\max(\text{fitness_value})$ **then**
- 12: Best_fitness $\leftarrow \max(\text{fitness_value})$;
- 13: **end if**
- 14: **end while**
- 15: min_path is $P(t)$ with Best_fitness;
- 16: Smooth optimization operation to min_path;
- 17: **return** min_path;

ALGORITHM 1: IMGGA Algorithm.

nodes after the delete operation, which affects the feasibility of crossover and mutation operations in genetic operations, this process is only for the optimal paths obtained by each

run. As shown in Figure 5, the original optimal path is S-21-34-66-76-G, and the optimal path after delete operation is S-66-76-G. It is obvious that there are four turns in the original path. After the deletion operation, the redundant nodes 21 and 34 in the section from S point to sequence point 66 are deleted, and only two turns occur in the whole path, and the path length becomes shorter.

4.6. Algorithm Pseudocode. The pseudocode of the IMGGA algorithm is shown below. We can see that the time complexity of the whole algorithm is $O(O_F * (O_{ps} + O_{pc} + O_{pm}))$, where O_F represents the complexity of fitness function, O_{ps} represents the time complexity of selection operation, O_{pc} represents the complexity of crossover operation, and O_{pm} represents the time complexity of mutation operation. In other words, the time complexity of IMGGA is determined by both genetic operator and fitness function. Compared with the traditional roulette method, the hierarchical selection method proposed in this paper has less time complexity. Although the fitness function considers multiple indicators, the measurement method of each index is relatively simple. In conclusion, the time complexity of IMGGA is not higher than that of traditional GA.

5. Simulation Experiment and Evaluation

In this section, to verify the rationality and optimization degree of the designed improved multiobjective genetic algorithm, MATLAB R2018b software was used to establish a grid model and carry out simulation experiments on the algorithm.

5.1. Experimental Result. The simulation experiment is first carried out in the grid environment model of 10×10 , 15×15 , and 20×20 to verify the universality of the algorithm in different maps. The parameters set in the experiment are as follows: population number: NP = 100; maximum evolutionary algebra: max_gen = 50; crossover probability: PC = 0.8; mutation probability: PM = 0.2; path length proportion: $a = 6$; path safety proportion: $b = 1$; and path energy proportion: $c = 3$. The experimental results are shown in Figures 6(a)–7(a). In Figure 6(a), there are 40 obstacle grids, Figure 6(b) has 90 obstacle grids, and Figure 7(a) has 160 obstacle grids. The dotted line represents the running path before delete operation, and the solid line represents the running path after delete operation. According to the statistical results in Table 1, the delete operation designed by this algorithm can effectively reduce the path length. Among them, the number of turns of the running paths with delete operation in environment 2 and environment 3 is also reduced.

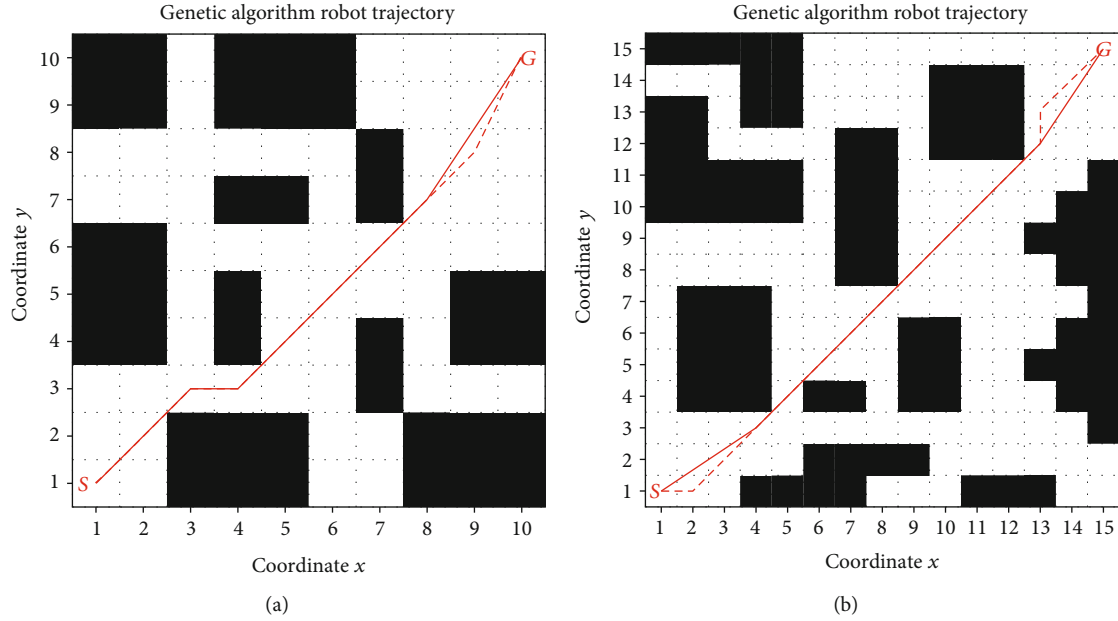


FIGURE 6: Optimal running path. (a) Environment 1. (b) Environment 2.

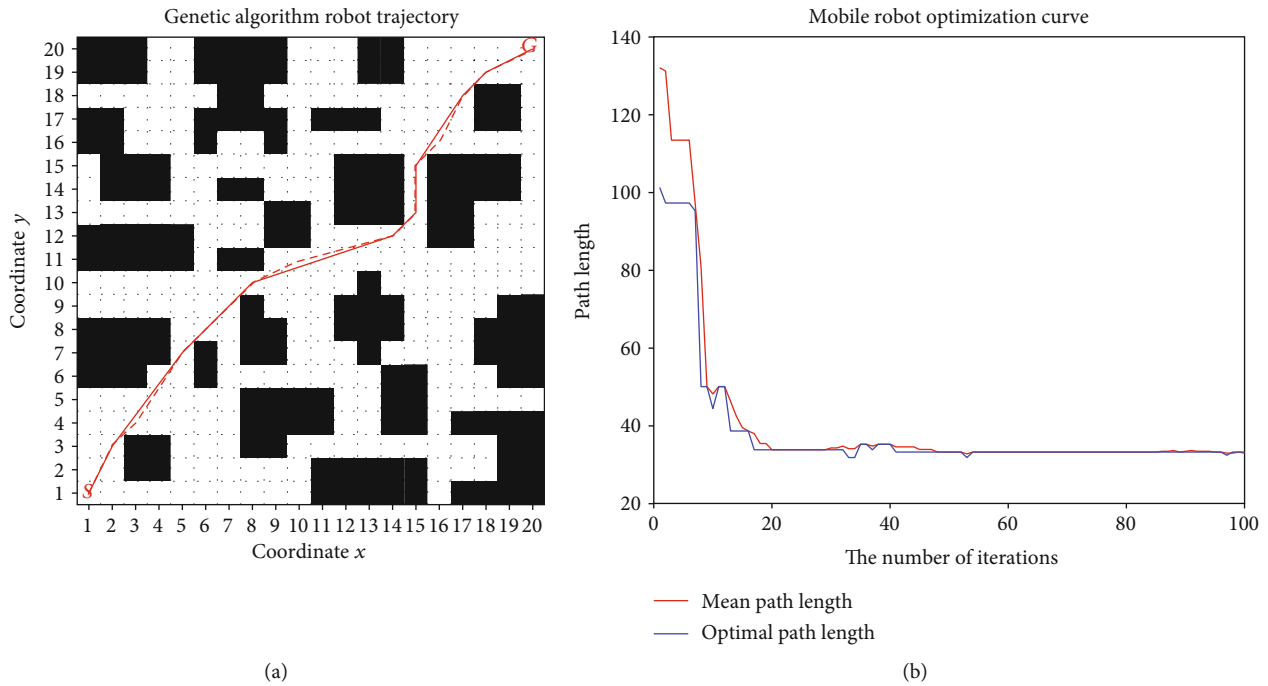


FIGURE 7: Environment 3. (a) Optimal running path. (b) Optimization curve.

Although the number of turns of environment 1 has not changed, it can be clearly seen from Figure 6(a) that the angle of the third turning of the running path with deletion operation is significantly smaller. Therefore, this operation can effectively optimize the path and further ensure the efficient operation of the robot. It is an indispensable link in the operation of the entire algorithm.

Taking environment 3 as an example, the influence of the maximum evolution algebra on the algorithm running

results is analyzed. Ensure that other parameters remain unchanged, set different maximum evolutionary generation values, and run 10 times, respectively. The experimental results are shown in Table 2, where “-” indicates that the algorithm did not obtain the optimal solution after running 10 times. It can be seen that the larger the maximum evolution algebra, the better the search ability and search stability of the algorithm, but the timeliness is poor, and the algorithm runs longer. In order to search the optimal running path of

TABLE 1: Comparison of run path data with or without delete operation.

Operating environment	Path length		Number of turns	
	No deletion operation	With deletion operation	No deletion operation	With deletion operation
Environment 1	13.1356	13.0908	3	3
Environment 2	20.3848	19.9390	3	2
Environment 3	28.5724	28.4733	11	8

TABLE 2: The influence of maximum evolution algebra on the running result of the algorithm.

Maximum evolution algebra	The number of runs to get the optimal solution	The total number of occurrences of the optimal solution in the results of 10 runs	Optimal running time (s)
20	—	0	0.59834
50	5	2	0.71417
100	3	4	0.80262
200	2	5	0.98274

the mobile robot as quickly as possible, this paper selects 50 as the maximum evolution algebra value.

Figure 7(b) shows the evolution curve of the algorithm running in environment 3 of Figure 7(a). It can be seen that in this environment, the algorithm designed in this paper tends to be stable when it evolves to the 10th generation, and the algorithm has a fast convergence speed. Moreover, under the regulation of the genetic operation, the path length did not fall into the local optimal immediately but fluctuated slightly and found the global optimal path successfully. It can be proved that the initial population quality of the algorithm designed in this paper is very high, and the fitness function is set reasonably, which is helpful for the algorithm to converge quickly. In the late stage of evolution, the diversity of the population is guaranteed while the stability of the population is maintained. It avoids the problem that the traditional genetic algorithm easily falls into the local optimal.

5.2. Algorithm Comparative Analysis. In order to show the superiority of the proposed algorithm (IMGA) in solving the path planning problem of the mobile robot, IMGA is compared with the improved genetic algorithms in literature [37] and literature [38].

Under the premise of the same control parameters, multiple simulation experiments were carried out in the environment mentioned in literature [37]. The optimal running path of the algorithm in this paper is shown in Figure 8(c). Figure 8(a) is the optimal operation of the basic genetic algorithm (GA), and Figure 8(b) is the optimal operation of the improved genetic algorithm proposed in literature [37]. The optimal path length, turn times, and tangency with obstacles

of the three methods are recorded in Table 3. Through the analysis of the operation results, it can be seen that the path obtained by the GA has a large number of turns and redundant path segments, and there are two places pass through the tiny gap between the obstacles, as shown in the blue circle of Figure 8(a). Although the improved algorithm proposed in literature [37] eliminates some redundant path segments and unnecessary turns, there is still a place that passes through the minimal gap between two obstacles, which is very unfavorable for the actual operation of mobile robots. Considering the path length, path security, and path energy consumption, IMGA can avoid the situation of blue circles in Figures 8(a) and 8(b). The path length is reduced by 17% compared with GA, and there are only three places that are tangent to obstacles. Therefore, under the condition of ensuring the optimal path length, the algorithm IMGA in this paper can effectively improve the operation safety of the mobile robot and reduce the number of turning movements of the robot. The improved effect is better than other improved algorithms.

Then, IMCA and Bezier curve smoothing algorithm (BCA) proposed by literature [38] are compared in a more complex environment under the same parameters. The optimal running path is shown in Figure 9. To eliminate the influence of random factors on the algorithm, the above two algorithms were independently executed 30 times, and the statistical results are recorded in Table 4. BCA first obtains the segmented Bezier curves control points (blue square dots in Figure 9) through genetic operations and then plans the optimal smooth path according to the optimization criteria, which is actually a combination of Bezier curve and GA. According to Table 4, compared with BCA, IMGA can obtain a running path with a smaller path length, and the algorithm running time is only five thousandths of BCA. However, as can be seen from Figure 9, the optimal path planned by IMGA is a broken line. Compared with the BCA algorithm, the broken line is smoother and the path length is shorter. However, there are three (one in the middle is not tangent) times tangent to the barrier grid, and the optimal operation path obtained by BCA planning is a smooth curve that is not tangent to any obstacle. Therefore, IMGA still has space for improvement in path safety and smoothness, but it has great advantages in environments with strong real-time requirements.

6. Conclusion

In this paper, an improved multiobjective genetic algorithm is designed to study the path planning problem of the mobile robot. The algorithm uses the heuristic median insertion method during the creation of the initial population, which effectively improves the quality of the initial population and speeds up the convergence rate of the algorithm, and the path length, path safety, and path energy consumption are used as evaluation indicators to establish a multiobjective fitness function, which minimizes the path length while ensuring that the robot always moves toward the target point, avoids unnecessary path energy consumption, and improves the safety of planned paths to a certain extent. Finally, the

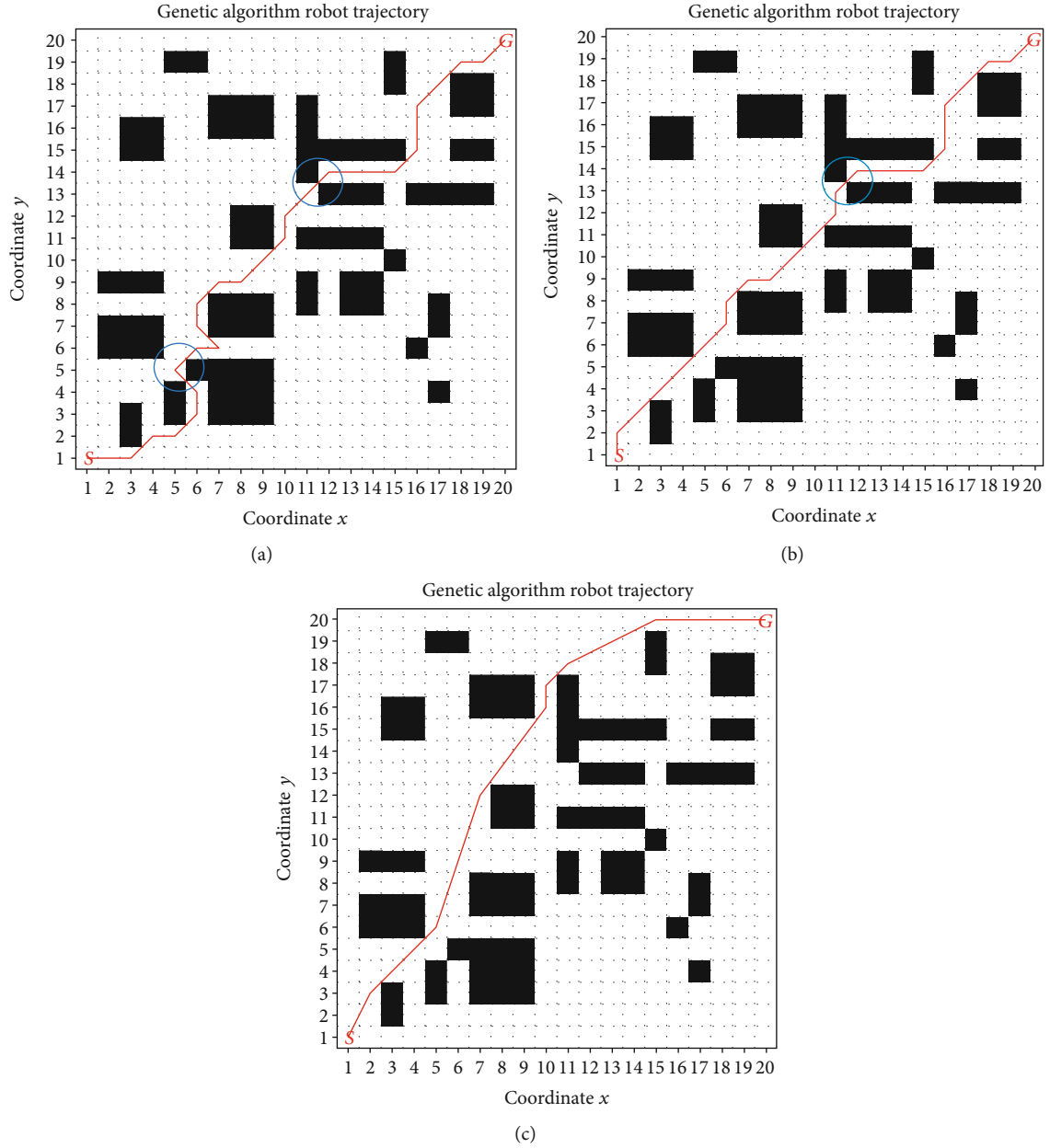


FIGURE 8: Optimal running path. (a) Optimal running path of GA. (b) Optimal running path of literature [37]. (c) Optimal running path of IMGA.

TABLE 3: Path planning data statistics in environment 4.

Path planning algorithm	Optimal path length	Number of turns	Number of tangency
GA	35.7990	20	10
Literature [37]	29.7990	13	8
IMGA	29.6896	7	3

improved genetic operation is used to further prevent the algorithm from falling into local optimum prematurely. Simulation experiments show that the algorithm can run in environments of different scales and complexity. Compared with other improved algorithms, this method has faster conver-

gence speed and shorter path length. Therefore, this algorithm has certain feasibility and superiority in the field of path planning problems for the mobile robot. However, there is still space for improvement in the security and smoothness of the optimal path planned. This is because the algorithm takes the path length as the main optimization objective and pays more attention to the operation efficiency in the planning process and adopts a relatively simple method to measure the three indicators of fitness function.

There are still many meaningful topics to be discussed in this algorithm, and the future research directions are as follows: (1) better design of fitness function to improve the overall quality of the planning path, especially the path safety on the premise of ensuring operation efficiency; (2)

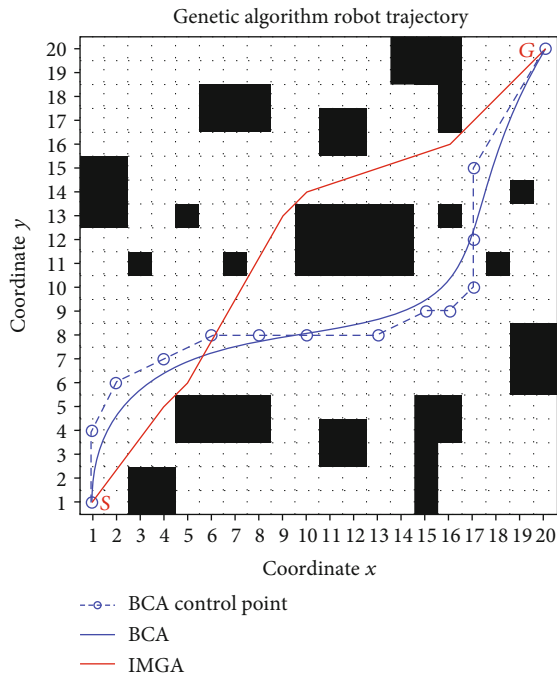


FIGURE 9: Comparison chart of algorithm optimal path.

TABLE 4: Path planning data statistics in environment 5.

Path planning algorithm	Optimal path length	Optimal running time(s)
IMGA	27.8721	0.7016
BCA	29.9416	136.4794

combining the algorithm with Bezier curve or other curve smoothing methods, the polygonal path is planned to be a smoother curve path; (3) increase complexity design, for example, consider the specific model and kinematics factors of mobile robot, design more complex operation environment, and add dynamic obstacles; and (4) apply path planning algorithm to solving scenarios of practical problems, such as express shunting robot, family service robot, and automatic driving vehicle.

Data Availability

A large number of data are included in the paper, and other relevant information can be obtained from the corresponding authors through e-mail.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Science and Technology Project of Emergency Management in Jiangsu Province

(Grant No. YJGLYF202017), in part by the Research Projects in Education Informatization in Jiangsu Province (Grant No. 20180012), in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYCX18_2366), in part by the Yangzhou Science and Technology (Grant Nos. YZ2018209 and YZ2019133), in part by the Yangzhou University Jiangdu High-End Equipment Engineering Technology Research Institute Open Project (Grant No. YDJD201707), and in part by the Open Project in the State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University (Grant No. #1907).




References

- [1] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control," *Robotics and Autonomous Systems*, vol. 89, pp. 95–109, 2017.
- [2] Y. Xue, "Mobile robot path planning with a non-dominated sorting genetic algorithm," *Applied Sciences*, vol. 8, no. 11, p. 2253, 2018.
- [3] H. Zhang, W. Lin, and A. Chen, "Path planning for the mobile robot: a review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.
- [4] J. Chen, S. Xie, H. Li, J. Luo, and K. Feng, "Robot path planning based on adaptive integrating of genetic and ant colony algorithm," *International Journal of Innovative Computing Information and Control*, vol. 11, no. 3, pp. 833–850, 2015.
- [5] H. Lu, Y. Li, S. Mu, D. Wang, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 99, 2017.
- [6] C. Ge, Z. Liu, J. Xia, and F. Liming, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2019.
- [7] X. Zhan, "Research on path planning method of humanoid robot based on improved genetic algorithm," *Journal of Physics: Conference Series*, vol. 1237, no. 2, pp. 22–28, 2019.
- [8] N. S. Utami, A. Jazidie, and R. E. A. Kadier, "Path planning for differential drive mobile robot to avoid static obstacles collision using modified crossover genetic algorithm," in *2019 International Seminar on Intelligent Technology and Its Application (ISITIA)*, pp. 282–287, Surabaya, Indonesia, 2019.
- [9] Q. Xiao, J. Li, and C. Xiao, "Research on performance optimization of several frequently-used genetic algorithm selection operators," in *International Conference on Applications and Techniques in Cyber Security and Intelligence*, vol. 580 of *Advances in Intelligent Systems and Computing*, pp. 90–97, Ningbo, China, 2018.
- [10] H. Lu, D. Wang, Y. Li et al., "CONet: a cognitive ocean network," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 90–96, 2019.
- [11] Z. Liu, Y. Xue, C. Zhou, H. Bai, and X. Cui, "Population initialization improvement of robot path planning based on genetic algorithm," *Machine Tool & Hydraulics*, vol. 47, no. 21, pp. 5–8, 2019.
- [12] N. Wang, H. Guo, B. Chen, C. Cui, and X. Zhang, "Design of a rotary dielectric elastomer actuator using a topology optimization method based on pairs of curves," *Smart Materials and Structures*, vol. 27, no. 5, article 055011, 2018.

- [13] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 107–119, 2019.
- [14] M. Albayrak and N. Allahverdi, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1313–1320, 2011.
- [15] L. Fang, W. Susilo, Y. Ren, C. Ge, and J. Wang, "Chosen public key and ciphertext secure proxy re-encryption schemes," *International Journal of Digital Content Technology and Its Applications*, vol. 4, no. 9, pp. 151–160, 2010.
- [16] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, pp. 1–1, 2020.
- [17] X. Zhang, Y. Zhao, N. Deng, and K. Guo, "Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 91, 2016.
- [18] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.
- [19] L. Chen and J. R. Chen, "Mobile robot path planning based on improved genetic algorithm," *Software Guide*, vol. 18, no. 4, pp. 24–27, 2019.
- [20] K. P. Cheng, R. E. Mohan, N. H. Khanh Nhan, and A. V. le, "Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot," *IEEE Access*, vol. 8, no. 1, pp. 121267–121284, 2020.
- [21] J. Duan and Q. Chen, "Research on path planning algorithm based on improved artificial potential field-genetic algorithm," *Foreign Electronic Measurement Technology*, vol. 38, no. 3, pp. 19–24, 2019.
- [22] X. Yi, W. S. Guo, and L. Zhao, "Robot motion based on adaptive selection operator combined with genetic algorithm," *Application Research of Computers*, vol. 37, no. 6, pp. 1–6, 2020.
- [23] B. Sun, P. Jiang, G. Zhou, and Y. T. Lu, "Application of improved genetic algorithm in path planning of mobile robots," *Computer Engineering and Applications*, vol. 55, no. 17, pp. 162–168, 2019.
- [24] K. Liang, Z. Chen, and X. Yan, "Simulation study on effective path planning for mobile robot," *Modern Electronic Technology*, vol. 41, no. 17, pp. 167–172, 2018.
- [25] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.
- [26] A. Tharwat, M. Elhoseny, A. E. Hassanien, T. Gabel, and A. Kumar, "Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm," *Cluster Computing*, vol. 22, no. S2, pp. 4745–4766, 2019.
- [27] J. C. Mohanta, D. R. Parhi, and S. K. Patel, "Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 1058–1070, 2011.
- [28] S. Ghosh, P. K. Panigrahi, and D. R. Parhi, "Analysis of FPA and BA meta-heuristic controllers for optimal path planning of mobile robot in cluttered environment," *IET Science, Measurement & Technology*, vol. 11, no. 7, pp. 817–828, 2017.
- [29] L. Wang, M. Li, J. Cai, and Z. Liu, "Research on mobile robot path planning by using improved genetic algorithm," *Mechanical Science and Technology*, vol. 36, no. 5, pp. 711–716, 2017.
- [30] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 41–50, 2014.
- [31] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: go beyond artificial intelligence," *Mobile Networks and Applications*, vol. 23, pp. 368–375, 2018.
- [32] F. Han, J. Jiang, Q. Ling, and B. Su, "A survey on metaheuristic optimization for random single-hidden layer feedforward neural network," *Neurocomputing*, vol. 335, pp. 261–273, 2019.
- [33] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes and Cryptography*, vol. 86, no. 11, pp. 2587–2603, 2018.
- [34] Y. Li, Z. Huang, and Y. Xie, "Path planning of mobile robot based on improved genetic algorithm," in *3rd International Conference on Electron Device and Mechanical Engineering (ICEDME)*, pp. 691–695, Suzhou, China, 2020.
- [35] J. Ma, Y. Liu, S. Zang, and L. Wang, "Robot path planning based on genetic algorithm fused with continuous Bezier optimization," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 981304, 10 pages, 2020.
- [36] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications*, vol. 115, pp. 106–120, 2019.
- [37] S. Huang, S. Jiang, and X. Zhu, "Multi-object tracking via discriminative appearance modeling," *Computer Vision and Image Understanding*, vol. 153, pp. 77–87, 2016.
- [38] Q. Ling, Y. Song, F. Han, D. Yang, and D. Huang, "An improved ensemble of random vector functional link networks based on particle swarm optimization with double optimization strategy," *PLoS One*, vol. 11, no. 11, article e0165803, 2016.

Research Article

Hierarchical Annotation Event Extraction Method in Multiple Scenarios

Shi Wang ¹, Zhujun Wang ^{1,2}, Yi Jiang ^{2,3} and Huayu Wang⁴

¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

²School of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225127, China

³State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

⁴Information Engineering College, Graduate School of Army Engineering University, Nanjing 210007, China

Correspondence should be addressed to Zhujun Wang; mx120180424@yzu.edu.cn

Received 28 September 2020; Revised 4 December 2020; Accepted 27 February 2021; Published 19 March 2021

Academic Editor: Paul Honeine

Copyright © 2021 Shi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the event extraction task, considering that there may be multiple scenarios in the corpus and an argument may play different roles under different triggers, the traditional tagging scheme can only tag each word once, which cannot solve the problem of argument overlap. A hierarchical tagging pipeline model for Chinese corpus based on the pretrained model Bert was proposed, which can obtain the relevant arguments of each event in a hierarchical way. The pipeline structure is selected in the model, and the event extraction task is divided into event trigger classification and argument recognition. Firstly, the pretrained model Bert is used to generate the feature vector and transfer it to bidirectional gated recurrent unit+conditional random field (BiGRU+CRF) model for trigger classification; then, the marked event type features are spliced into the corpus as known features and then passed into BiGRU+CRF for argument recognition. We evaluated our method on DUEE, combined with data enhancement and mask operation. Experimental results show that our method is improved compared with other baselines, which prove the effectiveness of the model in Chinese corpus.

1. Introduction

As a form of information, event is defined as the fact that some people or things interact at a certain time and place. As an important task in information extraction, event extraction is aimed at identifying triggers and arguments from unstructured text and expressing them in a structured way, which is the basic work of constructing knowledge graph. Natural language text contains many events and related arguments, as shown in Figure 1. Two events appear in ex1, and it describes two volleyball matches. “Eliminated” and “defeated” are the triggers of the event type of “competition behavior—win or lose.” “The National Women’s Volleyball Championship” plays the role of “competition name” in both events, “Tianjin Women’s volleyball team” plays the role of “loser” in the first event, and “Jiangsu women’s volleyball team” and “Shandong women’s volleyball team” played the role of “loser” and “winner,” respectively, in the second event.

The traditional event extraction model cannot reasonably distinguish the arguments under multiple scenario events. If there is more than one event in the sentence, the model cannot be extracted accurately.

There are two ways to implement event extraction task; one is based on pattern matching, and the other is based on machine learning. The method based on pattern matching mainly uses lexical symbol features, semantic features, and self-organizing constraints to extract events, and the key lies in the construction of event templates. However, the method based on pattern matching has poor cross domain and needs a lot of manual operation. With the continuous enrichment of corpus in recent years, researchers use machine learning method to extract events, especially deep learning and neural network. The traditional machine learning method uses the classified thought to transform the task of trigger classification and argument recognition into classification problem. The key point is the construction of classifier and the selection of fea-

Ex1 : The National Women's Volleyball Championship held in Jiangmen, Guangdong Province, has entered the final climax stage. All the top four teams have been produced this afternoon. After Tianjin Women's volleyball team was eliminated, there was a big surprise today. Jiangsu women's volleyball team was defeated 0-3 by Shandong women's volleyball team, and was not qualified for the top four.

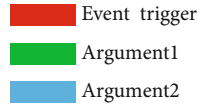


FIGURE 1: Trigger classification and argument recognition in event extraction.

tures. Chieu [1] introduced the maximum entropy model into event extraction task for the first time and realized the extraction of seminar notice and personnel management event. Llorens et al. [2] used conditional random field (CRF) to tag semantic roles, which improved the performance of the system. Support vector machine (SVM) and hidden Markov model (HMM) are also commonly used classification models.

Aiming at the problems of existing models, this paper proposes a pipeline model based on the pretrained model Bert. The main contributions of this paper are as follows:

- (i) An event extraction model based on the pretrained model Bert was designed. First, the event triggers in the corpus are labeled by pipeline, and then, the relevant arguments of each trigger are extracted
- (ii) Pipeline model extracts event triggers by hierarchical way and improves recall and accuracy of event recognition by data enhancement
- (iii) The model extracts the arguments from each identified event, respectively, which successfully solves the problem of argument overlapping. Through the use of window, the performance of the model is improved successfully, and the argument recognition errors in multiple scenarios are reduced

The rest of this paper is organized as follows. Section 2 reviews the related work on path event extraction, introduces a hierarchical annotation model for event trigger extraction and related argument recognition, and introduces each module of the model. Section 4 analyzes the experimental results of the current model through comparative experiments to verify the reliability of the model we proposed. Section 5 summarizes the paper and plans the direction of future work.

2. Related Work

Compared with traditional way of event extraction, more and more event extraction models based on neural network appear. Zheng et al. [3] proposed a tagging scheme, which transforms the task of information extraction into a tagging problem. However, one word can only be tagged once, which makes it difficult to extract from multievent sentences. According to the overlap of triples, Zeng et al. [4] divide sentences into normal, entity pair overlap, and single entity overlap and proposes an end-to-end model based on replication mechanism to jointly extract information, which solves the problem of entity overlap. However, this model can only

be used for a single word. If an entity has more than one word, it cannot extract the relationship accurately.

Traditional machine learning methods for event extraction need to design a large number of features manually and also need the support of external Natural Language Processing (NLP) tools. Based on neural network method, event extraction is modeled as an end-to-end model, which gets rid of the dependence on external NLP tools and uses word vectors with rich features as input, thus avoiding complex manual work. Nguyen and Grishman [5] studied the problem of event triggered word extraction in unbalanced corpus and used convolutional neural network to capture important feature information in sentences. Chen et al. [6] proposed a dynamic multipool convolutional neural network (DMCNN) to extract sentence level features. In Ghaeini et al.'s [7] paper, bidirectional recurrent neural network (Bi-RNN) is first used to detect events that can be words or phrases, which is the first attempt to extract multi-token events. Feng et al. [8] combined bidirectional long short-term memory (LSTM) and convolutional neural network to learn word representation and predict event triggers.

Although the neural network-based event extraction method has achieved good performance, due to the fact that there may be multiple scenario events in a sentence, the confusion of argument annotation and role overlap are also important problems in the current event extraction, which makes event extraction still a difficult NLP problem. Therefore, this paper focused on the research of hierarchical annotation event extraction method in multiple scenarios, which improves the recognition effect of event triggers and alleviates the confusion of argument.

3. Methodology

3.1. Model Flow Chart. In the traditional event extraction research, scenario switching under multiple scenario events leads to argument extraction confusion, and the traditional annotation scheme cannot solve the problem of argument overlap. In this paper, a pipeline model of event extraction based on pretrained model Bert is proposed to solve the chaotic problem of event argument extraction in multiple scenarios. The flow chart is shown in Figure 2, which is divided into three parts: pretrained model Bert, event trigger classification, and argument recognition.

3.2. Model Architecture. The architecture of the model in this paper is shown in Figure 3. The model represents the event extraction task as a pipeline model based on hierarchical tagging, which solves the problem that a word can only be

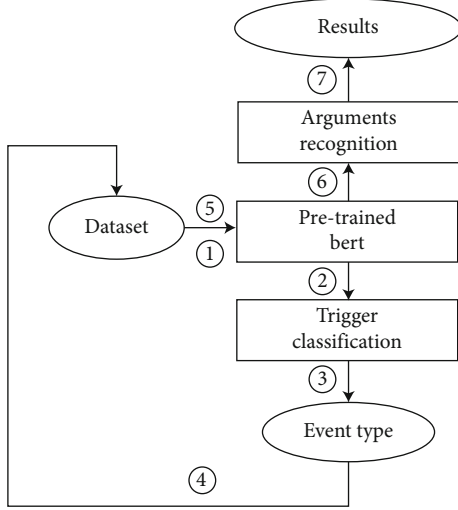


FIGURE 2: Flow chart of the model.

tagged once, resulting in argument overlap. The first stage of the model is trigger classification using a model based on pre-trained model Bert. If the trigger is identified in the sentence, the second stage is carried out. The extracted event types are placed in the front of the sentence as features and input into the argument extraction model to extract relevant arguments and identify roles.

3.3. Pretrained Model Bert. The word2vec [9] model considers that the meaning of a word is associated with the meaning of the word that appears around it, thus mapping each word into a vector. However, in natural language, a word may have different meanings, but the traditional word2vec model does not generate static vectors and ignores the context, which cannot solve the problem of polysemy. The pretrained model Bert [10] makes full use of context and solves the problem of polysemy.

The Bert model is shown in Figure 4(a). The model selects the editing module of transformer model [11] as the model of feature extraction for bidirectional coding. The module structure is shown in Figure 4(b). By adding attention mechanism to replace the traditional convolutional neural network and cyclic neural network, the coding features of each word can obtain the information of all words.

The most important content of transformer encoding module is self-attention mechanism, which inputs the encoded vector and calculates the relationship between the current token and the context and obtains the weighted sum as the output of the current word. This makes the output vector contain not only the meaning of the word itself but also the relationship with other words. The weight calculation formula is as follows:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1)$$

where Q , K , and V represent query vector, key vector, and value vector, respectively, and d_k is the dimension of the

input vector. After the inner product sum of current token's Q and each token's K , the weight is obtained by softmax, and then, V is weighted and summed by using the weight obtained in the previous step to get the output coding vector of the current token. However, the use of self-attention mechanism can only obtain one feature expression; transformer module uses multihead attention mechanism to map Q , K , and V to QW_i^Q , KW_i^K , and VW_i^V with different n projections, and the specific formula is as follows:

$$\begin{aligned} \text{attention}(Q, K, V) &= \text{concat}(\text{head1}, \text{head2}, \dots, \text{head}n), \\ \text{head}i &= \text{attention}\left(QW_i^Q, KW_i^K, VW_i^V\right). \end{aligned} \quad (2)$$

In the self-attention mechanism, the position of each word and the position between the included segments have an impact on the representation of the current word. Therefore, the position vector and segment vector are added to Bert. In addition, a normalization and residual link are added after each self-attention module and feedforward neural network module, which solves the problem of gradient dispersion and improves the training efficiency of the model.

3.4. GRU Layer. In recurrent neural networks, hidden state is always transmitted from front to back. However, in the event extraction, the hidden state of the current time step is associated with the previous time and the next moment, so the bidirectional recurrent neural network is needed to establish the correlation. The bidirectional GRU [12] model selected in this paper solves the problems of long-term memory and gradient in backpropagation, as shown in Figure 5. Compared with LSTM [13], GRU can achieve similar results, and it is easier to train and improve training efficiency.

The input and output structure of GRU is consistent with the traditional RNN [14], including the current input x_t and the hidden state h_{t-1} passed in the previous time step $t-1$. But different from RNN, GRU uses gating mechanism to control the hidden state of the previous moment, instead of receiving all the features of h_{t-1} like RNN. The two gates built into GRU are reset gate r and update gate z , and the formulas are as follows: $r = \sigma(W^r[h_{t-1}, x_t])$ and $z = \sigma(W^z[h_{t-1}, x_t])$. After getting the gating signal, the reset gate r is used to reset the hidden state of the previous time, i.e., $h'_{t-1} = h_{t-1} \odot r$. Then, the data range is reduced to $(-1, 1)$ by activating function \tanh , and the formula is expressed as $h'_t = \tanh(W h'_{t-1})$. Finally, according to the calculated update gate, two operations, forgetting and memorizing, are carried out at the same time. The specific formula is

$$h_t = (1 - z) \odot h_{t-1} + z \odot h'_t. \quad (3)$$

3.5. Conditional Random Field. CRF [14] is a sequence tagging algorithm, which outputs the target sequence after inputting a segment of sequence. In NLP annotation task, the input sequence is a piece of text, and the output sequence is the corresponding tag. Considering the correlation between adjacent tags, CRF obtains a global optimal tagging chain.

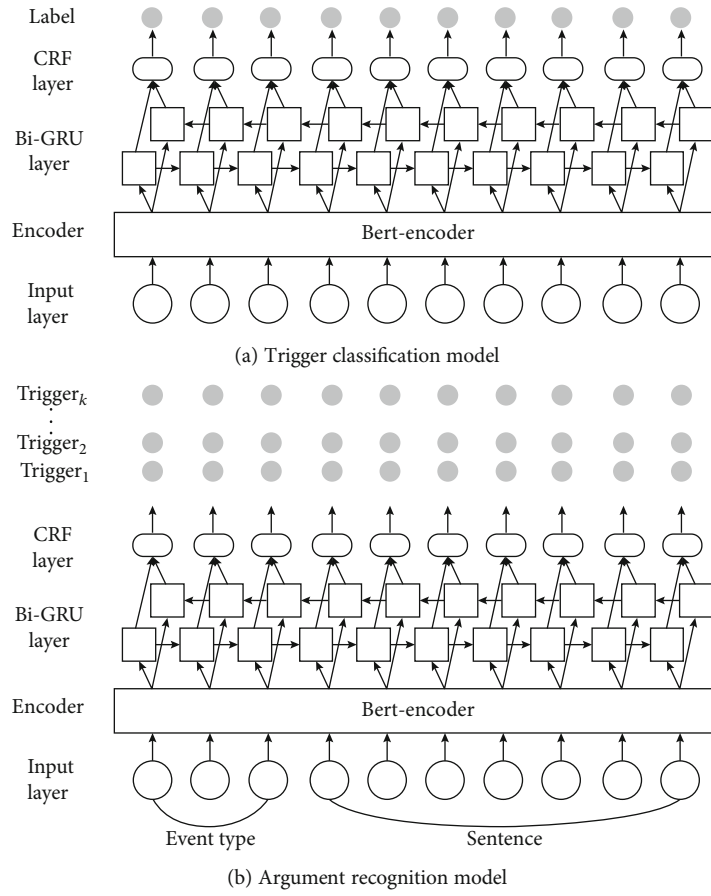


FIGURE 3: Event extraction model architecture.

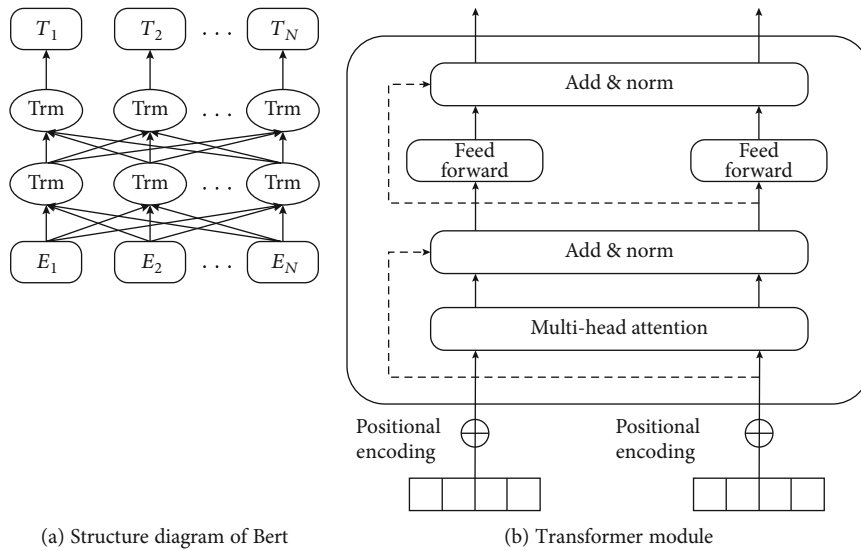


FIGURE 4: Pretrained model Bert.

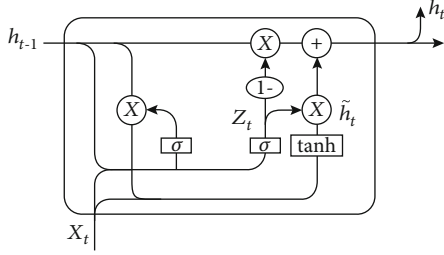


FIGURE 5: GRU module.

Set the matrix $P \in \mathcal{R}^{n \times N_t}$ as the fractional matrix output through the linear layer, where p_{ij} represents the probability that the i th word in the sentence is marked as the j th label. For the sentence $S = \{x_1, x_2, \dots, x_n\}$ and the corresponding tag $y = \{y_1, y_2, \dots, y_n\}$, CRF will give a specific score, the formula is as follows:

$$\text{score}(S, y) = \sum_{i=0}^n T_{y_{i-1}y_i} + \sum_{i=1}^n p_{i,y_i}, \quad (4)$$

where T is the transfer matrix, $T_{i,j}$ is a transfer probability from tag i to tag j , due to the special markers at the beginning and end of a sentence, and T is a square matrix with dimension $N_t + 2$. Then, the probability that the tag sequence of sentence S is y is

$$P(y | S) = \frac{\exp \{ \text{score}(S, y) \}}{\sum_{y' \in Y_s} \exp \{ \text{score}(S, y') \}}, \quad (5)$$

where Y_s represents all the tag sequences that sentence S can give. The logarithm of both sides of the formula is transformed into

$$\log(p(y | S)) = \text{score}(S, y) - \log \left(\sum_{y' \in Y_s} \exp \{ \text{score}(S, y') \} \right). \quad (6)$$

The loss function is defined as $\text{loss} = -\text{score}(S, y)$, and then, the decoded tag sequence is obtained by formula $y^* = \text{argmax}_{y' \in Y_s} \text{score}(S, y')$.

3.6. Trigger Classification. The model uses the BiGRU-CRF model based on pretrained model Bert to identify and classify the triggers. The input of the model is the vector pretrained by Bert, and the encoding features of each word contain the information of all words. The structure of BiGRU-CRF model is shown in Figure 6(a), which consists of three parts: encoding layer, BiGRU layer, and CRF layer.

Let $S = \{x_1, x_2, \dots, x_n\}$ be a sample input, where x_i is the i th word in the sentence, and sentence S is mapped to the matrix $W = \{e_1, e_2, \dots, e_n\}$ after random initialization and passed into the pretrained model Bert. The vector generated by Bert is mapped to the feature matrix $V \in \mathcal{R}^{n \times d}$, where n is the length of the sentence and d is the dimension of the

Trigger	O	B-victory	O	O	O	O
Sentence	Gasley won the British Grand Prix.					
Argument	B-champion	O	O	B-game	I-game	I-game

FIGURE 6: Tagging example.

word vector. Next, the feature matrix V is transferred to the BiGRU layer for further feature extraction, where the hidden state at step i is $h_i = (\vec{h}_i, \overleftarrow{h}_i)$. The calculation formula of hidden state vector is as follows:

$$\begin{aligned} \overrightarrow{\text{GRU}}(v_1, v_2, \dots, v_n) &= (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n), \\ \overleftarrow{\text{GRU}}(v_1, v_2, \dots, v_n) &= (\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n). \end{aligned} \quad (7)$$

After the hidden state h_i of each word is obtained, it is passed as input to the CRF layer for final label classification, and the final score matrix $P \in \mathcal{R}^{N_t \times n}$ is obtained. The k th column of the matrix P represents the fraction of each corresponding tag for input x_1 . According to the matrix P , the tag sequence y is obtained, and then, the trigger in the text is extracted and its type is determined.

3.7. Argument Recognition. The identification of argument is based on the extraction of triggers, and the extracted event types are spliced into the text as features for the next step. As shown in Figure 6(b), this model adopts hierarchical tagging scheme to label multiple events separately, avoiding the defect that one word can only be tagged once at a time. The overall structure of the model is similar to the extraction of triggers. The BiGRU-CRF model based on pretrained model Bert is selected as the extraction model. The difference is that the hierarchical extraction method is selected for the argument extraction. In order to avoid errors in multiscenario event corpus extraction, mask preprocessing is carried out before argument extraction.

The event type and text are spliced in the model, and the feature matrix V is generated in the pretrained model Bert. However, due to the existence of multiscene event corpus, it is necessary to mask the short sentences unrelated to the current event before it is passed into Bert. The calculation process of mask vector is shown in Algorithm 1.

In this paper, the hierarchical argument extraction model extracts arguments from the related trigger words extracted in the previous step and identifies their roles in the trigger. It is known that sentence $S = \{\text{event type}, x_1, x_2, \dots, x_n\}$ is randomly initialized to matrix $W = \{e_0, e_1, \dots, e_n\}$, and the goal is to extract relevant arguments for the current trigger. According to the mask M , the input of the pretrained model is input = $\{\text{event type}, W \odot M\}$. The vector input is mapped to the feature matrix V based on the pretrained model Bert. Next, the feature matrix is transferred to BiGRU layer for further feature extraction, where the hidden state at step i is h_i

```

Input: Sentence:  $S = \{x_1, x_2, \dots, x_n\}$ ; Event type: list; event trigger; triggeri
Output: mask vector:  $M_i$ 
 $M_i = [1] * \text{len}(S)$ ;
 $i, j = 0, 0$ ;
if length of event type is  $l$  then
    return the vector  $M_i$ 
end
other_trigger = list.pop(triggeri)
for triggerj in other_trigger do
    find  $i, j$  which is the index of  $S$  where  $s[i : j + 1]$  is the minimal clause containing triggerj;
    if triggeri is not in the clause  $s[i : j + 1]$  then
        change 1 of  $M[i : j + 1]$  to 0;
    end
end
return the vector  $M_i$ 

```

ALGORITHM 1: Calculation of mask vector for specified event trigger.

$= (\vec{h}_i, \overleftarrow{h}_i)$. Continue to pass in the CRF layer to calculate the score of each word corresponding to each tag, so as to further extract the relevant arguments of the current event.

3.8. Training and Optimization. This summary mainly introduces the learning and optimization details of the framework model. It is known that trigger classification model takes text S as input; the network with parameters θ outputs the event category vector O , where O_i is the value of i th position and indicates that the trigger is a fraction of type I . The optimization of argument recognition model is consistent with the trigger classification model, but the difference is that the event types identified in the previous step are added to the data set. The model in this paper maximizes the log likelihood of the data, and the optimization method used in the model is Adam proposed by Kingma and Ba [15]. The objective function is defined as

$$J(\theta) = \max \sum_{i=1}^{|D|} \sum_{t=1}^{S_i} \left(\log \left(p_t^{(i)} = y_t^{(i)} \mid S_j, \theta \right) \right) \cdot I + \alpha \left(\log \left(p_t^{(i)} = y_t^{(i)} \mid S_j, \theta \right) \right) \cdot (1 - I), \quad (8)$$

where $|D|$ represents the size of training data set, S_i represents the length of the i th sentence, $y_t^{(i)}$ represents the actual tag of the sentence, and $p_t^{(i)}$ is the tag based on the CRF score. α is the bias weight, and the larger the value, the greater the influence of relation label on the model. In addition, $I(O)$ is a conversion function used to distinguish the loss of the mark 'O' and the relation marker, which is defined as follows:

$$I(O) = \begin{cases} 1, & \text{if tag} = 'O', \\ 0, & \text{if tag} \neq 'O'. \end{cases} \quad (9)$$

4. Experiment

4.1. Experiment Settings

4.1.1. Data Set. In order to evaluate the performance of our method, the corpus of event extraction task in 2020 language and intelligent technology competition is selected as our data set. The data set DUEE contains 65 defined event type and 17000 Chinese sentences, including 12000 training sets, 1500 verification sets, and 3500 test sets.

4.1.2. Evaluation. Similar to previous work [16], we use the following criteria to determine the correctness of each prediction event trigger and argument extraction:

- (i) A trigger is correct if its event type and offsets match those of a reference trigger
- (ii) An argument is correctly identified if its event type and offsets match those of any of the reference argument mentions
- (iii) An argument is correctly identified and classified if its event subtype, offsets, and argument role match those of any of the reference argument mentions

Finally, we use Precision (P), Recall (R), and F-measure ($F1$) to evaluate the overall performance.

4.1.3. Hyperparameters. Our model consists of the pretraining layer, BiGRU layer, and CRF layer. The word embedding before the pretraining is generated by random initialization, and the dimension of word embedding is set to $D = 300$. The maximum length of a single sentence is limited to 300 words, dropout is 0.1, and Adam optimizer selects training learning rate of $1e - 3$ and batch size of data set of 32.

4.2. Tagging Scheme. In this paper, we use the "BIO" tagging scheme [17], where "B" (begin) represents the first word of the trigger or argument, I (inside) represents the subsequent word, and O (other) represents the unrelated word. Taking Figure 6 as an example, event type and argument roles have been predefined, and the extracted results are saved in a structured manner. In the case of triggers, the number of tags

for the event type is $N_t = 2 * |R| + 1$, where $|R|$ is the number of predefined event types, and Figure 3 shows an example of the marking method.

4.3. Experimental Results and Analysis

4.3.1. Experiment 1: Event Trigger Classification and Argument Recognition Based on Tagging Scheme. The model is based on the pretrained model Bert, and BiGRU layer and CRF layer are added for trigger classification and argument recognition. Four related neural network models are selected as baseline models to compare with the model proposed in this paper. The four control neural network models are as follows:

- (i) Bert: the data set is used to fine-tune the parameters of the Bert model, and finally, the sequence tag is obtained
- (ii) Bert-CRF: after fine-tuning the parameters of the Bert model using the data set, the conditional random field is added to constrain the related tags
- (iii) Bert-BiLSTM: it consists of Bert and a long short-term memory network layer
- (iv) Bert-BiLSTM-CRF: on the basis of Bert-BiLSTM, conditional random field is added for training

The experimental results of trigger recognition and argument extraction on DUEE by five models are shown in Tables 1 and 2.

Through the comparison of the results between the models in Tables 1 and 2, it can be seen that the addition of the pretrained model Bert improves the event extraction and argument recognition classification, and the average $F1$ score increased by 6.65%.

In Table 3, error analysis is conducted for all trigger classification and argument recognition results. The main causes of errors can be found as follows:

- (i) Trigger classification error: due to the fuzzy vocabulary of event triggers and inconsistent annotation in the data, classification errors occur
- (ii) Missing of trigger word recognition: similar to the multiple scenario event corpus, there are more than one event trigger. However, the model can only identify one or part of the triggers, but not all the triggers
- (iii) Argument classification errors: the model successfully extracted and marked arguments, but the classification was wrong
- (iv) Missing of argument recognition: there will be arguments with multiple roles under an event trigger, and the model will miss some argument when identifying the argument
- (v) Argument boundary segmentation error: event argument extraction is realized by tagging, and the tag will have boundary error

TABLE 1: Trigger classification results on DUEE.

Model	P (%)	R (%)	F (%)
BiLSTM-CRF	87.76	83.26	85.45
Bert	93.17	84.40	88.57
Bert-CRF	94.28	87.99	91.02
Bert-BiLSTM-CRF	94.86	88.03	91.31
Bert-BiGRU-CRF	94.62	88.69	91.56

TABLE 2: Argument recognition results on DUEE.

Model	P (%)	R (%)	F (%)
BiLSTM-CRF	75.27	76.18	75.72
Bert	86.29	81.96	84.07
Bert-CRF	86.86	79.14	82.82
Bert-BiLSTM-CRF	86.36	82.03	84.13
Bert-BiGRU-CRF	86.48	82.37	84.37

4.3.2. Experiment 2: The Influence of Corpus Distribution on Trigger Classification. The distribution of various types of events in the competition data set is not balanced, and the triggers in the same type of event corpus are unbalanced. For example, there are 605 articles on “organizational relationship resignation” and only 74 articles on “organizational behavior parade.” The F score of the two is 97.70% and 61.54%, respectively. Therefore, this experiment studies the influence of corpus distribution on the performance of trigger extraction. By analyzing the relation between corpus distribution and extraction performance, the training data set is enriched manually. The experimental results are shown in Table 4.

The results in Table 4 show that the distribution of corpus has an important impact on trigger recognition. The performance of event extraction is improved by knowledge enhancement of training set by manual supplement of data set. It also lays a better foundation for further argument recognition.

4.3.3. Experiment 3: The Influence of Adding Mask on Argument Recognition. Due to the frequent occurrence of argument recognition confusion in multiscene corpus, this paper chooses to add mask operation before argument extraction to reduce the confusion. In this experiment, we also choose the Bert-BiGRU-CRF model to extract and classify the argument and mask the operation before extracting the argument. The experimental results are shown in Table 5. The change of argument extraction results brought by whether or not to add mask operation is compared.

The results in Table 5 show that the mask operation improves the accuracy of argument extraction to some extent and alleviates the confusion in argument extraction. And the effect was improved most obviously in the Bert-BiGRU-CRF.

5. Conclusion

In this paper, a trigger classification and argument extraction model based on hierarchical annotation scheme is proposed.

TABLE 3: Event extraction error instance.

Type	Error instance
Trigger classification error	Pick up orders at 4 am! 51-year-old takeaway rider killed in a car accident Corpus label: disaster/accident-traffic accident Model tag: life-death
Missing of trigger word recognition	Three people have been killed and five injured in the collapse of Shenzhen sports center Corpus label: disaster/accident-collapse life-death Model tag: life-death
Argument classification errors	Lack of major general Braves hard won the G5, even if the public opinion is more optimistic about the rackets into the west Argument role: {winner: Braves} Role tag: {winner: Braves} {winner: rocket}
Missing of argument recognition	A 5.4 earthquake occurred in Gongxian country. Argument role: {magnitude: 5.4} {epicenter: Gongxian country} Role tag: {magnitude: 5.4}
Argument boundary segmentation error	The movie “Cold Pursuit” will be released nationwide on September 6! Argument role: {time: September 6} {movie: “Cold Pursuit”} Role tag: {movie: “Cold Pursuit” will be released nationwide on September 6”}

TABLE 4: Data augmentation performance of trigger classification.

Model	P (%)	R (%)	F (%)
Bert-BiLSTM-CRF (none)	94.86	88.03	91.31
Bert-BiLSTM-CRF (DA)	95.25	88.44	91.72
Bert-BiGRU-CRF (none)	94.62	88.69	91.56
Bert-BiGRU-CRF (DA)	95.47	88.74	91.98

TABLE 5: The effect of adding mask or not on argument recognition.

Model	P (%)	R (%)	F (%)
Bert-BiLSTM-CRF (none)	86.36	82.03	84.13
Bert-BiLSTM-CRF (mask)	86.73	82.85	84.74
Bert-BiGRU-CRF (none)	86.48	82.37	84.37
Bert-BiGRU-CRF (mask)	86.92	83.14	84.99

The event extraction task is completed by pipeline. Without complex NLP pretreatment, lexical features are extracted, and hierarchical tagging effectively alleviates the problem of argument overlap. The operation of adding mask before argument extraction reduces the confusion of argument extraction and proves the effectiveness of mask operation, which provides an effective event extraction model for multi-scene event corpus.

Compared with the traditional model, Bert-BiGRU can extract more than one expected event at the same time. For each event, different roles of the same argument can be distinguished accurately. However, the error in the process of event discrimination in pipeline model will lead to the error of argument extraction in the later stage, which leads to a wrong transmission. Therefore, the future work will focus on the model construction of joint extraction. For multiscene event extraction, a more reasonable segmentation method

can be used to improve the extraction performance of multi-event corpus. Combining knowledge enhancement [18, 19] is also a major research focus in the future.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by National Natural Science Foundation of China (61702234), and in part by the National Key Research and Development Program Grants 2017YFB1002300 and 2018YFC1700302.

References

- [1] H. L. Chieu and H. T. Ng, “A maximum entropy approach to information extraction from semi-structured and free text,” *National Conference On Artificial Intelligence*, vol. 1, pp. 786–791, 2002.
- [2] H. Llorens, E. Saquete, and B. Navarroc Colorado, “TimeML events recognition and classification: learning CRF models with semantic roles,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 725–733, Beijing, 2010.
- [3] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, “Joint extraction of entities and relations based on a novel tagging scheme,” in *Proceedings of the 55th Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1227–1236, 2017.
- [4] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, “Extracting relational facts by an end-to-end neural model with copy mechanism,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 506–514, Melbourne, Australia, 2018.
 - [5] T. H. Nguyen and R. Grishman, “Relation extraction: perspective from convolutional neural networks,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 39–48, Denver, Colorado, 2015.
 - [6] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “Event extraction via dynamic multi-pooling convolutional neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 167–176, Beijing, China, 2015.
 - [7] R. Ghaeini, X. Z. Fern, L. Huang, and P. Tadepalli, “Event nugget detection with forward-backward recurrent neural networks,” 2018, <https://arxiv.org/abs/1802.05672>.
 - [8] X. Feng, B. Qin, T. Liu et al., “A language-independent neural network for event detection,” *Science China Information Sciences*, vol. 61, no. 9, pp. 92–106, 2018.
 - [9] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, International Conference On Learning Representations, 2013.
 - [10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” 2018, <https://arxiv.org/abs/1810.04805>.
 - [11] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Neural Information Processing Systems*, vol. 1, pp. 5998–6008, 2017.
 - [12] K. Cho, B. Van Merriënboer, C. Gulcehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” 2014, <https://arxiv.org/abs/1406.1078>.
 - [13] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM – a tutorial into long short-term memory recurrent neural networks,” 2019, <https://arxiv.org/abs/1909.09586>.
 - [14] J. Lafferty, A. McCallum, F. Pereira et al., “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” *International Conference On Machine Learning*, pp. 282–289, 2001.
 - [15] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, <https://arxiv.org/abs/1412.6980>.
 - [16] Q. Li, H. Ji, and L. Huang, “Joint event extraction via structured prediction with global features,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 73–82, Sofia, Bulgaria, 2013.
 - [17] Z. Li, Q. Li, X. Zou, and J. Ren, “Causality extraction based on self-attentive bilstm-crf with transferred embeddings,” *Neurocomputing*, vol. 423, pp. 207–219, 2021.
 - [18] S. Bhatt, A. Sheth, V. Shalin, J. Zhao, and A. Sheth, “Knowledge graph semantic enhancement of input data for improving AI,” *IEEE Internet Computing*, vol. 24, no. 2, pp. 66–72, 2020.
 - [19] J. Liu, Y. H. Yang, and H. He, “Multi-level semantic representation enhancement network for relationship extraction,” *Neurocomputing*, vol. 403, pp. 282–293, 2020.

Review Article

A Survey of k Nearest Neighbor Algorithms for Solving the Class Imbalanced Problem

Bo Sun ¹ and Haiyan Chen ²

¹Department of Computer Science and Technology, Shandong Agricultural University, China

²College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

Correspondence should be addressed to Bo Sun; sunbo87@sdau.edu.cn and Haiyan Chen; chenhaiyan@nuaa.edu.cn

Received 17 January 2021; Accepted 18 February 2021; Published 3 March 2021

Academic Editor: Junwu Zhu

Copyright © 2021 Bo Sun and Haiyan Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

k nearest neighbor (k NN) is a simple and widely used classifier; it can achieve comparable performance with more complex classifiers including decision tree and artificial neural network. Therefore, k NN has been listed as one of the top 10 algorithms in machine learning and data mining. On the other hand, in many classification problems, such as medical diagnosis and intrusion detection, the collected training sets are usually class imbalanced. In class imbalanced data, although positive examples are heavily outnumbered by negative ones, positive examples usually carry more meaningful information and are more important than negative examples. Similar to other classical classifiers, k NN is also proposed under the assumption that the training set has approximately balanced class distribution, leading to its unsatisfactory performance on imbalanced data. In addition, under a class imbalanced scenario, the global resampling strategies that are suitable to decision tree and artificial neural network often do not work well for k NN, which is a local information-oriented classifier. To solve this problem, researchers have conducted many works for k NN over the past decade. This paper presents a comprehensive survey of these works according to their different perspectives and analyzes and compares their characteristics. At last, several future directions are pointed out.

1. Introduction

k nearest neighbor (k NN) [1] has simple implementation and supreme performance and can achieve comparable performance with more sophisticated classifiers including decision tree [2], artificial neural network [3], and support vector machine [4]. Therefore, k NN has been listed as one of the top 10 algorithms in data mining and machine learning [5, 6]. k NN has been utilized in many applications, such as pattern recognition [7], feature selection [8], and outlier detection [9]. For a test example with unknown class label, k NN makes a decision by employing the local information surrounding the test example. Concretely, k NN first simply stores all the training examples; then, in the classification phase, it takes the class occurring most frequently in the k ($k \geq 1$) nearest training examples of the test example as the

classification result. That is, k NN makes a decision according to the class distribution characteristics in the k neighborhood of a test example.

Nowadays, machine learning and data mining techniques are widely used in many aspects of the information society. However, for some applications such as medical diagnosis [10], system intrusion detection [11], and network fraud detection [12], the collected training example set is usually class imbalanced, i.e., there is a large difference among the sizes of different classes. For instance, in medical diagnosis data, the majority examples are descriptions of normal patients (negative examples), and only a small proportion of examples are representatives of special patients suffering a rare disease (positive examples). But if a special patient is erroneously classified as a normal patient, the best treatment time will be missed and serious consequences will be caused.

For computer network intrusion detection data, the majority examples denote the normal access data (negative examples) and only the minority examples denote the illegal access data (positive examples). Similarly, misclassifying illegal access as a legal one will lead to the disclosure of a unit's inner data or the steal of bank account information. From the above two instances, it can be seen that, in class imbalanced data, although the positive class is heavily outnumbered by the negative class, the positive class is usually the one in which we are more interested and is more important than the negative class. The positive class is also named the minority class while the negative class is also called the majority class.

Similar to classical classifiers such as decision tree, artificial neural network, and support vector machine, k NN is also proposed based on the assumption that a training set has approximately balanced class distribution, i.e., the classes have roughly the same number of training examples. In addition, these algorithms all employ the overall classification accuracy as the optimization objective in the classifier training phase, leading to their unsatisfactory performance on class imbalanced data. For k NN, it takes the majority class in the k neighborhood of a test example as the classification result; this majority voting-based classification rule further degrades its performance on a class imbalanced problem. This is because the positive examples are usually sparse in the k neighborhood of a test example [6], i.e., most examples in the k neighborhood are usually negative examples; thus, the positive examples are often misclassified as negative ones by k NN, leading to the poor classification performance for positive examples. For instance, in the binary classification problem shown in Figure 1 (circles denote negative examples, triangles denote positive examples, and the cross denotes a test example), when k equals 7, there are 4 negative examples (N1-N4) and 3 positive examples (P1-P3) in the k neighborhood; obviously, k NN classifies the test example as the negative class although it actually belongs to the positive class.

Experiments conducted in Reference [13] indicate that SMOTE oversampling integrated with Random Undersampling (RUS) [14] or SMOTE oversampling integrated with the cost-sensitive MataCost method [15] can both significantly improve the performance of C4.5 decision tree [2] on class imbalanced data. Unfortunately, these strategies do not work well for improving k NN in a class imbalanced scenario. The authors in [13] give the explanation from the following aspect: k NN makes a decision by investigating the *local* neighborhood of a test example, while the resampling and cost-sensitive strategies are *global* methods and are naturally inappropriate to k NN. Therefore, special methods for k NN need to be designed under the class imbalanced scenario.

As can be seen from the above illustration, improving k NN performance on imbalanced data is an important topic, which is of great significance to the expansion of its application fields and the enhancement of its practical utility. Over the past decade, researchers have conducted many works and proposed many methods. This paper tries to give a comprehensive survey of these works according to their perspectives and analyzes and compares their characteristics, which serves as a foundation for further study in this field.

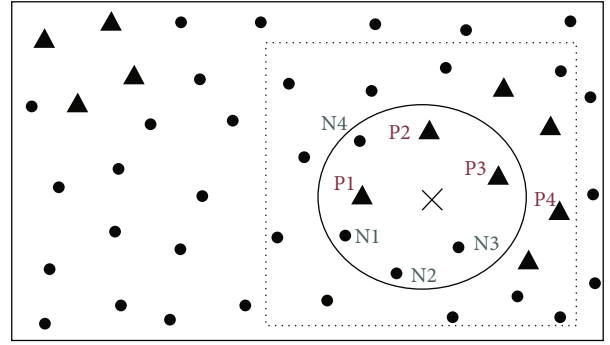


FIGURE 1: Classical k NN performs poorly on positive examples (cited from Reference [12]).

The rest of this paper is organized as follows. The weighting strategy-based methods are illustrated in Section 2, the local geometrical structure-based methods are illustrated in Section 3, Section 4 introduces the fuzzy logic-based methods and followed by a category of methods based on missing positive example estimation in Section 5, Section 6 presents methods based on novel distance metrics while Section 7 presents dynamic-sized neighborhood-based methods, and conclusions and future work are presented in Section 8.

2. Methods Based on Weighting Strategy

This section introduces a category of methods that assigns weights to training examples in the neighborhood of a test example. In general, these methods can be divided into 5 sub-categories as shown in the following.

2.1. Weighting Strategy Considering the Class Distribution around the Neighborhood. The authors in [12] claim that the reason for the unsatisfactory performance of k NN on the imbalanced data lies in the following: it only utilizes the local prior probabilities of each class in the neighborhood of a test example but does not employ the class distribution information around the neighborhood. In Figure 1, if the imbalanced class distribution around the test example's neighborhood is considered, i.e., the area surrounded by the dotted rectangle, then the test example can be correctly classified as the positive class because in this dotted area, the positive nearest neighbors of the test example are much more than the negative ones. Therefore, the classification performance of k NN can be improved if such local class distribution information is utilized.

Based on the above observation, a weighting-based method is proposed in [12] to assign a test example-dependent local weight to each class, i.e., the examples' weight in a class varies with the change of test examples rather than being a constant value. Concretely, for test example $x_i \in \mathbb{R}^d$, the weight w_i^l of examples in class C_l ($1 \leq l \leq L$, L is the total number of classes in a classification problem) is calculated as follows. For the $\lceil k/L \rceil$ number of nearest neighbors of x_i in class C_l , if they are erroneously classified by traditional k NN, it is likely that these $\lceil k/L \rceil$ neighbors belong to the minority class (the positive class) in the neighborhood of x_i ; in this case, the weight of class C_l is enlarged. Therefore,

the learned weights take into consideration the class distribution information around the neighborhood.

For the binary classification problem in Figure 1, $L = 2$, when k equals 7, we have $\lceil k/L \rceil = \lceil 7/2 \rceil = 4$; the test example's 4 nearest neighbors in positive class C_1 are P1 to P4, while its 4 nearest neighbors in negative class C_2 are N1 to N4. It can be seen that P1, P2, and P4 are misclassified to be negative examples as the majority members of their 7 nearest neighbors are all negative examples; thus, the weight of these positive examples should be enlarged. Based on the enlarged weight of positive class, in the classification phase of 7NN for the test example, the 3 positive neighbors P1 to P3 have much larger weight than the 4 negative neighbors, i.e., P1 to P3 make more contribution to the classification result; thus, the correct classification is achieved.

However, the shortage of this weighting-based method is that approximately k (the exact number is $\lceil k/L \rceil * L$) times of extra running of k NN is required around the neighborhood of each test example; thus, the computation cost is enlarged.

2.2. Weighting Strategy Based on Examples' Informativeness. The authors in [16] believe that some examples carry more information than other examples: if an example is close to the test example and far from examples of other classes, then it is considered to be more informative. Following this idea, it is easily seen from Figure 2 that the example with index 2 carries more information than the one with index 1. The reason is that the two examples have roughly the same distance to the test example (the "query point" in Figure 2), but the example with index 1 is nearer to the class boundary, i.e., it is closer to the other classes. Based on the above consideration, the authors propose two informative k NN algorithms: the local information-based version LI- k NN and the global information-based version GI- k NN.

2.2.1. The Idea of LI- k NN. LI- k NN first finds the k nearest neighbors $x_{t_1}^1, x_{t_1}^2, \dots, x_{t_1}^k$ of test example x_t in the training set, then employs the designed metric to evaluate the informativeness of each training example in the k neighborhood, i.e., the evaluation scope is *local*, and selects the first I ($I < k$) most informative examples $x_{t_1}^1, x_{t_1}^2, \dots, x_{t_1}^I$. Finally, the majority class among the class labels $y_{t_1}^1, y_{t_1}^2, \dots, y_{t_1}^I$ of these I examples is regarded as the classification result. That is to say, the weight of the I selected examples is set to 1 while the weight of the other $(k - I)$ neighboring training examples is set to 0.

2.2.2. The Idea of GI- k NN. After LI- k NN determines I most informative neighbors $x_{t_1}^1, x_{t_1}^2, \dots, x_{t_1}^I$ and then makes a decision for test example x_{t_1} , when classifying the next test example x_{t_2} , it separately determines the I most informative neighbors $x_{t_2}^1, x_{t_2}^2, \dots, x_{t_2}^I$ and does not utilize the informative neighbors $x_{t_1}^1, x_{t_1}^2, \dots, x_{t_1}^I$ of the previous test example x_{t_1} . However, GI- k NN believes that some informative neighbors of a test example may be also the members of other test examples' informative neighbors. For instance, two informative neighbors $x_{t_1}^1, x_{t_1}^2$ of test example x_{t_1} may be members of the informative neighbor sets of three test examples x_{t_2}, x_{t_3}

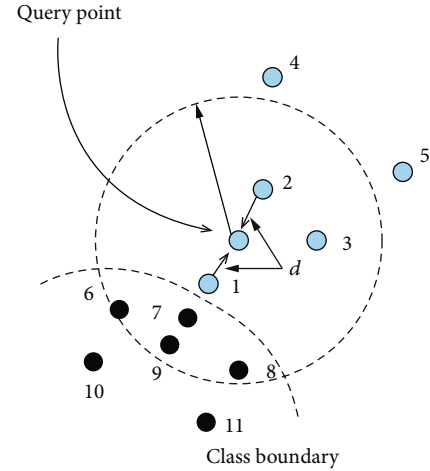


FIGURE 2: Schematic diagram of examples' information (cited from Reference [16]).

and x_{t_4} . In this case, training examples like $x_{t_1}^1, x_{t_1}^2$ are considered to be *global* informative, thus assigning a larger weight to these examples. Based on the above idea, GI- k NN tries to find the training examples with global informativeness and assigns larger weights to them compared with ordinary examples.

To summarize, LI- k NN is a local strategy as it determines the informative examples in the k neighborhood of a test example. GI- k NN is a global strategy as it evaluates the informativeness of all the training examples and then assigns larger weights to the examples that are globally more informative, and besides, these weights are fixed when classifying all the subsequent test examples.

Experimental results indicate that GI- k NN and LI- k NN are not very sensitive to the change of parameter I and can achieve comparable performance with SVM. One drawback of GI- k NN is that the robustness of its adopted informativeness metric needs to be enhanced when there exist noisy examples in the training set.

2.3. Class Confidence-Based Weighting Strategy. The class confidence-weighted (CCW) k NN method [17] is proposed to assign weights to training examples in the neighborhood. As shown in Figure 3, the real boundary between the negative class (denoted by blue triangles) and the positive class (denoted by red circles) is represented by the solid blue line. There are 4 negative examples and 1 positive example in the k (k equals 5 in this case) neighborhood of the test example (denoted by the solid green circle), and the nearest training example of the test example is a negative one. In this case, the classification result is certainly the negative class if the traditional majority voting-based classification rule is adopted. However, the test example actually belongs to the positive class and the negative examples in its neighborhood are also positive ones in reality. Thus, for each training example in the neighborhood, the probability that it belongs to its current class should be considered.

Based on this idea, for each training example $(x_t^j, y_t^j) = ((x_t^{j1}, x_t^{j2}, \dots, x_t^{jd}), y_t^j)$ ($j = 1, 2, \dots, k$) in the neighborhood of

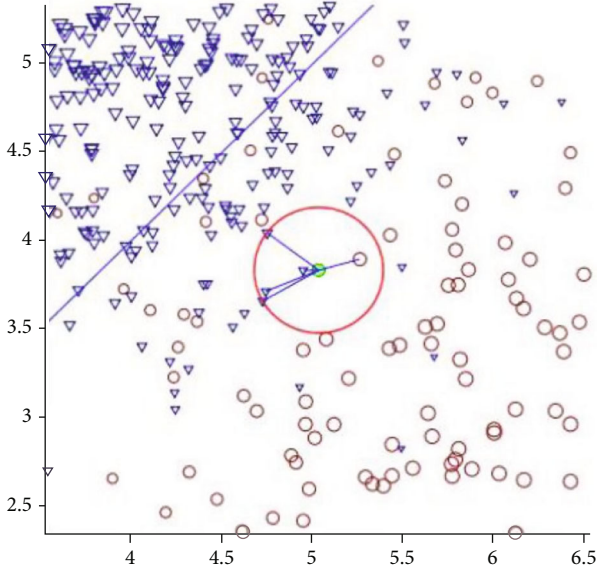


FIGURE 3: An instance of class confidence-weighted (CCW) method (cited from Reference [17]).

test example x_t , the confidence of its belonging to the current class y_t^j is calculated according to its attributes' values $(x_t^j, x_t^{j_2}, \dots, x_t^{j_d})$ and then this confidence serves as the weight w_t^j of this training example. For the instance shown in Figure 3, the mixture model and Bayesian network are employed in [17] to calculate the corresponding confidences of the 4 negative examples in the neighborhood and the values are 0.0245, 0.0173, 0.0171, and 0.0139, respectively. In addition, the calculated confidence of the positive example in the neighborhood is 0.1691. Obviously, the sum of the confidences of the 4 negative examples is much smaller than the confidence of the positive example, i.e., the weights of the 4 negative examples are much smaller than that of the positive example. In this way, the positive example in the neighborhood has much more influence on the classification result than the negative ones, ensuring that the test example can be correctly classified as a positive one. The conducted experiments also indicate that the class confidence-weighted method can correct k NN's inherent bias to negative examples.

On the other hand, the class confidence-weighted method has to calculate the class confidence for each training example in the neighborhood, which increases the computation cost to some extent.

2.4. Weighting Strategy Based on Nearest Neighbor Density. A nearest neighbor density-based weighted class-wise k NN (WckNN) algorithm is proposed in [18], and its basic idea is as follows.

First, the k nearest neighbor density of test example x_t is determined in each class.

This is implemented by constructing a k radius sphere $S_{l,k}(x_t)$ that takes test example x_t as its center and contains at last k nearest examples from class C_l ($1 \leq l \leq L$); then, the volume $V_{l,k}(x_t)$ of this k radius sphere is used to denote the

density: $1/V_{l,k}(x_t)$. It is not hard to see that, for a test example, its k radius sphere in the positive class usually has much larger volume than the one in the negative class due to the sparse distribution of positive class examples. As the radius of k radius sphere $S_{l,k}(x_t)$ is determined by the distance $d_{l,k}(x_t)$ between test example x_t and its k th nearest neighbor in class C_l , thus $d_{l,k}(x_t)$ is often used to approximately denote the volume of sphere $S_{l,k}(x_t)$.

Second, the posterior probability of test examples belonging to each class is calculated based on the above k nearest neighbor density, which is shown in

$$P(C_l | x_t) = \frac{\beta_l}{d_{l,k}(x_t)}, \quad l = 1, 2, \dots, L. \quad (1)$$

In formula (1), the weight β_l of class C_l is obtained by employing a certain convex optimization technique to optimize a nonlinear metric on the training set, and from this formula, we have the following observations. (a) For a class balanced data, two classes C_1 and C_2 have equal weights ($\beta_1 = \beta_2$); in this case, if examples in class C_1 are more densely distributed around test example x_t , i.e., $V_{1,k}(x_t) < V_{2,k}(x_t)$ and $d_{1,k}(x_t) < d_{2,k}(x_t)$, then the probability of x_t belonging to C_1 is larger than that of C_2 . (b) For an imbalanced data, compared with negative class C_n , positive class C_p is more likely to be sparsely distributed around x_t , i.e., $d_{C_p,k}(x_t) > d_{C_n,k}(x_t)$; fortunately, the effect of this imbalanced distribution can be overcome by assigning larger weight for the positive class, i.e., $\beta_p > \beta_n$.

At last, the class having the largest posterior probability is considered as the classification result: $y = \arg \max_l P(C_l | x_t)$, $l = 1, 2, \dots, L$.

In terms of complexity, when classifying test example x_t , WckNN needs to run one time of k NN on each class to determine the k nearest neighbor density in this class. Thus, L (the total number of classes) times running of k NN are needed to classify a test example.

2.5. Weighting Strategy Integrated with Self-Adaptive k . The methods introduced above all use a constant k value, i.e., for each test example, k is the sum of the number of its positive neighbors and the number of its negative ones. Thus, the number of neighbors is not considered separately for each class.

To further improve the performance of weighted k NN methods, the authors in [19] propose to integrate the self-adaptive k technique with the example weighting strategy. In terms of weight determination, the positive examples are assigned larger weights than the negative ones; in terms of neighborhood size, the positive class is given small neighborhood size k_p while the negative class is given relative large neighborhood size k_n , i.e., $k_n > k_p$. In this way, the test example's k_p positive neighbors and k_n negative neighbors constitute its neighborhood with size $k_p + k_n$.

Accordingly, the classification result is determined by two aspects: (a) the weighted sum of the test example's k_p

positive neighbors: $W_p(x_t) = \sum_{i=1}^{k_p} w_t^i$, and (b) the weighted sum of the test example's k_n negative neighbors: $W_n(x_t) = \sum_{j=1}^{k_n} w_t^j$. The class with a larger value is the corresponding decision result. To sum up, the self-adaptive k -based weighted k NN is simple and flexible. As to the formula used in the assignment of each class's neighborhood size, more efforts need to be made to ensure that it is theoretically sound.

3. Methods Based on Local Geometric Structure of Data

An algorithm named class conditional nearest neighbor distribution (CCNND) is presented in [20], which alleviates the class imbalanced problem by using the local geometric structure of data, and its basic idea is as follows.

3.1. Calculating the k Nearest Neighbor Distances in Each Class. For each training example x_m in class C_l ($1 \leq l \leq L$), the distances to its k nearest neighbors (without considering itself) in class C_l are calculated: $d_{l,k}(x_m) = (\text{dist}(x_m, x_{m1}^l), \text{dist}(x_m, x_{m2}^l), \dots, \text{dist}(x_m, x_{mq}^l))$, where $\text{dist}(x_m, x_{mq}^l)$ is the distance between x_m and its q th ($1 \leq q \leq k$) nearest neighbor x_{mq}^l in class C_l .

3.2. Making Decisions Based on the k Nearest Neighbor Distances of Test Example in Each Class. First, for test example x_t , the distances to its k nearest neighbors in class C_l ($1 \leq l \leq L$) are calculated: $d_{l,k}(x_t) = (\text{dist}(x_t, x_{t1}^l), \text{dist}(x_t, x_{t2}^l), \dots, \text{dist}(x_t, x_{tk}^l))$, where $\text{dist}(x_t, x_{tq}^l)$ is the distance between x_t and its q th nearest neighbor x_{tq}^l ($1 \leq q \leq k$) in class C_l .

Second, for each class, the number of its training examples with larger k nearest neighbor distances than the test example is determined: $N_l(x_t) = \{x_m \mid (x_m \in C_l) \wedge (d_{l,k}(x_m) > d_{l,k}(x_t))\}$, $l = 1, 2, \dots, L$, where $d_{l,k}(x_m) > d_{l,k}(x_t)$ is equivalent to $(\text{dist}(x_m, x_{m1}^l) > \text{dist}(x_t, x_{t1}^l)) \wedge (\text{dist}(x_m, x_{m2}^l) > \text{dist}(x_t, x_{t2}^l)) \wedge \dots \wedge (\text{dist}(x_m, x_{mk}^l) > \text{dist}(x_t, x_{tk}^l))$. It can be seen that the more such examples of a class have, the closer its class center to the test example is, i.e., the more likely the test example belongs to this class. Thus, the classification result is denoted as $y_t = \arg \max_{l \in \{1, 2, \dots, L\}} N_l(x_t)$.

The conducted experiments demonstrate that, compared with the classical resampling and cost-sensitive methods, CCNND can achieve comparable or even better performance. As shown in Figure 4, the decision boundary obtained using CCNND is closer to the real boundary than that obtained using SVM and nearest neighbor. In addition, another advantage of CCNND is that it still works when the imbalance degree in a training set changes with time, e.g., in the case of online streaming data [21]. Therefore, CCNND can be applied in streaming data such as the oil and natural gas industrial data.

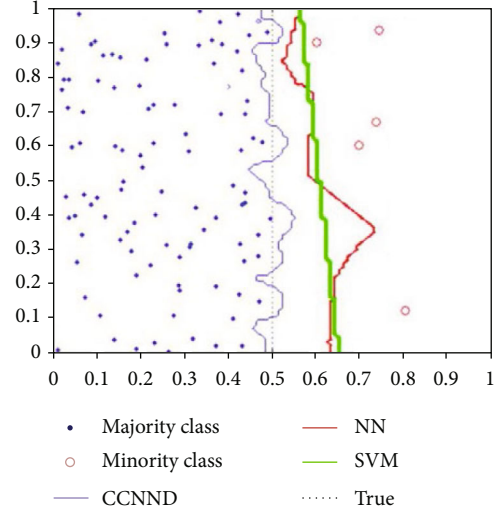


FIGURE 4: The class boundary obtained by the CCNND algorithm (cited from Reference [20]).

4. Fuzzy Logic-Based Methods

In fuzzy logic-based [22] classification methods, the membership of belonging to each class is assigned to an example rather than a crisp class label, which can preserve abundant classification information and thus make a full classification. Based on this conclusion, a fuzzy weighted k NN algorithm is proposed in [22] by integrating the advantages of both fuzzy logic and weighted k NN, which is the first method introduced in Subsection 4.1, and the second method in Subsection 4.2 is a further improvement of fuzzy k NN itself.

4.1. Fuzzy Weighted k NN Algorithm. The fuzzy weighted k NN in [22] improves the weighted k NN method by utilizing the advantage of fuzzy logic, and it has the following three steps.

4.1.1. Determining the Class Membership of Each Example. The membership of example $x \in \mathbb{R}^d$ for class C_l ($1 \leq l \leq L$) is calculated using formula (2), where n_{C_l} is the number of training examples belonging to class C_l in the k neighborhood of example x and $C(x)$ is the true class label of x .

$$\mu_{C_l}(x) = \begin{cases} 0.51 + \left(\frac{n_{C_l}}{k}\right) * 0.49, & \text{if } C_l = C(x), \\ \left(\frac{n_{C_l}}{k}\right) * 0.49, & \text{otherwise.} \end{cases} \quad (2)$$

For instance, in binary classification, if the true class of example x_1 is class C_1 and there are 4 neighbors belonging to class C_1 in its 5 neighborhood (i.e., $k = 5$), then the membership of x_1 for C_1 is $\mu_{C_1}(x_1) = 0.51 + (4/5) * 0.49 = 0.902$ while the one for C_2 is $\mu_{C_2}(x_1) = (1/5) * 0.49 = 0.098$.

4.1.2. Determining the Weight of Each Class. The weight w_l of class C_l ($1 \leq l \leq L$) is calculated using formula (3), where $N(C_l)$ denotes the number of examples in C_l , i.e., the size of

this class. It is easy to see that the positive class is assigned a weight of 1 while the negative class is assigned a weight less than 1, and the more examples in the negative class the smaller its weight.

$$w_l = \frac{1}{(N(C_l)/\min\{N(C_1), N(C_2), \dots, N(C_L)\})}, \quad l = 1, 2, \dots, L. \quad (3)$$

4.1.3. Making a Decision Based on the Class Memberships and the Class Weights. The class membership $\mu_{C_l}(x_t)$ of test example x_t for class C_l ($1 \leq l \leq L$) is calculated using formula (4), where $\mu_{C_{l,j}}(x_t)$ is the class membership of x_t 's j th ($j = 1, 2, \dots, k$) nearest neighbor for class C_l and w_j is the weight of the class to which this neighbor belongs.

$$\mu_{C_l}(x_t) = \frac{\sum_{j=1}^k w_j * \mu_{C_{l,j}}(x_t)}{\sum_{j=1}^k w_j}, \quad l = 1, 2, \dots, L. \quad (4)$$

At last, the decision is the class having the largest membership: $y(x_t) = \arg \max_{\{l=1,2,\dots,L\}} \mu_{C_l}(x_t)$.

4.2. Self-Adaptive k -Based Fuzzy k NN. Although the weighted fuzzy k NN introduced in Subsection 4.1 can achieve good performance, the fuzzy k NN algorithm itself can not accurately compute examples' class membership under the class imbalanced scenario. To solve this problem, an improved fuzzy k NN algorithm based on self-adaptive k strategy is proposed in [23] and it contains the following steps.

4.2.1. Determining the Neighborhood Size k for Each Class. The basic idea is to use relatively large neighborhood for the negative class and small neighborhood for the positive class. Concretely, the neighborhood size of each class is determined using formula (5). Where $N(C_l)$ is the number of training examples in class C_l , λ is a constant (e.g., take the value 1) with the purpose of preventing the value of k_{C_l} from being too small.

$$k_{C_l} = \min \left\{ \lambda + \left\lceil \frac{k * N(C_l)}{\max\{N(C_l) \mid l = 1, 2, \dots, L\}} \right\rceil, k, N(C_l) \right\}, \quad (5)$$

where $l = 1, 2, \dots, L$.

4.2.2. Calculating the Class Membership of Training Examples According to the Obtained Neighborhood Size. Formula (6) adopted here is different from formula (2): the corresponding class's k value k_{C_l} is utilized when calculating the class membership of example x , and $C(x)$ is the true class label of x .

$$\mu_{C_l}(x) = \begin{cases} 0.51 + \left(\frac{n_{C_l}}{k_{C_l}}\right) * 0.49, & \text{if } C_l = C(x), \\ \left(\frac{n_{C_l}}{k_{C_l}}\right) * 0.49, & \text{otherwise,} \end{cases} \quad (6)$$

where $l = 1, 2, \dots, L$.

4.2.3. Determining the Class Membership of the Test Example. The class membership of test example x_t for class C_l ($l = 1, 2, \dots, L$) is calculated using formula (7), where $\mu_{C_l}(x_{t,j}^l)$ is the class membership that example $x_{t,j}^l$ belongs to class C_l , and $x_{t,j}^l$ is the j th ($1 \leq j \leq k_{C_l}$) neighbor of test example x_t in class C_l . It can be seen from formula (7) that, in fact, test example x_t 's class membership for class C_l is the distance weighted sum of the corresponding class memberships of x_t 's k_{C_l} nearest neighbors.

$$\mu_{C_l}(x_t) = \frac{\sum_{j=1}^{k_{C_l}} \mu_{C_l}(x_{t,j}^l) \left(1 / \|x_t - x_{t,j}^l\|^{2/(p-1)}\right)}{\sum_{j=1}^{k_{C_l}} \left(1 / \|x_t - x_{t,j}^l\|^{2/(p-1)}\right)}, \quad (7)$$

where l takes values from $\{1, 2, \dots, L\}$ and p is an integer and is larger than 1.

At last, the class having the largest membership is the classification decision: $y(x_t) = \arg \max_{\{l=1,2,\dots,L\}} \mu_{C_l}(x_t)$.

By adopting different neighborhood sizes for different classes, this self-adaptive k -based fuzzy k NN can effectively alleviate the adverse influence of negative examples in the neighborhood of a positive example, making the obtained class membership more objective and thus improving the classification performance of fuzzy k NN on imbalanced data.

5. Methods Based on Missing Positive Data Estimation

The class imbalanced problem is regarded as a missing positive data estimation problem in [24]. From this perspective, a method called Fuzzy-based Information Decomposition (FID) is proposed and its main idea is as follows.

- (1) t ($t > 0$) number of synthetic positive examples are generated, and at the beginning, all their attributes have missing values
- (2) The values of each attribute are estimated according to the current training set

Concretely, for the s th ($s = 1, 2, \dots, d$, d is the dimension of training data) attribute attr^s :

- (1) Dividing all the available values on attr^s into t intervals

According to the values of the current training set on attr^s (i.e., $\text{attr}_1^s, \text{attr}_2^s, \dots, \text{attr}_N^s$) and the number t of positive examples to be generated, t intervals are obtained: $q_1 = [a, a + h]$,

$q_2 = (a + h, a + 2h], \dots, q_t = (a + (t - 1) * h, b]$, where attr_i^s is the value of the i th ($i = 1, 2, \dots, N$) training example on attribute attr^s , N is the total number of current training examples, a and b are, respectively, the minimum and maximum values among $(\text{attr}_1^s, \text{attr}_2^s, \dots, \text{attr}_N^s)$, and h is the step length: $h = (b - a)/t$.

- (2) Generating a synthetic attribute value for each interval

For the m th ($1 \leq m \leq t$) interval q_m , a synthetic value $\text{attr}_{q_m}^s$ of attribute attr^s is generated in the following way. The fuzzy membership $\mu(x_i, q_m)$ of training example x_i 's ($i = 1, 2, \dots, N$) attribute value attr_i^s with respect to interval q_m is calculated, which is used as the weight w_i^m of example x_i in estimating the m th missing value of attribute attr^s . For instance, if the attribute value attr_j^s of example x_j ($j = 1, 2, \dots, N$) is a "neighbor" to the center $((a + (m - 1) * h) + (a + m * h))/2$ of interval q_m , i.e., their distance is less than the step length h , then the corresponding fuzzy membership $\mu(x_j, q_m)$ is calculated and served as the weight $w_j^m = \mu(x_j, q_m)$ of example x_j ; otherwise, the weight of example x_j is set to 0: $w_j^m = 0$. Therefore, the m th estimated value for attribute attr^s can be represented in formula (8).

$$\text{attr}_{q_m}^s = \sum_{j=1}^N w_j^m \cdot \text{attr}_j^s = \sum_{j=1}^N \mu(x_j, q_m) \cdot \text{attr}_j^s. \quad (8)$$

The weights satisfy that $\sum_{j=1}^N w_j^m = 1$.

That is to say, only when an example's attribute value is close to the center of interval q_m ($m = 1, 2, \dots, t$) can it effectively influence the calculation of the m th synthetic attribute value $\text{attr}_{q_m}^s$. Thus, the m th estimated value for attribute attr^s is the weighted sum of these effective training examples on this attribute.

The advantage of FID is that it can deal with data with arbitrary dimension as it *separately* generates the missing values for each attribute. Traditional methods like Random OverSampling (ROS) [14] and Clustering Based OverSampling (CBOS) [25] have the tendency of overfitting due to the replication of existing positive examples; for methods like SMOTE [26] and Majority Weighted Minority (MWM) oversampling [27], an approximate positive example needs to be selected before generating a synthetic positive example using linear interpolation. However, these traditional methods have poor performance when the positive examples in the original training set are not enough. Fortunately, FID can overcome this problem as it generates the synthetic values separately for each attribute. As to the disadvantage of FID, when calculating a synthetic value for an attribute, the memberships of all training examples' values on this attribute with respect to the current interval need to be computed, leading to high computation and time complexity in the case of large training set size.

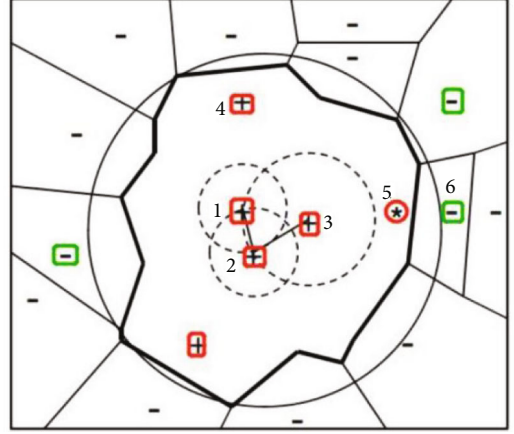


FIGURE 5: Schematic diagram of positive exemplar (cited from Reference [28]).

6. Novel Distance Metric-Based Methods

Euclidean distance is usually adopted as the metric to evaluate the similarity between two examples. However, this metric does not treat the positive and negative examples separately in the calculation of distance. To make up this shortcoming, the following works, respectively, present a novel distance metric that is sensitive to positive examples.

6.1. Distance Metric for Exemplar Positive Examples. The authors in [28] propose a method called k exemplar-based Nearest Neighbor (k -ENN), which improves the classification performance to positive examples by extending each exemplar positive example from a point in the feature space to a Gaussian ball. In detail, the principle of k -ENN is as follows.

6.1.1. Determining the Exemplar Positive Examples. For each positive example x_i^p ($1 \leq i \leq n_p$, n_p is the total number of positive examples in the training set) in training set D_{tr} , find its nearest positive class neighbor x_j^p ($i \neq j$) and compute their distance $r_i^p = \text{dist}(x_i^p, x_j^p)$. In this way, a Gaussian ball centered at x_i^p and with a radius of r_i^p is constructed, i.e., a neighborhood S_i of x_i^p is obtained. As x_j^p is the nearest neighbor of x_i^p in positive class, there are only two positive examples in Gaussian ball S_i : x_i^p and x_j^p , and the other negative examples occurring in S_i are "false positives." k -ENN considers positive example x_i^p as an exemplar positive example if the false positive rate fp_i in its Gaussian ball S_i is less than a threshold: $fp_i < \tau$.

For instance, Figure 5 displays the Gaussian balls of three positive examples numbered 1 to 3, which are denoted using dashed circles. It is easy to see that there is no negative example (denoted by symbol "-") in each Gaussian ball; thus, the three positive examples are all exemplar ones.

6.1.2. Defining the Distance Between the Test and Training Examples. When classifying test example x_i , its distance to each training example x_i ($1 \leq i \leq N$) needs to be computed.

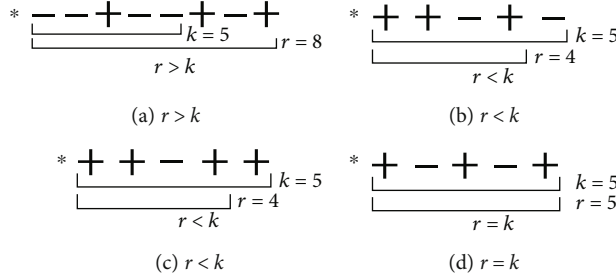


FIGURE 6: Schematic diagram of neighborhood in PNN.

In k -ENN, if x_t is an exemplar positive example, then the distance is defined as

$$d(x_t, x_i) = \text{dist}(x_t, x_i) - r_i^p. \quad (9)$$

If x_t is not an exemplar positive example, i.e., an ordinary positive example or a negative one, then the distance is still the Euclidean distance:

$$d(x_t, x_i) = \text{dist}(x_t, x_i). \quad (10)$$

The distance in formula (9) subtracts the radius of the Gaussian ball of exemplar positive example x_i ; in reality, it is the distance between test example x_t and the boundary of x_i 's Gaussian ball. In this way, the distance from exemplar positive examples to the test example is reduced such that these exemplar positives are given more attention in the classification phase, consequently improving the classification performance for positive examples.

6.2. Distance Based on Examples' Weights. The example weighting-based distance is proposed in [29]; it considers the relative importance of each training example x_i ($1 \leq i \leq N$) when classifying test example x_t , which is implemented by utilizing training examples' weights rather than simply computing the Euclidean distance.

$$d_w(x_t, x_i) = \frac{\|x_t - x_i\|}{w_i}. \quad (11)$$

It can be seen from formula (11) that, by assigning larger weights for positive training examples than for negative ones, the distances from positive examples to the test example can be reduced, which has the effect of improving the chance of positive examples being selected into the neighborhood, thus rectifying the inherent bias of k NN for negative examples and improving the classification performance of positive examples. As to the weights w_i ($i = 1, 2, \dots, N$) assigned to the training examples, they are obtained by employing a gradient ascend technique to optimize the G-mean [30] metric.

7. Methods Based on Dynamic-Sized Neighborhoods

The methods introduced above construct the neighborhoods with equal size for all the test examples: traditional k NN constructs the neighborhood by finding k nearest training exam-

ples for a test example, while self-adaptive k strategy-based k NN constructs the neighborhood by finding k_p positive neighboring examples and k_n negative neighbors for each test example, and in a classification problem, the number ($k_p + k_n$) of nearest neighbors does not change with respect to test examples. Different from these methods, the methods illustrated in this section construct test example-dependent neighborhood sizes, i.e., for a classification problem, the neighborhood size changes with different test examples. The aim of doing so is to ensure the existence of sufficient positive examples in the neighborhood, and the positive examples in the neighborhood of a test example are all closer to the test example and have similar posterior class probabilities to it. Based on whether parameters are required in determining the dynamic neighborhood size, this kind of method can be divided into two categories.

7.1. Dynamic-Sized Neighborhood k NN with Parameters

7.1.1. Positive-Biased Nearest Neighbor PNN. The Positive-biased Nearest Neighbor (PNN) algorithm is designed in [31] to improve the sensitivity of k NN to positive class. PNN first dynamically constructs the neighborhood of a test example and then adjusts the classification result according to the local class distribution in the neighborhood.

- (1) Constructing the “ $k/2$ Positive Nearest Neighborhood ($k/2$ -PNN)” for each test example to expand the neighborhood for decision making

Compute the distances from test example x_t to all the training examples and rearrange the training examples in ascending order according to their distances, then find the $\lceil k/2 \rceil$ number of positive nearest neighbors of x_t that constitute the neighborhood $S_{k/2}^p(x_t)$. $S_{k/2}^p(x_t)$ usually contains more examples than x_t 's k neighborhood $S_k(x_t)$, i.e., $r = |S_{k/2}^p(x_t)| > k$. As displayed in Figure 6(a), the test example is denoted using symbol “*” and the positive and negative examples are denoted using symbols “+” and “-,” respectively. When $k = 5$, the neighborhood having $\lceil k/2 \rceil = \lceil 5/2 \rceil = 3$ positive training examples contains $r = 8$ examples in total, in this case $r > k$, and an “extended neighborhood” $S_{k/2}^p(x_t)$ is obtained for test example x_t .

- (2) Making decision based on whether “ $k/2$ -PNN” is a positive subconcept

If the ratio of positive examples in $S_{k/2}^p(x_t)$ is much higher than that in the overall training set D , then $S_{k/2}^p(x_t)$ is considered to be a positive class subconcept and $\lceil k/2 \rceil / k > 1/2$ is set as the posterior probability of the test example for the positive class, i.e., x_t is classified as a positive class example. Otherwise, if $S_{k/2}^p(x_t)$ is not a positive subconcept, then the ratio (i.e., $\lceil k/2 \rceil / r$) of positive examples in $S_{k/2}^p(x_t)$ is regarded as the posterior probability of x_t for the positive class; in this case, the probability is usually less than 0.5, i.e., x_t is classified as a negative example.

For instance, if the neighborhood in Figure 6(a) is a positive class subconcept, then the probability of the test example with respect to the positive class is $P(+|x_t) = \lceil k/2 \rceil / k = 3/5 > 1/2$; thus, the classification result of PNN is the positive class; otherwise, the probability is $P(+|x_t) = \lceil k/2 \rceil / r = 3/8 < 1/2$; thus, the test example is classified as a negative example. For the case that there are less than k examples in $S_{k/2}^p(x_t)$ as shown in Figures 6(b) and 6(c), which rarely occurs under the class imbalanced scenario, it indicates that the positive examples are densely distributed around the test example and the corresponding probability is $P(+|x_t) = \lceil k/2 \rceil / r = 3/4 > 1/2$, i.e., the decision of PNN is the positive class. For the case that $S_{k/2}^p(x_t)$ has the same size with k neighborhood $S_k(x_t)$ as shown in Figure 6(d), PNN degrades to k NN.

Experiments in [31] indicate that the simple and effective decision bias of PNN can better classify positive examples, and PNN usually outperforms k -ENN [28] and achieves comparable performance with CCW- k NN [17] mentioned in previous sections. In terms of efficiency, PNN has a much lower computation cost than the two methods that require a “training phase”: (1) k -ENN needs to determine all the exemplar positive examples in the training phase to expand its decision boundary while (2) CCW- k NN computes the weight of each training example by using the mixture model and the Bayesian network. Therefore, both the two methods have a high computation cost. In addition, PNN also outperforms oversampling techniques like SMOTE as well as cost-sensitive strategies like MetaCost.

7.1.2. k Rare-Class Nearest Neighbor Algorithm. k Rare-class Nearest Neighbor (k RNN) is proposed in [32], which has a similar idea with PNN introduced in the previous subsection. k RNN also constructs a test example-dependent dynamic-sized neighborhood and then adjusts the posterior probability of the test example according to the positive examples’ distribution in the extended neighborhood. The differences between k RNN and PNN mainly lie in the following two aspects.

- (1) For test example x_t , its neighborhood constructed by k RNN contains at least k' positive examples, where k' is set to be a constant and takes values 1 or 3 in most cases
- (2) When calculating the test example’s posterior probability of belonging to the positive class, the local and global confidence intervals of the positive class are both utilized, making the obtained probability $P(+|$

$x_t)$ more accurate than the one (i.e., $\lceil k/2 \rceil / k$) obtained by PNN

It is experimentally demonstrated that k RNN significantly improves the classification performance of k NN for positive class and often outperforms the resampling and cost-sensitive strategies employing base classifiers like decision tree and support vector machine.

7.2. Dynamic-Sized Neighborhood k NN without Parameters

7.2.1. Gravitational Fixed Radius Nearest Neighbor (GFRNN). The methods introduced above often have many parameters and complex structure [33], leading to their high time complexity. These methods include the class weighted k NN [12], the examples’ informativeness-based k NN (e.g., LI- k NN [16]), the class confidence-weighted k NN [17], the class conditional nearest neighbor distribution (CCNND) [20] as well as k -ENN [28], and PNN [31] in the previous subsection. In addition, the global information is not fully utilized in these methods.

To overcome these drawbacks, a gravitational fixed radius nearest neighbor (GFRNN) algorithm is proposed in [34], which is inspired by the concept of gravitation in classical dynamics. GFRNN is formed by introducing the “gravitation between example pair” into the fixed radius nearest neighbor method. Concretely, GFRNN operates as follows.

- (1) The distance between each example pair is first calculated, and then, their average value is adopted as the neighborhood radius R of a test example, which is shown in

$$R = \frac{1}{2} N(N-1) \sum_{x_i, x_j \in D} \text{dist}(x_i, x_j). \quad (12)$$

In formula (12), D denotes the training set. Thus, the neighborhood of test example x_t can be described in formula (13), which is constituted by training examples having a distance no more than R with x_t .

$$S(x_t) = \{x_i \mid (x_i \in D) \wedge (\text{dist}(x_i, x_t) \leq R)\}. \quad (13)$$

- (2) Computing the gravitation between each training example in the neighborhood and the test example, which can be achieved using

$$f(x_t, x_i) = G \frac{m_{x_t} m_{x_i}}{d(x_t, x_i)^2}, \quad i = 1, 2, \dots, |S(x_t)|. \quad (14)$$

To simplify the computation, both the gravitational constant G and the mass m_{x_t} of test example x_t are set to 1, where mass m_{x_t} is essentially the weight of test example x_t . Thus, only the mass m_{x_i} of training example x_i in the neighborhood needs to be determined. To balance the effects of the positive

and negative examples in the neighborhood, the mass of each training example is calculated using formula (15), where D_p (D_n) denotes the set of positive (negative) examples in training set D . It is easy to see that the mass (weight) of positive examples in the neighborhood is the ratio between the number of negative examples and that of positive examples in training set D , i.e., the class imbalance ratio IR. In addition, the mass (weight) of negative examples is set to 1 in GFRNN.

$$m_{x_i} = \begin{cases} \text{IR}, & \text{if } (x_i \in S(x_t)) \wedge (x_i \in D_p), \\ 1, & \text{if } (x_i \in S(x_t)) \wedge (x_i \in D_n). \end{cases} \quad (15)$$

- (3) Making a decision according to the gravitation of training examples in the neighborhood to the test example

$$F(x_t) = \sum_{i=1}^{n_{\text{pos}}(x_t)} \frac{\text{IR}}{d(x_t, x_i)} - \sum_{j=1}^{n_{\text{neg}}(x_t)} \frac{1}{d(x_t, x_j)}, \quad (16)$$

where $n_{\text{pos}}(x_t)$ and $n_{\text{neg}}(x_t)$ in formula (16) are, respectively, the numbers of positive and negative examples in the neighborhood $S(x_t)$ of test example x_t . Formula (16) indicates that GFRNN makes a decision in this way: if the gravitation sum of positive examples in $S(x_t)$ is larger than that of the negative examples, then x_t is classified to be a positive example; otherwise, x_t is classified as a negative example.

It can be learned from the above illustration that, in determining the neighborhood for a test example, GFRNN does not require any parameter and only utilizes this *global* information: the average distance among training example pairs. In addition, another *global* information, i.e., the class imbalance ratio IR in the training set, is used as the weight of positive neighbors. To sum up, GFRNN has the following advantage: it can effectively address the class imbalanced problem and does not require the initialization or adjustment of any parameters, which further extends the family of kNN classification algorithms. On the other hand, GFRNN only employs the overall class imbalance ratio IR in the training set to set weights for positive neighbors but does not utilize any *local* information concerning training examples, which can be seen as its disadvantage. To solve this problem, the following two works, respectively, present a solution.

7.2.2. Two Improvement Algorithms for GFRNN

(1) *The First Improvement Algorithm.* An entropy and gravitation-based dynamic radius nearest neighbor (EGDRNN) is proposed in [35]; its differences with GFRNN mainly lie in the following two aspects.

- (a) Neighborhood radius determination

EGDRNN determines the radius of test example x_t 's neighborhood by first computing its average distance

$\text{avgdist}_{D_p}(x_t)$ to the positive examples D_{pos} in training set D and its average distance $\text{avgdist}_{D_n}(x_t)$ to the negative examples D_{neg} in training set D , respectively, and then taking the sum of these two values as the neighborhood radius, which is shown in formula (17).

$$R_{x_t} = \text{avgdist}_{D_p}(x_t) + \text{avgdist}_{D_n}(x_t) = \frac{1}{n_{\text{pos}}} \sum_{i=1}^{n_{\text{pos}}} \text{dist}(x_t, x_i) + \frac{1}{n_{\text{neg}}} \sum_{j=1}^{n_{\text{neg}}} \text{dist}(x_t, x_j). \quad (17)$$

Therefore, the radius determined by EGDRNN for a test example depends on the location of test example with respect to the positive and negative classes and varies with different test examples.

- (b) Weighting strategy for examples in neighborhood

In addition to IR, EGDRNN also introduces the information entropy concept to make examples in different locations have different degrees of importance. Concretely, for training example x_i ($i = 1, 2, \dots, |S(x_t)|$) in neighborhood, EGDRNN computes its information entropy $E(x_i)$ using formula (18). C_1 and C_2 denote the positive and negative classes, respectively; $p(x_i, C_1)$ denotes the probability of example x_i belonging to the positive class while $p(x_i, C_2)$ denotes the probability of example x_i for the negative class, where probability $p(x_i, C_1)$ is calculated using the proportion of positive examples in the k neighborhood of example x_i while $p(x_i, C_2)$ is calculated using the corresponding proportion of negative examples. It can be seen that the smaller the information entropy of x_i the higher the certainty degree of its belonging to a certain class; otherwise, the larger the information entropy, the lower the certainty degree, i.e., it is closer to the decision boundary.

$$E(x_i) = -p(x_i, C_1) \ln p(x_i, C_1) - p(x_i, C_2) \ln p(x_i, C_2). \quad (18)$$

To sum up, for a test example, the gravitation sum of examples in its neighborhood is calculated as follows:

$$F(x_t) = \sum_{i=1}^{n_{\text{pos}}(x_t)} \frac{\text{IR} * E(x_i)}{d(x_t, x_i)} - \sum_{j=1}^{n_{\text{neg}}(x_t)} \frac{E(x_j)}{d(x_t, x_j)}. \quad (19)$$

Formula (19) demonstrates that EGDRNN pays more attention to the positive examples in the neighborhood as well as the examples that are close to the class boundary. Experimental results indicate that EGDRNN not only achieves a high classification accuracy but also has the lowest time cost among the comparison algorithms.

(2) *The Second Improvement Algorithm.* The improvements made in [33] lie in the following aspects. When determining

the weight for neighboring training example x_i ($i = 1, 2, \dots, |S(x_i)|$), in addition to IR, the gravitation from other examples x_j ($x_j \in D, x_j \neq x_i$) to the current example x_i is also considered to calculate its weight. The authors in [30] believe that, for a training example, the larger the sum of gravitation from other examples, the denser its surrounding examples (local information), and they regard such training example as unimportant and assign relatively low mass (weight) for it. In this way, both the *global* example information (i.e., IR) and this *local* information are utilized in determining the weights for neighbors.

8. Conclusion

k NN is a simple and effective base learning algorithm and can achieve comparable classification performance with more complex classifiers such as decision tree and artificial neural networks. However, k NN does not work well on imbalanced data due to the usage of overall classification accuracy as its optimization objective as well as its majority voting-based classification rule. To solve this problem, researchers have conducted many works and proposed a lot of solutions. This paper gives a comprehensive survey of these works according to their adopted perspectives and analyzes and compares their characteristics. What is more, there are still some problems that deserve further study in this field. For instance, we list three of them in the following:

- (1) Most algorithms introduced in this paper mainly consider the case that there is only one positive class in imbalanced data; thus, in the case of two or more positive classes, how to adjust these algorithms to make them work is an important problem
- (2) For the global information-based algorithm GI- k NN, how to improve the robustness of its adopted information metric to noisy training examples needs to be investigated
- (3) For the online streaming data in which the class imbalance degree can change with time, only the class conditional nearest neighbor distribution algorithm CCNND introduced in Section 2 is applicable to this scenario. Thus, more efforts need to be made to make other methods introduced in this paper also suitable to online streaming data

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Natural Science Foundation of Shandong Province (ZR2018QF002).

References

- [1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc, New York, 2006.
- [2] A. Trabelsi, Z. Elouedi, and E. Lefevre, "Decision tree classifiers for evidential attribute values and class labels," *Fuzzy Sets and Systems*, vol. 366, pp. 46–62, 2019.
- [3] G. Villarrubia, J. F. de Paz, P. Chamoso, and F. D. la Prieta, "Artificial neural networks used in optimization problems," *Neurocomputing*, vol. 272, pp. 10–16, 2018.
- [4] B. Richhariya and M. Tanveer, "A reduced universum twin support vector machine for class imbalance learning," *Pattern Recognition*, vol. 102, p. 107150, 2020.
- [5] C. W. Wang and Y. L. Yang, "Nearest neighbor with double neighborhoods algorithms for imbalanced data," *International Journal of Applied Mathematics*, vol. 50, no. 1, pp. 1–13, 2018.
- [6] J. Derrac, S. García, and F. Herrera, "Fuzzy nearest neighbor algorithms: taxonomy, experimental analysis and prospects," *Information Sciences*, vol. 260, pp. 98–119, 2014.
- [7] Y. Lin, J. Li, M. Lin, and J. Chen, "A new nearest neighbor classifier via fusing neighborhood information," *Neurocomputing*, vol. 143, pp. 164–169, 2014.
- [8] C. H. Park and S. B. Kim, "Sequential random k-nearest neighbor feature selection for high-dimensional data," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2336–2342, 2015.
- [9] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.
- [10] Y. Lee, P. J. Hu, T. Cheng, T. C. Huang, and W. Y. Chuang, "A preclustering-based ensemble learning technique for acute appendicitis diagnoses," *Artificial Intelligence in Medicine*, vol. 58, no. 2, pp. 115–124, 2013.
- [11] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Iterative Boolean combination of classifiers in the ROC space: an application to anomaly detection with HMMs," *Pattern Recognition*, vol. 43, no. 8, pp. 2732–2752, 2010.
- [12] H. Dubey and V. Pudi, "Class based weighted k-nearest neighbor over imbalance dataset," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 305–316, Berlin, 2013.
- [13] G. Kovacs, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Applied Soft Computing*, vol. 83, no. 1, p. 105662, 2019.
- [14] P. Kaur and A. Gosain, "Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise," in *ICT Based Innovations.*, pp. 23–30, Springer, Singapore, 2018.
- [15] E. Nashnush and S. Vadera, "EBNO: evolution of cost-sensitive Bayesian networks," *Expert Systems*, vol. 37, no. 3, p. 12495, 2020.
- [16] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, "IKNN: informative k-nearest neighbor pattern classification," in *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 248–264, Berlin, 2007.
- [17] W. Liu and S. Chawla, "Class confidence weighted kNN algorithms for imbalanced data sets," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 345–356, Berlin, 2011.

- [18] S. Ando, "Classifying imbalanced data in distance-based feature space," *Knowledge and Information Systems*, vol. 46, no. 3, pp. 707–730, 2016.
- [19] H. Patel and G. S. Thakur, "A hybrid weighted nearest neighbor approach to mine imbalanced data," in *Proceedings of the 12th International Conference on Data Mining*, pp. 106–110, Athens, 2016.
- [20] E. Kriminger, J. C. Principe, and C. Lakshminarayan, "Nearest neighbor distributions for imbalanced classification," in *Proceedings of The 2012 International joint conference on neural networks (IJCNN)*, pp. 1–5, Brisbane, 2012.
- [21] S. Ramírez-Gallego, S. García, and F. Herrera, "Online entropy-based discretization for data streaming classification," *Future Generation Computer Systems*, vol. 86, pp. 59–70, 2018.
- [22] P. Harshita and T. Singh, "Classification of imbalanced data using a modified fuzzy-neighbor weighted approach," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 1, pp. 56–64, 2017.
- [23] H. Patel and G. S. Thakur, "An improved fuzzy k-nearest neighbor algorithm for imbalanced data using adaptive approach," *IETE Journal of Research*, vol. 65, no. 6, pp. 780–789, 2019.
- [24] S. Liu, J. Zhang, Y. Xiang, and W. Zhou, "Fuzzy-based information decomposition for incomplete and imbalanced data learning," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1476–1490, 2017.
- [25] P. Lim, C. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (ECO-Ensemble) for imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 47, no. 9, pp. 2850–2861, 2017.
- [26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2011.
- [27] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE: majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2012.
- [28] Y. X. Li and X. Z. Zhang, "Improving k nearest neighbor with exemplar generalization for imbalanced classification," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Berlin, 2011.
- [29] Z. Hajizadeh, M. Taheri, and M. Z. Jahromi, "Nearest neighbor classification with locally weighted distance for imbalanced data," *International Journal of Computer and Communication Engineering*, vol. 3, no. 2, pp. 81–86, 2014.
- [30] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineerin*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [31] X. Z. Zhang and Y. X. Li, *A positive-biased nearest neighbour algorithm for imbalanced classification*, Proc of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin, Springer, Heidelberg, 2013.
- [32] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, "KRNN: k rare-class nearest neighbour classification," *Pattern Recognition*, vol. 62, pp. 33–44, 2017.
- [33] B. Nikpour, M. Shabani, and H. Nezamabadi-pour, "Proposing new method to improve gravitational fixed nearest neighbor algorithm for imbalanced data classification," in *2017 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 6–11, Kerman, March 2017.
- [34] Y. J. Zhu, Z. Wang, and D. Q. Gao, "Gravitational fixed radius nearest neighbor for imbalanced problem," *Knowledge-Based Systems*, vol. 90, pp. 224–238, 2015.
- [35] Z. Wang, Y. Li, D. Li, Z. Zhu, and W. Du, "Entropy and gravitation based dynamic radius nearest neighbor classification for imbalanced problem," *Knowledge-Based Systems*, vol. 193, p. 105474, 2020.

Research Article

A Novel Pricing Mechanism for User Coalition in Blockchain

Yi Jiang ¹, Xu Liu ^{1,2} and Jun Dai¹

¹School of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225127, China

²School of Business, Victoria University, Melbourne VIC 3011, Australia

Correspondence should be addressed to Xu Liu; 1438239097@qq.com

Received 21 September 2020; Revised 15 October 2020; Accepted 23 October 2020; Published 28 November 2020

Academic Editor: Junwu Zhu

Copyright © 2020 Yi Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the blockchain platform is widely used as a new trading way, both participants and transaction volume in the blockchain projects have been growing by leaps and bounds. The generic mechanisms of ranking transaction priorities are heavily dependent on the transaction fees the users append into each transaction; then, all transactions are ranked in the nonincreasing order according to the transaction fee amounts, and the selected transactions will be packed into a new created block in order based on the ranking results. However, more complex influence factors more than transaction fees on transaction priority ranking results are not taken into consideration in the generic transaction priority ranking mechanisms, and a single user is the objective to create transactions in these mechanisms. In order to optimize the generic transaction priority ranking mechanisms and enrich transaction creation modes, a novel user-coalition-based transaction pricing mechanism (UCTPM) is proposed, and the user coalition quality score, user coalition contribution degree, and the transaction type demand degree are formulated and introduced into the UCTPM mechanism. Our research findings indicate that the UCTPM mechanism satisfies the economic attributes of budget balanced, individual rationality, and incentive compatibility when the user coalition contribution degree increases through theoretical proof and experimental analysis. Moreover, the UCTPM mechanism allows all the transactions to be processed more efficiently by experimental analysis.

1. Introduction

In contrast to the traditional financial systems that heavily rely on centralized supervision, digital cryptocurrencies use blockchain as the technical solutions providing a decentralized distributed ledger economic model for the current financial market and enabling participants to freely join this kind of open-market platform without constraints. Due to various excellent characteristics of blockchain technologies such as data immutability, information traceability, and node pseudonyms, more participants have enjoyed such new transaction methods provided by blockchain platforms. [1, 2]

With the emergence and constant improvement of various blockchain projects such as Bitcoin and Ethereum, the digital cryptocurrencies issued by different blockchain projects have attracted a large number of participants to contribute to the computational power in mining activities which is to compete for being the first valid participant solving the proof-of-work-based cryptographic puzzle then to receive a certain amount of lucrative cryptocurrency as a mining

reward. In detail, among the blockchain projects where the consensus process is based on proof of work (PoW), new blocks are normally generated by the participants called miners who continuously perform massive hash calculations to obtain the full solution for the hash puzzles, and the miner who firstly gains the full solution is granted the right to add these selected transactions into a new created block; then, he or she can get the transaction fees as extra mining profit in addition to the fixed block reward once the new block is committed as valid to the main chain [3, 4].

During the transaction confirming process, all the participants can create transactions; when a node firstly generates one transaction and broadcasts it to other nodes within the whole blockchain network, a certain amount of transaction fee is needed to append along with the generated transaction (even though the transaction fee is not enforced to append along with the transaction); once the transaction is verified to be valid, then it will be added into the mempool waiting in queue to get processed by the miner who wins the mining competition; then, some transactions are selected to append

into the new created block based on certain rules (generally from the highest transaction fee down to the lowest fee). Once the full solution of the cryptographic puzzles is found, the newly mined block is added to the main chain and propagated through the network. The transaction fee, as an economic incentive method of blockchain, is one of the important and effective approaches to guarantee nodes of participation in mining activities and transaction verification during the whole transaction confirming processes. Among the generic methods of transaction selection, miners normally prefer these transactions with higher fees to append into a new block in order to optimize the total profit while mining; as a result, users are stimulated to increase the transaction fees to some extent with the purpose of getting transactions to be confirmed as soon as possible [5]. Moreover, users can perform strategic bidding by observing and analysing the dynamic status of transaction queuing states in the mempool to estimate the transaction threshold fee (that is, the lowest transaction fee among all the transactions is being added into the new block), thus fluctuating the overall profit of miners and affecting the efficiency of blockchain network transaction confirming processing as consequences [6].

As it has been considered about the continuous improvement and widespread use of smart contracts in blockchain, the transaction types have been expanded. In the current PoW-based blockchain projects, the common transaction types can be divided into two main categories as token transfer class and smart contract operation class. In detail, the token transfer class covers transactions with different amounts of token and lock-time settings (that is, the expected time for the transaction to be added into a new block), while the smart contract operation class includes calling, updating, and creating. Beyond these transaction types above, participants also can convert the main chain tokens into the side chain tokens in order to bring more functions for the whole network under the applied advanced smart contracts; therefore, the shifting evolution of smart contracts results in a wider variety of transaction types [7, 8]. With all the nodes involved in the blockchain network are the behaviour subjects who are involved in transaction verification, transaction creation, mining reward competition, and other activities via the blockchain platform, and all the historical activity information is recorded in the relevant code fragment in the blockchain; this makes it possible to use such activity information as a basis to measure the quality and contribution of each node in the network.

In addition, for most mechanisms used in the transaction confirming process in the blockchain, incentive compatibility is of great importance for any participants to behave truthfully when bidding the transaction fees, and the rare resource in the transaction confirming process is the number of transactions which can be packed successfully during each round of mining. In this paper, we strive to study the factors affecting the transaction confirming process, and we declare that the participants who are involved in the blockchain network are user, user coalition, and miners for simplification.

As the fast development and application of blockchain technologies and the widespread usage of the digital economic system with such a technical foundation, a mechanism

where participants are encouraged to join more activities and behave honestly is necessary for the blockchain market [9]. Based on the analysis above, we proposed a user-coalition-based transaction pricing mechanism (UCTPM); this mechanism consists of two algorithms: user coalition weighted transaction priority (UCWTP) algorithm and transaction priority rank-based pricing (TPRP) algorithm, and each of the algorithms corresponds to transaction priority determination and pricing for the winning user coalitions, respectively. Although there are several researches that bring the transaction size, historical transaction volume, and user waiting costs to calculate the transaction priority results, the calculation modes of these factors are not given in the general second price auction model. [10] It needs to be emphasized that, in the UCWTP mechanism, the user coalition quality score, coalition contribution degree, and transaction type demand degree are formulated in detail and introduced to optimize the transaction priority algorithms that only focus on the single transaction fee. In addition, we consider that the participants join the user coalitions as the transaction creation mode, which is fairly a novel transaction creation mode, and we believe that the main advantage to consider such a new transaction creation mode is that the efficiency of the transaction confirming process can be improved due to the transactions that are selected by batch instead of by individual, also it is verified via the experiments conducted in this paper.

Whereafter, it is also proved through theoretical proof that the UCTPM mechanism proposed in this paper satisfies the economic attributes of budget balanced, individual rationality, and incentive compatibility under the circumstance that the user coalition contribution degree increases. Besides, it shows that the UCWTP algorithm in the proposed UCTPM mechanism can help to reduce the impact of a single transaction fee on transaction priority ranking by an average of 5.17% when compared to the traditional transaction priority algorithm through experimental analysis.

Our contribution can be summarized as the following three:

- (1) Design a new UCTPM mechanism to optimize the traditional transaction priority algorithms by introducing the user coalition quality score, coalition contribution degree, and transaction type demand degree
- (2) Adopt user coalition transaction creation model instead of an individual transaction creation model to enrich the transaction creation modes for the blockchain
- (3) Prove that nodes under the UCTPM mechanism can be effectively stimulated to participants in the activities such as transaction creation, transaction verification, and computing power competition for bookkeeping right, so as to improve the contribution degree and quality score

The rest of the paper is structured as follows: Section 2 gives a brief review of related work. Section 3 defines the basic

concepts that are used in the UCTPM mechanism model. Section 4 describes the problem details. Section 5 formally establishes the models of the UCWTP algorithm and the TPRP algorithm that both consisted of the UCTPM mechanism. Section 6 presents the detailed algorithms of UCWTP and TPRP and theoretically verifies that the UCTPM mechanism satisfies the economic attributes of individual rationality, budget balanced, and incentive compatibility as the user coalition contribution degree increases. In Section 7, the proposed UCWTP algorithm can effectively help to reduce the impact of the single transaction fee on transaction priority through the experimental analysis, thus motivating user coalition to actively join the activities in the blockchain platform, and the transaction confirming process time is also reduced under the user coalition transaction creation mode. Finally, this paper is summarized and some future work is discussed in Section 8.

2. Related Work

During the initial stage of the Bitcoin project, users need to provide the default transaction fee, and miners are required to process the transactions according to the system settings even for the zero fee transactions, whereas users prefer to offer a higher transaction fee to attract miners in order to reduce the transaction processing waiting time with the growth of transaction volume and users [11]. As recorded in the official document of Bitcoin, the transaction fees are not required for some certain types of transactions though the zero transaction fees cannot be financially sustainable with the development of Bitcoin [12].

As the emergence of transaction fees has led to the transformation of blockchain from a simple structure of mining to a market-based ecological structure, some researches and methodologies have been studied about the influence factors on transaction fees. The game model is established in Kasahara and Kawahara's research to analyse the impact of factors on transaction fees including the transaction flow-in rate, flow-out rate, and user waiting time [13]. Moreover, several mechanisms such as providing a fixed transaction fee to users and miners and the Vickrey-Clarke-Groves (VCG) auction-based pricing model are compared to analyse the equilibrium transaction fee, due to the fee that essentially depends on the cost of involving in the infrastructure for miners to compete in mining [14]. However, the fixed transaction fee is infeasible in the long run according to the static partial equilibrium model that is designed for the economic analysis of the transaction fee [15].

In addition, the Bitcoin market attribute of incentive compatibility is deeply studied, and the monopoly pricing model and random sampling pricing model are established with the purpose of avoiding user utility improvement by lying the transaction fees, and the results of theoretical verification show that the monopoly pricing mechanism is near incentive compatibility under the condition of enough users and transaction volume [16].

The researches on transaction fees introduced above are more focused on the user perspective, but the transaction fees are crucial to miners as well. Transaction fees as the extra

reward are the economical incentive to sustain mining operations [17]. Furthermore, when miners cannot profit from the mining process, they may switch from the current blockchain platform to another one for mining [18]. Since the past few years that Bitcoin was invented, the fixed block reward plays a much more important role in the whole mining revenue system; the average revenue is approximately 14 BTC per block with the fixed block generation reward 12.5 BTC in 2017; however, the situation starts to change somehow with the decreasing fixed block reward per 4 years, and the highest overall revenue is about 19.79 BTC per block; all the changes indicate that the transaction fees start to play a key role in the blockchain market and can affect the miner's revenue to a certain extent [19, 20].

Nevertheless, transaction fees not only are an important part of the blockchain market economy that determines the revenue of both users and miners but also have great influence on transaction priority ranking results which are the key basis for the transaction confirming process. Under different priority rules, the user transaction fee decisions are various [21]. Moreover, users prefer to increase the transaction fees for the higher transaction priority in the transaction confirming process where only transaction fees directly affect priority rank results; meanwhile, users with lower transaction fees are discouraged to create transactions due to prolonged waiting time, thus leading to negative influence on the blockchain development [22, 23].

3. Concept Definition

Definition 1 (user coalition). Any node who creates transactions via the blockchain platform is called a user, and such a node can be a miner as well. User coalition is a group with n ($n > 1$) users to create transactions, and in writing, the user coalition is set as CL_k , $k \in CL_k = 1, 2, \dots, k$. In particular, any users in the user coalition create their own transactions severally and form a transaction set as the coalition's transaction set.

Definition 2 (miner). The nodes who contribute to the computational power in the mining process are called miners. And there is only one miner m who will obtain the transaction fees and block rewards during each round of mining competition; moreover, the transactions of winning user coalition are processed by m .

Definition 3 (user coalition utility). Denote that the utility of user coalition k ($\forall k \in CL_k$) is $u_{CL_k} = (V_k - p_k)$; that is, the utility is the difference value between the overall valuation of the transactions and actual payment of any k .

Definition 4 (individual rationality). If there is no loss of profit for any k who participates in the mechanism, that is, $u_{CL_k} \geq 0$ ($\forall k \in CL_k$), this demonstrates that the mechanism satisfies the attribute of individual rationality.

Definition 5 (budget balanced). If all the income of the mechanism comes from the total payment of winning user

coalition without any external personnel, and the expenditures of the mechanism are all used to pay the miner who processes the transaction without any value transfer, that is, $\sum_{k \in W} p_k = \text{pay}_m$, this illustrates that the mechanism satisfies the attribute of budget balanced, and the attribute of budget balanced ensures that there is no value to be transferred in or out from the mechanism.

Definition 6 (incentive compatibility). Denote the truthful bid as b_k , the lying bid as b'_k for $k (\forall k \in \text{CL}_k)$ when $k (\forall k \in \text{CL}_k)$ bids for the transaction set, and the corresponding utilities are u_{CL_k} and u'_{CL_k} , respectively. The mechanism is incentive compatible if one of the following two conditions is true: (1) it satisfies $\forall k \in \text{CL}_k, b_k > b'_k \longrightarrow u_{\text{CL}_k} \geq u'_{\text{CL}_k}$ when k lowers the transaction fee bid, or (2) there exists a strategy α_k for k to increase the transaction fee bid and strategy α_k^* for k to increase the quality score, when it satisfies $\forall k \in \text{CL}_k, u(\alpha_k^*, \alpha_{-k}^*) \geq u(\alpha_k, \alpha_{-k}^*)$ under the conditions that k 's contribution degree increases. Both of the conditions indicate that any k cannot improve the utility by lying the transaction fee bid. Moreover, the attribute of incentive compatibility indicates that k satisfies the current transaction fee bid under certain conditions, so that the incentive of the lying transaction fee bid is deprived.

4. Problem Description

In this section, we formalize the transaction confirming problem in detail to lay the foundation for the establishment of the UCWTP mechanism, and the basic parameters involved in the problem are listed in Table 1.

Assume there are n nodes in the blockchain network and denote U as the set of nodes, written as $U = \{1, 2, \dots, n\}$. And there are two types of roles involved in the transaction confirming process, respectively, miner m who wins the mining competition during the current round and user coalition CL_k , and $\text{CL}_k \in U, m \in U$.

According to the real situation of blockchain, there is only one m who is verified to be valid for packing transactions into a new created block to gain the transaction fees during each round of mining competition. Because the user coalition k 's actual payment is affected by the discount factor in different priority ranks, in order to ensure the profit of miner m , m can set the unit reserve price c_m for transactions packed into a new block according to the actual mining cost. User coalition transaction volume, estimated ranking discount, and c_m are reported to the mechanism by m .

Assume there are $|\text{CL}_k|$ user coalitions in the whole network and each user coalition is required to submit the transaction report which contains transaction information including transaction fee bid set $B_k = [b_1, b_2, \dots, b_k]$ and transaction size set $S_k = [s_1, s_2, \dots, s_k]$. Denote the report submitted by user coalition CL_k as a triple $\text{Req}_k = (V_k, B_k, S_k)$, where $|B_k|$ represents the total volume of transactions in the set due to the arbitrary bid corresponding to a certain transaction, and V_k represents k 's valuation on the transactions, and V_k also can represent the maximum price k is willing to pay:

TABLE 1: Basic parameters used in this paper.

Notation	Explanation
CL	Set of user coalition
α_k	Transaction type demand
B_k	Transaction fee bid from user coalition
φ_k	User coalition quality score
V_k	Transaction valuation from user coalition
con_k	User coalition contribution degree
P_k	User coalition actual payment
u_{CL_k}	User coalition utility
S_k	Transaction set size
θ_k^j	The discount factor of k in rank j

$$V_k = \begin{cases} [v_1, v_2, \dots, v_n], & S_k \neq 0, \\ 0, & S_k = 0. \end{cases} \quad (1)$$

The blockchain transaction confirming processing is shown in detail in Figure 1; firstly, users join the coalition and report the transaction information to the UCTPM mechanism; miner m who wins the mining competition reports his/her own information as well; then, the transaction priority ranking results are available according to the UCTPM mechanism. The winning user coalition needs to make the payment according to the priority ranking results; thus, the essence of the UCWTP algorithm can be seen as to determine the winner set whose transaction sets are packed into a new block. Lastly, m gains the total transaction fees paid by winning user coalitions and broadcasts the new block to the network.

5. Design of UCTPM Mechanism

Our UCTPM mechanism consists of the UCWTP model and the TPRP model. The UCWTP model determines the winner determination problem, that is, determining whose transaction sets are packed into the new block, while the TPRP model is used for the actual payment of winning user coalitions.

5.1. UCWTP Transaction Priority Model. Assume that the arbitrary user is involved in the blockchain participant in the activities such as transaction creation, transaction verification, and extending the main chain, and these historical activities are recorded, so that such information can be used as a basis for calculating the quality score of the user coalition. Besides, the user coalition contribution degree and valid transaction volume are also considered the influencing factors for the user coalition quality score in our model, and the quality score calculation model is as follows:

$$\varphi_k = \left(\phi_0 + \text{con}_k \sum_{m=1}^n (|tx_k m|) \right)^\mu. \quad (2)$$

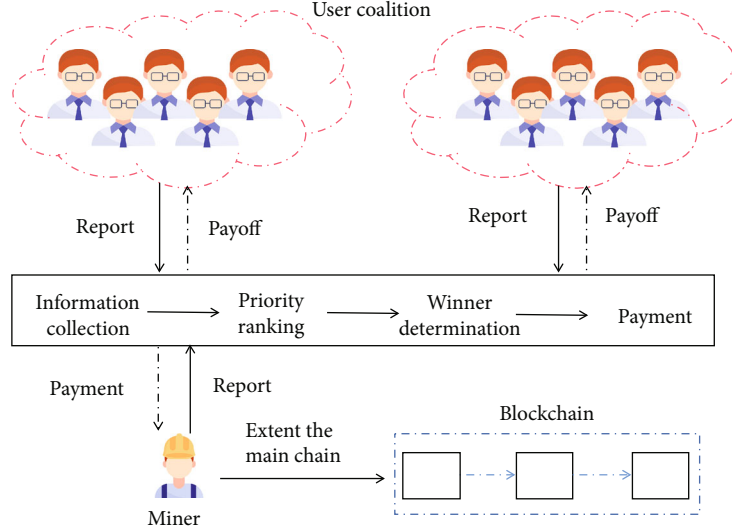


FIGURE 1: Basic transaction process diagram of blockchain.

In equation (2), ϕ_0 is the initial quality score for arbitrary user coalition k and ϕ_0 is a constant value. $|tx_k m|$ represents the valid transaction volume of k during the transaction confirming processing for the current round (for example, the n th round), which can be divided into two parts precisely: firstly, the total volume of transactions that have been recorded into the main chain during the $n-1$ rounds; secondly, the verified transaction volume for current round n . And μ is the constrained parameters where $0 < \mu < 1$. In addition, con_k is k 's contribution degree and can be formulated as

$$\text{con}_k = \frac{1}{n} \left(\sum_{m=1}^n \Omega_{km} \times (P_{km} + 1) \right). \quad (3)$$

In equation (3), Ω_{km} represents the number of transactions verified by k at the m th round of transaction confirming processing, and denote $P_{km} = \{0, 1\}$ to indicate whether there are users in the coalition who wins the mining competition at the m th round, where $P_{km} = 1$ means there is such a miner, $P_{km} = 0$ otherwise.

Except for user coalition quality ϕ_k which can affect the results of transaction priority ranking, the different transaction types created by k should influence the ranking results as well. User coalition can implement different functions and achieve various purposes by creating various transactions, and we roughly divide the common transaction types into five categories: transactions with or without lock-time, also with the smart contract operation, creation, and other transaction types:

- $\text{type}_t \leftarrow \text{type}_1$, if transaction with lock-time,
- $\text{type}_t \leftarrow \text{type}_2$, if transaction without lock-time,
- $\text{type}_t \leftarrow \text{type}_3$, if creation of smart contract,
- $\text{type}_t \leftarrow \text{type}_4$, if operation of smart contract,
- $\text{type}_t \leftarrow \text{type}_5$, if others.

Considering that the volume of a certain transaction type in the total volume that transactions are recorded on the main chain can reflect the demand degree of this certain type of transaction, thus we formula the transaction type demand degree based on type_t for arbitrary user i from k as

$$\gamma_i = \left(1 + \frac{\sum |\text{type}_t|}{\text{Num}} \right)^{1/\varepsilon}. \quad (4)$$

In equation (4), type_t reflects the total volume of this certain transaction type recorded on the main chain, Num represents the overall transaction volume on the main chain, and parameter $1/\varepsilon$ help to prevent the boost increase of certain transaction types in the blockchain network from severely impacting transaction priority ranking results. Because we consider the scenario that users join the coalition for transaction creation, the transaction type demand degree for k is developed on the average value of equation (4) as shown below:

$$\alpha_k = \frac{\sum_{i \in \text{CL}_k} \gamma_i}{|\text{CL}_k|}. \quad (5)$$

The methods that used to reduce the impact of transaction fee bid B_k on priority ranking results should not only focus on the introduction of user coalition quality score ϕ_k but also consider the impact of B_k on the transaction priority ranking results which should be gradually decreased when B_k increases under the condition that user coalition contribution degree con_k keep increasing. Therefore, we adopt a new method to calculate the transaction fee bid as the following:

$$b'_k = B_k \times e^{-(\sigma \text{con}_k)}. \quad (6)$$

The parameter $\sigma (0 < \sigma < 1)$ is used to restrain the abrupt attenuation of b'_k when con_k increases. In addition,

only the transaction sets that meet the requirement $B_k/|B_k| \geq c_m$ can be processed in order to guarantee the profit of m , where c_m is unit reserve price for m . In consequence, we proposed the UCWTP calculation model based on equations (2), (5), and (6):

$$B'_k = \frac{b'_k \times \varphi_k \times \alpha_k}{S_k}. \quad (7)$$

The new transaction priority ranking results for user coalition can be sorted as $B'_1 \geq B'_2 \geq \dots \geq B'_k$ according to equation (7), and the corresponding priority rank is $r_1 \geq r_2 \geq \dots \geq r_k$. Meanwhile, m is supposed to pack the transaction sets into a new created block by the rank from high to low until the block size cap. And denote $x_{ck} = \{0, 1\}$ to indicate the transaction set packing results, $x_{ck} = 1$ means the transaction set is packed into the new block, and $x_{ck} = 0$ otherwise, so that the winning user coalition set can be denoted as $W = \{\forall k \in CL_k \mid x_{ck} = 1\}$.

5.2. TPRP Pricing Model. The user coalition can obtain different transaction priority ranks by the comprehensive results of transaction fee bid, contribution degree, quality scores, and transaction type demand degree, and the actual payment of the winning user coalition based on various ranks is different, which also should be irrelevant to the user coalition transaction fee bid; otherwise, the user coalition will always have the incentive to lie the transaction fee under the situation of becoming the winner. Thus, we propose the TPRP pricing model that is based on the transaction priority ranks and on the purpose of depriving the correlation between payment and user coalition owning transaction fee bids:

$$p_k = \theta_k^j \times |B_k| \times c_m. \quad (8)$$

In equation (8), $|B_k|$ is the volume of transactions created by user coalition, and θ_k^j is the discount factor of k when k ranks j , and also, $\theta_k^j \leq \theta_k^g$ when $r_j \geq r_g$:

$$\theta_k^j = \beta \times r_k^j \longrightarrow (0, 1]. \quad (9)$$

The transaction set of winning user coalition is processed by m ; then, m broadcasts the new block to the whole blockchain network after the selected transaction sets are packed into the new generated block; then, m receives the total transaction fees from winning user coalitions as a reward:

$$\text{pro}_m = - \sum_{k \in W} p_k \quad (10)$$

The winning user coalition will also be profit after the new block is considered to be valid as it appended onto the main chain; therefore, the expected revenue V_k is obtained by k , and the utility of such a user coalition can be formulated as

$$u_{CL_k} = xc_k(V_k - p_k). \quad (11)$$

6. Algorithm and Attribute Proof

6.1. UCWTP Algorithm and TPRP Algorithm. The UCTPM mechanism proposed in this paper mainly includes two parts: the transaction priority ranking algorithm and the winning user coalition pricing algorithm. The essence of the transaction priority ranking algorithm is used for winner determination so that the specific miner needs to pack the selected transaction sets into the new created block, and the transaction priority ranking algorithm is called the UCWTP algorithm, while the TPRP algorithm is the solution to the winning user coalition payment issue.

Algorithm 1 describes the process of winner determination, and we set each round of priority ranking process as static; that is, once the transaction sets are selected for the transaction priority ranking process, these sets are not affected by any changes due to new transaction sets arriving in the mempool. First of all, Steps 2 to 5 indicate that the transaction priority ranking result of the selected user coalition is calculated based on the coalition quality score, transaction type demand degree, transaction fee, and contribution degree; Step 7 shows that the new transaction sequence is sorted in a nonincreasing order; finally, Steps 8 to 11 determine the winning user coalition sets according to the priority ranking result and the block capacity limitation.

Algorithm 2 describes the pricing process for the winning user coalition; Steps 2 to 6 illustrate that these user coalition needs to make the actual payment according to the transaction priority ranking result and the transaction volume, so that the user coalition actual payment is irrelevant to user coalition own transaction fee bid. In addition, the lower the rank is, the higher the user coalition payment would be due to the discount factor; nevertheless, when the user coalition wants to lower the actual payment by reaching the higher transaction priority rank, the user coalition will prefer to improve the unit quality score instead of the unit transaction fee. As a result, it deprives the motivation for the user coalition to improve utility by lying the transaction fee bid.

6.2. Attribute Proof

- (1) If the lower bid b'_k results in the failure of the transaction set being packed into the new block, then $xc_k = 0$ according to equation (11); thus, $u'_{CL_k} = 0$
- (2) If the lower bid b'_k results in lower transaction priority rank r_j , then $\theta_k^j \geq \theta_k^g$. And the user coalition actual payment only depends on the transaction priority rank and the transaction volume and is irrelevant with the transaction fee bid; that is, when $\theta_k^j \geq \theta_k^g$, there exists $p_k^j \geq p_k^g$, so that, $u_{CL_k} - u'_{CL_k} = V_k - p_k^g - (V_k - p_k^j) = p_k^j - p_k^g \geq 0$

Input: Reports submitted by user coalition: $R_{cl} = \{r_1, r_2, \dots, r_n\}, r_n = \langle V_k, B_k, S_k \rangle$
 Reports submitted by miner: $R_m = \langle c_m \rangle$
Output: W : the sets of winning user coalition

- 1 **initialize** $sum \leftarrow 0, W \leftarrow N, L \leftarrow \emptyset \leftarrow$ maximum size of a block
- 2 **for all** $k \in W$ **do**
- 3 $b'_k = B_k \times e^{-(\sigma con_k)}$
- 4 $\varphi_k = (\phi_0 + con_k \sum_{m=1}^n (|tx_k m|)^\mu)$
- 5 $B'_k \leftarrow (b'_k \times \varphi_k \times \alpha_k / S_k)$
- 6 **end for**
- 7 **sort** W in non-increasing order of $B'_k: B'_1 \geq B'_2 \geq \dots \geq B'_k$
- 8 **for all** $i \in W$ **do**
- 9 **if** $sum \leq C$ **then**
- 10 $sum + = S_i$
- 11 $W \leftarrow W \setminus \{i\}, L \leftarrow L \cup \{i\}$
- 12 **end if**
- 13 **end for**
- 14 $W \leftarrow \emptyset$
- 15 $W \leftarrow L$
- 16 **return** W

ALGORITHM 1: UCWTP.

Input: Reports submitted by user coalition: $R_{cl} = \{r_1, r_2, \dots, r_n\}, r_n = \langle V_k, B_k, S_k \rangle$
 Reports submitted by miner: $R_m = \langle c_m \rangle$
 W : the sets of winning user coalition

Output: p the payment sets of winning user coalition

- 1 **initialize** $p \leftarrow \emptyset$
- 2 **sort** W in non-increasing order of $B'_k: B'_1 \geq B'_2 \geq \dots \geq B'_k$
- 3 **for all** $i \in W$ **do**
- 4 $p_i = \theta'_i \times |B_i| \times c_m$
- 5 $p \leftarrow p \cup p_i$
- 6 $W \leftarrow W \setminus \{i\}$
- 7 **end for**
- 8 **return** p

ALGORITHM 2: TPRP.

Theorem 9. *Arbitrary user coalition k cannot improve utility by lowering the transaction fee bid in the UCTPM mechanism.*

Proof. Assume when the user coalition k bids the transaction fee b_k to rank r_g and obtains the utility u_{CL_k} , while the lower bid b'_k to rank r_j and obtains the utility u'_{CL_k} .

Therefore, the conclusion can be reached through the above analysis that arbitrary user coalition k cannot improve utility by lowering the transaction fee bid in the UCTPM mechanism.

Theorem 10. *Arbitrary user coalition k prefers to improve unit quality score instead of unit transaction fee bid to improve utility when con_k increases.*

Proof. The user coalition quality score is calculated according to equation (2): $\varphi_k = (\phi_0 + con_k \sum_{m=1}^n (tx_k m)^\mu)$.

When applied substitution as $TX_k \leftarrow \sum_{m=1}^n (|tx_m k|)$, then we can get $\varphi_k = (\phi_0 + con_k \times TX_k)^\mu$. And deriving φ_k to get $\varphi'_k(con_k) = \mu TX_k$. As φ_k is the monotone increasing function of con_k due to $|tx_k m| \geq 0$ and $\mu \geq 0$, thus $\mu TX_k \geq 0$, so that $\varphi'_k(con_k) \geq 0$.

Besides, the effect of user coalition contribution degree con_k on transaction fee bid B_k can be written as $b'_k = B_k \times e^{-(\sigma con_k)}$ according to equation (6).

Same as above, if $B \leftarrow b'_k$, then $B = B_k \times e^{-(\sigma con_k)}$ stands, and deriving B to get $B'(con_k) = -\sigma B_k e^{-(\sigma con_k)}$. For arbitrary k 's transaction fee bid $B_k > 0$, and $con_k \geq 0$, $\sigma > 0$, thus $B'(con_k) \leq 0$, so that, b'_k is the monotone decreasing function

of con_k . And the inequation $\varphi'_k(\text{con}_k) \geq B'(\text{con}_k)$ shows that the priority ranking that arbitrary user coalition obtains through improving the unit quality score is higher than improving unit transaction fee bid.

Denote the strategy set of user coalition k as $A_k = \{\alpha, \alpha'\}$, where α means the user coalition's strategy of improving the unit quality score, while α' means the strategy of improving the unit transaction fee bid. We only consider the pure strategy action instead of mixed strategy action for simplification; that is, k only chooses one of specific strategies from strategy set A_k . Denote the utility is u_{CL_k} when k chooses the strategy of α to obtain the priority rank r_g , while utility is u'_{CL_k} when k chooses the strategy of α' to obtain the priority rank r_j , thus $r_g \geq r_j$ based on the analysis above. Considering comprehensively with equation (8), it shows that $p'_k \geq p_k^g$ when $\theta_k^j \geq \theta_k^g$, so that $u_{\text{CL}_k} - u'_{\text{CL}_k} = V_k - p_k^g - (V_k - p_k^j) = p_k^j - p_k^g \geq 0$, which means arbitrary user coalition will not choose strategy of α' to improve the utility. According to the analysis above, the conclusion can be summed up that arbitrary user coalition k cannot improve utility by improving the transaction fee bid in the UCTPM mechanism.

- (1) When $b'_k < b_k$, according to Theorem 9, arbitrary k has the utility $u_{\text{CL}_k} - u'_{\text{CL}_k} \geq 0$, which means k cannot improve the utility through lowering transaction fee bid
- (2) When $b'_k > b_k$, according to Theorem 10, for arbitrary k prefers to improve the unit quality score rather than unit transaction fee bid for the higher priority rank to improve utility, which means there is no such a k that chooses the strategy of improving unit transaction fee for extra utility when the coalition contribution degree increases

Theorem 11. *The UCTPM mechanism satisfies the attribute of incentive compatibility when con_k increases.*

Proof. Denote the utility of user coalition k is u_{CL_k} when k bids transaction fee b_k , while u'_{CL_k} when k bids transaction fee b'_k .

From the analysis of (1) and (2), it can be proved that the UCTPM mechanism satisfies the attribute of incentive compatibility when con_k increases.

- (1) When $xc_k = 1$, which means the transaction set for this user coalition is added into the new block, thus $u_{\text{CL}_k} = V_k - p_k$. And user coalition k 's actual payment p_k is not higher than B_k based on equations (8) and (9), so that $p_k \leq B_k$. In addition, $V_k \geq B_k$ holds for users who use blockchain as the transaction platform; otherwise, they prefer the

off-chain methods. Therefore, the utility of k is $u_{\text{CL}_k} = xc_k(V_k - p_k) = V_k - p_k \geq 0$

- (2) When $xc_k = 0$, which means the transaction set of k is not appended onto the main chain, and also, there is no need for such a k to pay, so that $u_{\text{CL}_k} = 0$

Theorem 12. *The UCTPM mechanism satisfies the attribute of individual rationality.*

Proof. The user coalition k 's utility is written as $u_{\text{CL}_k} = xc_k(V_k - p_k)$ according to equation (11).

The conclusion can be summarized from the analysis above that the UCTPM mechanism is individual rational according to analyses (1) and (2).

Theorem 13. *The UCTPM mechanism satisfies the attribute of budget balanced.*

Proof. According to the payment function defined by the UCTPM mechanism for winning user coalition and miner, the sum of all participants in the mechanism is zero; that is, $\text{pro}_m + \sum_{k \in W} p_k = 0$; thus, the UCTPM mechanism satisfies the attribute of budget balanced.

7. Experiment

The simulation experiment codes are implemented in Python language, and the test hardware is Intel® Core™ i5-4200U CPU+4 GB memory, and Windows 10+Python 3.7+Pycharm 2019 are used for the test software. We conduct the simulation experiments to mainly explore two contents based on the researches in this paper: Firstly, the transaction priority ranking results influenced by the UWCTP algorithm are explored. Secondly, the influences of the UCTPM mechanism on transaction processing time are explored. The experimental results show the feasibility and effectiveness of the UWCTP mechanism, and the detailed experimental contents are as follows:

- (1) Exploring the influence of the increasing transaction fee and valid transaction volume on transaction priority scores as user coalition contribution degree increases
- (2) Observing the influence differences of the UCWTP algorithm and traditional transaction priority algorithm on transaction priority ranking results
- (3) Exploring the transaction processing time differences caused by different transaction creation modes (transactions created by a single user and user coalition) under different transaction volumes in the mempool

In order to explore the influence of the increasing transaction fee bid and valid transaction volume on priority results, we generate the transaction fee within the range of

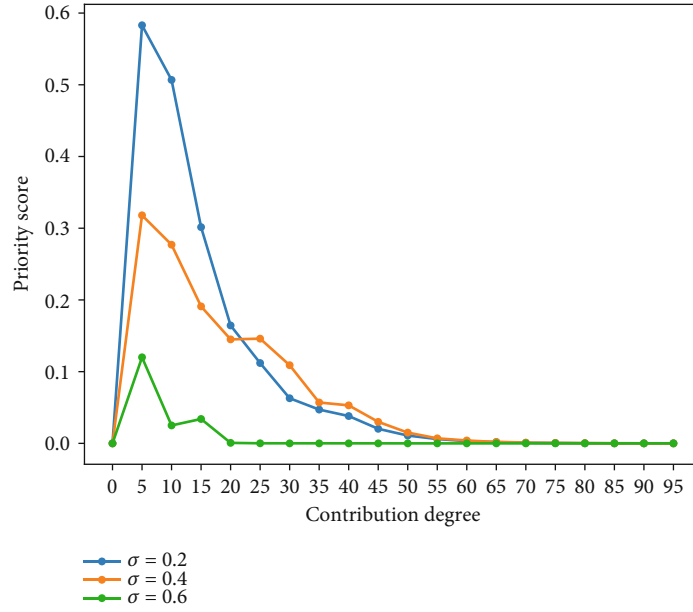


FIGURE 2: Influence of transaction fee on priority score.

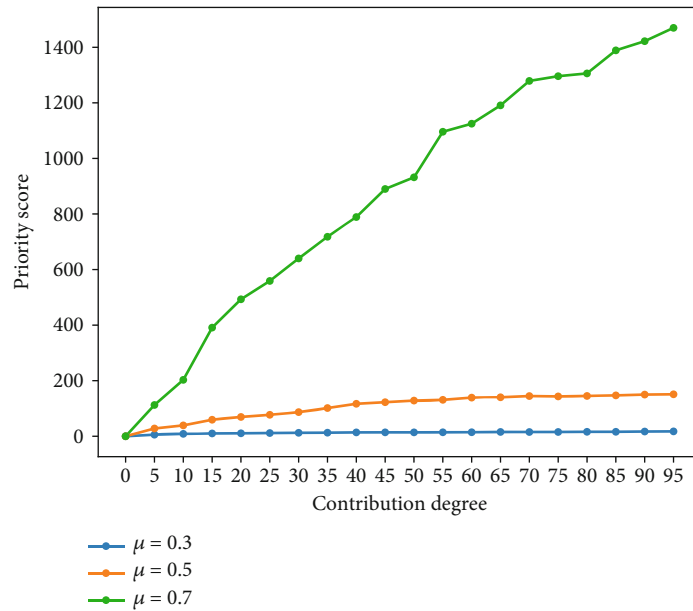


FIGURE 3: Influence of valid transaction volume on priority score.

[0, 3]/BTC based on the real Bitcoin situation [24] and generate the valid transaction volume within the range of [1, 1000] on the basis of assuming that the maximum valid transaction volume is 1000 in the experimental situation. In all the experiments, the random number generator of Python is used to generate the experimental data, and the average number is applied after the experimental data are generated 500 times with the purpose of reducing the influence of randomness on experimental results. From Figure 2, it can be observed that the priority score corresponds to the increasing transaction fee bid which goes up first, then drops down and tends to 0 as the contribution degree increases with the step of 5, and the results about the transaction fee influence on

priority score under different sets of σ share the same tendency. Meanwhile, we conduct the other experiment about the influence of increasing valid transaction volume on the priority score. As the result shows in Figure 3, the overall priority score goes up as the contribution degree increases; besides, the upward trend becomes faster when μ is higher. Moreover, as the value range of the transaction fee is much smaller than the valid transaction volume, there is a huge difference in the score result value, but the result comparison will not be affected; instead, the result value difference indicates that when user coalition increases transaction volume rather than transaction fee bid, the significant higher priority score will be obtained.

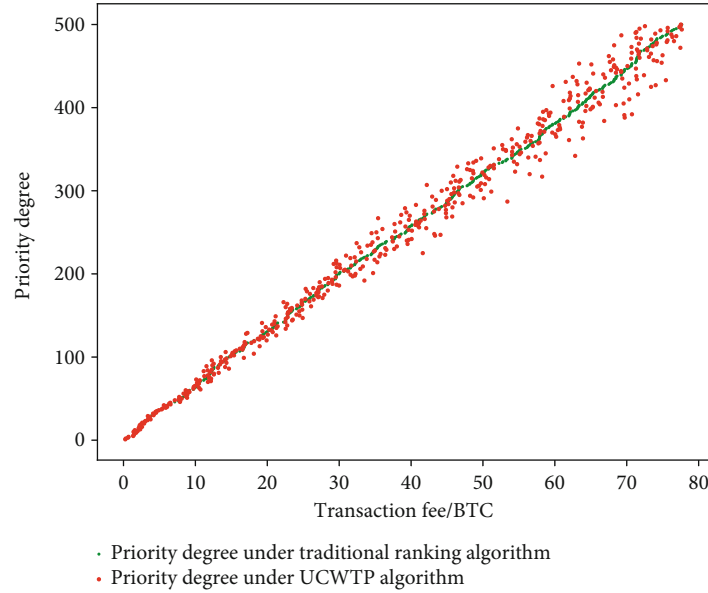


FIGURE 4: Impact of valid transaction volume and transaction fees on priority degree.

In order to make further impact comparisons of the UCWTP algorithm and traditional transaction priority algorithm on the priority ranking results, we conduct the second experiment as the result shows in Figure 4, and we randomly generate 500 transaction fees in the range of $[0, 80]$ /BTC as the user coalitions' transaction fee bid to imitate the actual Bitcoin transaction situation as possible. As it shows in Figure 4, the transaction fee bid is linearly related to the transaction priority degree under the traditional transaction priority algorithm; as a result, user coalition can obtain a higher transaction priority rank through improving transaction fee bid directly. However, with the UCWTP algorithm, the user coalition quality score increases with the growth of valid transaction volume of the user coalition, and the influence of single transaction fee on the transaction priority degree is reduced by an average of 2.34%, and it can be reduced by up to 12.2%; this is mainly because the ranking deviation obtained by the UCWTP algorithm will be reduced when the transaction fee of user coalition is low. However, it is interesting to notice that the impact of a single transaction fee on the priority degree is reduced by an average of 5.17% when referring to the annual ratio of approximately 21.66% for Bitcoin transaction packing (that is, the annual average ratio of the confirmed transactions to the total number of transactions in the blockchain platform from April 2019 to April 2020 [19]).

The experiment results above are consistent with the conclusion of Theorem 10; it is further proved that the arbitrary user coalition can obtain a higher transaction priority rank through improving the quality score when the coalition contribution degree increases; thus, the user coalition prefers to improve the quality score rather than lying to improve the transaction fees on the purpose of optimizing the utility. Besides, the experiment results reflect that the UCTPM mechanism can effectively incentivize users in coalitions to actively participate the blockchain's activities including transaction creation, transaction verification, competition

for bookkeeping right, and other activities that help to improve coalition quality scores and contribution degree; sequentially, we believe that the development of the blockchain can be helped to promote in the long run by adopting the UCTPM mechanism.

In addition to the two experiments we conducted for exploring the impacts of the UCWTP algorithm on transaction priority ranking results, we also implement the experiment on transaction confirming process time with the different transaction creation modes. In detail, we set up two different transaction creation modes to conduct the contrast experiments based on the researches in this paper, which are transactions created by a single user and by user coalitions, respectively. Also, the increment step of transaction volumes is set up as 5000 and end up by 60000 in this experiment. Although when compared with the transaction volume in the real Bitcoin situation, the experimental transaction volume is smaller, the experimental results as shown in Figure 5 can still reflect that the overall transaction confirming process time for the transactions created by user coalition is much faster than transactions created by a single user. Under the condition of the increased transaction volume, the transaction processing time of transactions created by a single user grows faster than that of transactions created by user coalition, and the maximum difference in transaction processing time between these two transaction creation modes is up to 6.9347 s when transaction volume reaches 60000. From the transaction processing time growth trend of Figure 5, we predict that if the UCTPM mechanism would be operated in the real blockchain project such as Bitcoin, the overall transaction confirming process time could be significantly reduced with the help of the UCTPM mechanism, because the user coalition transaction creation mode helps to effectively reduce the total time it takes miners to decide which transaction should be packed into a new block under the condition that no other factors affecting transaction confirming process time should be taken into consideration.

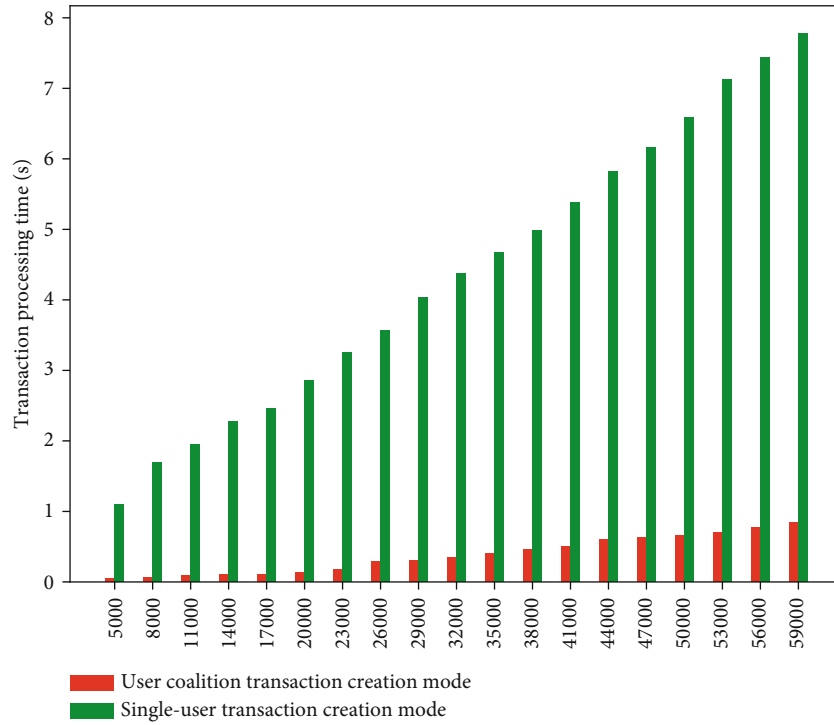


FIGURE 5: Transaction processing time differences between different transaction creation modes.

8. Conclusion and Future Work

Considering the issues of single-user transaction creation modes and limited influence factors on transaction priority algorithms in the current blockchain research fields, we propose a novel pricing mechanism called the UCTPM mechanism to optimize the traditional transaction priority algorithm and enrich the transaction creation modes. In addition, the user coalition quality score, user coalition contribution degree, and transaction type demand degree are introduced into the UCTPM mechanism to reduce the impact of the transaction fee on transaction priority ranking results.

In our proposed UCTPM mechanism, it is proved through theoretical proof and experimental analysis that the UCTPM mechanism can deprive the motivation of user coalition lying the transaction fees for extra utility as the user coalition contribution increases. From the aspect of the UCWTP algorithm, it demonstrates that the UCWTP algorithm can help to reduce the impacts of transaction fees on the transaction priority ranking results by an average of 5.17%, and the impacts can be reduced by up to 12.2% when compared to the traditional transaction priority ranking algorithm. While in terms of the performance of the UCTPM mechanism, the transaction confirming process time is greatly shorter when the transactions are created by user coalition instead of individual users from our related experiment results; it is also confirmed from the theoretical proof that the UCTPM mechanism satisfies the attributes of incentive compatibility, individual rationality, and budget balanced.

Although the outcomes of the researches conducted by this paper provide a new approach for the transaction prior-

ity ranking in blockchain, also enriching the modes of user transaction creation, the calculation models of unit reserve price for the miner who is granted the bookkeeping right, and the discount factor when the user coalition reaches certain ranks is not formalized in detail; thus, our future work will focus on the dynamic unit reserve price and discount factor formalization based on the actual blockchain transaction situation.

Data Availability

The (experimental data) data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872313, in part by the Key Research Projects in Education Informatization in Jiangsu Province under Grant 20180012, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18_2366, in part by the Yangzhou Science and Technology Bureau under Grants YZ2018209 and YZ2019133, in part by the Yangzhou University Jiangdu High-End Equipment Engineering Technology Research Institute Open Project under Grant YDJD201707, and in part by the Open Project in the State

Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, under Grant 1907.

References

- [1] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, "Blockchain technology in healthcare: a systematic review," *Healthcare*, vol. 7, no. 2, p. 56, 2019.
- [2] A. Al Shehabi, *Bitcoin transaction fee estimation using mempool state and linear perceptron machine learning algorithm*, [M.S. thesis], San José State University Library, 2018.
- [3] Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, 2006.
- [4] K. Chung, H. Yoo, D. Choe, and H. Jung, "Blockchain network based topic mining process for cognitive manufacturing," *Wireless Personal Communications*, vol. 105, no. 2, pp. 583–597, 2019.
- [5] J. Derks, J. Gordijn, and A. Siegmann, "From chaining blocks to breaking even: a study on the profitability of bitcoin mining from 2012 to 2016," *Electronic Markets*, vol. 28, no. 3, pp. 321–338, 2018.
- [6] D. Easley, M. O'Hara, and S. Basu, "From mining to markets: the evolution of bitcoin transaction fees," *Journal of Financial Economics*, vol. 134, no. 1, pp. 91–109, 2019.
- [7] D. Macrinici, C. Cartoceanu, and S. Gao, "Smart contract applications within blockchain technology: a systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, pp. 2337–2354, 2018.
- [8] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and ethereum: a brief overview," in *2018 17th international symposium infoteh-jahorina (infoteh)*, pp. 1–6, East Sarajevo, Bosnia-Herzegovina, March 2018.
- [9] N. Houy, *The economics of Bitcoin transaction fees*, Gate WP, 1407, 2014.
- [10] J. Li, Y. Yuan, and F. Wang, "A novel GSP auction mechanism for ranking bitcoin transactions in blockchain mining," *Decision Support Systems*, vol. 124, p. 113094, 2019.
- [11] G. Huberman, J. Leshno, and C. C. Moallemi, *An economic analysis of the bitcoin payment system*, Columbia Business School Research Paper No. 17-92, 2019.
- [12] K. Kerem, "Near zero bitcoin transaction fees cannot last forever," in *Proceedings of the International Conference on Digital Security and Forensics (DigitalSec2014)*, pp. 91–99, Ostrava, Czech Republic, June 2014.
- [13] S. Kasahara and J. Kawahara, "Effect of bitcoin fee on transaction confirmation process," 2016, <http://arxiv.org/abs/1604.0103>.
- [14] S. O. Lavi Ron and Z. Aviv, "Redesigning bitcoin's fee market," in *The World Wide Web Conference*, pp. 2950–2956, New York, NY, USA, May 2019.
- [15] M. Malte and B. Rainer, "Trends, tips, tolls: a longitudinal study of bitcoin transaction fees," in *In International Conference on Financial Cryptography and Data Security*, pp. 19–33, Springer, Berlin, Heidelberg, September 2015.
- [16] G. Huberman, J. Leshno, and C. C. Moallemi, "Monopoly without a monopolist: an economic analysis of the bitcoin payment system," in *Bank of Finland Research Discussion Paper*, p. 27, Bank of Finland (Suomen Pankki), 2017.
- [17] D. Azzolini, F. Riguzzi, and E. Lamma, "Studying transaction fees in the bitcoin blockchain with probabilistic logic programming," *Information*, vol. 10, no. 11, p. 335, 2019.
- [18] E. Erdin, M. Cebe, K. Akkaya, S. Solak, E. Bulut, and S. Uluagac, "A Bitcoin payment network with reduced transaction fees and confirmation times," *Computer Networks*, vol. 172, p. 107098, 2020.
- [19] BlockchainApril 2020, <https://www.blockchain.com/charts/n-transactions>.
- [20] A. Richard and R. Hitchens, "Efficient power markets," in *Transforming Climate Finance and Green Investment with Blockchains*, pp. 93–98, Elsevier, 2018.
- [21] J. Li, Y. Yuan, and F. Wang, "Bitcoin fee decisions in transaction confirmation queueing games under limited multi priority rule," in *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 134–139, Zhengzhou, China, China, Nov. 2019.
- [22] H. Nicolas, *From mining to markets: the economics of bitcoin transaction fees*, GATE WP, 2014.
- [23] Y. Yuan and F. Wang, "Blockchain and cryptocurrencies: model, techniques, and applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1421–1428, 2018.
- [24] *Bitcoin transaction fees soar 550% in a month, BCH, dash transactions much cheaper*, 2020, October 2020, <https://news.bitcoin.com/>.

Research Article

Efficient Planning and Solving Algorithm of S-Shape Acceleration and Deceleration

Zhijie Li ^{1,2,3,4}, Ligang Cai ^{2,4} and Zhifeng Liu ^{1,2}

¹Institute of Advanced Manufacturing and Intelligent Technology, Beijing University of Technology, Beijing 100124, China

²Beijing Key Laboratory of Advanced Manufacturing Technology, Beijing University of Technology, Beijing 100124, China

³College of Mechanical and Electronics Engineering, North China Institute of Aerospace Engineering, Langfang 065600, China

⁴Machinery Industry Key Laboratory of Heavy Machine Tool Digital Design and Testing Technology, Beijing 100124, China

Correspondence should be addressed to Zhifeng Liu; lzf@bjut.edu.cn

Received 17 September 2020; Revised 5 October 2020; Accepted 6 November 2020; Published 26 November 2020

Academic Editor: Junwu Zhu

Copyright © 2020 Zhijie Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

S-shape acceleration and deceleration are the most widely used flexible acceleration and deceleration method in the current CNC system, but its velocity solution equation contains irrational terms, which create a more complicated solution process. When analyzing the solution process of the S-shape acceleration and deceleration directly, using a traditional numerical solution method, the phenomenon of “solving the interval jump” arises, which is the main reason for low efficiency and poor stability of the solution. According to the S-curve profile and solution, the concept of separating the curve profile recognition from the velocity solution was proposed, and a method of quickly identifying the interval of the solution location was introduced. Through the method mentioned above, the complete acceleration and deceleration curve parameters can be obtained through a one-time plan and a one-time solution, and the solution efficiency and stability are guaranteed; solving the Newton problem depends too much on the initial value of Newton velocity, which not only retains the speed advantage of the Newton method but also uses the downhill factor to ensure its convergence. Through the simulation comparison and analysis, the efficiency, stability, and universality of the method are verified.

1. Introduction

The ability of acceleration and deceleration is an important index in evaluating the performance of CNC systems, which directly affects the processing efficiency and quality of CNC equipment. High-speed and high-quality machining demands higher requirements for the acceleration and deceleration control abilities of CNC systems. The CNC equipment must be able to move from the current position point to the next position point quickly and accurately, while also ensuring the smooth operation of the equipment and avoiding impact, out of step, over the range, or oscillation.

The commonly used acceleration and deceleration algorithms in medium and low-speed equipment include trapezoids [1, 2] and exponential acceleration and deceleration algorithms [3, 4]. The two algorithms are simple in programming, good in real-time performance, and high

in processing efficiency, but they all have the problems of abrupt change and discontinuity of the acceleration curve, which makes them not suitable for high-speed and high-accuracy machining [5].

In high-speed equipment, flexible acceleration and deceleration algorithms are often used, and their acceleration and deceleration function expressions are continuous, to reduce the impact on mechanical components [6]. S-shape acceleration and deceleration algorithm is the most widely used in flexible acc/dec algorithms at present. This method is based on the jerk constraint. The curves of displacement, velocity, and acceleration are composed of cubic, quadratic, and linear functions, respectively, and all of them are continuous without deviation [7, 8].

The mathematical programming and solution of the S-shape acc/dec method have attracted considerable attention. The complete S-shaped acc/dec curve is composed of 7 segments, with many parameters. In most cases, the

velocity solution equation is nonlinear, which makes its solution complex. Altintas [9] presented a detailed seven segment model of S-shape acc/dec, without elaborating its efficient solution. Barre [10] studied the influence of the acc/dec control in a high-speed machining process and analyzed the control algorithm of the seven-segment S-shape acc/dec. Shi [11] summarized eight possible cases of S-shape acc/dec in the planning, according to the characteristics of the curve segment. Nam [12] used the binary iteration to find the numerical solution, but the calculation amount was uncertain, and the solution efficiency was not ideal. Yang [13] adopted the Newton iteration, but the preconditions and convergence of Newton iteration were not clarified.

Many scholars simplified the S-shaped acc/dec curve in order to reduce the complexity of the solution. Wang and others [14, 15] introduced a look-ahead control algorithm based on the S-shaped acc/dec curve with assuming that the velocity curve is symmetrical. In reference [16], the uniform acceleration and deceleration segments were removed, and the model was simplified to 5 segments. Although the above method reduced the complexity of the solution, it limited the dynamic performance of the machine tool. Xia [17] used the S-shape acc/dec based on the reference table to plan the velocity, which reduced the calculation time while limiting the flexibility of S-shape acc/dec.

Presently, the planning of NURBS curve interpolation velocity based on S-shape acc/dec has become a hot research topic, and the solution mainly adopts the matching displacement by changing the velocity step by step [18, 19]. The main idea of this method is to use the assumed key velocity value to calculate the corresponding displacement, match it with the actual displacement, and then determine the curve profile according to the matching situation. For small line segments, the calculation needs to be repeated more than three times, which reduces the efficiency of the solution.

In order to reduce the complexity in S-shaped acc/dec curve planning and solutions, some scholars introduced the artificial intelligence algorithms such as genetic algorithms [20] and particle swarm optimization algorithms [21]. Although the above artificial intelligence algorithm has strong robustness, it needs to calculate the fitness of each iteration. Compared with the traditional dichotomy and Newton iteration, it has a large amount of calculation and low competitiveness in real-time.

For S-shaped acc/dec curves, the more ideal solution method should have smaller calculation complexity and higher solution accuracy. However, there is no general perfect solution at present.

In this paper, the characteristics of the S-shaped acc/dec curve were analyzed. Focusing on the disadvantages of low efficiency and poor stability of traditional algorithms, a more concise, efficient, and stable general solution was proposed. In this method, the curve profile planning and velocity solution were completely separated. On the basis of proving the continuity of displacement's segmented function about velocity, the interval boundary

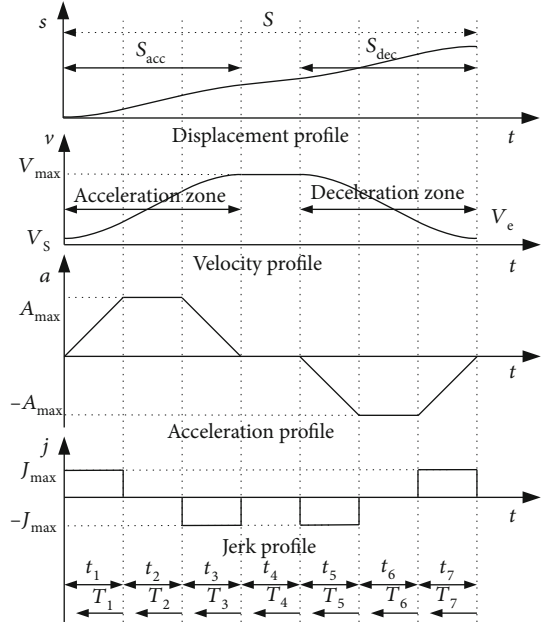


FIGURE 1: Profile of S-shape acceleration and deceleration.

points were used to analyze the interval of velocity, so as to complete the profile planning quickly. After obtaining the interval of the profile solution, the modified Newton method was introduced, which can make full use of the Newton method to accelerate up to the convergence speed, and at the same time, the downhill factor guarantees the convergence of the iteration.

2. Profile Characteristics of S-Shape Acceleration and Deceleration

The name of the S-acceleration and deceleration curves are derived from the fact that the acceleration curve is approximately S-shaped. This type of S-acceleration and deceleration processes includes the S-acceleration process, deceleration process, and the uniform connection between acceleration and deceleration. Acceleration processes includes the accelerated acceleration segment, uniform acceleration segment, and decelerated acceleration; the deceleration process includes the accelerated deceleration segment, uniform deceleration segment, and decelerated deceleration segment. There are 7 sections in total, as shown in Figure 1. The corresponding mathematical expression is as follows:

In the follow formula, $t_i (i = 1, 2, \dots, 7)$ is the running time of each segment. Command $T_1 = t_1$, $T_2 = t_1 + t_2$, and so on, $T_7 = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7$, A is the maximum acceleration, J is the maximum jerk, V_s is the initial velocity, V_e is the final velocity, and V_{\max} is the maximum velocity. According to the characteristics of the S-acceleration and deceleration curve, $t_1 = t_3$, $t_5 = t_7$.

$$J(t) = \begin{cases} J_1 = J & 0 \leq t < T_1 \\ J_2 = 0 & T_1 \leq t < T_2 \\ J_3 = -J & T_2 \leq t < T_3 \\ J_4 = 0 & T_3 \leq t < T_4 \\ J_5 = -J & T_4 \leq t < T_5 \\ J_6 = 0 & T_5 \leq t < T_6 \\ J_7 = J & T_6 \leq t < T_7 \end{cases},$$

$$a(t) = \begin{cases} a_1 = Jt & 0 \leq t < T_1 \\ a_2 = A & T_1 \leq t < T_2 \\ a_3 = A - J(t - T_2) & T_2 \leq t < T_3 \\ a_4 = 0 & T_3 \leq t < T_4 \\ a_5 = -J(t - T_4) & T_4 \leq t < T_5 \\ a_6 = -A & T_5 \leq t < T_6 \\ a_7 = -A + J(t - T_6) & T_6 \leq t < T_7 \end{cases},$$

$$v(t) = \begin{cases} v_1 = V_s + \frac{1}{2}Jt^2 \\ v_2 = V_1 + Jt_1(t - T_1) & V_1 = V_s + \frac{1}{2}JT_1^2 \\ v_3 = V_2 + Jt_1(t - T_2) - \frac{1}{2}J(t - T_2)^2 & V_2 = V_1 + Jt_1t_2 \\ v_4 = V_3 & V_3 = V_2 + \frac{1}{2}Jt_1^2 \\ v_5 = V_4 - \frac{1}{2}J(t - T_4)^2 & V_4 = V_3 \\ v_6 = V_5 - Jt_5(t - T_5) & V_5 = V_4 - \frac{1}{2}Jt_5^2 \\ v_7 = V_6 - Jt_5(t - T_6) + \frac{1}{2}J(t - T_6)^2 & V_6 = V_5 - Jt_5t_6 \end{cases}.$$

Displacement of the acceleration section:

$$S_{\text{acc}} = S_1 + S_2 + S_3 = (v + V_s)(2t_1 + t_2)/2. \quad (2)$$

Displacement of the deceleration section:

$$S_{\text{dec}} = S_5 + S_6 + S_7 = (v + V_e)(2t_5 + t_6)/2. \quad (3)$$

Total displacement:

$$S_T = S_{\text{acc}} + S_{\text{dec}} = (v + v_s)(2t_1 + t_2)/2 + (v + v_s)(2t_5 + t_6)/2. \quad (4)$$

If the actual displacement is $S > S_T$, then there is a uniform velocity segment and running time:

$$t_4 = \frac{S - S_T}{v}. \quad (5)$$

Whether there is a uniform acceleration section or a uniform deceleration section in the process of acceleration and deceleration is decided by V_s , V_e , v , A , and J .

$$v - V_s = (t_1 + t_2)A, \quad v - V_e = (t_5 + t_6)A, \quad (6)$$

$$t_1 = t_3 = \frac{A}{J}, \quad t_2 = \frac{v - V_s}{A} - \frac{A}{J} \quad v > V_s + \frac{A^2}{J}, \quad (7)$$

$$t_1 = t_3 = \sqrt{\frac{v - V_s}{J}}, \quad t_2 = 0 \quad v \leq V_s + \frac{A^2}{J}, \quad (8)$$

$$t_5 = t_7 = \frac{A}{J}, \quad t_6 = \frac{v - V_e}{A} - \frac{A}{J} \quad v > V_e + \frac{A^2}{J}, \quad (9)$$

$$t_5 = t_7 = \sqrt{\frac{v - V_e}{J}}, \quad t_6 = 0 \quad v \leq V_e + \frac{A^2}{J}. \quad (10)$$

Substituting the above time expression into the displacement formula:

- (1) While $v > V_s + (A^2/J)$ & $v > V_e + (A^2/J)$, then $S_T = (1/A)v^2 + (A/J)v - (V_s^2 + Ve^2/2A) + (V_s + V_e/2J)A$
- (2) While $v > V_s + (A^2/J)$ & $v \leq V_e + (A^2/J)$, then $S_T = (v + V_s/2)((A/J) + (v - V_s/A)) + (v + V_e)\sqrt{v - V_e/J}$
- (3) While $v \leq V_s + (A^2/J)$ & $v > V_e + (A^2/J)$, then $S_T = (v + V_e/2)((A/J) + (v - V_e/A)) + (v + V_s)\sqrt{v - V_s/J}$
- (4) While $v \leq V_s + (A^2/J)$ & $v \leq V_e + (A^2/J)$, then $S_T = (v + V_s)\sqrt{v - V_s/J} + (v + V_e)\sqrt{v - V_e/J}$

The expression (13) is a quadratic equation of one variable, and the exact solution can be obtained by using the root formula, while the expressions (14)-(16) contain the irrational part, so it is difficult to obtain the analytical solution.

3. Direct Numerical Solution

3.1. Dichotomy. Dichotomy is the most common method in solving nonlinear equations. Its main process in solving S-shape acceleration and deceleration is shown in Figure 2.

Step 1. input the initial velocity V_s , the final velocity V_e , maximum acceleration A , maximum jump A , required moving displacement S , and maximum limiting velocity V_{max} .

Step 2. compare the value of $V = V_{\text{max}}$, $V_s' = V_s + (A^2/J)$, and $V_e' = V_e + (A^2/J)$.

Step 3. call formulas (7)-(10) to calculate the time of each segment t_i and put them into formulas (13)-(16) to calculate the displacement S_T .

Step 4. if $S_T \leq S$, call formula (5) and obtain the running time of each segment and the overall curve profile. The calculation is finished.

If in step 4 $S_T > S$, the binary algorithm will be executed in step 5 to step 8:

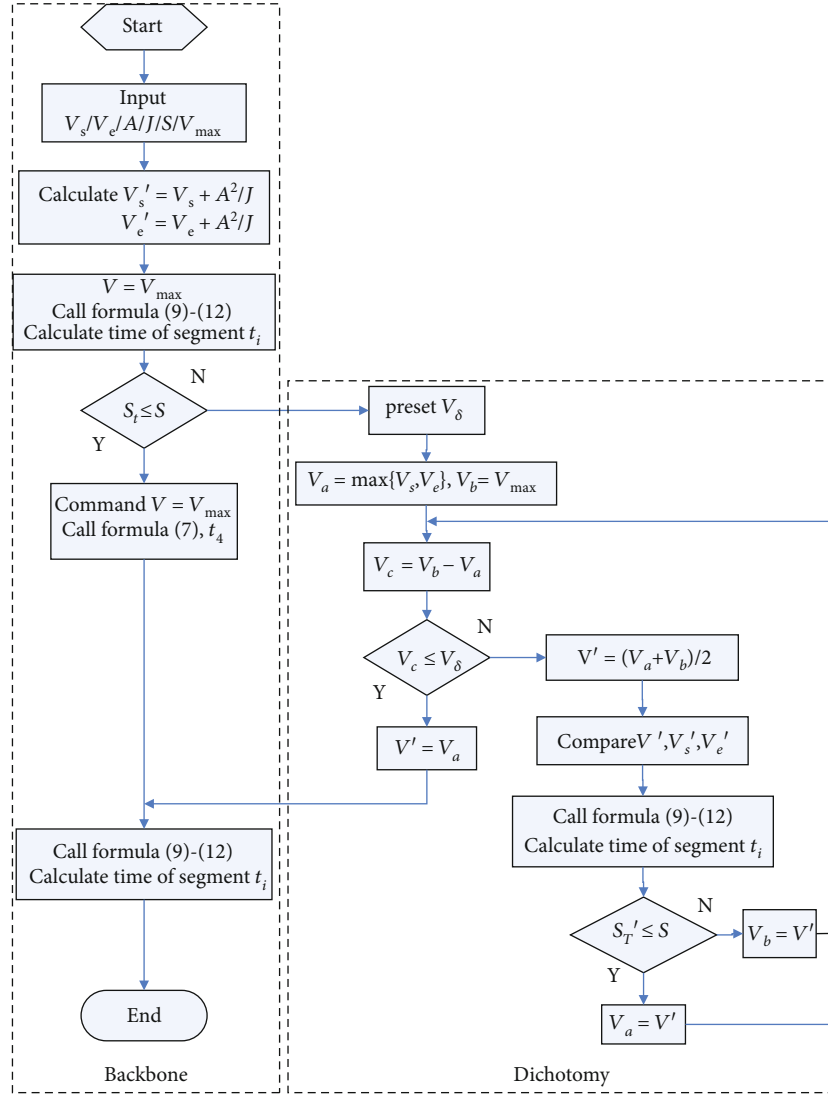


FIGURE 2: Solving the S-shape acceleration flow chart with dichotomy.

Step 5. take the initial interval $[V_a, V_b]$, where $V_a = \max\{V_s, V_e\}$ and $V_b = V_{\max}$ and set the velocity accuracy V_ε .

Step 6. determination of termination conditions: if $(V_b - V_a) \leq V_\varepsilon$, the calculation will be stopped.

Step 7. for iterative calculation, take $V' = (V_b + V_a)/2$, compare V' , $V_s' = V_s + (A^2/J)$, and $V_e' = V_e + (A^2/J)$, call formulas (7)-(10) to calculate the time of each segment, and substitute them into formulas (13)-(16) for calculation S_T . If $S_T \leq S$, then set $V_a = V'$; otherwise, set $V_b = V'$.

Step 8. turn to step 6.

3.2. Newton Iteration

3.2.1. Principle. For nonlinear equation $f(v) = S_T - S = 0$, V_0 is assumed to be an approximate solution of V^* ; $f(v)$ is expanded in the Taylor form at V_0 , and the following formula is obtained:

$$f(v) \approx f(V_0) + f'(V_0)(v - V_0) = 0. \quad (11)$$

$V_1 = V_0 - f(V_0)/f'(V_0)$ $f'(V_0) \neq 0$ is obtained as the first approximation of V^* ; V_k is assumed as the current point, and $f(v)$ can be expanded in the Taylor form at V_k :

$$f(v) \approx f(V_k) + f'(V_k)(v - V_k) = 0. \quad (12)$$

Then,

$$V_{k+1} = V_k - f(V_k)/f'(V_k) \quad f'(V_k) \neq 0, k = 0, 1 \dots \quad (13)$$

3.2.2. Iteration Process. The first four steps are the same as the first four steps of dichotomy, so it will not be described here. Start with step 5:

Step 5. V_0 is taken as the initial value, N is the maximum number of iterations, and V_ε is taken as the accuracy requirements.

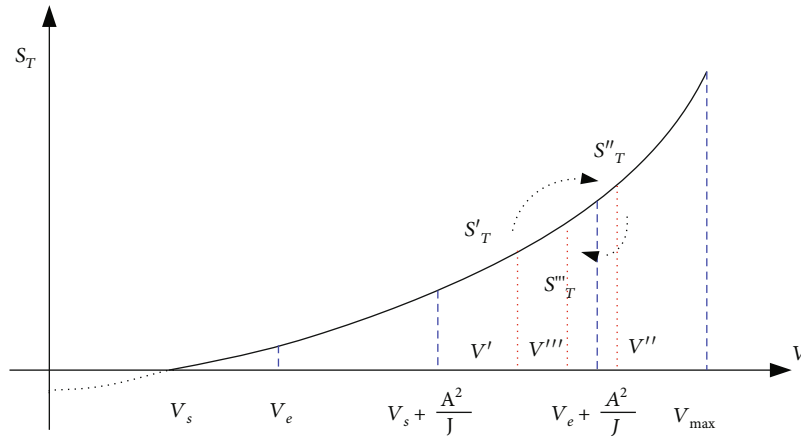


FIGURE 3: Schematic diagram of interval jump in the solving process.

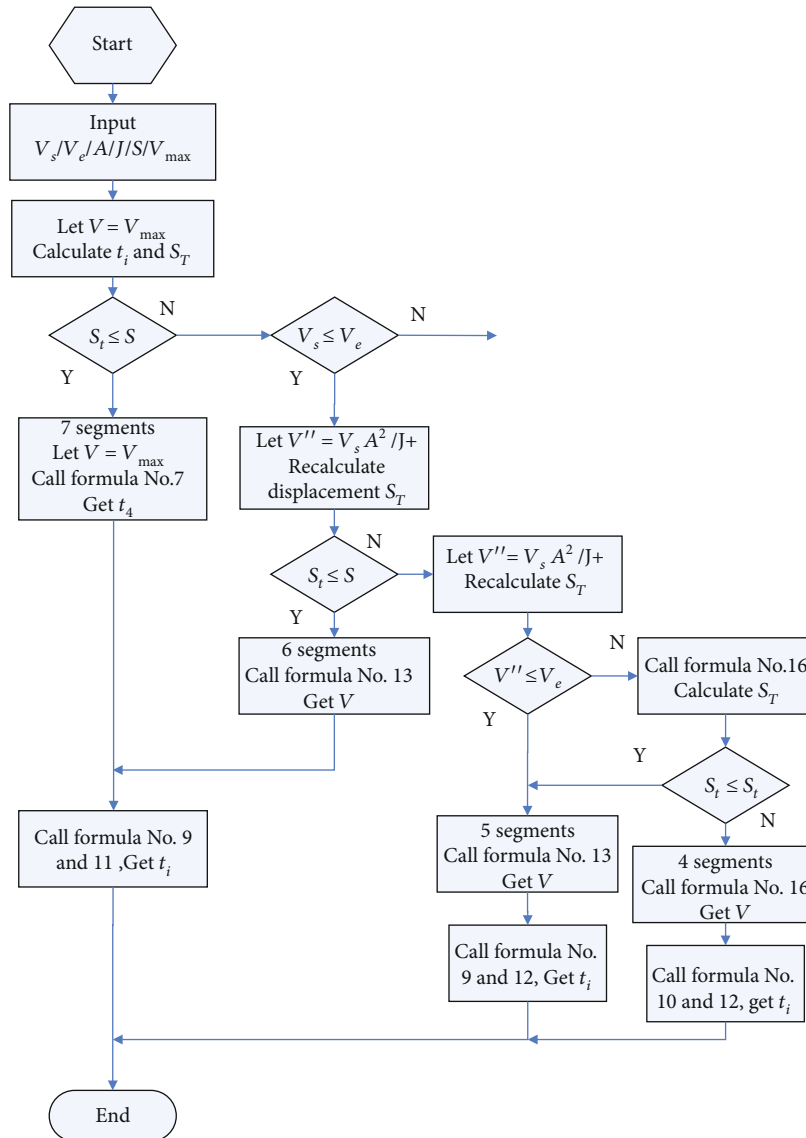


FIGURE 4: Flow chart of solution by matching with velocity step by step.

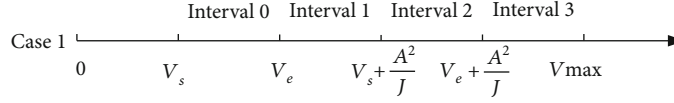


FIGURE 5: Case 1 with initial velocity less than the final velocity.

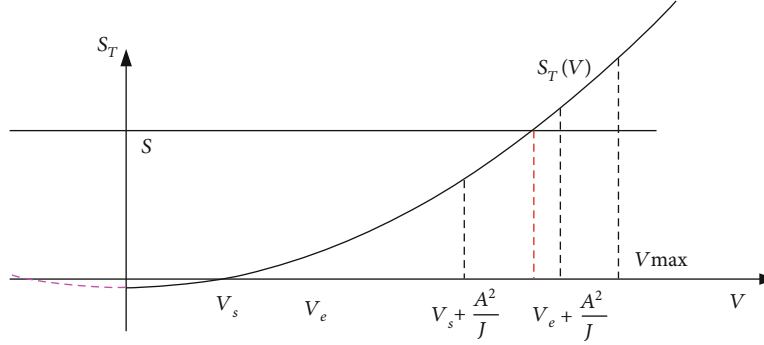
FIGURE 6: $S_T(v)$ Function curve.

TABLE 1: The summary of the curve profile planning of case 1.

Serial number	$F_{is} * F_{ie} \leq 0, f_{is} * f_{ie} \leq 0$ true or false			Expression for calculate t_i	Composition of curve profile
	$F_{1s} * F_{1e}$	$F_{2s} * F_{2e}$	$F_{3s} * F_{3e}$		
1	T	F	F	10/12	4 segments: $t_1/t_3/t_5/t_7$
2	F	T	F	9/12	5 segments: $t_1/t_2/t_3/t_5/t_7$
3	F	F	T	9/11	6 segments: $t_1/t_2/t_3/t_5/t_6/t_7$
4	F	F	F	9/11/7	7 segments: $t_1/t_2/t_3/t_4/t_5/t_6/t_7$

Step 6. calculate $V_{k+1} = V_k - f(V_k)/f'(V_k)$.

Step 7. if $|V_{k+1} - V_k| < V_e$, the calculation will be stopped.

Step 8. if $k = N$, stop; otherwise, $k = N + 1$ and turn to step 6.

4. Phenomenon of Solving Interval Jump

For the numerical solution, whether dichotomy or the Newton method, the solution interval is selected according to the initial value of the velocity in the first step, and then the displacement expression of this interval will be calculated. The solution obtained is not the final velocity solution, and the solution interval will be selected again according to the velocity value, and the iterative calculation will be carried out repeatedly until the termination condition is met. The process will cause the velocity value to jump at different intervals, and the displacement expression needs to be selected repeatedly, which will affect the solution efficiency, as shown in the following figure:

Figure 3 shows the case of $V_s < V_e < V_s + (A^2/J) < V_e + (A^2/J) < V_{max}$. The blue dotted line represents the segment point of the interval. From formula (13) to formula (16), it can be seen that the expression of displacement velocity is different in different segments. Taking dichotomy as an example, assuming that the required solution V^* is slightly less than $V_e + (A^2/J)$, the solution obtained after the first iter-

ation is $V' = V_a + V_b/2 = V_e + V_{max}/2(A^2/J)(A^2/J)$. Red indicates the corresponding position of the iterative solution. At this time, formula ((3)-(14)) is applicable, and the next iteration is carried out $V'' = V_a + V_b/2 = V' + V_{max}/2(A^2/J) \max]$. At this time, formula (16) is applicable, and the third iteration is carried out $V''' = V' + V''/2 \in [V_s + (A^2/J), V_e + (A^2/J)]$. In this case, (14) applies again.

This kind of iterative solution jumps between intervals, which causes different displacement expressions to be called back and forth, which leads to the instability of the solution process and affects the solution efficiency. In order to solve this kind of jump problem, the analytic programming method is introduced; that is, the interval of the expected solution is estimated first, and then the solution is calculated.

5. Solution by Matching with Velocity Step by Step

The main idea of this method is to use the assumed key velocity value to calculate the corresponding displacement, match it with the actual displacement, and then determine the curve profile according to the corresponding situation.

Take acceleration segment as example:

Step 1. assuming that the velocity can reach the required maximum velocity, the maximum velocity is substituted into

the displacement formula, and the calculated displacement is compared with the actual displacement. If the calculated displacement is smaller than the actual displacement, then there is a uniform velocity segment:

Step 2. if the calculated displacement is larger than the actual displacement, assume that there is just a uniform acceleration segment and substitute the assumed velocity value into the displacement formula and compare the calculated displacement with the actual displacement. If the calculated displacement is smaller than the actual displacement, it means that there is a uniform acceleration segment, then use the displacement formula including the uniform acceleration segment to solve the actual velocity value.

Step 3. if the calculated displacement is larger than the actual displacement, it means that there is no uniform acceleration section. At this time, the displacement formula without uniform acceleration section is used to solve for the actual velocity value.

The specific flow of the algorithm is shown in Figure 4.

The advantage of this method is that you can use the assumed velocity to plan first; then, you can solve for the actual velocity and avoid the jump in value ranges, increasing the stability of the solution, but there are also obvious disadvantages: the displacement is calculated and matched step by step with the assumed velocity. For the small line segment,

According to the definition of the function continuity, let

$$S_{Ti}(v) = \begin{cases} S_{T1} = (v + V_s)\sqrt{\frac{v - V_s}{J}} + (v + V_e)\sqrt{\frac{v - V_e}{J}} & v \leq V_s + \frac{A^2}{J} \leq V_e + \frac{A^2}{J} \leq V_{\max} \\ S_{T2} = \frac{v + V_s}{2} \left(\frac{A}{J} + \frac{v - V_s}{A} \right) + (v + V_e)\sqrt{\frac{v - V_e}{J}} & V_s + \frac{A^2}{J} \leq v \leq V_e + \frac{A^2}{J} \leq V_{\max} \\ S_{T3} = \frac{1}{A}v^2 + \frac{A}{J}v - \frac{V_s^2 + V_e^2}{2A} + \frac{V_s + V_e}{2J} & V_s + \frac{A^2}{J} \leq V_e + \frac{A^2}{J} \leq v \leq V_{\max} \end{cases} \quad (14)$$

$y = s(v)$ exist in a certain field of point v_0 ; when $v \rightarrow v_0$, if the limit of the function $s(v)$ exists and is equal to its function value at point v_0 , that is $\lim_{v \rightarrow v_0} s(v) = s(v_0)$, then function $y = s(v)$

is continuous at point v_0 . If $V_s' = V_s + (A^2/J)$ and $V_e' = V_e + (A^2/J)$, it can be proved that the function is continuous at the points V_s' and V_e' .

It can be seen from Figure 6 that the curve increases monotonously in the range of (V_s, V_{\max}) . Let $f_i(v) = S_{Ti}(v) - S$, ($i = 1, 2, 3$), S is the actual displacement, and there is a unique solution to $f_i(v) = S_{Ti}(v) - S = 0$ ($i = 1, 2, 3$), and with the difference of the displacement s value, the interval of solution of the piecewise function equation will change.

In the initial conditions, the initial velocity, the final velocity, the maximum acceleration, and the maximum jerk are given. The end values of three intervals are

the calculation needs to be repeated more than three times, which reduces the efficiency of the solution.

6. Interval Identification

To solve the problem of low efficiency of solution by matching with velocity, a new velocity planning method is proposed here.

From formula (5) to formula (10), it can be seen that when the initial velocity, final velocity, and maximum acceleration are given, the value of V directly affects the size and existence of each segment time t_i , and the actual s determines whether the actual velocity can reach the maximum allowable value.

When $V_s < V_e$, $V_e \leq V_s + (A^2/J)$, and $V_e + (A^2/J) < V_{\max}$, because of $\max\{V_s, V_e\} \leq V \leq V_{\max}$, interval 0 in Figure 5 is meaningless, and V may exist in the intervals 1-3.

The velocity value interval is divided into six sections and interval 0-5 by five key numerical points. The time expression (8) and formula (10) are suitable in interval 1, and the corresponding displacement expression is formula (16); the time expressions (7) and (10) are suitable in interval 2, and the corresponding displacement expression is formula (14); the time expressions (7) and (9) are suitable in interval 3, and the corresponding displacement expression is formula (15). Then, the piecewise function is obtained:

substituted into $f_i(v) = S_{Ti}(v) - S$, respectively. F_{is} and

F_{ie} are function values of the endpoints of the i -th interval.

$$\begin{aligned} F_{1s} &= f_1(v|v = V_e) = S_{T1}(v|v = V_e) - s \\ F_{1e} &= f_1\left(v\left|v = V_s + \frac{A}{J}\right.\right) = S_{T1}\left(v\left|v = V_s + \frac{A}{J}\right.\right) - s \\ F_{2s} &= f_2\left(v\left|v = V_s + \frac{A}{J}\right.\right) = S_{T2}\left(v\left|v = V_s + \frac{A}{J}\right.\right) - s \\ F_{2e} &= f_2\left(v\left|v = V_e + \frac{A}{J}\right.\right) = S_{T2}\left(v\left|v = V_e + \frac{A}{J}\right.\right) - s \\ F_{3s} &= f_3\left(v\left|v = V_e + \frac{A}{J}\right.\right) = S_{T3}\left(v\left|v = V_e + \frac{A}{J}\right.\right) - s \\ F_{3e} &= f_3(v|v = V_{\max}|_{T2}|v = V_{\max}|) \end{aligned} \quad (15)$$

From the properties of the continuity function, $F_{1e} = F_{2s}$, $F_{2e} = F_{3s}$.

In the next step, the function values of $f_i(v) = S_{Ti}(v) - S$ at the end points of each group of intervals are multiplied by each other to determine whether the product result is less than zero. There are two situations for the result:

(1) If $F_{is} * F_{ie} > 0$, the S_T and S curves do not intersect in this interval, and there is no solution in this interval. If there is no solution in the three intervals, the velocity is taken as V_{\max} , which indicates that the system has a uniform segment, the time is t_4 , and the curve is composed of seven segments. The t_i can be obtained by substituting the velocity value into the solutions of formulas (4) and (5)

(2) If $F_{is} * F_{ie} \leq 0$, it means that the S_T curve and S line intersect in this interval, and the equation has a solution, which is the actual value of velocity V that the system can achieve. At this time, substituting V into equation (7)–(10) can calculate the running time of each segment

Table 1 is the summary of the curve profile planning of case 1. The first three lines show the three cases of $F_{is} * F_{ie} \leq 0$ when i equals 1,2,3, respectively. The fourth line shows that $i = 1, 2, 3$ cannot meet the situation of $F_{is} * F_{ie} \leq 0$.

Formulas (14)-(16) can be solved by the dichotomy and Newton method listed above to get the velocity value. Then, the value is brought into the corresponding time calculation formula to get the running time of each segments.

As shown in Table 2, there are 12 possibilities in the comparison of five key values of the initial velocity V_s , the final velocity V_s , $V_s + (A^2/J)$, $V_e + (A^2/J)$, and V_{\max} .

Next, case 2 and case 3 in Table 2 will be discussed, and the interval distribution is shown in Figure 7.

Case 2. Interval 1 and interval 2 make sense.

The time expression (8) and equation (10) in interval 1 are suitable, and the displacement expression is the same as equation (17).

The time expression (7) and equation (10) in interval 2 are suitable, and the displacement expression is the same as equation (18).

Case 3. Only interval 1 makes sense. The time expression (8) and equation (10) in interval 1 are suitable, and the displacement expression is the same as equation (17).

Case 4 and case 5 express the situation when $V_e > V_s + (A^2/J)$, as shown in Figure 8.

Case 4. interval 1s and 2 make sense.

The time expression (7) formula (10) in interval 1 is applicable, and the displacement expression is the same as formula (14).

Time expression (7) and equation (9) in interval 2 are applicable, and displacement expression is the same as equation (13).

TABLE 2: Possibility of key point value comparison.

Case no.	Description of value
1	$V_s < V_e < V_s + \frac{A^2}{J} < V_e + \frac{A^2}{J} < V_{\max}$
2	$V_s < V_e < V_s + \frac{A^2}{J} < V_{\frac{A^2}{J_{\max}}}$
3	$V_s < V_e < V_{\frac{A^2 A^2}{J J_{\max}}}$
4	$V_s < V_s + \frac{A^2}{J} < V_e < V_e + \frac{A^2}{J} < V_{\max}$
5	$V_s < V_s + \frac{A^2}{J} < V_e < V_{\frac{A^2}{J_{\max}}}$
6	$V_s = V_e < V_s + \frac{A^2}{J} < V_{\max}$
7	$V_s = V_e < V_{\frac{A^2}{J_{\max}}}$
8	$V_e < V_s < V_e + \frac{A^2}{J} < V_s + \frac{A^2}{J} < V_{\max}$
9	$V_e < V_s < V_e + \frac{A^2}{J} < V_{\frac{A^2}{J_{\max}}}$
10	$V_e < V_s < V_{\frac{A^2 A^2}{J J_{\max}}}$
11	$V_e < V_e + \frac{A^2}{J} < V_s < V_s + \frac{A^2}{J} < V_{\max}$
12	$V_e < V_e + \frac{A^2}{J} < V_s < V_{\frac{A^2}{J_{\max}}}$

Case 5. Interval 1 makes sense.

The time expression (7) and formula (10) in interval 1 are suitable, and the displacement expression is the same as formula (15).

(1) When $V_s = V_e$, there are only two cases, case 6 and 7, as shown in Figure 9

Case 6. Interval 1 and 2 make sense.

The time expression (8) and equation (10) in interval 1 are suitable, and the displacement expression is the same as equation (17).

The time expression (7) and equation (9) in interval 2 are suitable, and the displacement expression is the same as equation (24).

(2) When $V_s > V_e$, the possible value range of velocity is from case 8 to case 12. The actual solution process is the same as that of $V_s < V_e$, and only 1 and 2 need to be interchanged. In consideration of space, it will not be expanded here

(3) There are four other possibilities, which are not listed in the bid, as shown in Table 3

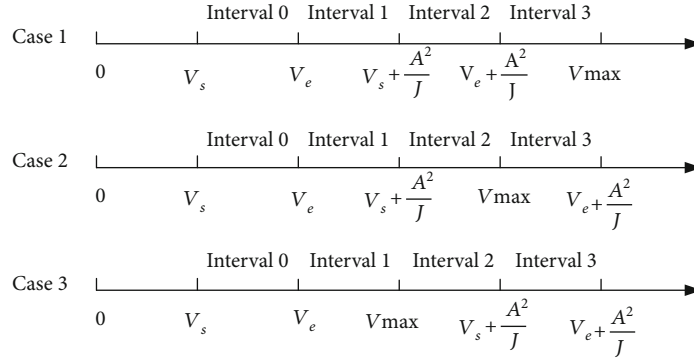


FIGURE 7: Interval partition graph of cases 1-3.

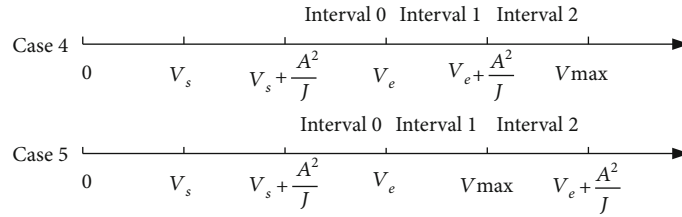


FIGURE 8: Interval partition graph of cases 4 and 5.

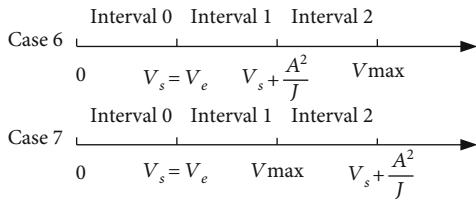


FIGURE 9: Interval partition graph of case 6 and 7.

TABLE 3: Only acceleration section or deceleration section.

Case no.	Description of value
13	$V_s < V_s + \frac{A^2}{J} < V_{\max}$
14	$V_s < V_{\frac{A^2}{J_{\max}}}$
15	$V_s < V_e + \frac{A^2}{J} < V_{\max}$
16	$V_e < V_{\frac{A^2}{J_{\max}}}$

Case 7. Only interval 1 makes sense. The time expression (8) and equation (10) in interval 1 are suitable, and the displacement expression is the same as equation (17).

The so-called S-shape means that the shape of acceleration is similar to the S-shape, case 13—case listed in Table 3. There is only acceleration section or deceleration section, and acceleration is similar to half circle. According to the definition of the S-shaped acceleration and deceleration curve, these four cases do not meet the definition, but these four cases do exist in small displacement, especially in

the speed sensitive area, so they are listed here. The calculation method is the same as above and will not be described here.

7. Comparison of Solution Methods

7.1. Match with Velocity VS Interval Identification. The first step when using the velocity method is to assume that the curve has seven complete segments, and that the time and velocity values are substituted into the formula to calculate the displacement. If the calculated displacement is less than the actual displacement, the one with the larger value in $V_s + A^2/J$ and $V_e + A^2/J$ will be substituted into the equation again to calculate the displacement, and the calculated displacement and the actual displacement will be compared again. If the calculated displacement is greater than or equal to the actual displacement, then the actual velocity and running time can be obtained by solving the equation.

If the calculated displacement is smaller than the actual displacement, it is necessary to replace the smaller value of $V_s + A^2/J$ and $V_e + A^2/J$ into the displacement equation again, calculate and compare the displacement for the third time, and then plan out the profile of the curve. Finally, the actual velocity can be obtained by solving the equation.

The specific steps of the interval identification method are as follows: ① input the initial value, compare 5 key speed values, and find the corresponding case No.; ② in the corresponding case No., substitute the value to judge $F_{is} * F_{ie} \leq 0$; ③ solve the equation in the interval of $F_{is} * F_{ie} \leq 0$ and get the velocity; and ④ replace the speed V with the time calculation formula and get the whole curve profile.

Figure 10 shows the comparison of the processes of the two methods. In summation, if the curve has 7 or 6 segments, the calculation steps and calculation amounts of the two

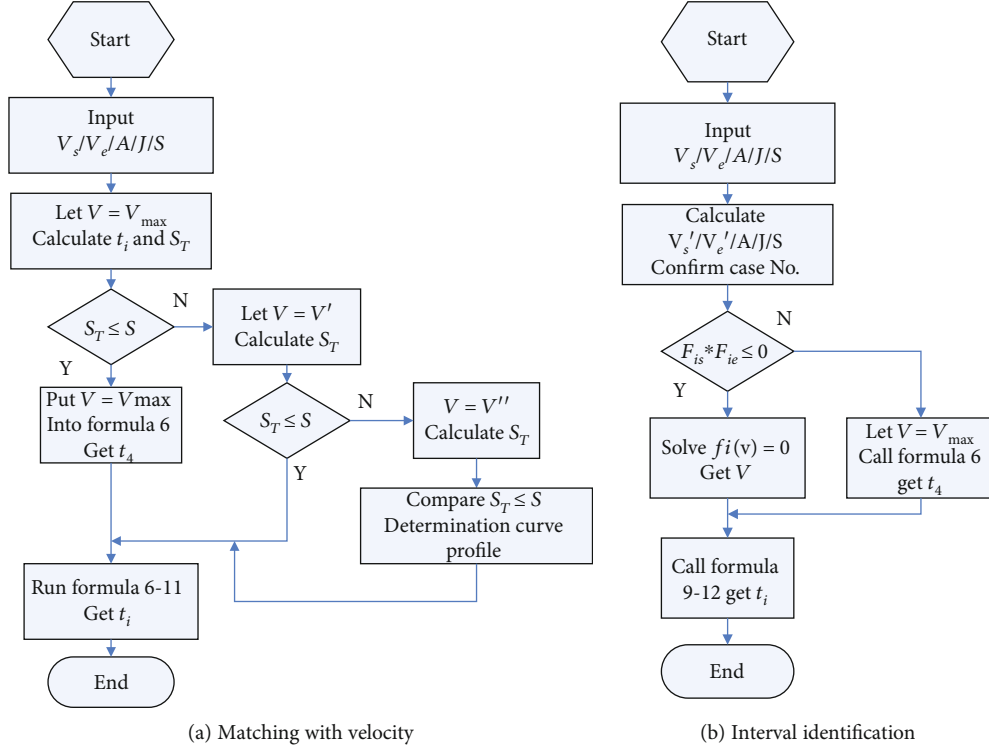


FIGURE 10: Comparison of matching with velocity and interval identification.

methods are basically the same. If the curve has 4 or 5 segments, the calculation steps and calculation amounts of the interval identification method are less and therefore, the efficiency is higher.

7.2. Direct Numerical Solution vs Interval Identification. In order to verify the efficiency of the solution, various methods and algorithms are programmed with matlab2018a, and the displacement is taken as a variable between 0.0001 m and 0.018 m, and the interval between the value points is 0 and 1 mm. Computer configuration: processor Intel (R) core (TM) i5-4200u memory 8 g. The solution time of each algorithm under different displacement conditions is calculated, and the solution efficiency of each method is compared. The number of Newton iteration and dichotomy iteration termination is 10000. The specific parameter settings are shown in Table 4, and the solution efficiency is shown in Figure 11.

It can be seen from Figure 11 that the abscissa of either direct Newton iteration or interval identification+Newton iteration does not start from zero; that is, the red dotted area in the figure which indicates that Newton iteration does not return the velocity solution and directly jumps out of the iteration cycle after reaching the limit of iteration rounds. As mentioned in the previous paper, the convergence of the Newton iteration needs to be further verified. When the displacement value is about 1-3 mm, the number of Newton iterations is also significantly higher than the number after it is stable. When the displacement is more than 4 mm, all algorithms tend to be more stable as displacement increases.

TABLE 4: Parameters used in the simulation.

Symbol	Item	Value
V_s	Initial velocity	20 mm/s
V_e	Final velocity	30 mm/s
V_{\max}	Maximum velocity	100 mm/s
A_{\max}	Maximum acceleration	600 mm/s ²
J_{\max}	Maximum jerk	3×10^4 mm/s ³
V_e	Iteration termination precision	1×10^{-6} mm/s
T_s	Interpolation period	1 ms
S	Displacement	0.01-18 mm
V_o	Initial velocity of iteration	15 mm/s

In order to see the solution efficiency of the four methods more clearly and intuitively, the abscissa in Figure 11 is intercepted through 7-12 mm part, resulting in Figure 12.

Regarding solving efficiency, the efficiency of the interval identification method is obviously better than that of the direct iterative method when the solving process of various algorithms is stable. The efficiency of interval identification plus Newton iteration is the highest, and then the efficiency of direct Newton iteration is about one third of that of interval identification plus Newton iteration. The efficiency of dichotomy is obviously lower than that of Newton iteration. The efficiency of direct dichotomy is about one seventh of interval identification plus the Newton iteration. From the point of view of solving stability, the interval identification method is also better than the direct

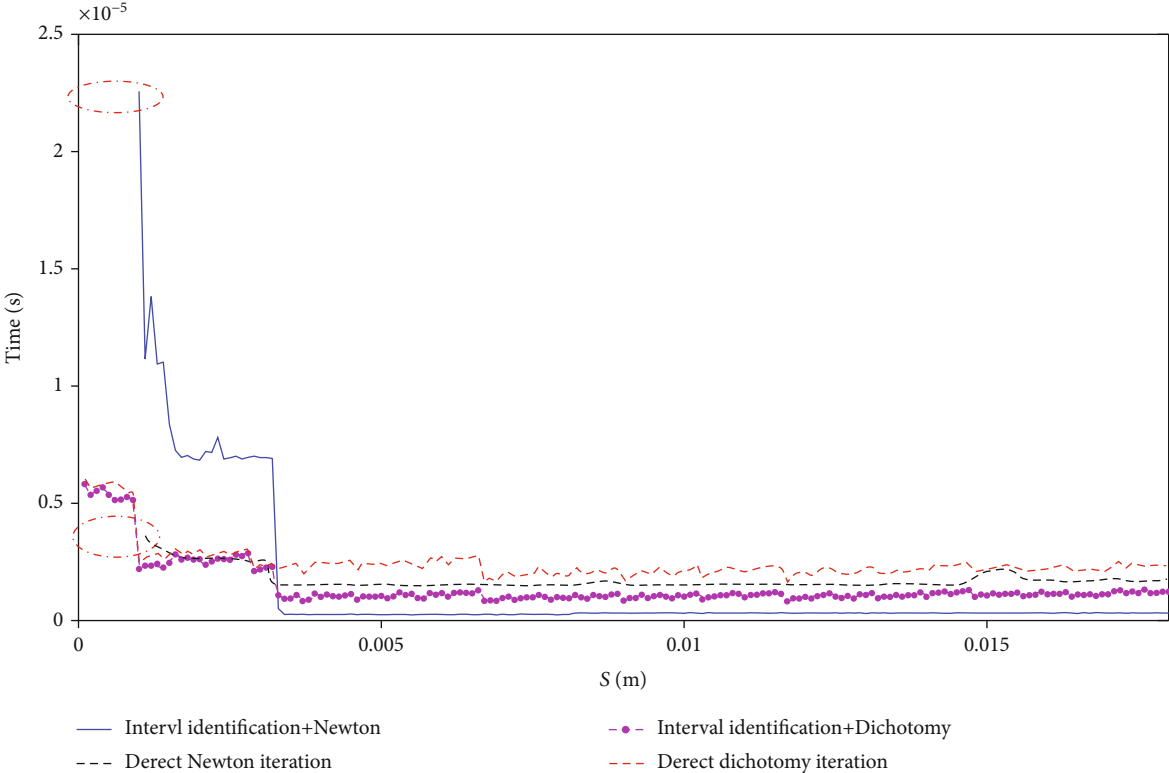


FIGURE 11: Solution efficiency comparison.

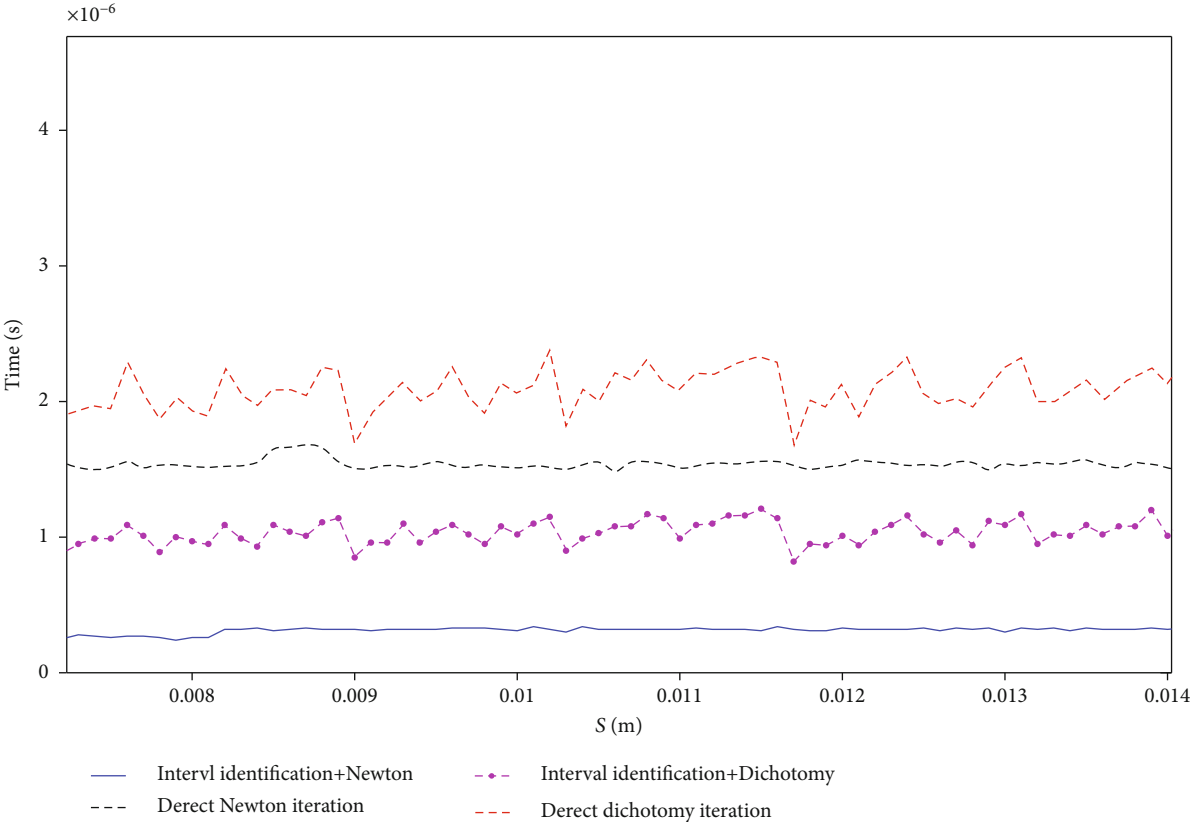


FIGURE 12: Local amplification of efficiency comparison graph after stabilization.

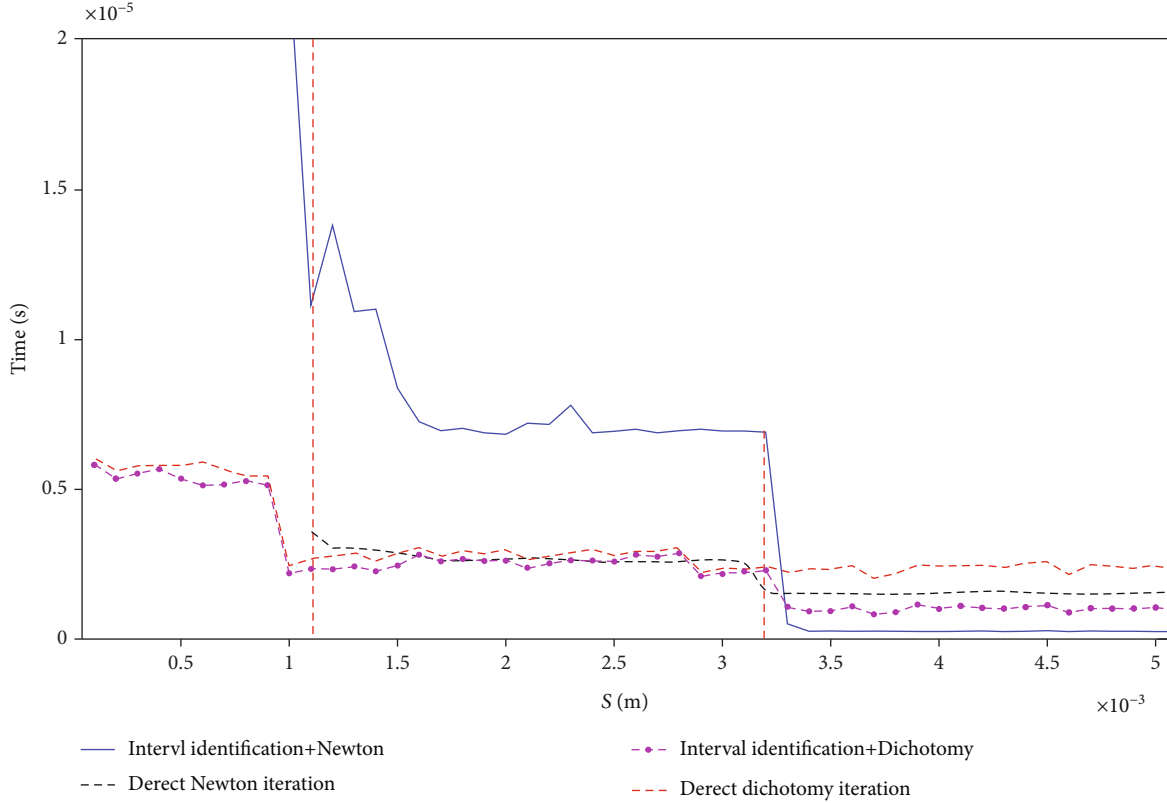


FIGURE 13: Comparison of solution efficiency when small displacement.

iteration method. As mentioned earlier in the paper, the interval identification avoids the phenomenon of iteration jumping back and forth in different value intervals, thus improving the stability.

To sum up, the interval identification+Newton iteration method is the best both in solving efficiency and solving stability.

Figure 11 is the abscissa which is truncated at 0-5 mm and enlarged locally to get Figure 13. It can be seen that the efficiency of all solutions fluctuates within a certain boundary, and the Newton iteration is the obvious. In the figure, the red dot dash appears near two key points, which are S_{T1} and S_{T2} in formula (17), corresponding to the interval boundary points in the interval identification method. This shows that the expression change caused by the change of displacement length has an obvious effect on the iteration efficiency. The smaller the displacement, the more time-consuming the solution.

8. Interval Identification plus Modified Newton Iteration

In order to prevent the divergence of Newton iteration, a requirement is attached to the iteration process to keep its monotonicity.

$$|f(V_{k+1})| < |f(V_k)|. \quad (16)$$

In order to meet this requirement, Newton iteration is modified.

$$\begin{cases} V_{k+1} = V_k - \frac{f(V_k)}{f'(V_k)} \\ V_{k+1} = \lambda V_{k+1} + (1 - \lambda)V_k \end{cases}, \quad (17)$$

Where $\lambda(0 < \lambda \leq 1)$ is the downhill factor, and formula (18) is obtained after the transformation of the above formula:

$$V_{k+1} = V_k - \lambda \frac{f(V_k)}{f'(V_k)} \quad k = 0, 1, \dots \quad (18)$$

In the iteration, it starts from $\lambda = 1$ and reduces by half until it meets the requirements (16).

After the modified Newton iteration is introduced, the iteration parameters and initial conditions are set according to Table 4. Under the same conditions, the efficiency curve Figure 14 is obtained.

Only Newton iteration can be seen in Figure 14, because the iteration diverges until the number of iterations jumps out of the cycle. The record time of 0.01 seconds in the figure is the time taken to jump out of the cycle. Figure 13 is partially enlarged to obtain Figure 15.

It can be seen from Figure 15 that, when the displacement is large, the solution efficiency of interval identification plus

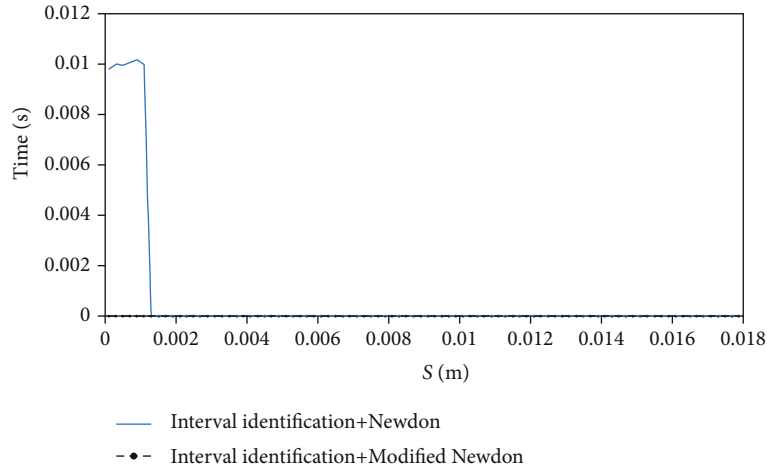


FIGURE 14: Comparison between Newton iteration and Newton downhill method.

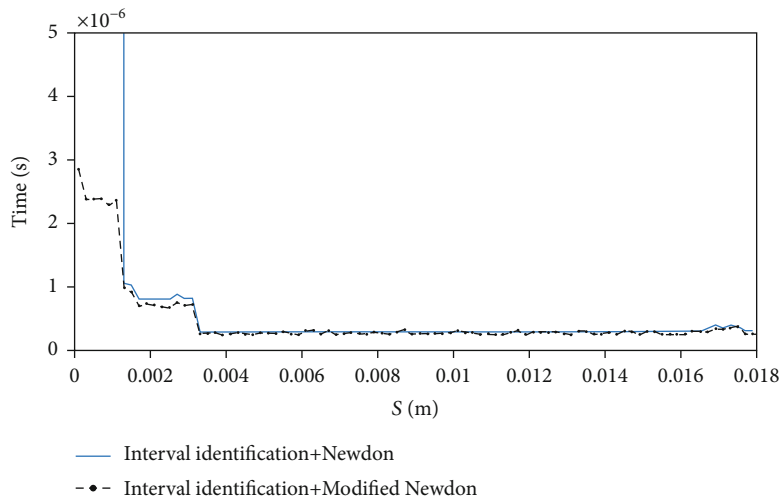


FIGURE 15: Local enlarged map of comparison between Newton and Newton downhill.

modified Newton method is very close to that of interval identification plus Newton iteration, which retains the characteristics of fast convergence of Newton iteration; when the displacement is small, the downhill factor ensures the convergence of iteration.

9. Conclusion

- (1) When analyzing the process of solving S-shape acceleration and deceleration directly by the traditional numerical solution method, the phenomenon of “solving interval jump” was concerned, which is a main reason for the low efficiency and poor stability of the solution
- (2) According to the S-curve profile and solution, the concept of separating the curve profile recognition from the velocity solution was introduced, and a plan of quickly identifying the interval of the solution location was put forth. With this method, the com-

plete acceleration and deceleration curve parameters can be obtained through one-time planning and one-time solutions, and the solution efficiency and stability are certain

- (3) The modified Newton method was introduced to solve the problem that the Newton iteration depends too much on the initial value of velocity, which not only retains the speed advantage of the Newton method but also uses the downhill factor to ensure its convergence
- (4) Finally, through the simulation comparison and analysis, the efficiency and stability of the interval identification and modified Newton method were verified

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (No. 51975019) and the Science and Technology Project of Hebei Education Department (No. QN2019193).

References

- [1] J. W. Jeon and Y. Y. Ha, "A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 133–139, 2000.
- [2] J. Hu, L. J. Xiao, Y. H. Wang, and Z. Wu, "An optimal feedrate model and solution algorithm for a high-speed machine of small line blocks with look-ahead," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9-10, pp. 930–935, 2006.
- [3] Y. P. Li and S. N. Wang, "Improved exponential acceleration and deceleration algorithm," *Machine Tool and Hydraulics*, vol. 1, pp. 39–40, 2006.
- [4] H. Li, H. Li, L. Song, Y. Yin, L. Huang, and W. Li, "Design of global sliding-mode controlled AC servo controller based on exponential acceleration/deceleration algorithm," in *2010 IEEE International Conference on Mechatronics and Automation*, pp. 1507–1511, Xi'an, China, 2010.
- [5] H. Jun, Y. Youpeng, C. Hao, and W. Huaming, "A fast nested look-ahead algorithm with S-shape acceleration and deceleration," *Acta Aeronautica Et Astronautica Sinica*, vol. 31, no. 4, pp. 842–851, 2010.
- [6] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation," *International Journal of Machine Tools & Manufacture*, vol. 41, no. 9, pp. 1323–1345, 2001.
- [7] J. H. Chen, S. S. Yeh, and J. T. Sun, "An S-curve acceleration/-deceleration design for CNC machine tools using Quintic Feedrate function," *Computer-Aided Design and Applications*, vol. 8, no. 4, pp. 583–592, 2011.
- [8] K. Z. Sun, X. F. Zhou, G. Zhang, and X. S. Chen, "An interpolation algorithm based on S-curve Feedrate profile and look-ahead function," *Applied Mechanics & Materials*, vol. 300-301, pp. 1389–1396, 2013.
- [9] Y. Altintas, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*, Cambridge University Press, London, 2000.
- [10] P. J. Barre, R. Bearee, P. Borne, and E. Dumetz, "Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 42, no. 3, pp. 275–293, 2005.
- [11] S. Chuan, Z. Tong, Y. Peiqing, and L. Qiang, "Study on S-shape curve acceleration and deceleration control on NC system," *China Mechanical Engineering*, vol. 18, no. 12, pp. 1421–1425, 2007.
- [12] S. H. Nam and M. Y. Yang, "A study on a generalized parametric interpolator with real-time jerk-limited acceleration," *Computer Aided Design*, vol. 36, no. 1, pp. 27–36, 2004.
- [13] Y. Liangliang, X. Shoujin, S. Weimin, and G. Hongwei, "Research on S type acceleration and deceleration time planning algorithm with beginning and end speed non-zero," *Journal of Mechanical Engineering*, vol. 52, no. 23, pp. 199–206, 2016.
- [14] Y. Chen, X. Ji, Y. Tao, and H. Wei, "Look-ahead algorithm with whole S-curve acceleration and deceleration," *Advances in Mechanical Engineering*, vol. 5, no. 10, Article ID 974152, 2015.
- [15] L. Wang and J. Cao, "A look-ahead and adaptive speed control algorithm for high-speed CNC equipment," *International Journal of Advanced Manufacturing Technology*, vol. 63, no. 5-8, pp. 705–717, 2012.
- [16] J. A. Tao, S. L. Gao, Q. N. You, and Q. Shi, "Look ahead and acceleration deceleration algorithms for micro-line blocks machining," *Journal of Computer-Aided Design & Computer Graphics*, vol. 22, no. 9, pp. 1570–1577, 2010.
- [17] Y. B. Xia and W. Wang, "S-curve acceleration/deceleration algorithm based on reference table," *Research Journal of Applied Sciences Engineering & Technology*, vol. 4, no. 2, pp. 131–134, 2012.
- [18] H. Ni, J. Yuan, S. Ji, C. Zhang, and T. Hu, "Feedrate Scheduling of NURBS Interpolation Based on A Novel Jerk-continuous ACC/DEC Algorithm," *IEEE Access*, vol. 6, pp. 66403–66417, 2018.
- [19] D. Li, J. Wu, J. Wan, S. Wang, S. Li, and C. Liu, "The implementation and experimental research on an S-curve acceleration and deceleration control algorithm with the characteristics of end-point and target speed modification on the fly," *The International Journal of Advanced Manufacturing Technology*, vol. 91, no. 1-4, pp. 1145–1169, 2017.
- [20] T.-C. Lu and S.-L. Chen, "Genetic algorithm-based S-curve acceleration and deceleration for five-axis machine tools," *International Journal of Advanced Manufacturing Technology*, vol. 87, no. 1-4, pp. 219–232, 2016.
- [21] F. Xu, H. Chen, X. Gong, and Q. Mei, "Fast nonlinear model predictive control on FPGA using particle swarm optimization," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 310–321, 2016.