Computing and Informatics, Vol. 39, 2020, 1–4, doi: 10.31577/cai_2020_1-2_1

A PREFACE TO THE SPECIAL ISSUE ON EMERGING AND INTELLIGENT INFORMATION SERVICES

Maozhen Li

Department of Electronic and Computer Engineering
Brunel University London, Uxbridge, UB8 3PH, UK

Zhijun Ding

Department of Computer Science and Technology
Tongji University, Shanghai, China

Information services have evolved from centralized monolithic systems to distributed and intelligent systems especially empowered with the emerging technologies such as big data, artificial intelligence, Internet of Things (IoTs). Papers included in this special issue mainly cover the topics of Petri net, cloud and fog computing, social networks, and deep neural networks. The following sections briefly introduce these papers.

Workflow has been playing an increasing role in modern information systems. Xiang and Liu [9] employed WFD-net, a special form of Petri net, to build a guard-driven reachability graph for correctness verification of data flows dealing with the cases that data might be missing, inconsistent, lost and redundant. According to Feng et al. [10], there is an increasing demand for business intelligence automatically extracted from event logs and process mining plays a critical role in business process management. This paper presents a new Petri net alignment to improve the efficiency in process mining. In [11], Cheng et al. presented a comprehensive learning particle swarm optimization algorithm based on fuzzy Petri net and applied it to diagnose the faults of a complex motor system. According to Li et al. [8], there is a growing research interest in imbuing robots not only with the capability of perception and planning but also the capability of learning. This paper models the behaviours of a robot into a Petri net which facilitates the learning capability of the robot. In [12], Teng et al. modelled processing mining with a logic Petri net with an aim to deal with structures which might be incomplete and concurrent to better reflect the reality of business processes.

The past few years have witnessed the emerging of edge computing in support of IoT applications close to the edge users. Li et al. [1] presented a novel computing platform which incorporates fog computing with cloud computing in support of healthcare services. He et al. [2] employed a stochastic Petri net to model resource scheduling in cloud data centers with an aim to improve energy efficiency. A single cloud-based data storage potentially suffers from data unavailability, vendor lock-in and data privacy leakage. In [3], Wang et al. focused on multi-cloud stage with an aim to minimize monetary cost and maximize data availability. Resource virtualization plays a critical role in resource provisioning in cloud computing platforms. In [4], Pang et al. presented an improved multi-objective particle swarm optimization algorithm to optimize virtual machine deployment.

In [5], Zhang et al. considered the strength of user relationship, the similarity of entities, and the degree of user interest in recommendation systems. Interestingly, user relationship can be inferred from user interactions on social networks. In [6] Pang et al. focused on recommendation systems, and introduced stability variables and time-sensitive factors to solve the problem of user interest drift, and improve the accuracy of prediction. With a penetration of social networks, it is highly important to detect malicious information which can be hidden in data flows. In this work, Yuan and Sun [7] employed a divide and conquer strategy and proposed an intervention algorithm based on subgraph partitioning to search for influential nodes to block or release clarification.

In [13], Zhu et al. employed the Long Short-Term Memory (LSTM) deep neural network model to learn user travel patterns. The LSTM was further optimized with Quantum Genetic Algorithm (QGA). In [14], Wang et al. proposed the deep convolution and correlated manifold embedded distribution alignment (DC-CMEDA) model, which is able to realize the transfer learning classification between and among various small datasets, and greatly shorten the training time in forest fire smoke prediction. With an increasing use of mobile devices such as mobile phones, Wang et al. [15] focused on mobile sensing and proposed a particle swarm optimization algorithm for pedestrian step tracking.

We hope that the perspectives presented in this special issue would be of a great interest to the readers. We also expect the readers to contribute to this exciting and fast-growing research area.

Acknowledgements

We would like to thank Dr. Ladislav Hluchy, the Editor-in-Chief of Computing and Informatics (CAI) for his timely advice on this special issue. A big thanks also goes to Ms Viera Jablonska, the CAI journal editorial assistant for her great support in publication of the special issue.

REFERENCES

[1] Li, Z.—Wen, L.—Liu, J.—Jia, Q.—Che, C.—Shi, C.—Cai, H.: Fog and Cloud Computing Assisted IoT Model Based Personal Emergency Monitoring and Diseases Prediction Services. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 5–27, doi: 10.31577/cai_2020_1-2_5.

[2] He, H.—Zhao, Y.—Pang, S.: Stochastic Modeling and Performance Analysis of Energy-Aware Cloud Data Center Based on Dynamic Scalable Stochastic Petri Net. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 28–50, doi: 10.31577/cai_2020_1-2_28.

[3] Wang, P.—Zhao, C.—Liu, W.—Chen, Z.—Zhang, Z.: Optimizing Data Placement for Cost Effective and High Available Multi-Cloud Storage. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 51–82, doi: 10.31577/cai_2020_1-2_51.

[4] Pang, S.—Dong, D.—Wang, S.: Virtual Machine Deployment Strategy Based on Improved PSO in Cloud Computing. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 83–104, doi: 10.31577/cai_2020_1-2_83.

[5] Zhang, B.—Ya Zhang, Bai, Y.—Lian, J.—Li, M.: A Multi-Dimensional Recommendation Scheme for Social Networks Considering a User Relationship Strength Perspective. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 105–140, doi: 10.31577/cai_2020_1-2_105.

[6] Pang, S.—Yu, S.—Li, G.—Qiao, S.—Wang, M.: A Time-Sensitive Collaborative Filtering Algorithm with Feature Stability. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 141–155, doi: 10.31577/cai_2020_1-2_141.

[7] Yuan, D.—Sun, H.: Reverse Intervention for Dealing with Malicious Information in Online Social Networks. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 156–173, doi: 10.31577/cai_2020_1-2_156.

[8] Li, J.—Yang, R.—Ding, Z.—Pan, M.: A Method for Learning a Petri Net Model Based on Region Theory. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 174–192, doi: 10.31577/cai_2020_1-2_174.

[9] Xiang, D.—Liu, G.: Checking Data-Flow Errors Based on the Guard-Driven Reachability Graph of WFD-Net. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 193–212, doi: 10.31577/cai_2020_1-2_193.

[10] Feng, X.—Han, D.—Tian, Y.: Analysis and Application of Min-Cost Transition Systems to Business Process Management. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 213–245, doi: 10.31577/cai_2020_1-2_213.

[11] Cheng, X.—Wang, C.—Li, J.—Bai, X.: Adaptive Fault Diagnosis of Motors Using Comprehensive Learning Particle Swarm Optimizer with Fuzzy Petri Net. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 246–263, doi: 10.31577/cai_2020_1-2_246.

[12] Teng, Y.—Du, Y.—Qi, L.: A Logic Petri-Net Based Repair Method of Process Models with Incomplete Choice and Concurrent Structures. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 264–297, doi: 10.31577/cai_2020_1-2_264.

[13] Zhu, S.—Sun, H.—Duan, Y.—Dai, X.—Saha, S.: Travel Mode Recognition from GPS Data Based on LSTM. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 298–317, doi: 10.31577/cai_2020_1-2_298.

[14] Wang, Y.—Liu, X.—Li, M.—Di, W.—Wang, L.: Deep Convolution and Correlated Manifold Embedded Distribution Alignment for Forest Fire Smoke Prediction. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 318–339, doi: 10.31577/cai_2020_1-2_318.

[15] Wang, W.—Wang, C.—Wang, Z.—Zhao, X.: An Improved PDR Localization Algorithm Based on Particle Filter. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 340–360, doi: 10.31577/cai_2020_1-2_340.

Maozhen Li is Professor in the Department of Electronic and Computer Engineering, Brunel University London, UK. He is also Visiting Professor of Tongji University, Shanghai, China. He received his Ph.D. from the Institute of Software, Chinese Academy of Sciences in 1997. His main research interests include high performance computing, big data analytics and intelligent systems with applications to smart grid, smart manufacturing and smart cities. He has over 180 research publications in these areas including 4 books. He has served over 30 IEEE conferences and is on the editorial board of a number of journals including journal of Computing and Informatics. He is Fellow of the British Computer Society (BCS) and the Institute of Engineering and Technology (IET).

Zhijun Ding is currently Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. He received the M.Sc. degree from the Shandong University of Science and Technology, Tai'an, China, in 2001, and the Ph.D. degree from Tongji University, Shanghai, China, in 2007. He has published over 100 papers in domestic and international academic journals and conference proceedings. His research interests are in formal engineering, Petri nets, services computing, and mobile internet. He is a Senior Member of IEEE since 2015.

Computing and Informatics, Vol. 39, 2020, 5–27, doi: 10.31577/cai_2020_1-2_5

FOG AND CLOUD COMPUTING ASSISTED IOT MODEL BASED PERSONAL EMERGENCY MONITORING AND DISEASES PREDICTION SERVICES

Zhancui Li

Department of Magnetic Resonance Surgery
The 960 Hospital of Joint Logistics Support Force of PLA
Shandong Taian, China
e-mail: lizc1027@163.com

Longri Wen*, Jimin Liu, Quanqiu Jia

Department of Information Engineering
Shandong University of Science and Technology
Shandong Taian, China
e-mail: wenlr51@sdust.edu.cn, jiaqq1995@163.com, skdljm@126.com

Chengri Che

Department of Medical College Thoracic Surgery
Affiliated Hospital of Yanbian University
Jilin Yanji, China
e-mail: ycrche@ybu.edu.cn

Chengfeng Shi, Haiying Cai

Department of Maternal Healthcare, Maternal and Child Health Hospital
Department of Cancer Prevention and Treatment Institute
Shandong Taian, China
e-mail: chengfengshi1980@163.com, haiyingcai1981@163.com

* Corresponding author

Abstract. Along with the rapid development of modern high-tech and the change of people's awareness of healthy life, the demand for personal healthcare services is gradually increasing. The rapid progress of information and communication technology and medical and bio technology not only improves personal healthcare services, but also brings the fact that the human being has entered the era of longevity. At present, there are many researches focused on various wearable sensing devices and implant devices and Internet of Things in order to capture personal daily life health information more conveniently and effectively, and significant results have been obtained, such as fog computing. To provide personal healthcare services, the fog and cloud computing is an effective solution for sharing health information. The health big data analysis model can provide personal health situation reports on a daily basis, and the gene sequencing can provide hereditary disease prediction. However, the injury mortality and emergency diseases since long ago caused death and great pain for the family. And there are no effective rescue methods to save precious lives and no methods to predict the disease morbidity likelihood. The purpose of this research is to capture personal daily health information based on sensors and monitoring emergency situations with the help of fog computing and mobile applications, and disease prediction based on cloud computing and big data analysis. Through the comparison of test results it was proved that the proposed emergency monitoring based on fog and cloud computing and the diseases prediction model based on big data analysis not only gain more of the rescue time than the traditional emergency treatment method, but they also accumulate lots of different personal healthcare related experience. The Taian 960 hospital of PLA and the Yanbian Hospital as IM testbed were joined to provide emergency monitoring tests, and to ensure the CVD and CVA morbidity likelihood medical big data analysis, the people around Taian city participated in personal health tests. Through the project, the five network layers architecture and integrated MAPE-K Model based EMDPS platform not only made the cooperation between hospitals feasible to deal with emergency situations, but also the Internet medicine for the disease prediction was built.

Keywords: EMDPS, DML, PHR, EHR, IoT, fog computing, cloud computing, APC model

1 INTRODUCTION

Along with the rapid development of high technology and the improvement of personal healthcare, human beings have entered the era of longevity. However, the sudden death brings great pain to the family. From the age cohort analysis, the order of death causes is different. The top three causes of death in the children (1–14 years old) cohort were IM (injury mortalities), CA (cancer) and congenital abnormalities, accounting for 74.28 %, in the young adults cohort (15–44 years old) were IM, CA and CVD (cardiovascular disease), accounting for 75.97 %, in the middle-adult and aged (over 45 years old) cohort were CVD, CA and CVA

(cerebral vascular accident), accounting for 88.07 % [1]. Among them, most of CA and congenital anomaly belong to genetic disease, so the proposed EMDPS (emergency monitoring and diseases prediction service) model focused on the IM, CVD and CVA emergency situations monitoring and diseases morbidity likelihood analysis.

At present, despite the continuous development of medical technology and better rescue services, hospitals cannot accurately predict the DML (disease morbidity likelihood) and there is no advanced corresponding method for sudden death. In order to provide emergency forecasting and advanced rescue services the EMDPS model was proposed, and it is composed of the following modules.

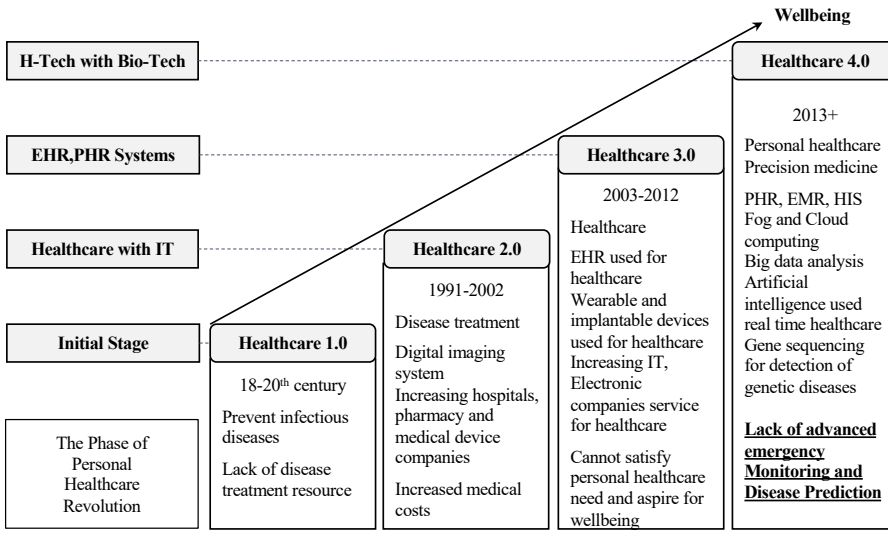


Figure 1. Revolution trend in China healthcare industry

The first is health information sensing module, the EMDPS model using PHR (personal health record) formation to capture information and screen emergency situation. The concept of PHR began to be used in 1978. PHR is personal health record generated in one's lifetime, it includes health-related information such as life log data, diagnostic record and genetic information and so on. Medical institutions use EMR (electronic medical record) and EHR (electronic health record) terms similar to PHR. EMR is a record used by a medical institution generated from medical information.

EHR is a generalized concept shared by multiple medical institutions [2]. The healthcare industry is developing rapidly, the healthcare technical level was upgraded from hospital-centered Healthcare 3.0 to current Healthcare 4.0 [3]. The difference between them is shown in Figure 1. At Healthcare 3.0, patients need to visit many hospitals and wait for a long time. Under the environment of Healthcare 4.0, the

mechanism of EHR, PHR and fog and cloud computing based real-time data capture and delivery method can solve the problems of Healthcare 3.0.

The second is fog computing assisted IoT (Internet of Things) based personal daily life health information capture and monitoring module. Recently, the main services using PHR have been extended to diagnostic records, exercise information, and gene sequencing based application fields. Along with the development of IoT, linking multiple healthcare devices can collect a variety of personal health information, and the integrated applications should provide more services for personal healthcare [2]. Fog computing has emerged as an active medical service solution because it contributes to the continuous monitoring of the health of remote patients and the detection of emergency situations. In addition, fog computing can reduce latency and communication costs, which is usually a huge problem in cloud computing. Fog computing is used to analyze, classify and share medical information between users and medical service providers [3]. The IoT based location position management model that makes use of the captured data resources could ensure the patient's personal information security and simplify the management. Even in emergencies, an efficient IoT healthcare service model can quickly respond using patient location information, so that hospital staff can locate patients in real time [4]. Subnet generation scheme, which collects and processes healthcare information to servers, provides a large amount of healthcare information through IoT devices connected by users. By assigning attribute values to the healthcare information sent to the server, a subnetwork is constructed according to the attribute values, and the related information between the subnetworks is extracted as seeds, and grouped into hierarchical structures. The server utilizes the deep operation of grouped medical information to extract optimized information and improve the observation speed and accuracy of decision making [5]. For the future Point-of-Care detection model the different sensing technologies were analyzed in detail, and that provided a path for the design and development of healthcare point detection device and data acquisition and processing in the future [6]. For the health information safe storage and accurate analysis, it is needed to transmit a large amount of captured data to the health cloud platform.

The third is cloud based health information delivery and secure store module. The personal healthcare cloud platform is a platform that can browse one's own health records ubiquitously and input health information and management independently. It is also a platform that can safely store and manage personal and family health information in one's lifetime. This platform not only provides a reliable technical basis for the capture and utilization of personal health information, but also provides a reliable experimental environment for the realization of precision medicine. A novel architecture for mobile group and cloud computing for healthcare could reduce costs, improve efficiency and reduce errors. At the same time, it could provide better consumer care and services for patients in the field of healthcare information to make them have universal transparency [7]. Cloud-MHMS (Cloud-based M-Health Monitoring System) puts the forward framework, which is used to achieve universal health information monitoring [8]. In

terms of storage management, data analysis and data security management of health information, there are advanced management models and algorithms proposed [9, 10].

The fourth is health big data analysis based emergency monitoring and disease prediction module. Personal healthcare platform as a new medical service technology or application tool not only improves the accuracy of diagnosis and disease prediction, but it is also improving the quality of life. Many data analysis methods have been applied in the field of disease prediction [11, 12]. Among them, the APC (Age Period Cohort) model and ANN (Artificial Neural Network) based disease prediction has made remarkable achievements. The APC model based on age, period and cohort analysis needs fewer original data attributes in disease prediction, but it better reflects the trend of health status than other models. The proposed APC model establishes a prospective cohort model by analyzing the three impact factors of test result for the DML prediction.

The last one is health situation visualization and rescue service module. It is the purpose of the research that captures personal health information in real time and provides healthcare services according to a personal health situation. The MPR (medication possession ratio) monitoring application is a supervisor medication related decision making method to enhance the analysis function of personal health records [13, 14], and it is a good reference for our research. In order to provide a personal healthcare more quickly, effectively and accurately, a personal health situation visualization EMS (Emergency Monitoring Services) and DPS (Disease Prediction Services) applications were proposed.

2 THE EMDPS NETWORK ARCHITECTURE

The fog and cloud computing assisted IoT architecture is a network scenario where everything is connected and uniquely identified over the global information and communication infrastructure [15]. The traditional IoT architectures can be decomposed into three layers, as shown in Figure 2.

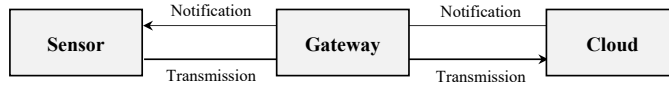


Figure 2. IoT-based system for a remote patient monitoring

The first layer is sensor, it is used for health information capture and delivery. The second layer is gateway, for the health information monitoring, it only acts as a relay between sensor and cloud. Gateway provides continuous, conventional and safe communication services with sensors using different network protocols such as Wi-Fi, ZigBee, and Bluetooth. The third layer is cloud, a broad health big data analysis, it safely stores amount of personal daily health sensory data and accurately predicts diseases morbidity likelihood. The latency is critical impact factor of IoT

network performance and the case core network was added in Cisco fog and IoT distributed architecture, as shown in Figure 3. The core network could provide paths to carry and transfer data and network information between numerous subnetworks and protect against network threats [16].

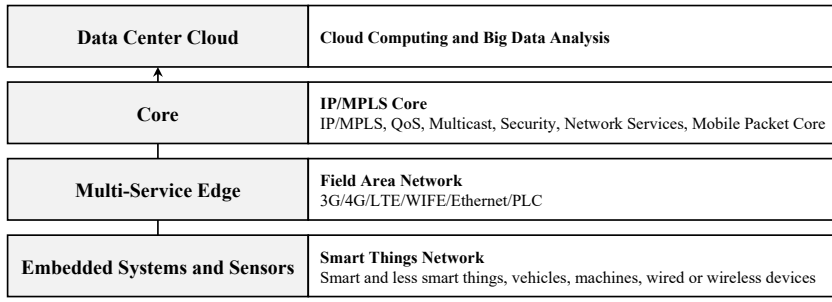


Figure 3. Cisco fog and IoT distributed architecture

Considering the impact factor of network latency in emergency situations and information feedback from the rescue service center, the mobile edge network layer is added to the proposed EMDP architecture, as shown in Figure 4.

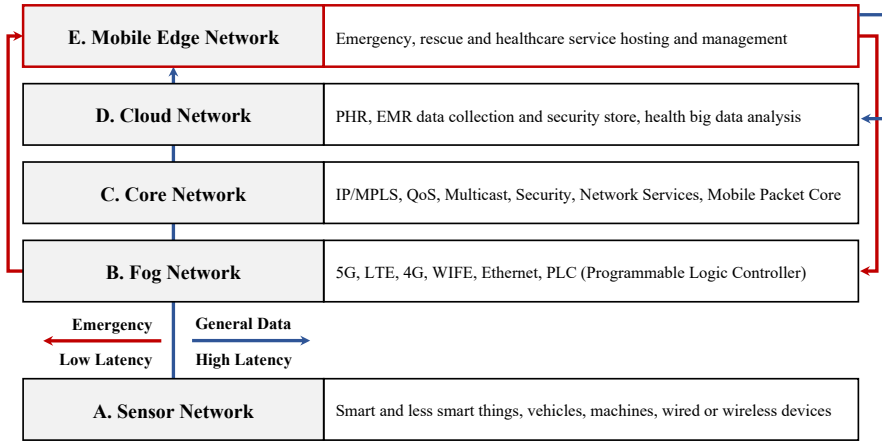


Figure 4. The proposed EMDPS network architecture

The EMDP architecture consists of four layers. The emergency ubiquitously monitoring finds a way in the fog network layer, the diseases morbidity likelihood is analyzed in the cloud network layer, and the mobile edge network layer is an actuator which provides the service feedback to the fog and cloud network layer.

2.1 Sensor Network Layer

The sensor device focused on arrhythmia detection for the IM, CVD and CVA emergency situations monitoring in the EMDP model, and PHR related information from the personal daily health information was captured by additional devices such as smart watch and flexible and stretchable physical sensors. Recently, flexible and stretchable physical sensors such as temperature, pressure, and strain sensors that can measure and quantify electrical signals generated by human activities are attracting a great deal of attention as they have unique characteristics [17]. There are ECG (Electrocardiogram) [18], RESP (Respiration), and NIBP (Non-Invasive Blood Pressure), SPO2 (Surplus Pulse O2) could be detected using smart and less smart sensor devices [19].

2.2 Fog Network Layer

Using the fog network core character of low latency and location awareness, the very large number of fog nodes could receive emergency information in real time, wireless, and heterogeneous ways from the sensor network layer and send it to the mobile edge network layer [21, 22, 23]. In this way, EMDPS model could save a lot of time to deal with emergency situations. For disease prediction, the fog layer sends periodic PHR to the cloud network layer using a standardized data format.

2.3 Core Network Layer

The core network is similar like in traditional networks and provides paths to carry and transfer data and network information between numerous subnetworks. The traffic profile is the critical variation between IoT and traditional core network layers [16]. The core network layer not only provides the best network from fog to cloud network layer but also provides QoS and data transmission security for the EMDPS model.

2.4 Cloud Network Layer

The main task of the cloud network is to receive the PHR from fog network and the EHR from the medical institutions, and store it safely in a standardized form and manage it. The cloud based PHR platform configuration was proposed in the previous research and offered the way how to provide healthcare services for aged cohort [23, 9, 10]. In order to ensure a more efficient transmission and network architecture the customized architecture [24], a novel architecture [7] and fine grained access architecture [25] for the cloud network were referenced for the EMDPS cloud network design. For the disease prediction, the APC (Age Period Cohort) model based health big data analysis services were provided in this layer.

2.5 Mobile Edge Network Layer

Recently, many mobile applications for the personal healthcare have appeared on the market. The IoT application healthcare system [26], a distributed movement prediction scheme [27] and the mobile phone based blood glucose management system [28, 8] were researched for our mobile application services.

At the emergency situation, the mobile edge network layer analyses received monitoring results from the fog network layer and delivers it to the family members, neighborhood service center and special mobile vehicle service center and nearby hospitals. Whenever received any acknowledgement from these service centers, the mobile application sends a feedback to the user and fog network layer as soon as possible.

Normally, it will receive periodical PHR and EHR analysis result from the cloud and provide health visualization service for the users. Because of the high morbidity likelihood of diseases such as CVD and CVA, it will send the related analysis reports and notify on the health situation the user medical institutions and family members. When any request from the user healthcare supporter is received, it sends a feedback to the cloud platform for the deep analysis and it decides on the further healthcare service.

3 THE EMDPS PLATFROM ARCHITECTURE

The IBM's MAPE-K model is an alternative computing model that provides automated management components for computational units and specifies system behaviors. The MAPE-K (Monitor-Analysis-Plan-Execute plus Knowledge) model is specified on the level of four different computing components: Monitor, Analyze, Plan, and Execute with the access to a partially or fully shared knowledge base, as shown in Figure 5. The Monitor captures health information from sensors, and it is the closest to the sensor devices, also it can determine events attributes [29]. The Analyze selects the data formation and analysis model. The Plan is in charge of selecting or generating a procedure for the system. The Execute provides necessary changes in the system and determines the behavior of the system [30].

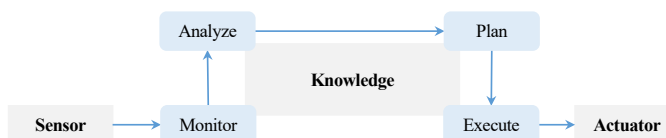


Figure 5. IBM's MAPE-K model

To fulfill the desired closed-loop behavior for resource management, the enhanced MAPE-K model was proposed in HICH (Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT) [15], in which a new component System

Management is integrated. The four MAPE-K components are enabled with feedback in the model. The feedback is received from Execute, System Management is used to periodically tune the computing components with respect to the inputs and the computations in the model, as shown in Figure 6.

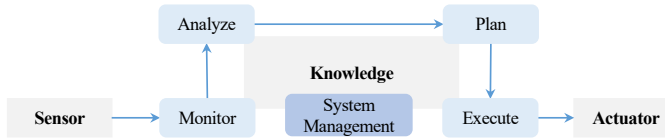


Figure 6. The HICH enhanced MAPE-K model

To efficiently deal with emergencies the mobile edge network layer was added to the EMDPS architecture. According to the characteristics of the proposed EMDPS architecture, the MAPE-K model was proposed, also based on the HICH model, integrating the Analyze separated by fog and cloud layer, and EMS and DPS as the Plan. As the result, it not only retains the MAPE-K model original function but also implements the HICH model system management function. Focused on the emergencies and network latency, the EMDPS model using distributed computing method replaced the original Plan components using EMS and DPS, as shown in Figure 7.

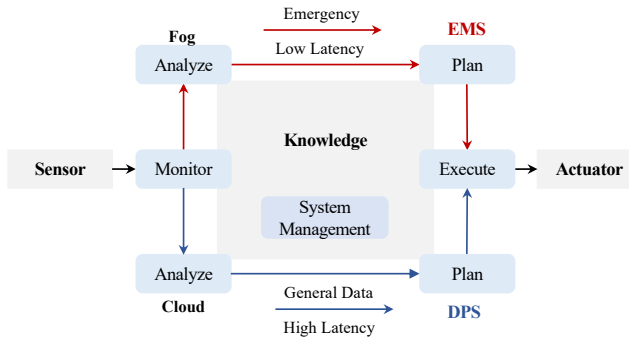


Figure 7. The EMDPS integrated MAPE-K model

3.1 EMDPS Platform Architecture

The proposed EMDPS platform is designed to provide a personal health care and related medical services. The service users are divided into three different groups. The main user group includes people and patients who want to receive their personal health care through the platform. The second user group is the family members and the personal health manager. The third user group represents medical institutions,

healthcare centers and emergency response service providers. All the platform users are connected and communicate through the web site and mobile applications.

The main user group are people who use the personal smart devices to detect physical fitness and healthy ingredients, such as BPG (Blood Pressure Gauge), BGM (Blood Glucose Meter), and so on. And these personal health information will be sent to the gateway server of the residence community health care center. The EMDPS platform partitions the health data analytics into two parts: the emergency data analysis running on fog nodes and PHR based diseases prediction analysis in the cloud.

The EMDPS platform provides EMS and DPS on mobile edge node and the knowledge is distributed to different layers. The EMDPS deploys EMS and DPS closed System Management base of emergency situations. The preselection could be determined according to the emergency monitoring parameters. The EMDPS Platform can be used to different resource system management, although the focus is on the network traffic management to efficiently deal with a personal health emergency situation and accuracy prediction of diseases morbidity likelihood, as shown in Figure 8.

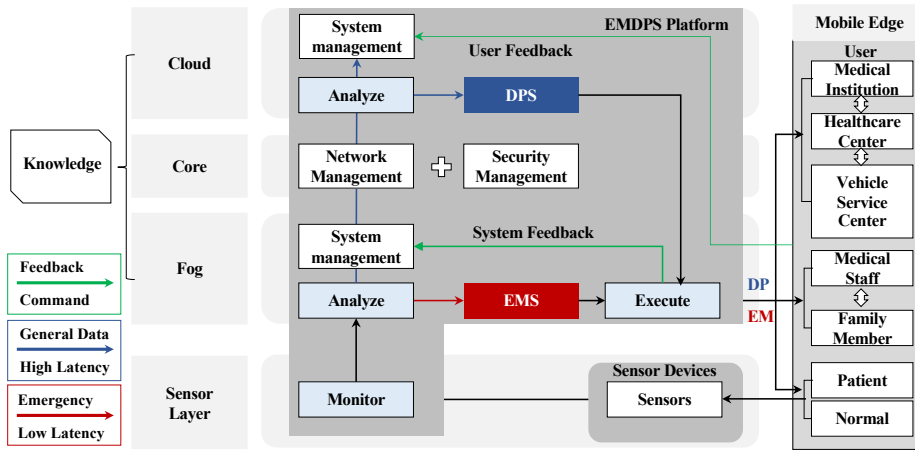


Figure 8. The proposed EMDPS platform configuration

In order to solve the problem of the network delay, the EMDPS platform uses the proposed five-layer network architecture. To deal with EM and DP situations correctly, the EMDPS platform uses integrated MAPE-K model.

3.2 Emergency Monitoring Method

The sensors could periodically capture personal health information and transmit the monitoring report to the gateway (fog layer), see the report format in Table 1. There are two types of monitoring reports from the Monitor component, the one is

emergency and the other is PHR. When it is an emergency, the emergency report with blood pressure, location and remarks will be sent to the fog node as soon as possible.

Manual	**Situation**		**Systolic Pressure**		**Remarks**	
Monitor	Emergency	PHR	High	Normal, Low	IM	Diseases
	1	0	1	0	1	0
	Diastolic Pressure		**Location**		**Family**	
	Low	Normal, High	System Located	Personal Located	Contact	None
	1	0	1	0	1	0

Table 1. Emergency report on sensor layer monitor

As soon as the fog receives the emergency report from the sensor layer, it will analyze the personal information mapping to the captured report information and send EMS as a plan to the mobile edge layer for the user, the report format is presented in Table 2. There are several previous researches for the fog computing and health data analysis [31], for the goal of an efficient emergency monitoring report and accuracy service, which defined several critical factors for the rescue.

Manual	**Situation**		**Blood Pressure**			**Location**		
Analyze	Emergency		Systolic	Diastolic	None	Map Add		
	1		sy	di	null	ad		
	Age		**Gender**			**EMS**		
	Age	None	Male	Female	Others	FM	HEC	VSC
	ag	Null	1	0	null	null	null	null

Table 2. Emergency report on fog layer analysis

In the emergency situation, EM method uses five impact factors for the rescue. The first is blood pressure, it will determine the user health situations of IM or disease. The second is gender, the third is age. The range of normal blood pressure varies according to sex and age factor and all the information could be helpful for rescue service. The fourth is user location information, it will be send to the service center in real time and used to contact rescue service staff and family member. The last one is EMS, it will contact FM (family member), HEC (hospital emergency center) and VS (vehicle service center) such as DIDI vehicle service center. The emergency monitoring report will also send system feedback to the cloud for the continuous rescue service and EMS mapping with platform PHR and EHR.

3.3 Disease Prediction Method

It is difficult to share personal EHR from the medical institutions, so the daily personal health information collected by EMDPS platform and medical reports col-

lected by medical research institutes are used as data sources for disease prediction test. The DPS focus on CVD and CVD morbidity likelihood prediction in this research to prevent sudden death from the diseases. In order to better share personal health information and easy to exchange utilization, the standardized PHR model is proposed based on general PHR, hospital EHR and medical reports; see the report format in Table 3.

Manual	**Situation**	**Gender**	**Age**	**Temperature**		**Puls Rate**
Analyze	PHR	Null	Null	Null		Null
General	0					(Times/M)
	Height	**Weight**	**BMI**	**Blood Pressure**		**Respiratory Rate**
	Null	Null	Null	sy	di	Null
	(cm)	(kg)	(Kg/m2)	Null	Null	(Times/Min)
Habits	**Physical Training**			**Eating**		
	1. Daily, 2. More than Once a Week, 3. Occasionally, 4. No Exercise.			1. Balanced, 2. Buckwheat, 3. Vegetarian, 4. Halophilic, 5. Oil Loving, 6. Sugar Tolerance.		
	Smoking			**Drinking**		
	1. Never, 2. Quit, 3. Smoking.			1. Never, 2. Occasionally, 3. Often, 4. Daily.		
Diseases	**CVD**			**Heart Disease**		
	1. None, 2. Ischemic Stroke, 3. Cerebral Hemorrhage, 4. Subarachnoid Hemorrhage, 5. Transient Ischemic Attack, 6. Others.			1. None, 2. Myocardial Infarction, 3. Angina Pectoris, 4. Revascularization of Coronary Artery, 5. Congestive Heart Failure, 6. Anterior Cardiac Pain, 7. Others.		
	CVA			**Others**		
	1. None, 2. Dissecting Aneurysms, 3. Arterial Occlusive Diseases, 4. Others.			Null		
Cases of Treatment	**Personal Medical History**					
	Cause of	Null				
	Family Medical History					
	Cause of	Null				
Table format based on National Standards for Basic Public Health Services (Third Edition). National Health and Family Planning Commission, Feb 2017.						

Table 3. Disease prediction report on cloud analysis

The APC (Age Period Cohort) is a generalized model proposed in 1939. It has three impact factors: age, period and cohort. The age factor impacts the results with the personal physiological changes and the accumulation of social experiences or social status changes. The period has impact on the result considering the lifetime and living surrounding. The cohort changes result from the cross-impact of the personal experience and social layer.

However, the APC model is a generalized linear model. There is a complete linear relationship between age and period and cohort variables, the period equals sum of the age and cohort value. The model design matrix is a singular matrix with non-full rank. The matrix is irreversible, so the unique solution of model parameters cannot be obtained. Therefore, as there is an "unrecognizable problem", a large number of parameter estimates exist. The method has been proposed [32],

in which IE (intrinsic estimator) proposed by Fu does not need prior information assumptions, and it is close to the results of traditional generalized linear model. The endogenous factor estimator is convergent and unique, which is suitable for APC model parameter estimation [33].

The APC model based on time series can achieve the goal of disease prediction through the coefficients of each cohort. The basic form of the APC model is Formula (1).

$$\ln[E(r_{ijk})] = \ln(\Theta_{ijk}/N_{ijk}) = \upsilon + \alpha_i + \theta_j + \gamma_k + \varepsilon_{ijk} \tag{1}$$

where $E(r_{ijk})$ is the expected value of disease morbidity likelihood for the age cohort (i), the period (j) and the birth cohort (k), the Θ_{ijk} represents the expected value of disease morbidity likelihood for the i age cohort observed in the j period, and N_{ijk} represents the population in the corresponding age, period and birth cohort. The υ is the intercept of the regression model, the α_i is the impact of the i^{th} age cohort, the θ_j is the impact of the j^{th} period, the γ_k is the impact of the k^{th} cohort and the ε_{ijk} is the random error.

The proposed disease prediction uses the IE integrated APC model and the 7 284 males' and 8 593 females' physical examination reports as the test sample data. In the test result, the CVA and CVD were marked as "2", and normal as "1". The whole test was carried out with STATA 15, and the final results are shown in Table 4.

In the Age cohort, both men and women have a tendency to increase the risk of disease with age, but women may suffer from CVA and CVD earlier than men, which is related to the fact that women are more likely to suffer from hypertension and other diseases. Overall, Age growth has a significant impact on the DML. In the Period cohort, we can see the DML downward trend, which means the personal healthcare service and the people's awareness of health is growing.

In the birth cohort, we see a more complex result. It means that the male's DML shows a growth trend, on the contrary, there is a downward trend regarding females. The results are closely related to the personal lifestyle, bad habits such as smoking and drinking can lead to diseases, females pay more attention to health. Despite of that a higher CVA and CVD morbidity likelihood was noticed. According to the PHR analysis, the CVA and CVD morbidity likelihood was obtained. The proposed APC model is more suitable for personal disease prediction, but even having the insufficient sample data will lead to prediction accuracy.

The traditional APC model can predict disease through the age, period and birth cohort. It is observed from the data set that the smoking and drinking is the influence factor causing CVA and CVD.

For the test, 500 male and 100 female data were selected from the source data. Each group of data consists of two different user data of the same age, period and birth, but one of them is smoking or drinking. Calculate the $\ln[E(r_{ijk})]$ of each data according to Formula (1), and then take the $\ln[E(r_{ijk})]$ difference of two different

Cohort	Male		Female	
	Coef.	OIM Std.Err.	Coef.	OIM Std.Err.
Age_40	−0.03007	0.06761	−0.04307	0.07887
Age_45	−0.02556	0.05431	−0.01834	0.05186
Age_50	−0.04382	0.05478	−0.04237	0.05198
Age_55	−0.0494	0.04889	−0.04465	0.05058*
Age_60	−0.05107	0.03423	0.00053	0.03582
Age_65	−0.0458**	0.02369	0.00703	0.0257*
Age_70	0.02934*	0.01697	0.015703	0.01706
Age_75	0.033***	0.0165	0.01408	0.01402
Age_80	0.04023*	0.02303	0.01339	0.019***
Age_85	0.091***	0.03575	0.02566	0.03144
Period_2005	0.028***	0.01247	0.01473	0.0127
Period_2010	−0.00376	0.00604	−0.00692	0.00538
Period_2015	−0.018***	0.01254	−0.015***	0.01234
Birth_1925	−0.01836	0.04652	0.00896	0.0464
Birth_1930	−0.03144	0.02871	0.02014	0.02917
Birth_1935	−0.03102	0.0194	0.01277	0.02001
Birth_1940	−0.0104	0.0177	0.0109	0.0181
Birth_1945	0.00385	0.02324	0.00641	0.02345
Birth_1950	0.03012	0.03204	−0.0296	0.03218
Birth_1955	0.0289	0.41947	−0.048***	0.04226
Birth_1960	0.01233	0.05335	−0.01994	0.05121
Birth_1965	0.123***	0.0585	−0.00241	0.05694
Birth_1970	0.1084**	0.05547	−0.066***	0.05755
Birth_1975	0.256***	0.10157	0.02922	0.0662
Birth_1980	0.22342	0.12878	0.07748	0.06619
(*** $p < 0.0001$; ** $p < 0.001$; * $p < 0.05$; OIM.std.err: The square root of Coef's variance. The Coef's calculates based on the observed information matrix in the maximum likelihood estimation.)				

Table 4. Disease prediction report on cloud analysis

user data in each cohort, and adjust the effect of these factors on the results using the Formula (2).

$$\varepsilon_{ijk} = \sum_{1}^{n} (\ln[E_A(r_{ijk})] - \ln[E_B(r_{ijk})])/n \tag{2}$$

where ε_{ijk} is the impact value of smoking on the results, $\sum_1^n(\ln[E_A(r_{ijk})] - \ln[E_B(r_{ijk})])$ is the result of different habit of n groups in the same period, age and cohort, and n is the total number of groups. The impact value of drinking on the result can be calculated in the same way.

The influence of single factor (smoking or drinking) and multiple factors (drinking and smoking) to the test result is not simply cumulative outcome. In this case, we selected 200 male and 100 female groups sample data for the test. The influence of smoking using Formula (3) and the influence of drinking using Formula (4) also the multiple factors using Formula (5). The influence of smoking and drinking on diseases is shown in Table 5.

$$\ln_{smoke} = \ln(\theta_{ijk}/N_{ijk}) = \upsilon + \alpha_i + \theta_j + \gamma_k + \varepsilon_{smoke}, \tag{3}$$

$$\ln_{drink} = \ln(\theta_{ijk}/N_{ijk}) = \upsilon + \alpha_i + \theta_j + \gamma_k + \varepsilon_{drink}, \tag{4}$$

$$\ln_{s+d} = \ln(\theta_{ijk}/N_{ijk}) = \upsilon + \alpha_i + \theta_j + \gamma_k + \varepsilon_{s+d}. \tag{5}$$

Habit	**Male**		**Female**	
	Coef.	OIM Std. Err.	Coef.	OIM Std. Err.
Smoke	0.10369	0.06192	0.12469	0.06197
Drink	0.09635	0.07362	0.10394	0.10377
Smoke + Drink	0.13681	0.09684	0.14332	0.08251

Table 5. The influence of smoking and drinking on diseases

Different prediction models can be selected by classifying the user data, but the accuracy has not been improved as expected after adding the intercept influence factors. The less control data groups of the same period, age and cohort cause the inaccuracy of the result simulation. At the same time, the lack of the number of women smoking and drinking control groups and the disproportion with men lead to the inaccuracy of the result. Today, the number of women smoking and drinking control groups is not enough.

4 EMPDS PERSONAL HEALTCARE SERVICES

The implemented healthcare services based on proposed EMDPS model are shown in Figure 9.

The EMDPS platform consists of two main services.

The one is the emergency monitoring service. In order to provide this service, the management service node has been setup in the campus big data center and hospital data center, and also the residential community center.

At first, wearable devices collect users' health information and the detected information will be automatically stored in the mobile by personal health care application. The application will intelligently judge the collected health information. Such as blood pressure, it will judge systolic blood pressure and diastolic blood pressure checking whether the measured values are in the normal range. If the blood pressure passes beyond the standard value reaching the hazard value then the application will trigger the emergency and sends the analyze request to the service node.

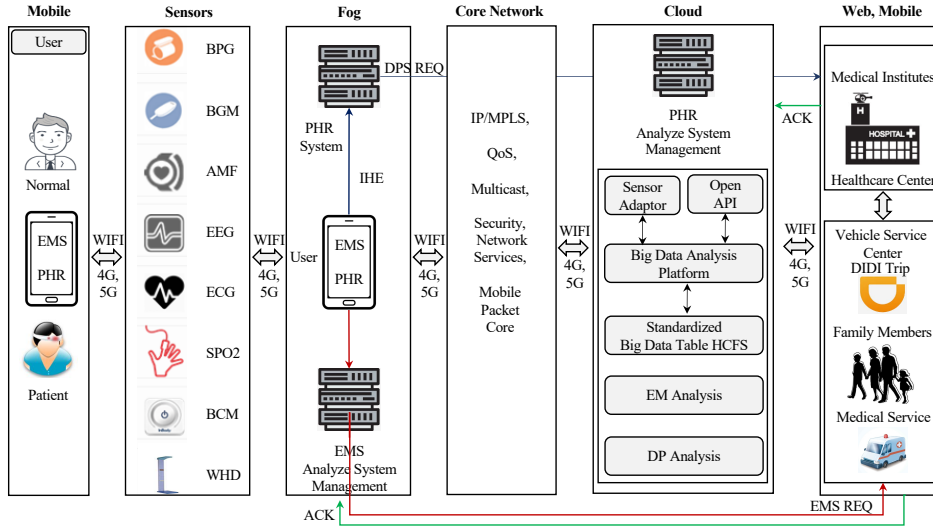


Figure 9. The proposed EMDPS healthcare service configuration

Secondly, when the fog service node receives EMS service it will send an EMS report to the medical institutes and a family member and also to a special vehicle service center.

At last, when the service center confirms the information, they will respond to the fog service node and request the stored user's personal health information and corresponding scheme from the cloud node.

The other one is the diseases prediction service. Because a large amount of a personal health information is collected every day and personal network conditions are different, it needs the support of cloud computing to safely store and analyze the daily health big data.

At first, the fog node analyzes the users' health information and if there is none emergency the collected data is sent to the cloud platform. Secondly, the collected personal daily health information will be modified to a standardized format and encrypted stored in the cloud storage. Finally, the results of APC based disease prediction model analysis will be sent to related medical institutions and to a personal user.

5 TEST RESULT COMPARISON

The EMS focuses on rescue time reduction, and the DPS is focused on diseases accurate prediction and effective personal healthcare services. So the test results are compared with the traditional services regarding the time of the rescue service and modified healthcare services.

5.1 Emergency Monitoring Test Results

The emergency treatment service is analyzed in the following four steps. At first, it is emergency detection. The second step is a call for help to medical institutes and contacting a family member, and also a special healthcare vehicle service. The third step is transportation by the ambulance to the hospital. The last step is preparation of the rescue. The statistical results of the time spent in the whole process are shown in Table 6.

In this test, the distance between CVD patient and hospital was five kilometers and when systolic blood pressure is higher than the standard value by 10 points or diastolic blood pressure is lower than the standard value by 10 points the emergency situation will be triggered.

Emergency Treatment	**Traditional Rescue Service**		**EMS Based Rescue Service**	
Emergency Detection	Seek help	>= 1 minutes	Sensors	<= 30 Seconds
Call for Help	1. Call 120 first aid 2. Contact family members 3. Special vehicle service	>= 1 minutes	Fog System	<= 30 seconds
		No service		<= 60 seconds
Moving (5 km)	Round Trip Delay	>= 30 minutes	Mobile APP, Nearly One Way Delay	<= 18 minutes
Preparation of Rescue	Situation analysis	>= 3 minutes	Fog System	Gear up
Total	Nearly 35 minutes		Hardly 19 minutes	

Table 6. Comparative analysis of rescue service time

5.2 PHR and EHR Based Healthcare Services

The structure of ontology based cohort DB was provided for the PHR and EHR services [34], and the smart sensor and mobile device based PHR general services and medical institutions providing EHR extended services are proposed, as shown in Figure 10. In this case the additional healthcare services can provide an effective emergency monitoring and diseases prediction services about a personal daily life for medical institutes.

6 CONCLUSIONS

The goal of this research is the cross-application of the advanced information and communication technology and medical technology to provide EMS and DPS, and use these services to reduce the sudden mortality rate.

The proposed EMDPS platform embodies three aspects of the technological advantages. The first one is the network service, the fog computing layer in the proposed network architecture can solve network latency problem, and the core layer provides advanced data security and delivery service, and the mobile edge computing layer provides real-time healthcare services at hand. The second one is the proposed

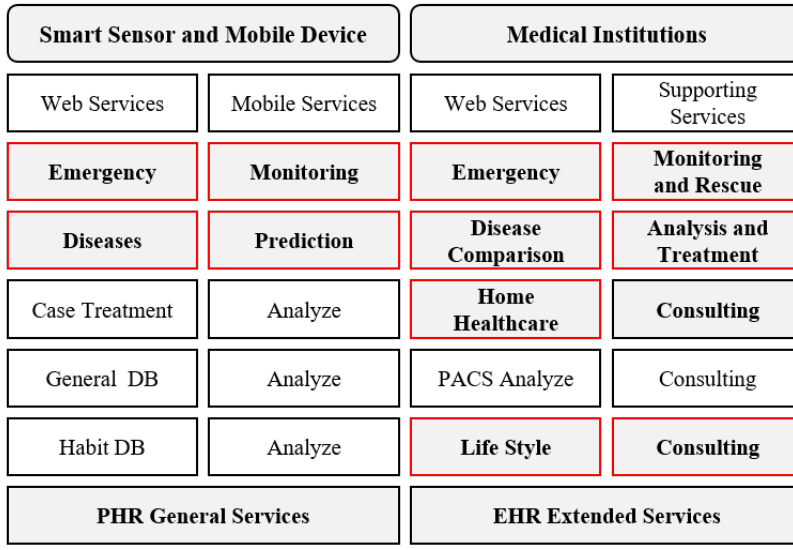


Figure 10. The PHR and EHR based healthcare services

EMDPS platform architecture, it is focused on the EM and DP services based on EMDPS integrated MAPE-K model. From the EM method, DP method, the plan management and the system management the platform has greatly advanced. The last one is mobile EMDP service model, it can provide real-time healthcare services ubiquitously.

Although the proposed EMDPS is a scientific platform advanced in specific services, still more detailed and deep research is needed to achieve the accurate disease prediction and prevention. Hence, the goal of the extended future research is to develop a biotechnology integrated healthcare platform to get closer to the precision medicine.

Acknowledgements

This paper was supported by the Scientific Research Foundation of Shandong Natural Science (No. ZR2013DMOII), Shandong Provincial Key R & D (No. 2016GGX10-5013) and Shandong University of Science and Technology for Recruited Talents.

REFERENCES

[1] Ding, S. M.: Shandong Annual Conference on Diseases and Health Status of Residents. The State Council Information Office of the People's Republic of China. Available at: http://www.scio.gov.cn, 2017 (in Chinese).

[2] Kim, D. H.—Kim, S. S.: PHR Related Recent Research Trend from the Smart Medical Platform Activity Point. Journal of Communication and Networks, Korea, Vol. 35, 2018, pp. 10–17.

[3] Kumari, A.—Tanwar, S.—Tyagi, S.—Kumar, N.: Fog Computing for Healthcare 4.0 Environment: Opportunities and Challenges. Computers and Electrical Engineering, Vol. 72, 2018, pp. 1–13, doi: 10.1016/j.compeleceng.2018.08.015.

[4] Jeong, Y. S.: An Efficient IoT Healthcare Service Management Model of Location Tracking Sensor. Journal of Digital Convergence, Vol. 14, 2016, No. 3, pp. 261–267, doi: 10.14400/JDC.2016.14.3.261 (in Korean).

[5] Jeong, Y. S.: Subnet Generation Scheme Based on Deep Learing for Healthcare Information Gathering. Journal of Digital Convergence, Vol. 15, 2017, No. 3, pp. 221–228, doi: 10.14400/JDC.2017.15.3.221 (in Korean).

[6] Lopez-Barbosa, N.—Gamarra, J. D.—Osma, J. F.: The Future Point-of-Care Detection of Disease and Its Data Capture and Handling. Analytical and Bioanalytical Chemistry, Vol. 408, 2016, pp. 2827–2837, doi: 10.1007/s00216-015-9249-2.

[7] Kumar, R.—Ganapathy, G.—Kang, J.: A Novel Architecture for Mobile Crowd and Cloud Computing for Health Care. International Journal of Advanced Culture Technology, Vol. 6, 2018, No. 4, pp. 226–232, doi: 10.17703//IJACT2018.6.4.226.

[8] Xu, B. Y.—Xu, L. D.—Cai, H. M.—Jiang, L. H.: Architecture of M-Health Monitoring System Based on Cloud Computing for Elderly Homes Application. Second International Conference on Enterprise Systems, Shanghai, China, 2014, pp. 45–50, doi: 10.1109/ES.2014.11.

[9] Sun, J.—Wang, X. J.—Wang, S. P.—Ren, L. L.: A Searchable Personal Health Records Framework with Fine-Grained Access Control in Cloud-Fog Computing. PLoS One, Vol. 13, 2018, No. 11, Art. No. e0207543, doi: 10.1371/journal.pone.0207543.

[10] He, X. Z.—Zhao, L.: A Data Management and Analysis System in Healthcare Cloud. 2013 International Conference on Service Sciences (ICSS), 2013, pp. 164–169, doi: 10.1109/ICSS.2013.27.

[11] Sun, Y. Y.—Wang, Y. H.—Li, M. M.—Cheng, K. L.—Zhao, X. Y.—Zheng, Y.—Liu, Y.—Lei, S. Y.—Wang, L.: Long-Term Trends of Liver Cancer Mortality by Gender in Urban and Rural Areas in China: An Age-Period-Cohort Analysis. BMJ Open, Vol. 8, 2018, Art. No. e020490, 8 pp., doi: 10.1136/bmjopen-2017-020490.

[12] Åström, F.—Koker, R.: A Parallel Neural Network Approach to Prediction of Parkinson's Disease. Expert Systems with Applications, Vol. 38, 2011, No. 10, pp. 12470–12474, doi: 10.1016/j.eswa.2011.04.028.

[13] Sargent, D. J.: Comparison of Artificial Neural Networks with Other Statistical Approaches: Results from Medical Data Sets. Cancer, Vol. 91, 2001, No. 8 Suppl., pp. 1636–1642.

[14] Perai, A. H.—Nassiri Moghaddam, H.—Asadpour, S.—Bahrampour, J.—Mansoori, G.: A Comparison of Artificial Neural Networks with Other Statistical Approaches for the Prediction of True Metabolizable Energy of Meat and Bone Meal. Poultry Science, Vol. 89, 2010, No. 7, pp. 1562–1568, doi: 10.3382/ps.2010-00639.

[15] Azimi, I.—Anzanpour, A.—Rahmani, A. M.—Pahikkala, T.—Levorato, M.—Liljeberg, P.—Dutt, N.: HiCH: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT. ACM Transactions on Embedded Computing Systems, Vol. 16, 2017, No. 5s, Art. No. 174, 20 pp., doi: 10.1145/3126501.

[16] Nandyala, C. S.—Kim, H. K.: From Cloud to Fog and IoT-Based Real-Time U-Healthcare Monitoring for Smart Homes and Hospitals. International Journal of Smart Home, Vol. 10, 2016, No. 2, pp. 187–196, doi: 10.14257/ijsh.2016.10.2.18.

[17] Trung, T. Q.—Lee, N. E.: Flexible and Stretchable Physical Sensor Integrated Platforms for Wearable Human-Activity Monitoring and Personal Healthcare. Advanced Materials, Vol. 28, 2016, No. 22, pp. 4338–4372, doi: 10.1002/adma.201504244.

[18] Wan, J.—Al-awlaki, M. A. A. H.—Li, M. S.—O'Grady, M.—Gu, X.—Wang, J.—Cao, N.: Wearable IoT Enabled Real-Time Health Monitoring System. EURASIP Journal on Wireless Communications and Networking, Vol. 1, 2018, Art. No. 298, 10 pp., doi: 10.1186/s13638-018-1308-x.

[19] Woo, M. W.—Lee, J. W.—Park, K. H.: A Reliable IoT System for Personal Healthcare Devices. Future Generation Computer Systems, Vol. 78, 2018, No. 2, pp. 626–640, doi: 10.1016/j.future.2017.04.004.

[20] Islam, M. S.—Islam, M. T.—Almutairi, A. F.—Beng, G. K.—Misran, N.—Amin, N.: Monitoring of the Human Body Signal Through the Internet of Things (IoT) Based LoRa Wireless Network System. Applied Sciences, Vol. 9, 2019, No. 9, pp. 1884–1901, doi: 10.3390/app9091884.

[21] Kharel, J.—Reda, H. T.—Shin, S. Y.: Fog Computing-Based Smart Health Monitoring System Deploying LoRa Wireless Communication. IETE Technical Review, Vol. 36, 2019, No. 1, pp. 69–82, 2019, doi: 10.1080/02564602.2017.1406828.

[22] Samarah, S.—Al Zamil, M. G. A.—Rawashdeh, M.—Hossain, M. S.—Muhammad, G.—Alamri, A.: Transferring Activity Recognition Models in FOG Computing Architecture. Journal of Parallel and Distributed Computing, Vol. 122, 2018, pp. 122–130, doi: 10.1016/j.jpdc.2018.07.020.

[23] Wen, L. R.—Yang, S. M.—Lee, B. M.: Cloud Platform Based Mobile Service for Aging Gerneration Healthcare Management. International Journal of Multimedia and Ubiquitous Engineering, Vol. 11, 2016, No. 11, pp. 235–246.

[24] Ganatra, N. P.—Patel, R. S.: Proposed Customized Architecture of Mobile Cloud Computing in Health Care Domain. International Journal of Advanced Research in Computer Science, Vol. 8, 2017, No. 5, pp. 876–879.

[25] Zhang, W.—Lin, Y. P.—Wu, J.—Zhou, T.: Inference Attack-Resistant E-Healthcare Cloud System with Fine-Grained Access Control. IEEE Transactions on Services Computing (Early Access), 2018, pp. 1–14, doi: 10.1109/TSC.2018.2790943.

[26] Isa, I. S. M.—Musa, M. O. I.—El-Gorashi, T. E. H.—Lawey, A. Q.—Elmirghani, J. M. H.: Energy Efficiency of Fog Computing Health Monitoring Applications. 2018 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, Romania, 2018, doi: 10.1109/ICTON.2018.8473698.

[27] Cao, G.—Liu, J.: An IoT Application: Health Care System with Android Devices. In: Gervasi, O. et al. (Eds.): Computational Science and Its Applications –

ICCSA 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9786, 2016, pp. 563–571, doi: 10.1007/978-3-319-42085-1_46.

[28] Zamanifar, A.—Nazemi, E.—Vahidi-Asl, M.: DMP-IOT: A Distributed Movement Prediction Scheme for IoT Health-Care Applications. Computers and Electrical Engineering, Vol. 58, 2017, pp. 310–326, doi: 10.1016/j.compeleceng.2016.09.015.

[29] Vizcarrondo, J.—Aguilar, J.—Exposito, E.—Subias, A.: MAPE-K as a Service-Oriented Architecture. IEEE Latin America Transactions, Vol. 15, 2017, No. 6, pp. 1163–1175, doi: 10.1109/TLA.2017.7932705.

[30] IBM Corporation: An Architectural Blueprint for Autonomic Computing. White Paper, Third Edition, 2005.

[31] Bonomi, F.—Milito, R.—Natarajan, P.—Zhu, J.: Fog Computing: A Platform for Internet of Things and Analytics. In: Bessis, N., Dobre, C. (Eds.): Big Data and Internet of Things: A Roadmap for Smart Environments. Springer, Cham, Studies in Computational Intelligence, Vol. 546, 2016, pp. 169–186, doi: 10.1007/978-3-319-05029-4_7.

[32] Chernyavskiy, P.—Little, M. P.—Rosenberg, P. S.: Correlated Poisson Models for Age-Period-Cohort Analysis. Statics in Medicine, Vol. 37, 2017, No. 3, pp. 405–424, doi: 10.1002/sim.7519.

[33] Fu, W. J.: Ridge Estimator in Singular Design with Application to Age-Period-Cohort Analysis of Disease Rates. Communications in Statistics – Theory and Methods, Vol. 29, 2000, No. 2, pp. 263–278, doi: 10.1080/03610920008832483.

[34] Wen, L. R.—Yang, S. M.—Lee, B. M.: Healthcare Platform and Big Data Analysis Based Personal Fitness Healthcare Service Model. International Journal of Bio-Science and Bio-Technology, Vol. 8, 2016, No. 5, pp. 115–128, doi: 10.14257/ijbsbt.2016.8.5.11.

Zhancui Li received her B.Sc. degree from the Military Medical University, Beijing, China, in 2003. She is currently working as a senior technologist at the Department of Magnetic Resonance Surgery, the 960 Hospital of Joint Logistics Support Force of PLA, Shandong Taian, China. Her main research includes medical image processing and medical imaging diagnosis.

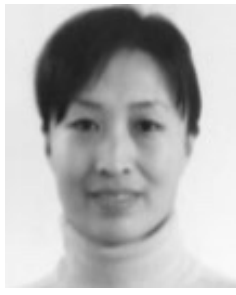
Longri Wen received his Ph.D. degree in computer science from the Soongsil University, South Korea, in 2017. He is currently working as Assistant Professor at the Shandong University of Science and Technology, Taian Shandong, China. His main research includes precision medicine, ICT and medical technology and bio technology integrated application.

Jimin Liu received his Ph.D. degree in geodesy and surveying engineering from the Shandong University of Science and Technology. He is currently working as the Director of the Information Engineering Department and Professor at the Shandong University of Science and Technology. His main research includes data mining and machine learning algorithms.

Quanqiu Jia received his B.Sc. degree in computer science from the Shandong University of Science and Technology. He is currently pursuing his M.Sc. degree at Shandong University of Science and Technology. His main research includes medical data mining and big data analysis.

Chengri Che received his M.Dr. degree in medical management from the Chungnam National University, South Korea in 2002. He is currently working as the Director of thoracic surgery and Professor at the Yanbian University Affiliated Hospital. He is the first batch of outstanding young and middle-aged talents of the Jilin province in China.

Chengfeng Shi received her B.Sc. degree from the Taishan Medical University, Shandong Taian, China, in 2010. She is currently working at the Department of Maternal Healthcare, Maternal and Child Health Hospital, Shandong Taian, China. Her main research includes personal healthcare and disease prevention technology.



Haiying Cai received her B.Sc. degree from the Jining Medical University, Jining Shandong, China, in 2008. She is currently working at the Department of Cancer Prevention and Treatment Institute, Shandong Taian, China. Her main research includes cancer disease prediction and disease prevention technology.

Computing and Informatics, Vol. 39, 2020, 28–50, doi: 10.31577/cai_2020_1-2_28

STOCHASTIC MODELING AND PERFORMANCE ANALYSIS OF ENERGY-AWARE CLOUD DATA CENTER BASED ON DYNAMIC SCALABLE STOCHASTIC PETRI NET

Hua He

School of Mathematics and Statistics, Shandong University of Technology
266 Xincunxi Road, Zhangdian District, 255000 Zibo City, Shandong, China
e-mail: huahe@sdut.edu.cn

Yu Zhao

Institute of Rural Development, Shandong Academy of Social Sciences
56 Shungeng Road, 250002 Jinan, China
e-mail: yuzhaosdass@foxmail.com

Shanchen Pang

College of Computer Science and Technology, China University of Petroleum
66 Changjiangxi Road, Huangdao District, 266580 Qingdao City, Shandong, China
e-mail: shanchenpang@sohu.com

Abstract. The characteristics of cloud computing, such as large-scale, dynamics, heterogeneity and diversity, present a range of challenges for the study on modeling and performance evaluation on cloud data centers. Performance evaluation not only finds out an appropriate trade-off between cost-benefit and quality of service (QoS) based on service level agreement (SLA), but also investigates the influence of virtualization technology. In this paper, we propose an Energy-Aware Optimization (EAO) algorithm with considering energy consumption, resource diversity and virtual machine migration. In addition, we construct a stochastic model for Energy-Aware Migration-Enabled Cloud (EAMEC) data centers by introducing Dynamic Scalable Stochastic Petri Net (DSSPN). Several performance parameters are defined

to evaluate task backlogs, throughput, reject rate, utilization, and energy consumption under different runtime and machines. Finally, we use a tool called SPNP to simulate analytical solutions of these parameters. The analysis results show that DSSPN is applicable to model and evaluate complex cloud systems, and can help to optimize the performance of EAMEC data centers.

Keywords: Stochastic Petri net, QoS, energy efficiency, performance evaluation, cloud computing

Mathematics Subject Classification 2010: 68M20

1 INTRODUCTION

Cloud computing can provide a convenient access to shared configurable resources (e.g. servers, storage, network, applications and services) to consumers by cloud providers directly deploying geographically distributed cloud data centers around the world [1]. As the important underlying infrastructure of cloud computing, the scale of data centers becomes lager, and has received increasing attention in the improvement of both performance and quality of service (QoS) requirements. But the research on energy consumption is still insufficient [2].

Statistically, the electricity energy consumption of data centers is estimated up to 40 % of total U.S. energy consumption, and the energy cost is accounted for 42 % of the total operating expense of data centers [3]. Hence, the improvements of energy efficiency are crucially important for cloud data centers. Cloud providers need to insure that their profits and return on investment are not rapidly falling owing to increased energy costs, while satisfying the QoS requirement of consumers based on service level agreement (SLA). In addition, improving energy efficiency can reduce resource consumption, release negative effects of environmental pollution, and achieve sustainable development in cloud data centers. Nevertheless, there still remain a range of challenges in realizing, modeling and performance evaluation resources scheduling of cloud data centers with the energy efficient way [2].

Firstly, the physical resources (e.g. PMs) in underlying infrastructure of cloud data centers are heterogeneous. It means that service capacities and energy consumptions can vary with the resource types. Secondly, based on virtualization technology, cloud data centers can provide multiple virtual machine (VM) instances on fewer PMs for multiple consumers simultaneously. Although energy consumption can be reduced by switching idle PMs off or to a low-performance levels (e.g. using DVFS), the performance may be significantly degraded when multiple VMs are running on the same PM in cloud data centers [4]. In other words, the key is to find an appropriate trade-off between energy efficiency and QoS guarantee. Moreover, VMs can dynamically migrate from one PM to another, which will help to improve resource utilization, realize load balancing, and decrease failure rate by avoiding hot

spots. Finally, but one of the most important aspects, the properties of cloud data centers (such as large-scale, dynamics, heterogeneity and diversity) make the system performance evaluation becoming more and more complicated. But for now, little attention has been the focus on how to provide an intuitive model description and effective analysis method for cloud data centers.

Stochastic Petri Net (SPN) is a graphic modeling and analysis tool for distributed systems [5]. DSSPN is an extension of SPN, which introduces enabling predicates and random switches to describe the firing conditions of immediate transitions. Compared with SPN, DSSPN can provide rich semantics to depict the scheduling process by allowing tokens and the labels on arcs to be expressed by a tuple $\langle R_k, PM_j \rangle$. In this paper, we can better model and evaluate some important parameters of cloud system by introducing DSSPN. Moreover, system bottlenecks can be well detected through oberservation and analysis token backlogs in places.

The state explosion is the main difficulty of Petri nets. DSSPN not only can effectively reduce the scale of state space by merging transitions and places with equivalent transformation, but the refined technology can dynamically adjust model based on enabling predicates and random switches according to run-time states of the system. That is, immediate transitions can be disabled/enabled by setting parameters of enabling predicates and random switches without the model reconstruction to realize scalability.

Based on the above discussions, this paper is dedicated to design and model resource scheduling for cloud data centers, which can realize energy efficiency and avoid the degradation of performance. The main contributions of this study are organized as follows:

1. We abstract a task scheduling and VMs allocation model of energy-aware migration-enabled cloud data centers ($EAMEC$).
2. In order to improve energy efficiency and ensure performance by avoiding hot spots in clusters, we put forward an Energy-Aware Optimization (EAO) algorithm.
3. Based on Dynamic Scalable Stochastic Petri Net (DSSPN), we establish the stochastic model of $EAMEC$ [6]. Furthermore, we evaluate some performance parameters (such as task backlogs, throughput, utilization, and energy consumption) of $EAMEC$ by adopting EAO algorithm.
4. To validate the proposed approach and algorithm, we conduct extensive experiments through simulations, and receive performance results under different resources or different runtime.

The rest of this paper is organized as follows: Section 2 briefly discusses the related literature. The system model formal description is discussed in Section 3. Section 4 constructs a stochastic model for $EAMEC$ data centers based on DSSPN, and proposes an EAO algorithm to enhance resource utilization and impair energy

consumption. Section 5 elaborates the parameters used in the performance analysis, and the experimental setup used for simulations is demonstrated in Section 6. Finally, we make a conclusion and discuss the future research.

2 RELATED WORK

Performance analysis usually concentrates on interrelation of system configuration, system load and performance indicators, which has already attracted some attention in the industry and academia. We divide the methods of performance analysis into three categories: measurement method, simulation method and model method.

Applying some measuring instruments, or measurement and simulation approaches, or measuring procedures, can directly attain the performance indicators and closely related quantities of systems. Then, performance indexes could be figured out by the corresponding calculation. An extensible cloud simulator CloudSim was proposed, which could evaluate the overall performance and also the energy consumption with taking into account I/O workload in a data center [7]. Based on CloudSim, a cloud framework CloudSimNFV was introduced to simulate several scheduling algorithms for resource allocation, and energy consumption was further evaluated [8]. Performance measurement framework (PMF) in virtualized cloud was studied to quantify the performance of profit and loss, and the significance of optimization for the application deployment was exposed [9]. The VM consolidation algorithm, which can consolidate VMs to PMs based on input task, was proposed to decrease the energy consumption by reducing the amount of active PMs, and is evaluated in CloudSim simulator to verify its effectiveness [10]. Based on Dynamic Voltage Frequency Scaling (DVFS), a cloud service framework with several power aware VM provisioning schemes was demonstrated to model the request process of VMs for real-time applications within data centers [11].

Measurement and simulation approaches are the most direct and effective way for the performance evaluation. However, the two approaches only can be applied to subsistent running systems, and are extremely time-consuming. Moreover, none of the two approaches is suitable for large scale and complicated cloud systems, especially involving numerous parameters in dynamic environments [6]. Therefore, neither of the measurement and simulation approaches is capable of finding out performance bottlenecks.

To overcome these challenges, some researchers propose some model methods to analyze and to evaluate the system performance. Based on network, a stochastic queuing approach is introduced to analyze the performance of migration-enabled clouds in error-prone environment, and to evaluate the performance metrics with different load conditions [1]. The cloud center is modeled as a M/G/m/m+r queuing system with single task arrivals and the finite task buffer, and a transformed analytical model based on Markov chain is proposed to obtain the probability distribution of the response time, blocking probability, and number of tasks [12]. The complex cloud system is divided into multiple submodels, and the interactive continuous

time Markov chain (CTMC) is introduced to study some important performance parameters of cloud data centers, such as task blocking probability, total waiting time, and time delay of users' service requests [13]. Multi-layered graph models are proposed to analyze various data center network (DCN) topologies and to compare the classic robustness metrics under different failure scenarios. In addition, based on the percentage change in the graph structure, a new metric named deterioration is also presented to quantify the DCN robustness [14].

Compared with the above works, we mainly focus on the energy-aware strategies for migration-enabled cloud data centers, and firstly introduce Dynamic Scalable Petri Net (DSSPN) to model and evaluate some important performance parameters (e.g. task backlog, average throughput, average reject ratio, resource utilization, and so on) of the proposed cloud system under different runtime and various quantities of PMs.

3 SYSTEM MODEL

The Energy-Aware Migration-Enabled Cloud (*EAMEC*) is a kind of green clouds, which can provide virtual configurable services on data centers by integrating themselves into networks. In addition, it can also increase the electricity efficiency in buildings, which accounts for about 40 % of the total energy consumption [15]. Figure 1 shows the process of task scheduling and VM provision in *EAMEC*.

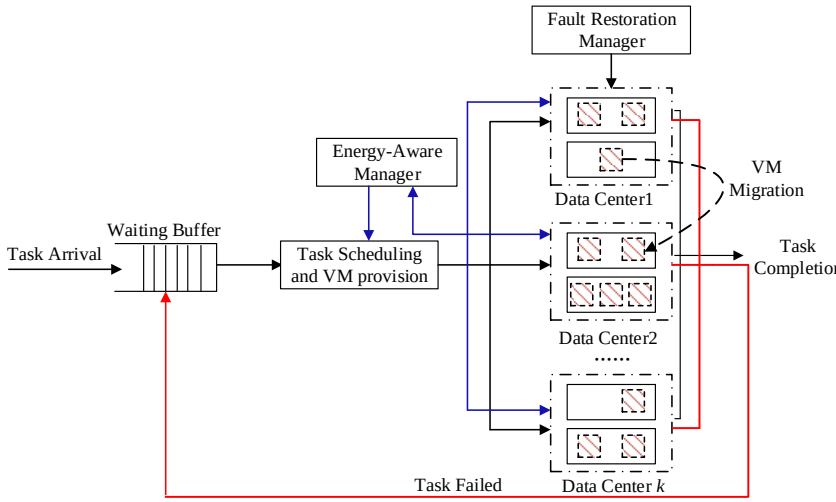


Figure 1. Task scheduling and VM provision in *EAMEC*

In this paper, the service model of the *EAMEC* system is the Infrastructure as a Service (IaaS). Without a loss of generality, we assume that the *EAMEC* system consists of k clusters (denoted as DC_i), each of which hosts np_i heterogeneous ma-

chines (i.e. PMs), such as high performance computers, workstations, and so on. Where, $k \in N^+$, $np_i \in N^+$, $i \in N^+$, $N^+ = \{1, 2, 3, \dots\}$. The the total number of PMs in the system is defined as follows:

$$tnp = \sum_{i=1}^{k} np_i. \tag{1}$$

Note that, in order to facilitate the analysis, it is assumed that these heterogeneous PMs (denoted as PM_j) have the same functionality, just for different capabilities in both CPU and memory. For example, a data center consists of two PMs, the CPU capability of each PM is 1 000 MIPS, while another might be 800 MIPS. Where, $j \in \{1, 2, \dots, tnp\}$, MIPS is the unit of CPU.

Each PM_j can host at most mv instances of VM simultaneously, and the amount of VM instances concurrently running on PM_j is denoted as nv_j. The total number of VM instances running in the system is defined as follows:

$$tnv = \sum_{j=1}^{tnp} nv_j. \tag{2}$$

The set of VM in the system is expressed by $VM = \left\{VM_j^{nv_j} | VM_1^1, \dots, VM_1^{nv_1}, \dots, VM_{tnp}^1, \dots, VM_{tnp}^{nv_{tnp}}\right\}$, where $j \in \{1, 2, \dots, tnp\}$. Let vpu_j^l indicate the processor utilization of l^{th} VM on machine PM_j, and pc_j expresses the processor capacity of machine PM_j. Where, $vpu_j^l \geq 0$, and $0 \leq \sum_{l=1}^{nv_j} vpu_j^l \leq pc_j$. Then the processor utilization PPU_j of machine PM_j during a given period of time is:

$$PPU_j = \left(\sum_{l=1}^{nv_j} vpu_j^l\right) / pc_j, \quad 0 \leq PPU_j \leq 1. \tag{3}$$

For the problem formulation, the following constraints are taken into consideration:

1. Tasks submitted to waiting buffer by users are independent, and they need to be allocated across the pool of VMs. The arrival rate of new tasks is λ, and it obeys the exponent distribution.
2. The capacity of waiting buffer is C, in which the tasks are served on the "first come, first served" (FCFS) basis. The VM is allocated in slots, each of which has the same length and denoted as ΔT. Where, $C \in N^+$, and $\Delta T \geq 0$.
3. Machines in the same cluster are homogeneous, while machines in different clusters might be heterogeneous. Each machine can be turned on or off, or configured to operate at low-performance levels (e.g. using DVFS) independently.
4. Each VM can be dynamically started and stopped on a PM according to the incoming tasks' requirements, and may lead to failure due to breakdown at runtime. Then failed VM can be repaired by the normal function. Let γ indicate

the failure rate, and η express the repair rate. In addition, both of them obey exponent distribution. Where, $\gamma \geq 0$, $\eta \geq 0$.

5. Once virtual machine VM_j^l malfunctions, the task running on VM_j^l will be broken down, and then resubmitted to the waiting buffer. The ratio of resubmission is $\beta\,(\beta \geq 0)$, which obeys exponent distribution.
6. The service rate of each VM_j^l is uniform, and is expressed by u_j^l. Note, that the capacity of VM_j^l is determined by PM. That is, if VM_j^l hosted on PM_j, and the service rate of PM_j is u_j, then $u_j^l = u_j$. Where, $l \in 1, 2, \ldots, nv_j$, and $j \in 1, 2, \ldots, tnp$.
7. Each PM applies Dynamic Voltage and Frequency Scaling (DVFS) to achieve an appropriate trade-off between energy efficiency and performance. All machines have two service levels with different service rates. For $\forall PM_j$, if $n\,(PM_j)\,/m_j < \delta$, the service level of PM_j is 1, that represents serving at normal mode. Otherwise the service level of PM_j is 2, with providing a lower service mode to save energy. Where, $n\,(PM_j)$ indicates the amount of VMs currently running on PM_j, δ expresses the control threshold, $0 < \delta < 1$ (see below).

In order to analyze properly, this paper only considers the energy consumed by processors. There are several reasons [1]:

1. In cloud data centers, the total energy consumption is determined by CPU, memory, disk storage and network components. Compared to other resources, the energy consumption of CPU is dominant in cloud data centers. Therefore, we focus on the energy consumed by CPU in this paper.
2. The energy consumption by machines can be accurately described by a linear relationship between the energy consumption and CPU utilization.
3. The main goal of this paper is to reveal how the energy-aware strategies influence the energy consumption in *EAMEC* data centers.

The studies have shown that the relationship between energy consumption and CPU utilization can be described by a linear function, even when DVFS is applied [16]. This is because that DVFS is only applied on CPU, which can adjust the voltage and frequency of CPU based on the number of states. In addition, these studies discover that the energy consumption of an idle PM is approximately 70 % of the power consumed under a fully utilization [16]. Hence, the PM can be switched to the leisure mode for energy conservation. The linear function is defined as follows:

$$P(u) = k \times P_{wm} + (1 - k) \times P_{wm} \times u = P_{wm} \times (0.7 + 0.3u). \tag{4}$$

Where, P_{wm} indicates the energy consumption under normally working condition, $k = 70\,\%$, while u is the CPU utilization.

As mentioned above, the energy consumption at peak rate is much higher than other rates. To reduce the energy consumption, VMs hosting on the PMs at the peak rate can be migrated to those PMs with other service rates. The VM migration

can avoid hot spots, and implement load balancing of cloud data centers. Moreover, it can also improve the resource utilization, and decrease the failure rate caused by machine errors.

4 STOCHASTIC MODELING BASED ON DYNAMIC SCALABLE STOCHASTIC PETRI NET

As discussed above, we introduce Dynamic Scalable Stochastic Petri Net (DSSPN) to model the process of task scheduling in *EAMEC* data centers. DSSPN is an expanded formation of Stochastic Petri Net (SPN) with similar firing rules and dynamics. For the limitation of space, we will not work it here in detail. Figure 2 shows a DSSPN model with abstract subnets of the task scheduling in an *EAMEC* data center. Figure 3 further describes the DSSPN model with detailed flow of each subnet. It should be noted that the scheduling or decision is expressed by the enabling predicates and random switches associated with the transitions.

4.1 DSSPN Model of EAMEC

The DSSPN model of the task scheduling in Energy-Aware Migration-Enabled Cloud is defined as (Figures 2 and 3): $EAMC = (P, T, F, K, W, \lambda, TS, G, E, f, g, M_0)$ where P is the set of places, T is the set of transitions, consisting of immediate transitions and timed transitions. F expresses the set of arcs, K indicates the set of capacities combined with places, W is the set of weights, λ is the set of average fired rates mapping to timed transitions. TS denotes the set of types, G is a function that maps places or transitions to types. E is a function to set values for types, f and g are enabling predicates and random switches associated with transitions, respectively. M_0 represents the initial marking which models the initial status of a system.

The elements in the DSSPN model *EAMEC* are defined as follows:

1. $P = \{p_{wq}\} \cup \{p_{mj}, s_{wmj}, s_{smj}, q_{wmj}, q_{smj}, s_{errj}, s_{resj}\}$. Where, P is a finite set, and $j \in \{1, 2, \ldots, tnp\}$.
2. $T_I = \{t_{mj}, t_{sj}, t_{wtsj}, t_{stwj}, t_{ij}, t_{errj}, t'_{errj}\}$. Where, T_I is the set of immediate transitions, $i, j \in \{1, 2, \ldots, tnp\}$, and $i \neq j$.
3. $T_T = \{t_c, s_{1j}, s_{2j}, t_{repj}, t_{resj}\}$. Where, T_T is the set of timed transitions, and $j \in \{1, 2, \ldots, tnp\}$.
4. The elements of finite set F, G and E signify arcs and labels on arcs, respectively (Figures 2 and 3). The detailed descriptions of K, λ, G, E, f, g and M_0 will be explained later.
5. $TS = \{R_k, PM_j, \langle R_k, PM_j >\}$, where $k \in N^+$, and $j \in \{1, 2, \ldots, tnp\}$.

The definitions of places and transitions included in $EAMEC$ are described as follows:

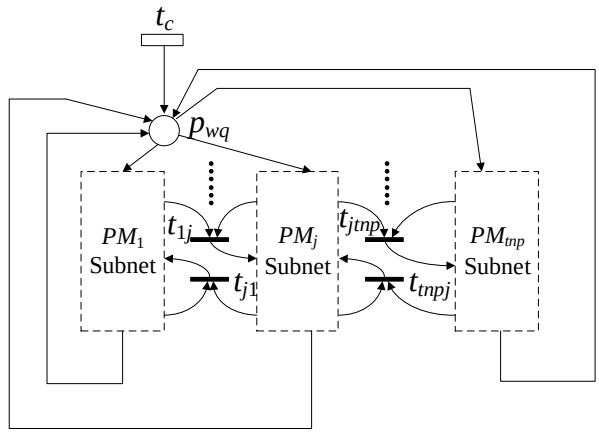


Figure 2. The DSSPN model of task scheduling in *EAMEC*

(a) t_c: a timed transition with exponent distribution. It expresses assignment requests of VMs submitted by users, and fires with rate of λ. Once t_c fired, there is a task entering the place p_{wq}. Its enabling predicate is $f(t_c)$:

$$f(t_c) : M(p_{wq}) < C. \tag{5}$$

(b) p_{wq}: a place expresses the waiting buffer, and is used to store task/VM provision requests. The capacity of p_{wq} is C, i.e., $p_{wq} = C$. For example, if $G(p_{wq}) = R_k$, then the type of each token in place p_{wq} is R_k, and $E(p_{wq}) \in N^+$ (indicates the size of tasks in the unit MB). That is, the attribute of tokens in a place are a tuple $\langle R_k, E(p_{wq})\rangle$. Where, $k \in N^+$, and $j \in \{1, 2, \ldots, tnp\}$.

(c) p_{mj}: a place corresponds to machine PM_j. The tokens in p_{mj} indicates the current amount of available VMs on p_{mj}. The initial marking is $M_0(p_{mj}) = nv_j$. In addition, $K(p_{mj}) = nv_j$, $G(p_{mj}) = PM_j$, and $E(p_{mj}) \in [1, nv_j]$.

(d) t_{wj}, t_{sj}: immediate transitions which are combined with place p_{mj}, p_{wq}, s_{wmj}, and s_{smj} to express allocation strategies of VMs for task requests in waiting buffer. Their enabling predicates and random switches will be described in the next section.

(e) s_{wmj}, s_{smj}: state places, representing the working states of machine PM_j. Where, s_{wmj} indicates PM_j working with normal state, while s_{smj} denotes PM_j working with leisure state. The machine provides different service rates under different states. All machines can provide different service rates under different states.

(f) t_{wtsj}, t_{stwj}: immediate transitions. Transition t_{wtsj} indicates that machine PM_j switches from normal state to leisure state, while t_{stwj} has the opposite

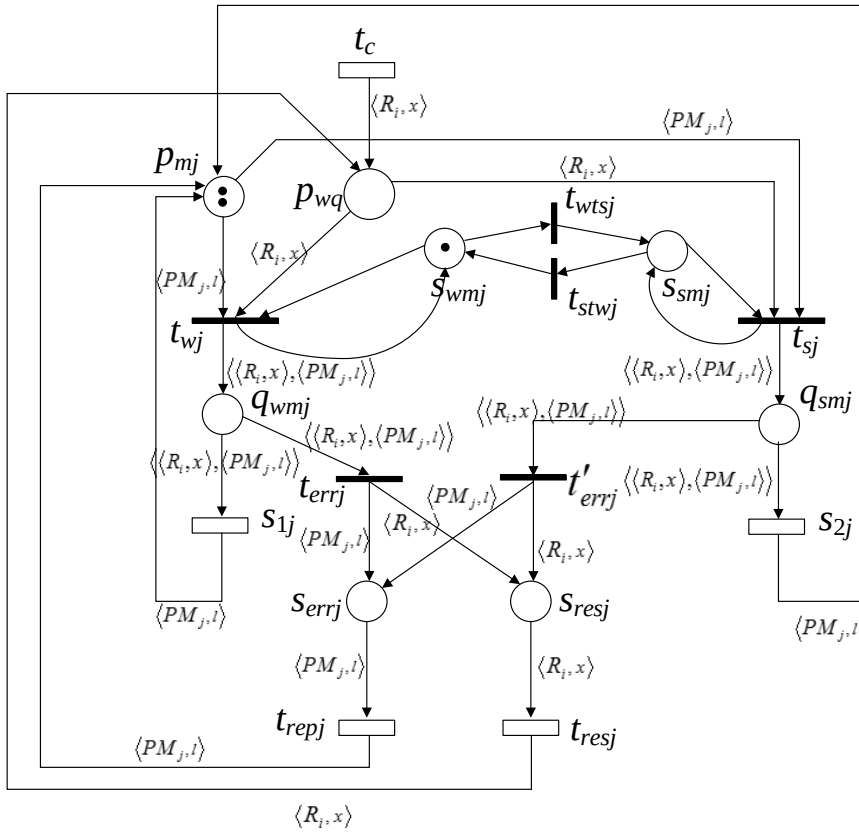


Figure 3. Detailed DSSPN model involves two subnets in EAMEC

meaning. The enabling predicates of t_{wtsj} and t_{stwj} are:

$$f\left(t_{wtsj}\right) : \left(M\left(q_{wmj}\right) = 0\right) \wedge \left(\frac{M\left(q_{wmj}\right) + M\left(q_{smj}\right)}{nv_j} < \delta\right), \tag{6}$$

$$f\left(t_{stwj}\right) : \left(M\left(q_{smj}\right) = 0\right) \wedge \left(\frac{M\left(q_{wmj}\right) + M\left(q_{smj}\right)}{nv_j} > \delta\right). \tag{7}$$

Where, δ is the threshold value for state transitions. When the number of VMs running on PM_j is less than $\delta \cdot nv_j$, the PM_j switches to leisure state in order to reduce energy consumption. Otherwise, the PM_j works with normal state to improve throughput and avoid the degradation in performance.

(g) q_{wmj}, q_{smj}: the tokens of q_{wmj} and q_{smj} indicate the amount of VMs running on PM_j working under normal condition and leisure condition, respectively.

(h) s_{1j}, s_{2j}: timed transition which expresses machine PM_j providing services with normal state or leisure state. The corresponding rate is exponent distribution, and $\lambda_{1j} = \mu_j$, $\lambda_{2j} = \mu_{sj}$, respectively. Where, μ_j denotes the service rate of PM_j with normal state, while μ_{sj} is the service rate with leisure state.

(i) s_{errj}, s_{resj}: places are used to buffer the failed VMs and resubmitted tasks due to machinery breakdown, respectively. Where, $K(s_{errj}) = n_j$, $n_j \in N^+$.

(j) t_{errj}, t'_{errj}: immediate transitions used to estimate whether faults of VMs occur. The failure rate is γ, the enabling predicates and random switches are:

$$g_{errj} = g'_{errj} = \begin{cases} \gamma, & \text{if } (M(S_{errj}) < K(S_{errj})) \wedge \left(M\left(s'_{errj}\right) < K\left(s'_{errj}\right)\right), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

(k) t_{repj}: timed transition indicates the resubmission of failed tasks, the resubmitted rate obeys exponent distribution, and $\lambda_{repj} = \beta$. Its enabling predicate is:

$$f(t_{resj}) : M(p_{wq}) < C. \tag{9}$$

4.2 VM Scheduling Algorithm

In cloud data centers, VMs are allowed to migrate from one PM to another, and can be completed in a very short time without suspending the services. However, VM dynamic migration will influence the performance of applications running on the VM. Moreover, the performance degradation and downtime during the migration process depend on behaviors of the application running on migrated VM, i.e., the capacity of memory is occupied by the application during execution [16]. The time depends on the capacity of memory occupied by the migrated VM and available network bandwidth. Since the overall objective of this paper is the energy consumption in cloud data centers, we only consider the energy consumed by migration. The energy consumption generated by machines in the leisure state is much less than that in the normal state. So only those VMs running on machines in normal state will be migrated, and the destination nodes are chosen from the machines in the leisure state. In this paper, we also propose an Energy-Aware Optimization (EAO) algorithm to allocate and migrate VMs in cloud data centers, shown in Algorithm 1.

The enabling predicates and random switches of transitions (Figures 2 and 3) are described as follows:

Algorithm 1 Energy-Aware Optimization (EAO) Algorithm

Input: the task requests in waiting buffer, the set of machines PM, the set of maximal VMs concurrently running on PMs nv, the set of available virtual machines VM, switching threshold δ, the set of working states S, the set of energy consumption in normal state $ER(PM(w))$, the set of energy consumption in leisure state $ER(PM(s))$, bandwidth B, and threshold δ;
for $j = 1$ to $|PM|$ **do**
  $count_j \leftarrow$the number of VMs running on PM_j;
  if $(\sum_{j=1}^{|PM|}(nv_j - count_j) \leq$ the number of task requests in the waiting buffer)
    no VM need to be migrated;
  else
    claculate the energy consumption EC_j of VM_j^l running on PM_j;
    for $i = 1$ to $|PM|$ **do**
      if $(i \neq j)$
        if $(((count_j + 1)/nv_j) < \delta)$
          calculate the energy consumption EC_i of VM_j^l running on PM_i;
          if $((TEC_j^l + EC_i) < EC_j)$
            migrate VM_j^l to PM_i;
if $(count_j < nv_j)$
    if $((count_j/nv_j) < \delta) \wedge (((count_j + 1)/nv_j)$
      choose an available VM from PM_j to allocate to the first task request in the waiting buffer;
if there is no match VM for above description
  for $j = 1$ to $|PM|$ **do**
    if $(count_j < nv_j) \wedge ((count_j/nv_j) < \delta)$
      choose an available VM from PM_j to allocate to the first task request in the waiting buffer;
else
  randomly choose an available VM to allocate to the first task request in the waiting buffer;

1. The enabling predicate and random switch of t_{wj} are respectively:

$$f(t_{wj}) : (M(s_{wmj}) = 1) \wedge \left(\frac{M(q_{wmj}) + 1}{nv_j} < \theta\right) \wedge \left(\sum_{j=1}^{tnp} M(s_{smj}) = 0\right) \wedge ((M(q_{wmj}) + M(q_{smj})) < nv_j), \tag{10}$$

$$g_{wj}(M) = \begin{cases} 1/|WEA(M)|, & j \in WEA(M), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

$$WEA(M) = \left\{ j | ((M(q_{wmj}) + 1)/nv_j) = \min_{\_} wec \right\}. \tag{12}$$

Where, θ is the upper threshold of machines, and $\min\_ wec$ in $WEA(M)$ is that $\min\{(M(q_{wmj})+1)/nv_j\}$, $j \in \{1,2,\ldots,tnp\}$, $|WEA(M)|$ indicates the amount of elements in $WEA(M)$. When the ratio of virtual machine on a machine is equal to or larger than θ, it is not allowed to create new VM instance on this machine.

2. The enabling predicate and random switch of t_{sj} are respectively:

$$f(t_{sj}) : \left(\left(\frac{M(q_{smj})+1}{nv_j} < \delta\right) \vee \left(\frac{M(q_{smj})}{nv_j} < \delta\right)\right) \wedge (M(s_{smj}) = 1), \\ \wedge ((M(q_{wmj}) + M(q_{smj})) < nv_j), \tag{13}$$

$$g_{sj}(M) = \begin{cases} 1/|SEA(M)|, & j \in SEA(M), \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

$$SEA(M) = \left\{ j \middle| \left((M(q_{smj})+1)/nv_j\right) = \min_{\_}\sec \right\}. \tag{15}$$

Where, δ represents the threshold of state transition, and $\min\_ \sec$ in $SEA(M)$ is $\min\{(M(q_{smj})+1)/nv_j\}$. $|SEA(M)|$ expresses the number of elements in $SEA(M)$.

3. t_{ij} combines with places q_{wmj}, q_{smj} and p_{mj} to describe the process of task scheduling and VM allocation, its enabling predicate and random switch are:

$$f(t_{ij}) : \left(M(p_{wq}) < \sum_{j=1}^{tnp} M(p_{mj})\right) \wedge (M(s_{smj}) = 1) \\ \wedge (M(q_{wmi})/nv_i > \delta) \wedge \left(\frac{M(q_{wmj}) + M(q_{smj}) + 1}{nv_j} < \delta\right) \\ \wedge \left(\left(\frac{M(q_{wmi})\cdot x}{B} + \frac{M(q_{wmi})\cdot x}{pc_j}\right) < \frac{M(q_{wmi})\cdot x}{pc_i}\right), \tag{16}$$

$$g_{ij}(M) = \begin{cases} \frac{1}{|MDP(M)|}, & \text{if } j \in MDP(M), i \neq j, \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

$$MDP(M) = \left\{ j \middle| \left(\frac{M(q_{wmj}) + M(q_{smj}) + 1}{nv_j} < \delta\right) \wedge (M(s_{smj}) = 1) \right. \\ \left. \wedge \left(\left(\frac{M(q_{wmi})\cdot x}{B} + \frac{M(q_{wmi})\cdot x}{pc_j}\right) < \frac{M(q_{wmi})\cdot x}{pc_i}\right) \right\}. \tag{18}$$

Where, $i \neq j$, and $i, j \in \{1,2,\ldots,tnp\}$. $|MDP(M)|$ expresses the amount of elements in $MDP(M)$.

5 PERFORMANCE ANALYSIS

As described above, we will further analyze the performance and energy consumption of *EAMEC* based on DSSPN model and the probabilities of stability. Compared with other approaches, such as Markov decision process, DSSPN can employ the integrated functions of Stochastic Petri Net Package (SPNP) to automatically deduce the probabilities of stability, without having to calculate them by stochastic math formulas. This is beneficial to model and evaluate the performance the cloud systems, because the number of states might reach thousands even if only consisting of few machines, shown in Table 1.

	1 Machine	2 Machines	3 Machines	4 Machines
Reachable states	283	569	1 088	1 594
Fired transitions	923	1 977	3 928	5 842

Table 1. Number of states and fired transitions

- At time t, the amount of running VMs on PM_j is:

$$nwv_j(t) = M\left(q_{wmj}(t)\right). \tag{19}$$

- At time t, the amount of idle VMs on PM_j is:

$$nsv_j(t) = M\left(p_{mj}(t)\right). \tag{20}$$

- At time t, the average queue length of system $AQL(t)$ is:

$$AQL(t) = \left(\sum_{y=0}^{t} M\left(p_{wq}(y)\right)\right)/t. \tag{21}$$

- At time t, the average throughtput of system $ATP(t)$ is:

$$ATP(t) = \left(\sum_{y=0}^{t}\sum_{j=1}^{tnp} nwv_j(y)\right)/t. \tag{22}$$

When the service transitions corresponding to machines are saturated, we can use the accumulating character of tokens in place p_{wq} to analyze the throughput of the $EAMEC$ systems. The average throughput is related to the capacity of waiting buffer, service rates of machines and the maximal number of available VMs.

- At time t, the average probability that machine PM_j works with leisure state $ASR_j(t)$:

$$ASR_j(t) = \left(\sum_{y=0}^{t} P\left(M\left(q_{wmj}(y)\right) \leq (\delta \cdot nv_j)\right)\right)/t \tag{23}$$

where $P\left(enabled\left(t_{erry}(y)\right)\right)$ is the enabling probability of error transition t_{errj} at time t, and $P\left(M\left(q_{wmj}(y)\right) \leq (\delta \cdot nv_j)\right)$ represents the probability that tokens are in place q_{wmj} is equal to or less than $\delta \cdot nv_j$.

- At time t, the utilization of machine PM_j is:

$$UR_j(t) = M\left(q_{wmj}(t)\right)/nv_j. \tag{24}$$

- Based on the Equation (5), we can deduce the average energy consumption of the system at time t:

$$AECR(t) = \frac{\sum_{y=0}^{t}\sum_{j=1}^{tnp}\left(0.7 + 0.3 \cdot UR_j(y)\right) \cdot P_{wmj}}{t \cdot tnp} \tag{25}$$

where P_{wmj} is the energy consumption of PM_j in normal state, and the unit is watt.

6 CASE STUDY AND SIMULATION

In this section, we make simulated experiments to study the applicability of DSSPN in the framework of *EAMEC* data centers. We consider a sample of data center on laptop with Intel i5-4210 multi-core processors. In addition, SPNP platform is used to automatically deduce the analytical solutions of performance for *EAMEC* model, as shown in Figures 4, 5, 6, 7, 8, and 9. The machines are depicted by places in DSSPN, the analysis results are calculated based on simulated analytical solutions of SPNP and the tokens in places. The processor specification is given in [18], which is illustrated in Table 2.

State Level	Normalized Service Rate	Energy Consumption
1	0.3333	0.279
2	0.5000	0.390
3	0.6666	0.570
4	1.0000	0.925

Table 2. Normalized specification for processors

The number of machines in the cloud data center varies from 1 to 4, the capacity of waiting buffer C is from 30 to 50, the threshold of switching transformation δ is 0.5. The arrival rate of task request is 20, the error rate is 0.2, the repair rate is 0.1, and the resubmitted rate of failed tasks is 0.3. In addition, we suppose that there are two classes of machines in the system. One can host 3 VMs at most with the state levels being 2 and 4. Another can host 2 VMs at most with the state levels being 1 and 3. For the convenient analysis, we assume that all rates obey the exponent distribution.

Figure 4 shows how the average queue length varies with the runtime t. When the system consists of 1 PM or 2 PMs, the arrival rate of tasks is larger than the

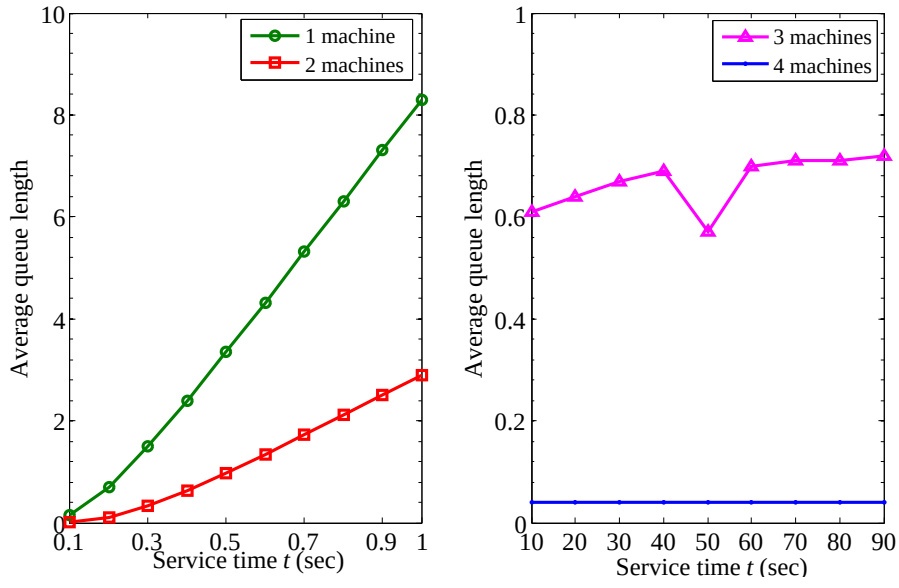


Figure 4. Simulation results of the average queue length

service rate of the system, so the queue length increases with the runtime t. And the queue length is gradually close to the capacity of the waiting buffer. When the system consists of 3 PMs or 4 PMs, and $0 \leq t \leq 1$, the tasks in the waiting buffer are less than the capacity of the system. At this point, there is no backlog of task requests in waiting buffer. In particular, Figure 4 b) illustrates that 3 PMs can satisfy the requirement of the system hardly without backlogs.

Figure 5 illustrates how the number of PMs affects the average throughput of the system. When the capacity of waiting buffer is 30, the arrival rate is 20 tasks per second, $0 \leq t \leq 1$, and the number of machines is 1 or 2, the throughput of system is limited by the processing capacity of the system. When the number of PMs is 3 or 4, the processing capacity of the system can satisfy the requirements of users without waiting.

Figure 6 shows how the number of PMs affects the average reject rate of the system. When the task requests are up to 30, the system will reject new requests submitted by users. When $1 \leq t \leq 10$, one machine can meet user demands, the average reject rate is up to 56.65 %, while the average reject rate is only 10.07 % in 2 machines scenario. In addition, the average reject rate increases over time.

Figure 7 shows how the number of machines impacts the average failure rate. When $1 \leq t \leq 10$, the average failure rate is higher, but gradually decreases over time. The figure also illustrates that the more machines in the system, the lower the

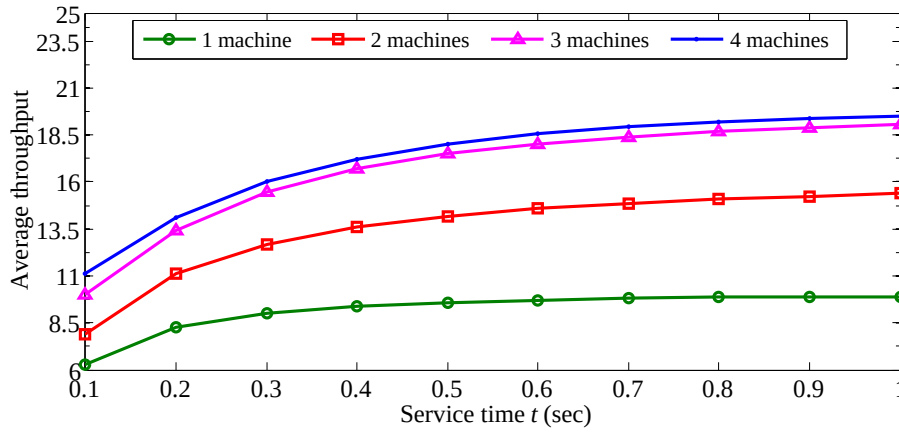


Figure 5. Simulation results of the average throughput

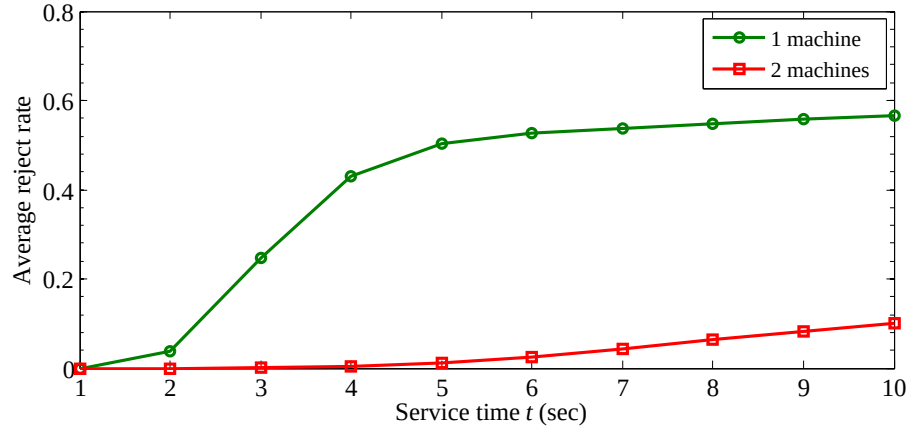


Figure 6. Simulation results of the average reject rate

average failure rate. This is due to the risk spread out by virtual machine dynamic migration.

Figure 8 shows how the number of PMs affects the average energy consumption of the sytem. When $0 \leq t \leq 1$, the average energy consumption is gradually close to a constant. The energy consumption increases with the number of machines included in the system. In 1 PM and 2 PMs scenarios, PMs provide services to peak. However, in 3 PMs and 4 PMs scenarios, the utilization of each machine is approaching a certain stability over time. In 3 PMs scenarios, machines serve in leisure state with the probability of 55.46 % on average, while in 4 PMs scenarios, the probability is 60.49 %.

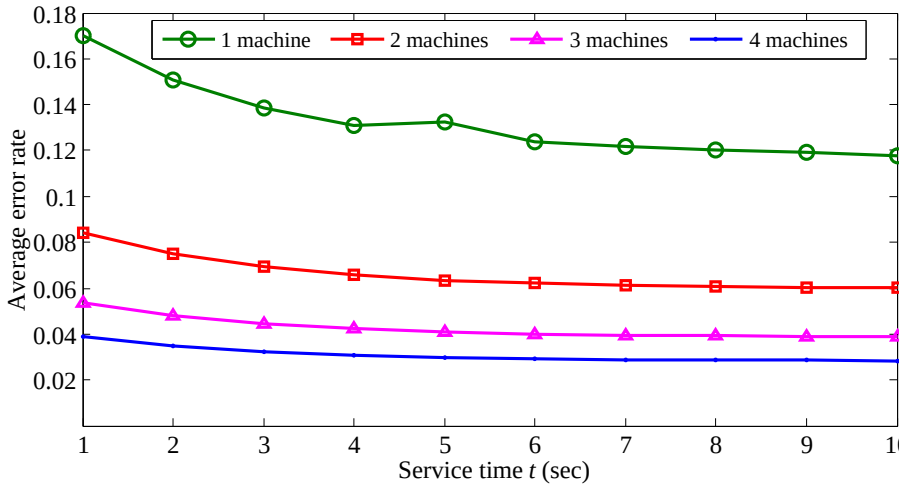


Figure 7. Simulation results of the average error rate

Figure 9 shows how the number of PMs affects the average resource utilization of the sytem. When $0 \leq t \leq 1$, the utilization in 1 PM scenario is the highest and up to 98.39 %. In 2 PMs scenario, the utilization is about 48 %, while the utilization is 69.04 % and 78 % for 3 PMs scenario and 4 PMs scenario, respectively. The reason for the reduced resource utilization of machines is that when the number

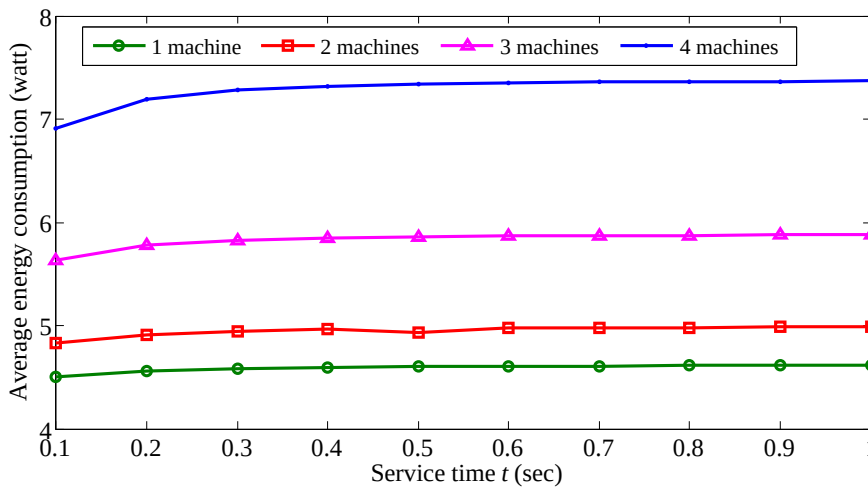


Figure 8. Simulation results of the average energy consumption

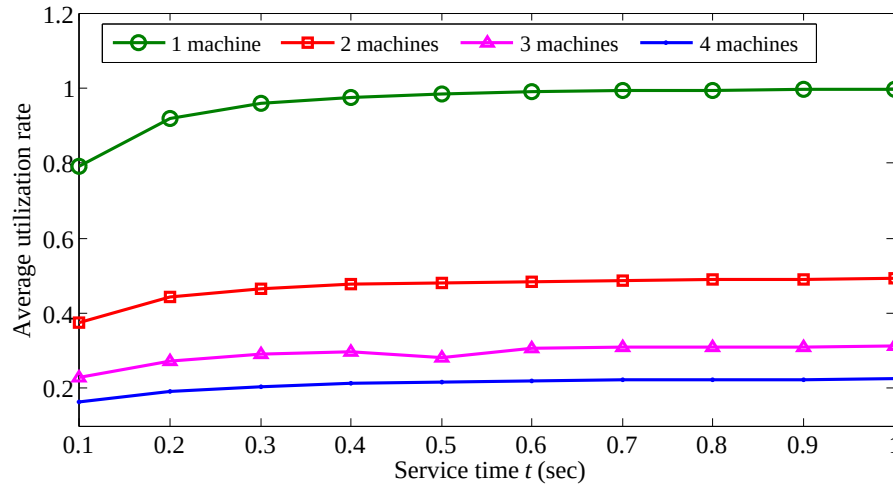


Figure 9. Simulation results of the average resource utilization

of machines increases the arrival rate of tasks and the capacity of waiting buffer remains unchanged.

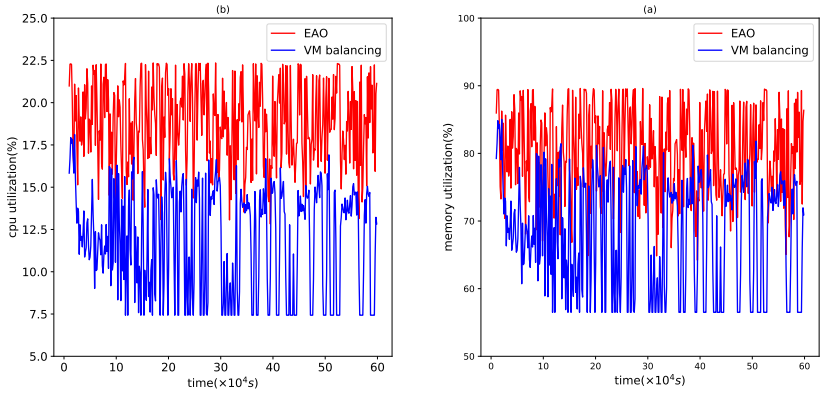
In the simulation, the queue length increases with the runtime t, and it is gradually close to the capacity of the waiting buffer. When the system consists of 3 PMs or 4 PMs, and $0 \leq t \leq 1$, the tasks in the waiting buffer are less than the capacity of the system. At this point, there is no backlog of task requests in the waiting buffer. That is, available resources are greater than the task demands under the maximum capacity of the waiting buffer. So the resource utilization rate goes up when more machines are used.

To demonstrate the effectiveness of EAO, we compare the approach with VM balancing based on the SCORE tool [19]. The VM balancing approach splits VM requests over multiple cloud datacenters, which can avoid the performance degradation by resource contention. But VM balancing causes the large energy consumption owing to a large number of active servers [20]. The experiment parameters are shown in Table 3.

	avgTaskPerJob	**avgJobDuration**	**avgCpuPerTask**	**avgMemPerTask**
Batch	30	50	0.3	0.2
Service	9	500	0.5	0.7

Table 3. Experiment parameters

Figure 10 shows the performance between EAO and existing VM balancing, at numMachines = 100, cpuPerMachine = 4, and memPerMachine is 1. Figures 10 a) and 10 b) show the average utilization of CPU and memory between EAO and the VM balancing during runtime, respectively. The minimal memory utilization of VM



a) b)

Figure 10. Experiment results between EAO and VM balancing

balancing is 49.49 %, but that of EAO is up to 64.20 %. Moreover, EAO has a 5.08 % higher maximum than VM balancing. The minimum of average CPU utilization for EAO is 12.82 %, but for VM balancing is 7.43 %. The maximum of average CPU utilization in EAO and VM balancing are 22.35 % and 18.12 %, respectively. Compared to VM balancing, the EAO approach achieves the improvements in resource utilization.

Scheduling	Shut-Down Policy	Runtime [s]	Energy Consumed [kwh]	Energy Saved [kwh]	Number of Jobs
EAO	always	604 800	1 314831	92.86	30 347
VM balancing	never off	604 800	1 470 019	0.00	27 918

Table 4. Energy-efficiency experiment

Except performance parameters, the energy-efficiency parameters are also included in the SCORE tool [19]. The results corresponding to Batch jobs between EAO and VM balancing are presented in Table 4. Note, that EAO approach further improves the performance of energy. Table 4 shows that the total energy consumption of EAO is approximately 10 % smaller than that of VM balancing. In addition, EAO completes more jobs than VM balancing at the same time. Our proposed approach takes into account both the energy consumption and resource utilization, while VM balancing only considers the performance of VM requests.

7 CONCLUSION

Based on DSSPN, this paper proposed a stochastic model to analyze the performance and energy consumption of *EAMEC* data centers, which can decrease systematic energy consumption by applying DVFS to CPUs. The *EAMEC* model depicts the logical relations among workload, failure and recovery of virtual machines, number of machines, VM dynamic migration and scheduling strategy. With the improving energy efficiency and increasing utilization, we proposed *EAO* algorithm to realize the dynamic migration of VMs. Then, we described the EAS strategy by setting enabling predicates and random switches of transitions in the model. Finally, a simulation tool called SPNP was introduced to work out the analytical solutions. The results could be used to analyze both the performance and energy consumption for cloud data centers. The simulation results also showed that DSSPN was convenient to model and evaluate complex cloud systems. Even if the states were up to thousands, we still obtained simulation results easily, without exhaustive complicated computations.

In the subsequent work, the authors plan to conduct their research on the model mechanism and formal semantics of dynamic Petri nets, and investigate the flexible modeling and dynamic optimization of service composition in large-scale mobile cloud system.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (grants No. 61702307, No. 61572523, and No. 61902222).

REFERENCES

[1] Xia, Y. N.—Zhou, M. C.—Luo, X.—Pang, S. C.—Zhu, Q. S.: A Stochastic Approach to Analysis of Energy-Aware DVS-Enabled Cloud Datacenters. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 45, 2014, No. 1, pp. 73–83, doi: 10.1109/TSMC.2014.2331022.

[2] Dinesh Reddy, V.—Gangadharan, G. R.—Rao, G. S. V. R. K.: Energy-Aware Virtual Machine Allocation and Selection in Cloud Data Centers. Soft Computing, Vol. 23, 2019, No. 6, pp. 1917–1932, doi: 10.1007/s00500-017-2905-z.

[3] Klass, A. B.—Wilson, E. J.: Energy Consumption Data: The Key to Improved Energy Efficiency. San Diego Journal of Climate and Energy Law, Vol. 6, 2015, pp. 69–115.

[4] Krzywda, J.—Ali-Eldin, A.—Carlson, T. E.—Östberg, P.-O.—Elmroth, E.: Power-Performance Tradeoffs in Data Center Servers: DVFS, CPU Pinning, Horizontal, and Vertical Scaling. Future Generation Computer Systems, Vol. 81, 2018, pp. 114–128, doi: 10.1016/j.future.2017.10.044.

[5] Molloy, M. K.: Discrete Time Stochastic Petri Nets. IEEE Transactions on Software Engineering, Vol. SE-11, 1985, No. 4, pp. 417–423, doi: 10.1109/TSE.1985.232230.

[6] He, H.—Pang, S.—Zhao, Z.: Dynamic Scalable Stochastic Petri Net: A Novel Model for Designing and Analysis of Resource Scheduling in Cloud Computing. Scientific Programming, Vol. 2016, 2016, Art. No. 9259248, 13 pp., doi: 10.1155/2016/9259248.

[7] Ouarnoughi, H.—Boukhobza, J.—Singhof, F.—Rubini, S.: Integrating I/Os in Cloudsim for Performance and Energy Estimation. ACM SIGOPS Operating Systems Review, Vol. 50, 2017, No. 2, pp. 27–36, doi: 10.1145/3041710.3041715.

[8] Yang, W.—Xu, M.—Li, G.—Tian, W.: CloudSimNFV: Modeling and Simulation of Energy-Efficient NFV in Cloud Data Centers. arXiv preprint arXiv:1509.05875, 2015.

[9] Bautista, L.—Abran, A.—April, A.: Design of a Performance Measurement Framework for Cloud Computing. Journal of Software Engineering and Applications, Vol. 5, 2012, No. 2, pp. 69–75, doi: 10.4236/jsea.2012.52011.

[10] Mishra, S. K.—Puthal, D. et al.: Energy-Efficient VM-Placement in Cloud Data Center. Sustainable Computing: Informatics and Systems, Vol. 20, 2018, pp. 48–55, doi: 10.1016/j.suscom.2018.01.002.

[11] Kim, K. H.—Beloglazov, A.—Buyya, R.: Power-Aware Provisioning of Virtual Machines for Real-Time Cloud Services. Concurrency and Computation: Practice and Experience, Vol. 23, 2011, No. 13, pp. 1491–1505, doi: 10.1002/cpe.1712.

[12] Khazaei, H.—Misic, J.—Misic, V. B.: Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems. IEEE Transactions on Parallel and Distributed Systems, Vol. 23, 2012, No. 5, pp. 936–943, doi: 10.1109/TPDS.2011.199.

[13] Khazaei, K.—Misic, J.—Misic, V. B.: A Fine-Grained Performance Model of Cloud Computing Centers. IEEE Transactions on Parallel and Distributed Systems, Vol. 24, 2012, No. 11, pp. 2138–2147, doi: 10.1109/TPDS.2012.280.

[14] Bilal, K.—Manzano, M.—Khan, S. U.—Calle, E.—Li, K.—Zomaya, A. Y.: On the Characterization of the Structural Robustness of Data Center Networks. IEEE Transactions on Cloud Computing, Vol. 1, 2013, No. 1, 14 pp., doi: 10.1109/TCC.2013.6.

[15] Clean Energy Australia Report 2015. Melbourne, VIC, Australia, Clean Energy Council, 2015.

[16] Kusic, D.—Kephart, J. O.—Hanson, J. E.—Kandasamy, N.—Jiang, G.: Power and Performance Management of Virtualized Computing Environments via Lookahead Control. Cluster Computing, Vol. 12, 2009, No. 1, pp. 1–15, doi: 10.1007/s10586-008-0070-y.

[17] Elbay, S. K.—Hegazy, I.—El-Horbaty, E.-S. M.: Live Migration Overhead-Aware Dynamic VM Consolidation Algorithm in Cloud Computing. Egyptian Computer Science Journal, Vol. 42, 2018, No. 4, pp. 75–88.

[18] Chandnani, L.—Kapoor, H. K.: Formal Approach for DVS-Based Power Management for Multiple Server System in Presence of Server Failure and Repair.

IEEE Transactions on Industrial Informatics, Vol. 9, 2012, No. 1, pp. 502–513, doi: 10.1109/TII.2012.2198656.

[19] Fernández-Cerero, D.—Fernández-Montes, A.—Jakóbik, A.—Kołodziej, J.—Toro, M.: SCORE: Simulator for Cloud Optimization of Resources and Energy Consumption. Simulation Modelling Practice and Theory, Vol. 82, 2018, pp. 160–173, doi: 10.1016/j.simpat.2018.01.004.

[20] Peng, Y.—Kang, D. K.—Al-Hazemi, F.—Youn, C.-H.: Energy and QoS Aware Resource Allocation for Heterogeneous Sustainable Cloud Datacenters. Optical Switching and Networking, Vol. 23, 2017, Part 3, pp. 225–240, doi: 10.1016/j.osn.2016.02.001.

Hua He received her Ph.D. degree in computer application technology from Tianjin University, China, in 2017. Her research interests include Petri nets, big data processing, performance analysis and cloud computing. She is currently Lecturer at Shandong University of Technology, Zibo, China.

Yu Zhao received his Ph.D. degree in Monte-Carlo modeling from the Brunel University London, Middlesex, UK, in 2014. His research interests include cloud computing, MapReduce. He is currently Assistant Research Fellow from the Shandong Academy of Social Sciences, Ji'nan, China.

Shanchen Pang received his Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2008. His research interests include theory and the application of Petri nets, service computing, trusted computing. He is currently serving as Professor at the China University of Petroleum, Qingdao, China.

Computing and Informatics, Vol. 39, 2020, 51–82, doi: 10.31577/cai_2020_1-2_51

OPTIMIZING DATA PLACEMENT FOR COST EFFECTIVE AND HIGH AVAILABLE MULTI-CLOUD STORAGE

Pengwei Wang, Caihui Zhao, Wenqiang Liu
Zhen Chen, Zhaohui Zhang

School of Computer Science and Technology
Donghua University
201620 Shanghai, China
e-mail: wangpengwei@dhu.edu.cn

Abstract. With the advent of big data age, data volume has been changed from trillionbyte to petabyte with incredible speed. Owing to the fact that cloud storage offers the vision of a virtually infinite pool of storage resources, data can be stored and accessed with high scalability and availability. But a single cloud-based data storage has risks like vendor lock-in, privacy leakage, and unavailability. Multi-cloud storage can mitigate these risks with geographically located cloud storage providers. In this storage scheme, one important challenge is how to place a user's data cost-effectively with high availability. In this paper, an architecture for multi-cloud storage is presented. Next, a multi-objective optimization problem is defined to minimize total cost and maximize data availability simultaneously, which can be solved by an approach based on the non-dominated sorting genetic algorithm II (NSGA-II) and obtain a set of non-dominated solutions called the Pareto-optimal set. Then, a method is proposed which is based on the entropy method to determine the most suitable solution for users who cannot choose one from the Pareto-optimal set directly. Finally, the performance of the proposed algorithm is validated by extensive experiments based on real-world multiple cloud storage scenarios.

Keywords: Data hosting, cloud storage, multi-cloud, multi-objective optimization, genetic algorithm

Mathematics Subject Classification 2010: 68-M02

1 INTRODUCTION

With the rapid development of Internet, mobile Internet, IoT and other related technologies [1, 2, 3], the explosive growth of data volume has become an important and challenging issue. From the statistical result, 8×10^5 PB of data were generated and replicated in the world by the year of 2000 and it is expected that this number will increase to 35 ZB by 2020 [4]. Storing such data volume has been an important and challenging issue for enterprises.

In recent year, cloud computing has become a popular computing paradigm for hosting and delivering services over the Internet [5, 6]. Cloud storage, the prominent service in cloud computing, is synonymous with pay-for-use pricing structures. Compared with the traditional storage mode, cloud-based data storage offers high availability, durability, and scalability. However, a single cloud-based data storage comes with the following risks.

Data unavailability. The first obstacle to the growth of Cloud Computing is the availability of a service [7]. The availability of data is also an indicator that users are most concerned about. Although cloud service providers (CSPs) have strict Service Level Agreement (SLA) for their services, some unpredictable events may cause services to be unavailable. These unpredictable events include server downtime, natural disasters, power failures, and so on. On August 8th, 2016, Google Cloud Storage and File Backup Server service terminals crashed, which brought huge economic losses to users. Coincidentally, in the afternoon of December 7$^{\text{th}}$, 2017, Alibaba Cloud's domain name resolution failed due to sudden large-scale traffic attacks. Unavailability of services can bring huge economic losses to users.

Vendor lock-in. Vendor lock-in is a major barrier to the adoption of cloud computing, due to the lack of standardization [8]. The vendor lock-in problem in cloud storage is the situation where users are vulnerable to price hike, availability decrement, or even to provider bankruptcy [7]. The reason why users give up migrating their data to a new provider who provides better service or lower price is the expensive bandwidth cost. Consequently, once a provider adjusts price, users are on the horns of a dilemma [27]. Moreover, the time it takes to migrate large amounts of data from one CSP to another is also huge [9].

Data privacy leakage. As data is stored with a third party, users want to avoid an untrusted CSP. If users put their data into a single cloud provider, their data is completely exposed to CSP, easily causing data privacy leakage. Data privacy leakage mainly includes cases where some untrusted CSP steal data without user permission. Malicious insiders of CSP can steal or corrupt the data and external attacks may also lead to data privacy leakage [10].

Recently, multi-cloud storage can mitigate the abovementioned risks with geographical providers and also provide benefits including adequate responsiveness, better load balance, and quick data recovery [11]. In multi-cloud storage, there

Symbols	Descriptions
C	List of cloud providers
N	Total number of cloud service
m	Number of the data splitting into
n	Number of the data storing
i	$i = 1, 2, \ldots, N$
S	Size of the file
P	Total cost of the data hosting scheme
P_{stor}	Total price of storage in scheme
P_{net}	Total out of out-bandwidth in scheme
P_{op}	Total price of operation in scheme
τ_t	Number of users access to the file
C_T	Total cost of a data file
C_i	Total cost of the i^{th} cloud service
P_{si}	Storage price of the i^{th} cloud service
P_{bi}	Out-bandwidth price of the i^{th} cloud service
P_{oi}	Operation price of the i^{th} cloud service
A_i	Availability of the i^{th} cloud service
A_{req}	Lowest limit of the data availability

Table 1. Symbol table

are many metrics with which users are concerned, especially low cost and high availability. The price of the same service across providers is different, and a provider offers different service with the same functionality while performance is directly proportional to price [11]. If users desire to enhance data availability, the more cost is incurred.

In multi-cloud storage, two main redundant strategies to categorize data distributed storage are replication and erasure coding [27]. For erasure coding, a data object is divided into m equal-size chunks and these chunks are used to generate $(n - m)$ encoded data chunks. Users can retrieve the original data through any m data chunks of these n chunks and tolerate any $0 \sim (n - m)$ cloud providers' shutdown at the same time. This strategy can reduce storage cost compared with data replication.

From a user's perspective, the key issue is to maximize data availability by minimizing data management cost that consists of *storage* cost and *network* cost (i.e., operation cost and out-bandwidth cost) [4]. The goal of optimizing multi-objective functions is to obtain the optimal data placement given cloud storage providers. In other word, the optimization problem is: **How to choose CSPs so as to minimize data management cost and maximize data availability?**

In this paper, an architecture in multi-cloud storage is presented at first. Next, a multi-objective optimization problem is defined to minimize monetary cost and maximize data availability. Then, an approach based on the non-dominated sorting genetic algorithm II (NSGA-II) [31] is given, whose goal is to effectively solve a multi-objective optimization problem and obtain a set of non-dominated solutions (i.e.,

list of cloud storage providers) and erasure coding parameters. Since CSPs with low (resp. high) storage cost may have low (resp. high) availability and high (resp. low) network cost, it is nontrivial to trade off data management cost and data availability. Some users can choose the data placement solution from the Pareto-optimal set directly. However, most users are still confused when they face the Pareto-optimal set. In order to recommend a suitable solution for such users, a method based on the entropy method is proposed. Finally, we demonstrate the performance of the proposed algorithm dealing with the real-world cloud storage providers from *CloudHarmony, cloudharmony* in a simulation.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents a cloud storage scenario to show the risks of reliance on a single cloud and discuss the benefits of multi-cloud storage. Section 4 formulates a data placement problem. The proposed algorithm is presented in Section 5. The performance of our proposed algorithms is shown via extensive experiments by using real-world cloud information in Section 6. Finally, Section 7 concludes the paper.

2 RELATED WORK

More and more users are hosting their data into multi-cloud not only to overcome the risks arising from reliance on a single cloud, but also to obtain higher availability, lower latency, and lower monetary cost [11]. Reducing monetary cost and enhancing data availability are two most important driving forces for users to host their data into the cloud. According to the optimized metrics, the existing studies can be divided into three categories, i.e., monetary cost optimization based on QoS metrics, data availability optimization based on QoS metrics, and cost-availability trade-off. In the following, we review and discuss them, respectively.

2.1 Monetary Cost Optimization Based on QoS Metrics

In this category, minimizing the monetary cost based on some QoS metrics is full or part of the work in the studies. Abu-Libdeh et al. [13] propose Redundant Array of Cloud Storage (RACS), a proxy striping user data across multiple providers to reduce the cost of switching providers. However, it does not propose a method to solve the data placement problem to meet any optimization goal. Papaioannou et al. [14] propose Scalia as inspired by RACS. It is a cloud storage brokerage solution for adaptive data placement, which minimizes the storage cost. Furthermore, Mansouri et al. [9] present an algorithm to find subsets of data centers to store original data and their replicas such that the storage cost is minimized while the expected availability is guaranteed.

But the above two tasks only consider a part of the monetary cost optimization: the cost of switching providers and storage cost. The cost of data storage management in the cloud should consist of residential cost (i.e. storage and data access operations), and network cost resulting from data transfer from CSP [4].

In [15], Hadji proposes a commodity flow solution to minimize the cost of storing data and latency to access data centers. However, the network and operation costs are ignored when users access their data. Ma et al. [16] adopt the ensemble of replication and erasure coding leading to low bandwidth cost, low storage cost, and low latency, but ignore the proper selection of CSPs. One function in CHARM, proposed by Zhang et al. [17], is to select the data placement configuration which contains several suitable clouds and an appropriate redundancy strategy to store their data with minimized data storage management cost and guaranteed availability. But the proposed algorithm to solve the minimization problem is a simple heuristic solution and cannot obtain the global optimal one. The studies in [15] and [17] only minimize the monetary cost at a certain point in the time slot. Mansouri et al. [18] propose the optimal offline algorithm to minimize the residential and migration costs in a time slot where the exact and known future workload is assumed.

There are also several studies to minimize the monetary cost based on QoS metrics for a geographical distributed cloud storage. Wu et al. [19] present a unified view of storage services in geographically distributed data centers called *SPANStore*. It aims to minimize the monetary cost and compute resources with the constraints of GET/PUT latencies, flexible consistency, and tolerate failures. Liu et al. [20] propose a multi-cloud service to minimize the payment cost while providing Service Level Objective (SLO) guarantee to customers. In order to minimize the payment cost, the authors propose a heuristic solution based on genetic algorithm to maximize the reservation benefit. However, these studies mainly focus on GET/PUT latency, this paper focuses on the optimization of data availability and monetary cost.

2.2 Data Availability Optimization Based on QoS Metrics

In addition to optimizing cost, increasing data availability is also an optimization objective in recent studies. As mentioned above, data replication and erasure coding are two main data redundancy strategies to improve data availability [11]. In [21, 22], the authors compare them. Here, we briefly introduce them.

Data Replication. Wei et al. [23] propose a novel model to capture the relationship between availability and the number of replica. It only calculates the minimal replica count for a given availability requirement instead of improving availability. DEPSKY, proposed by Bessani et al. [24], is a system that stores critical data with high availability through replication of the data on diverse clouds. Another algorithm in [9] is to provide the optimal data placement for chunks of an object across CSPs such that data availability is maximized under a given budget. It also prevents vendor lock-in through splitting a data object into multiple CSPs, but neglects how to determine the replica count. In [15], Hadji discusses the rational the number of chunks to be used to split the original data according to data center failure probabilities, number of replicas of each chunk, and expected data availability.

Erasure Coding. Mu et al. [25] introduce μLibCloud, a client-side library based on Apache libCloud to improve the data availability through erasure coding. But it cannot give any optimization model to provide an optimal data placement. In [13] and [14], the authors also choose erasure coding to enhance data availability and avoid vendor lock-in, but fail to provide the specific mathematical expressions. Zhang et al. [17] use erasure coding to store a data object and calculate data availability. Yet it is only a constraint in the optimization model. Similarly, Wang et al. [27] optimize the data availability through erasure coding.

2.3 Cost-Availability Trade-Off

The studies in the above two categories only minimize the monetary cost or maximize data availability based on some QoS metrics, but fail to optimize both at the same time.

Singh et al. [26] propose a secured cost-effective multi-cloud storage model to minimize the total cost of storing data, while maximizing QoS, but give many unreasonable assumptions and use cost minus QoS as the final optimization goal. Its experimental results are not convincing. Wang et al. [27] propose an ant colony algorithm-based approach to minimize the monetary costs and maximize data availability. However, for simplicity, the authors use the weights to calculate the integrated QoS value, which is the final optimization goal. This is not a true multi-objective optimization. Su et al. [28] propose a systematic model to formally formulate data placement in multi-cloud storage by using erasure coding. It can solve the data placement under complex requirements. However, in order to solve multi-objective optimization problem, the authors adopt Euclidean distance to obtain the best solution, in which the optimization weight for each objective is subjectively determined.

3 MOTIVATION

3.1 Cloud Storage Scenario

There are a large number of cloud service providers now providing storage services, and we select five most popular CSPs to obtain their pricing: Amazon S3, Microsoft Azure Cloud Storage, Alibaba Cloud Object Storage, Google Cloud Storage, and Century Cloud Object Storage, as shown in Table 2. We can see that there is heterogeneousness in the price of the same functional storage service provided by the same CSP in different regions. For example, Eastern Australia Microsoft Azure Cloud Storage has lower storage price but higher out-bandwidth price than that in Eastern USA and Northern Europe. The price models of the same functional storage service across CSPs are different. For instance, Amazon S3 in Oregon, USA, has lower storage price but higher GET request price than CenturyLink Cloud in Eastern USA.

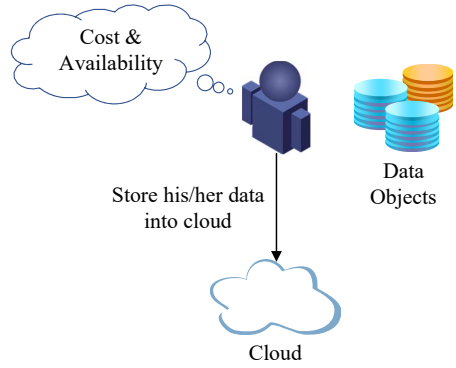


Figure 1. Cloud storage scenario

Today, more and more users are hosting their data into cloud to reduce the maintenance cost and improve data availability. Here, for clarity and conciseness, we abstract and simplify scenarios that users put their data into the cloud, as shown in Figure 1. It depicts that users store their data objects in the cloud with a series of requirement, which includes data access frequency, low monetary cost, high availability, and so on. Since users' data may contain common files, the user demands also include the data access frequency (DAF) to the data, that is, the number of times the data is retrieved within a unit period.

For example, users need to store 200 G files in the cloud, and require data availability not less than 99.99 %, and retrieve their data 0.3 times during a month. However, facing the complex cloud market, this user may choose a CSP with a lower storage cost and the availability greater than 99.99 %, i.e., Amazon S3 Paris. However, this choice is subject to higher out-bandwidth price than other CSPs when users retrieve their data, and also has risks like vendor lock-in, data unavailability, data privacy leakage, and so on.

3.2 Discussions

From the scenario, reliance on a single cloud has the abovementioned risks. Multi-cloud storage can mitigate these risks through distributing user data across multiple CSPs. Then, we discuss in detail the benefits if users in the above scenario put their data into multi-cloud.

Achieving high data availability. In the above scenario, users require data availability not less than 99.99 %. The availability of SLA in many CSPs is much greater than this value. However, it is common to hear that some well-known CSPs have experienced crash down at their data centers. As mentioned in the introduction, erasure coding and replication are two common data dispersion schemes in multi-cloud storage. Although replication can achieve higher availability than erasure coding, it also generates high storage cost. So in our paper,

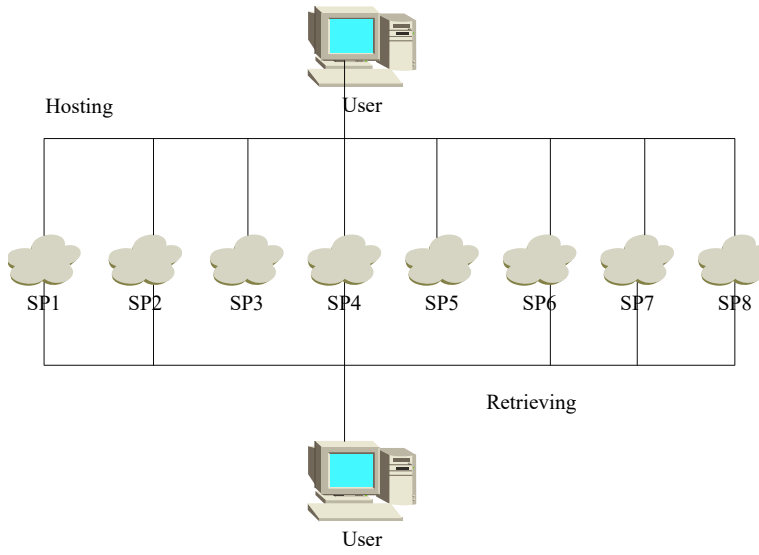


Figure 2. Erasure coding $(6, 8)$

we use erasure coding to improve data availability. As shown in Figure 2, users cannot tolerate more than 2 CSPs crash at the same time. We assume that the availability of CSPs in $(6, 8)$-erasure coding is to 99.99 %. According to the availability calculation method using erasure coding to propose in next section, the overall availability is 99.99997 %, which is larger than 99.99 % that is the availability of a single cloud.

Lowering data retrieving cost and avoiding vendor lock-in. The data retrieving cost consists of GET request cost and out-bandwidth cost. Due to the use of erasure coding, users can retrieve their data through the lowest request price and out-bandwidth price CSPs. For instance, assume that the above user puts their data into Amazon S3 in Oregon because of the low storage price, and out-bandwidth cost is $ 3. If the user puts their data into multi-cloud with a $(2, 3)$-erasure coding and CSPs are at Amazon S3 in Oregon, Azure in Eastern USA and Northern Europe, the data access can be satisfied by Azure in Eastern USA and Northern Europe and the out-bandwidth cost is $ 1.2. Apart from this, the most important phenomenon in the vendor lock-in is the high bandwidth costs brought by data migration when users face the bankruptcy of CSP or the emergence of a CSP with lower price and high availability or price hike of CSP. When such conditions emerge, users only need to pay for part of the entire data migration cost but are no longer subject to vendors.

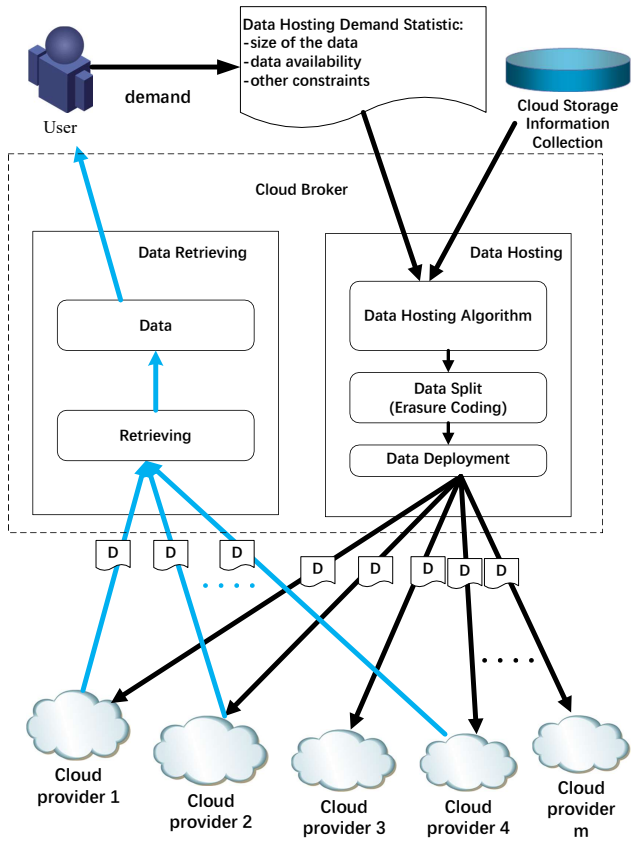


Figure 3. Multi-cloud storage framework

Protecting data privacy. As a result of erasure coding, each CSP only stores a chunk of user's original data object. Even if the cloud provider has malicious insiders or suffers external attacks, the attacker cannot recover user's original data object with a data chunk. To a certain extent, multi-cloud storage with erasure coding guarantees the privacy of user data.

Although multi-cloud storage has the abovementioned benefits, the trade-off between storage cost and retrieval cost and data availability brings a considerable challenge. Since high-availability CSPs impose enormous storage cost and retrieval cost on the user, it is a critical problem. Its solution answers how to choose suitable cloud storage providers and erasure coding parameters so as to minimize storage and retrieval cost while maximizing data availability.

CSP	Amazon S3			Microsoft Azure Cloud Storage			Alibaba Cloud Object Storage			CenturyLink Cloud		Google Cloud Storage
	Oregon	Seoul	Paris	USA East	Europe North	Australia East	China	USA West	Australia	USA	USA East	Asia Pacific
Storage price	0.0125	0.018	0.0131	0.0208	0.022	0.02	0.0226	0.02	0.0209	0.04	0.14	0.026
Out-bandwidth price	0.05	0.108	0.05	0.02	0.02	0.12	0.117	0.076	0.13	0.05	0.06	0.2
Get request price	0.004	0.0035	0.0042	0.004	0.0044	0.004	0.001	0.001	0.002	0.0	0.0	0.004

Table 2. Pricing of storage (in $/GB/month), out-bandwidth (in $/GB, and GET requests (in $/10K) of each CSP

4 SYSTEM MODEL AND PROBLEM DEFINITION

In this section, we briefly discuss problem statement, and then based on that, we formulate a data management model. Afterwards, we define a multi-objective optimization problem based on data management formulation.

4.1 Problem Statement

Figure 3 shows a scenario of how users put their data into multi-cloud storage. There are four components: *User Demand Statistic*, *Cloud Storage Information Collection*, *Data Retrieving* and *Hosting*. *Data Demand Statistic* collects user needs, which includes data size, required availability of data, and data access frequency. *Cloud Storage Information Collection* is used to collect the information of cloud providers from *CloudHarmony*, which is a third party website for collecting and monitoring cloud service information including the charges of services, attributes, services status, and so on.

Data Hosting and *Data Retrieving* are two core components in the framework. *Data Hosting* determines clouds in which the data should be deployed. *Data Retrieving* decides the clouds where the data of a user is to be retrieved from. These two components rely on the *erasure coding* which has been widely used in storage systems to provide high availability[17]. With the aid of (m, n)-erasure coding, the data object can be divided into m equal size chunks, and $(n - m)$ chunks can be encoded through m data chunks. The key property of *erasure coding* is that the original data can be recovered from any m data chunks [21]. In Figure 2, data is splitted and stored by $(6, 8)$-erasure coding, where any 6 of the 8 CSPs' data chunks can be used to recover the original data.

The primary objective of the above scenario is the optimization of data placement based on user needs, which is to choose CSPs and erasure coding parameters.

4.2 Problem Definition

To well describe a data management model, we introduce the following definitions. The symbols used in this article are listed in Table 1.

Definition 1 (Cloud Service Provider)**.** The data management model is represented as a set of independent cloud service providers $C = \{SP_1, SP_2, \ldots, SP_N\}$ where

each cloud service provider supplies the storage service. Each CSP has tuple: $CSP = \{P_{si}, P_{bi}, P_{oi}, a_i\}$, where:

1. P_{si} denotes the storage cost per unit size in CSP i;
2. P_{bi} is the out-bandwidth cost per unit size in CSP i;
3. P_{oi} defines the cost for GET requests in CSP i; and
4. a_i represents the probability of CSP i being available (i.e. availability).

Definition 2 (Data File)**.** We assume that a data file is related with a triples: $DF = \{S, \tau, A_{req}\}$, where:

1. S is the size of a data file that user stores;
2. τ denotes user data access frequency, which is equal the data access count during a time period; and
3. A_{req} defines user's required data file availability.

The objective is to choose CSP and erasure coding parameters (m, n) such that the total cost including storage and GET costs for data as well as the network cost is minimized; while the data availability is maximized. For simplicity, we assume that each CSP only stores one data chunk. It is worth noting that the following definition for availability and cost are similar to that in [17, 28], which is a universal way to define them for data hosting in erasure coding mode.

Definition 3 (Erasure Coding Parameters)**.** An (m, n)-erasure coding divides a data file into m equal size chunks, and encodes the m chunks into n ($n \geq m$) chunks which contain the m original equalized chunks and the $(n - m)$ parity chunks. Users can tolerate any $0 \sim (n - m)$ clouds simultaneously shut down.

Definition 4 (Data Availability)**.** Based on erasure coding, data availability is the sum of all cases that k CSPs are simultaneously available, where $k \in [m, n]$. This depends on the fact that outage occurrences are independent among CSPs [29]. We define $C' = \{SP_1 \times \mu_1, SP_2 \times \mu_2, \ldots, SP_N \times \mu_N\}$ ($|C'| = n$) as the service list of the n block choices, where $\{\mu_i \in \{0, 1\} \,|i = 1, 2, \ldots, N\}$, and μ_i is used to mark whether the i^{th} SP is chosen. $\Omega = \binom{|C'|}{k}$ means the number of cases that k cloud service providers are available, S_j^{Ω} denotes the j^{th} cloud services collection in Ω cases. The availability of the data file, denoted as A, can be calculated as follows:

$$A = \sum_{k=m}^{n} \sum_{j=1}^{\Omega} \left[\prod_{i \in S_j^{\Omega}} a_i \prod_{i \in C' \setminus S_j^{\Omega}} (1 - a_i) \right] \tag{1}$$

where $C' \setminus S_j^{\Omega}$ represents the CSPs that are not in S_j^{Ω}.

Definition 5 (Storage Cost)**.** The storage cost of a data file is equal to the storage cost of all data chunks in n CSPs. Since each CSP stores the data chunk of size

S/m, it can be defined as follows:

$$P_{stor} = \sum_{i \in C'} \frac{S}{m} P_{si}. \tag{2}$$

In fact, some CSPs use a tiered pricing scheme for storage. Taking AWS S3 in USA East as an example, the storage price is \$ 0.023 if the data size is less than 50 TB, and when the data size is between 50 TB and 450 TB, the storage price is \$ 0.022 [30]. Since we adopt erasure coding to divide data object in this work, the size of each data chunk is not too large. However, in the case that the data size is very large, we can use the threshold of each tier to calculate the storage cost, which is similar to the piecewise functions.

Definition 6 (Network Cost)**.** Due to erasure coding, users can retrieve the data file through any m data chunks from n clouds. In order to minimize the total network cost, we choose the m-cheapest clouds for data retrieving. It can be solved as follows:

$$P_{net} = \min_{j \in [1,\Omega]} \left(\sum_{i \in S_j^{\Omega}} \frac{S}{m} \tau_t P_{bi} \right). \tag{3}$$

Definition 7 (Operation Cost)**.** The operation cost is the cost of users' GET requests for retrieving the data file from the cheapest m CSPs. Thus, it can be calculated as follows:

$$P_{op} = \min_{j \in [1,\Omega]} \left(\sum_{i \in S_j^{\Omega}} \tau_t P_{oi} \right). \tag{4}$$

It is worth noting that the value of j in Equation (3) is equal with that in Equation (4).

Definition 8 (Total Cost)**.** The total cost of a data file C_T is the sum of *storage cost*, *operation cost*, and *network cost* and is defined as follows:

$$C_T = P_{stor} + P_{net} + P_{op}. \tag{5}$$

4.3 Optimization Problem

Given a data management model, we formalize a data placement optimization problem. It aims to maximize the availability of a data file and minimize the total cost. The overall optimization problem can be defined as follows:

$$\begin{cases} Maximize\ A, \\ Minimize\ C_T. \end{cases} \tag{6}$$

Subject to:

$$A \geq A_{req}.$$

In the above optimization problem, constraint 1) guarantees that the availability of a data file is not less than a user's required availability.

5 SOLUTION

In this section, we present a multi-objective optimization algorithm based on NSGA-II [31] to solve the multi-objective optimization problem defined in the previous section firstly. Then we use the entropy weight method to determine the weights of cost and availability, and find the most suitable data placement solution for users in the Pareto-optimal set.

5.1 Multi-Objective Optimization Algorithm

The proposed algorithm used to solve the optimization problem formulated above is mainly based on NSGA-II [31]. NSGA-II algorithm is one of the most popular multi-objective optimization algorithms. [2]. It has the advantages of fast running speed and good convergence of the solution set. Compared with NSGA, the previous generation algorithm, it uses a fast non-dominated sorting algorithm, which greatly reduces the computational complexity. The introduction of the elite strategy ensures that the individuals of the excellent population are not discarded in the iterative process, which can improve the accuracy of the optimization results. By using congestion degree, we not only overcome the defect of artificially specifying shared parameters, but also can take the congestion degree as the comparison standard among individuals in the population, so that the individuals can be evenly extended to the whole Pareto domain, which will ensure the diversity of the population. As mentioned before, how to place a user's data cost-effectively with high availability in multi-cloud environments is a hot and challenging multi-objective optimization problem. So we propose an NSGA-II-based algorithm NDP to solve this problem, which has been fully defined in Section 4. The NDP algorithm depicted in the pseudocode **Algorithm NDP** includes population initialization, individual fitness calculation, genetic operators (i.e. selection, crossover, and mutation), non-dominated sorting approach, and making new population based on an elitism approach.

Initialize Population. The first step of **NDP** is to generate an initial population. In our problem, the combination of CSPs is converted to the individual in the population. It is encoded in a binary array $[x_1, x_2, \ldots, x_N]$, where i^{th} CSP is chosen if $x_i = 1$. Specific to this optimization problem, each gene represents a CSP and the number of the genes whose value equals 1 is n (i.e. erasure coding parameter (m, n)). The algorithm **GenInd** presents how to generate an individual of the population. Firstly, it generates an integer array of length n randomly, whose elements are integers between 0–35 (lines 3–13). Then, the

Algorithm 1

Algorithm NDP: Getting the Pareto-optimal set

Require: The set of CSPs, C, the upper limit of n, ξ, and a user request $DF = \{S, \tau, A_{required}\}$

Ensure: A series of Pareto-optimal set, P

Initialize the parameters, N_p (population size), N_g (number of generation), p_c (cross probability), p_m (mutation probability);
Initialize P as empty;
Initialize population through $PopInitEQ$ or $PopInitDC$;
$g = 0$;
$Q_0 = Cpoy(P_0)$;
while $g \leq N_g$ **do**
  Cross the population P_g;
  Mutate the population P_g;
  for $i = 0$ to N_p **do**
    $fitness = \{0, 0\}$
    Calculate total cost (*totalCost*) and data file availability (*availability*) of data placement scheme represented by individual $P_g[i]$;
    $fitness = \{totalCost, availability\}$
    $P_g[i] = fitness$
  end for
  $F = Nondominated(P_g \cup Q_g)$;
  Calculate the crowding distance for each Pareto set in F;
  $P_{g+1} = []$;
  for $i = 0$ to $(|F| - 1)$ **do**
    if $|P_{g+1}| + |F_i| \leq N_p$ **then**
      $P_{g+1} = P_{g+1} \cup F_i$;
    else
      $P_{g+1} = P_{g+1} \cup F_i\,[1 : (N_p - |P_{g+1}|)]$
    end if
  end for
  $Q_{g+1} = Copy(P_{g+1})$
  $g = g + 1$;
end while
$paretoFront = []$;
for $i = 0$ to N_p **do**
  if $Q_g\,[i]\,.rank == 1$ **then**
    $paretoFront = paretoFront \cup Q_g\,[i]$;
  end if
end for
$P = P \cup paretoFront$;
return P;

Algorithm 2

Algorithm GenInd: Generating an individual

Require: The erasure coding parameter, m, n, the number of CSP, *length*, and the population size, N_p

Ensure: An individual, P_0

```
1: Initialize empty arrays index[n], P_0;
2: i = 0, j = 0;
3: for i = 0 to n do
4:    index[i] = Generate an integer (0 − length) randomly;
5:    for j = 0 to i do
6:       if index[i] == index[j] then
7:          break;
8:       end if
9:    end for
10:   if j == i then
11:      i++;
12:   end if
13: end for
14: for i = 0 to n do
15:   P_0[index[i]] = 1;
16: end for
17: return P_0;
```

corresponding position of the array representing the individual is modified to 1 (lines 14–16). In this optimization problem, the optimal data placement solution includes not only a list of CSPs but also erasure coding parameters. Based on the characteristic of our optimization problem, we propose two strategies for initializing population, as follows:

1. The idea of this strategy called **EQ** is to initialize an equal number of individuals for each erasure coding parameter. The procedure **PopInitEQ** is the pseudo code for this strategy. The individuals corresponding to all erasure coding parameters make up the entire population.
2. The second strategy called **DC** is inspired by the "divide-and-conquer" idea. The procedure **PopInitDC** is the pseudo code for this strategy. We run multiple **NDP** for different erasure coding parameter and the population belongs to a parameter in each run. Finally, we choose the best one from the results of all erasure coding parameters as the final data placement solution.

Crossover Operation and Mutation Operation. A crossover operation is used to generate new individuals through single-point or multi-point intersection. In our paper, we use a single-point crossover operator. Firstly, the algorithm pairs individuals in the population randomly. Then, it generates a point randomly and

two individuals exchange part of their genes at the mating point with crossover probability. The mutation operation is to avoid premature convergence of the population during the later iterations of the algorithm. In our paper, due to the binary encoding, the algorithm first generates a point randomly. Then, the individual's value at this point is modified to 1 if it is 0, and vice versa.

Calculate Fitness. In a genetic algorithm, calculating the fitness of individuals in the population is one of the important steps. In terms of our problem, each individual's fitness is a two-dimensional array, which contains cost and data availability of a data placement scheme represented by this individual. We first transform the individual into its corresponding data placement scheme. Then, data file availability and total cost are calculated through Equations (1), (2), (3), (4) and (5).

Make New Population. In order to maintain population distribution and diversity, the algorithm **NDP** first merges two generations of populations and the non-dominated set is constructed through a fast non-dominated sorting approach. An individual is a non-dominated one when no individual in the population is superior to this individual in all objective functions. These non-dominated individuals constitute a non-dominated set. Then, it calculates the crowding distance for each pareto set in a non-dominated set and sorts it in descending order. Finally, the algorithm selects individuals into new populations in turn from a non-dominant set.

When iterations end, **NDP** facilitates all individuals and selects individuals with a pareto rank of 1 to compose the optimal data hosting solutions.

Algorithm 3

Algorithm PopInitEQ: initializing the population according the first strategy

Require: The population size, N_p
Ensure: The Population, *Population*

```
1: Population = [];
2: for n = 2 to ξ do
3:    for m = 1 to n do
4:       for i = 1 to N_p do
5:          Population[i] = GenInd;
6:       end for
7:    end for
8: end for
9: return Population;
```

Algorithm 4

Algorithm PopInitDC: initializing the population according the second strategy

Require: The population size, N_p
Ensure: The Population, *Population*

```
1: Population = [];
2: for i = 1 to N_p do
3:     Population[i] = GenInd;
4: end for
5: return Population;
```

5.2 Approach to Determine the Most Suitable Solution

We get a Pareto-optimal set solution through **NDP**. For users who have very specific preferences and the ability to make choice, they can choose the data placement solution directly from the Pareto-optimal set. For example, one user wants to choose the solution with highest availability from the Pareto-optimal set, and would rather pay more cost. Another user wants to choose the solution with lowest cost, and accept a lower availability. However, most of users are still confused and to choose a solution from the Pareto-optimal set is difficult for them. In fact, regarding cloud storage, most users tend to choose the solutions that are more compromised on each metric, especially for cost and availability. However, there are many extreme solutions in the resulted Pareto-optimal set. For example, there always exist such solutions in the Pareto-optimal set: A [99.9999 %, $ 10] and B [99.1 %, $ 3], respectively. Although A has higher availability, its cost is also more expensive, and solution B is the opposite. Therefore, in order to recommend suitable data placement solutions for the users who cannot make choice from the Pareto-optimal set directly, the entropy based method is proposed. We calculate the QoS of each solution in the resulted Pareto-optimal set by determining the weights of the two metrics of cost and availability, and recommend the solution with the maximum QoS to the user.

There are many ways to determine the weights, which can be classified as subjective and objective weighting methods [32]. The former determine weight methods are based on the subjective value judgment of indices, including Delphi method, Analytic Hierarchy Process (AHP) method, least square method, and binomial coefficient method. The objective methods are based on the objective information (e.g. decision matrix), which includes principal component analysis, entropy method, deviation and mean square method, and multiple objective programming model [32].

Since the subjective weighting methods have strong subjective randomness and poor objectivity, in our paper, we use the entropy method to determine the weights of cost and availability, which is one of the most common objective methods. In the information theory, entropy is a measure of uncertainty [33]. The greater the amount of information, the less uncertainty and the smaller entropy, and vice versa. According to the characteristics of entropy, we can use it to judge the degree of

dispersion of an index. The greater the entropy, the higher the degree of dispersion of the index, and the greater the influence of the index on comprehensive evaluation.

Before introducing the application of an entropy method in our paper, we present the definition of the element in Pareto-optimal set:

Definition 9 (The Element in Pareto-Optimal Set)**.** The Pareto-optimal set is obtained through **NDP**, and it is represented as $P = \{P_1, P_2, \cdots, P_N\}$. Each element of Pareto-optimal set has triples: $P_i = \{P_i^{ep}, P_i^c, P_i^a\}$, where:

1. P_i^{ep} is the erasure parameter of the i^{th} element in Pareto-optimal set;
2. P_i^c denotes the i^{th} data placement solution's cost; and
3. P_i^a defines the i^{th} data placement solution's availability.

There are many studies using entropy to calculate weights [34, 35, 36], and the process of calculating weights based on entropy is very mature. In our paper, we calculate the weights of cost and availability through the following steps:

Step 1. Normalize cost and availability.

Owing to the fact that the availability (resp. cost) index is a positive (resp. negative) index, we use different normalization functions for them:

$$f_1\left(P_i^c\right) = \begin{cases} \frac{P_{max}^c - P_i^c}{P_{max}^c - P_{min}^c}, & \text{if } P_{max}^c \neq P_{min}^c, \\ 1, & \text{if } P_{max}^c = P_{min}^c, \end{cases} \tag{7}$$

where P_{max}^c (resp. P_{min}^c) means the maximum (resp. minimum) cost of all solutions in Pareto-optimal set P.

$$f_2\left(P_i^a\right) = \begin{cases} \frac{P_i^a - P_{min}^a}{P_{max}^a - P_{min}^a}, & \text{if } P_{max}^a \neq P_{min}^a, \\ 1, & \text{if } P_{max}^a = P_{min}^a, \end{cases} \tag{8}$$

where P_{max}^a (resp. P_{min}^a) means the maximum (resp. minimum) availability of all solutions in Pareto-optimal set P.

For simplicity, we combine the normalized cost and availability into an $N \times 2$ matrix A, and A_{ij} denotes i^{th} element's j^{th} index's value in the Pareto-optimal set, where $i \in \{1, 2, \cdots N\}$ and $j \in \{1, 2\}$.

Step 2. Calculate the proportion of the j^{th} index of the i^{th} element to this index as:

$$p_{ij} = \frac{A_{ij}}{\sum_{i=1}^{N} A_{ij}}. \tag{9}$$

Step 3. Calculate the entropy value of the j^{th} index as follows:

$$e_j = -k \sum_{i=1}^{N} p_{ij} \ln(p_{ij}) \tag{10}$$

where $k = \frac{1}{\ln(N)}$.

Step 4. Calculate the divergence of entropy as follows:

$$d_j = 1 - e_j. \tag{11}$$

Step 5. Calculate the weight of each index through:

$$w_j = \frac{d_j}{\sum_{j=1}^{2} d_j}. \tag{12}$$

Step 6. Calculate the integrated QoS value of each data placement solution according to the weights which are defined by step 5, as follows:

$$q_i = \sum_{j=1}^{2} w_j p_{ij}. \tag{13}$$

5.3 Discussion

In this section, we propose a solution to provide a data placement with low monetary cost and high availability for users. High data availability and low monetary cost are the two most important driving forces for users to host their data into cloud instead of the traditional storage mode. In fact, there are many metrics needed to be considered in cloud storage, such as availability, monetary cost, durability, data lock-in level, latency, and security. The proposed method can be easily extended to consider these metrics. Taking latency as an example, whether it is an optimization objective or a constraint, the algorithm **NDP** based on NSGA-II can well solve the optimization problem. Once the Pareto-optimal set is obtained, the entropy method can determine the most suitable data placement for the user who cannot make choice from the Pareto-optimal set.

6 PERFORMANCE EVALUATION

We implement the proposed data placement algorithms and carry out simulations by using real-world CSP information to evaluate its performance. In this section the goal is threefold. The first is to discuss the experimental setup in terms of CSP information dataset and parameter settings of algorithms. Second, to study the performance of the proposed algorithms through several scenarios. Although multi-cloud has become a research hotspot in recent years, there are not many studies on data storage optimization in multi-cloud environments. CHARM [17] and ACO [27] are two representative ones, which are the closest to this work. Thus, we compare the proposed method to them in this section.

Provider	Location	Specific Location
Amazon S3 (AM), Microsoft Azure (AZ), Google (GO), Alibaba (AL), CenturyLink (CL), and SoftLayer (SL)	USA (US), Europe (EU), Asia Pacific (AP), Australia (AU)	South (S), North (N), West (W), East (E), center (C), Mumbai (M), Seoul (S), Tokyo (T), Frankfurt (F), Ireland (I), Paris (P), London (L), Sydney (Sy), and so on.

Table 3. Element in CSP name

6.1 Experimental Setup

The real-world cloud providers' information is collected from *CloudHarmony* [12], which is a third-party platform to simplify the comparison of cloud services by providing reliable and objective performance analysis, reports, commentary, metrics, and tools. We use 35 CSPs in the experiments and among these, including 12 by Amazon S3 (AM), 4 by Microsoft Azure (AZ), 3 by Google (GO), 7 by Alibaba (AL), 5 by CenturyLink (CL), and 4 SoftLayer (SL). Each CSP has a name that consists of the element in Table 3 [4]. For example, the CSP with name AZ-EUN refers to the cloud provider of Microsoft Azure in the North of Europe. In our dataset, it is noted that Amazon S3 has two data centers in USA-West (i.e. Northern California (N), Oregon (O)) and USA-East (i.e., N. Virginia (N), Ohio (O)) region, respectively. For instance, AWS-USW-N denotes that the CSP of Amazon S3 is in Northern California in Eastern USA. Each CSP is also referred by a specification that consists of storage, out-going bandwidth and operation (i.e., GET requests) prices. Since the availability of SLA for each cloud provider is just what they claim, we also simulate the values of availability of each CSP in the interval of [95.0 %, 99.9 %].

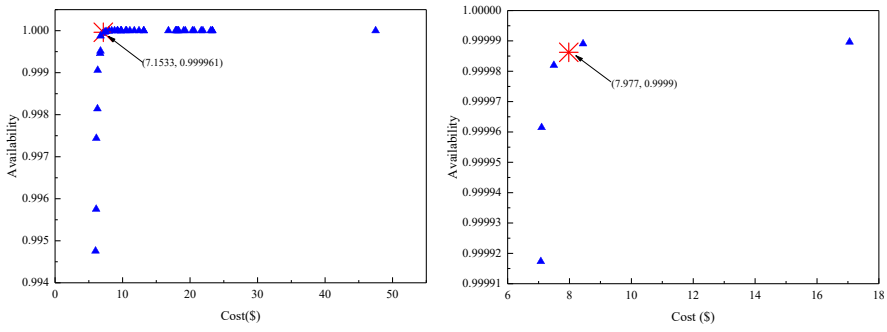
The programs for the proposed algorithm are coded in the Java language and run on an Intel Core^{TM} i7-6700 processor with 3.40 GHz CPU and 16 GB RAM. The settings for various parameters have a direct influence on the algorithm performance. Appropriate parameter values are determined by multiple experiments. Since we have two strategies for initializing the population, there are two final parameter settings, as shown in Table 4. The algorithm based on the first strategy is called **EQ**, the other is **DC**.

Algorithms	EQ	DC
Population Size	1 500	300
Generation Count	3 000	600
Mutation Rate	0.1	
Crossover Rate	0.9	

Table 4. The parameters of algorithms **EQ** and **DC**

Parameters Setting	Default	Range
Data size	200 GB	100–1 000 GB
DAF	0.3	0.0–1.0
Erasure coding (m, n)	$2 < n < 7, 0 < m < n$	

Table 5. Settings of simulation parameters



a) **EQ** b) **DC** with $(3, 5)$-erasure coding

Figure 4. Pareto-optimal set of two algorithms with data size 200 GB and DAF 0.3

6.2 Performance of the Proposed Algorithm

Before describing the performance of the proposed method, we first discuss the correctness of the model in this paper. In multi-cloud storage, providing a cost-effective and high-availability data placement for users is a research hotspot. In this paper, we first define the multi-objective optimization problem, which is to maximize data availability and to minimize the monetary cost, under the erasure coding mode in multi-cloud storage. Erasure coding is used to reduce storage cost and to improve availability, as compared to data replication. The definitions for data availability and cost are similar to that in [17, 28], which has become a common way to define them for data hosting in erasure coding mode. Then, in order to solve the multi-objective optimization problem, we propose a method based on NSGA-II. This algorithm is widely used to solve multi-objective optimization problems and can achieve good results. Since the resulted Pareto-optimal set usually contains many solutions, which makes users still confused and difficult to make choices, we adopt the entropy method to determine the most suitable solution for user from this set. The entropy method is a common method to objectively determine weight of each index based on the characteristics of the solution space. From the final results, the data placement solutions obtained by the entropy method can satisfy the user's requirement for compromise on all objectives. Furthermore, we compare our model with two representative studies. All the results show the effectiveness of our proposed model.

Due to different strategies for initializing the population, the results of **EQ** and **DC** are different. So we discuss the results gained by them in the following aspects.

6.2.1 Pareto Optimal Solution

The data placement problem is a dual-objective optimization problem in our paper. In general, there is no absolute or unique optimal solution in multi-objective problems. In this section, we study the Pareto optimal solutions of the proposed algorithms for our dataset with the default parameters in Table 5, as shown in Figure 4. Figure 4 a) shows the result of **EQ**. Since the algorithm **DC** separately solves Pareto optimal solutions under different erasure coding parameters, there are 15 Pareto optimal solutions. Due to the space limitation, we only depict the Pareto optimal solution under the $(3, 5)$-erasure coding, as shown in Figure 4 b).

An optimal data placement solution can be obtained through the method in Section 5.2. For **EQ**, this method can find the most suitable solution with maximum QoS value in Pareto optimal solution, which is marked in Figure 4 a). The data placement scheme represented by this point contains the chosen CSPs {AZ-USAE, AZ-EUN, AWS-USE-O, AWS-USW-O, AWS-EU-I} and erasure coding $(3, 5)$. The total cost and availability of this placement are \$ 7.1533 and 99.9961 % respectively. Each CSP in the result stores the size of 40 GB data, and 3 CSPs with the cheapest out-bandwidth price for GET requests.

For **DC**, the point marked in Figure 4 b) only represents the best solution under the $(3, 5)$-erasure coding. Intuitively, we have 15 solutions under this. We need to run the method in Section 5.2 again to gain the most suitable data placement scheme and results for all erasure coding parameters. The best data placement consists of:

1. the chosen CSPs {AZ-USAE, AZ-EUN, GO-AS, AWS-USE-O, AWS-EU-I, AL-USE};
2. total cost \$ 7.715 and availability 99.997 %; and
3. the $(4, 6)$-erasure coding.

6.2.2 Storage Mode Change

In fact, the DAF of a data object is time-varying in the cloud. In this section, we study the impact on data placement scheme with DAF varying from 0.0 to 1.0 with 0.05 interval. For clarity, Table 6 in Appendix summarizes the erasure coding parameters (i.e., storage mode) change with varying DAF. Whether **EQ** or **DC**, the data storage mode becomes the special erasure coding when DAF is greater than a certain value (i.e., the value of n is an integer multiple of m). For **EQ**, the storage mode is $(2, 4)$-erasure coding when DAF is greater than 0.55. For **DC**, the storage mode is $(1, 2)$-erasure coding (i.e., replication) when DAF is beyond 0.60.

The reason for this phenomenon is that high DAF requires expensive operation cost, especially network cost, and it accounts for a large proportion of the total

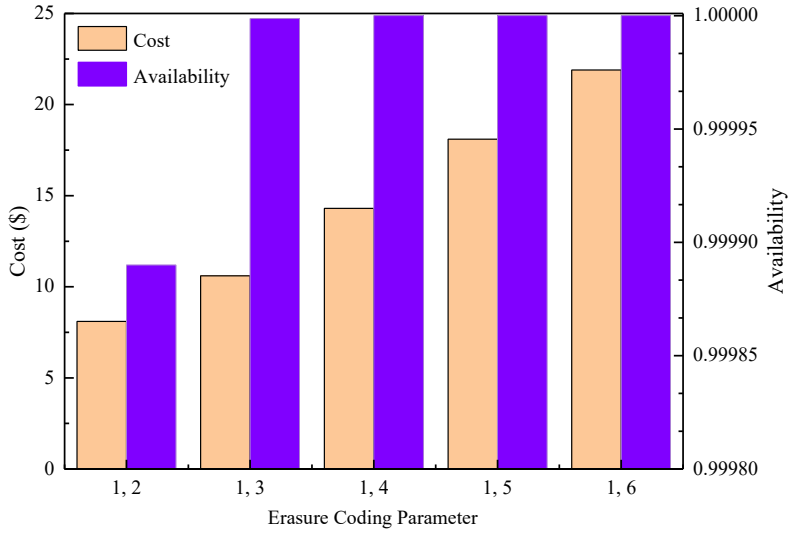
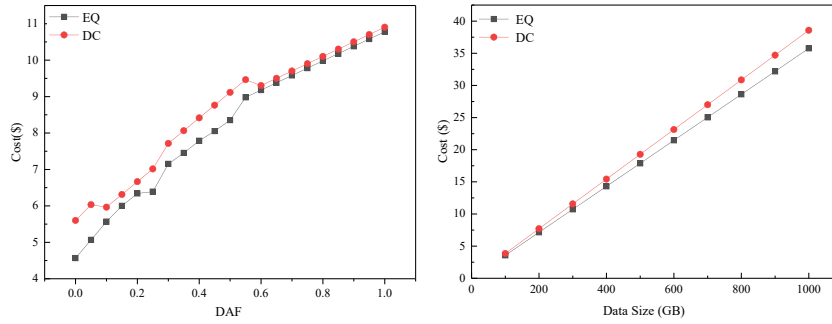


Figure 5. Cost and availability vs. erasure coding

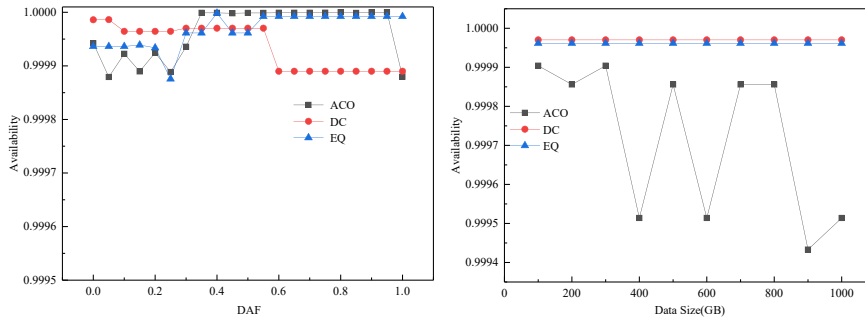
DAF	**EQ**	**DC**
0.0	$(3, 5)$	$(2, 4)$
0.05	$(3, 5)$	$(4, 5)$
0.10	$(3, 5)$	$(4, 5)$
0.15	$(3, 5)$	$(4, 5)$
0.20	$(4, 6)$	$(4, 6)$
0.25	$(4, 6)$	$(4, 6)$
0.30	$(3, 5)$	$(4, 6)$
0.35	$(3, 5)$	$(4, 6)$
0.40	$(3, 5)$	$(4, 6)$
0.45	$(3, 5)$	$(4, 6)$
0.50	$(3, 5)$	$(4, 6)$
0.55	$(2, 4)$	$(4, 6)$
0.60	$(2, 4)$	$(1, 2)$
0.65	$(2, 4)$	$(1, 2)$
0.70	$(2, 4)$	$(1, 2)$
0.75	$(2, 4)$	$(1, 2)$
0.80	$(2, 4)$	$(1, 2)$
0.85	$(2, 4)$	$(1, 2)$
0.90	$(2, 4)$	$(1, 2)$
0.95	$(2, 4)$	$(1, 2)$
1.0	$(2, 4)$	$(1, 2)$

Table 6. Erasure coding parameter (m, n) changing with varying DAF

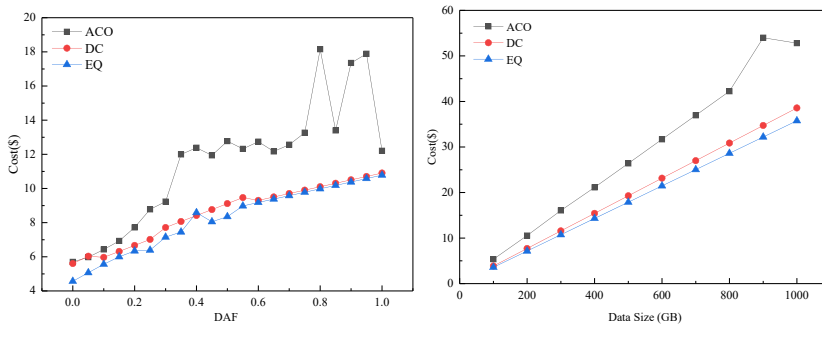


a) Cost vs. varying DAF b) Cost vs. varying data size

Figure 6. Total cost of data placement scheme of **EQ** and **DC** when the DAF and data size are varied



a) Availability vs. DAF b) Availability vs. data size



c) Total cost vs. DAF d) Total cost vs. data size

Figure 7. The total cost and availability comparison of data placement schemes from our proposed algorithms and **ACO**

cost. For example, in **EQ**, the network cost is \$ 1.8 which is 25.16 % of the total when DAF is 0.3, and the ratio increases by 6.9 % compared to the former when DAF equals 0.8. When DAF is high, the proposed algorithms explore CSPs with the cheaper out-going bandwidth price to handle high DAF. Therefore, it is really necessary to timely adjust the data placement scheme according to varying DAF. Data migration of varying DAF is beyond the scope of our paper, and we leave it as the future work.

6.2.3 Cost and Availability Performance

In this section, we evaluate the cost and availability performance of the proposed algorithms. Since each erasure coding has an optimal result for **DC**, we study the impact of erasure coding on cost and availability by varying it from $(1, 2)$ to $(1, 6)$ with data size 200 GB and DAF 0.3. As shown in Figure 5, as n increases as an erasure coding parameter, the availability gradually approaches 1. The reason is that the overall availability of a data object is equal to the probability that not more than $(n - m)$ CSPs crash at the same time. When n becomes larger, the data placement scheme can tolerate the simultaneous failure of more CSPs, and so the availability can be enhanced. At the same time, the total cost is higher with n. This is because of the storage cost of a data object growing with the number of replicas.

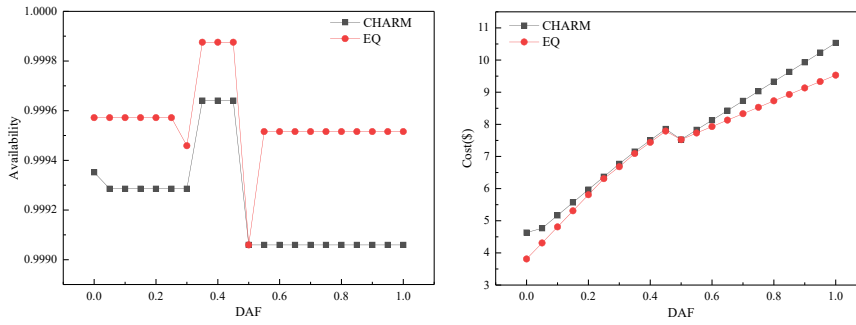
Figure 6 shows the impact of DAF and data size on the total cost of the optimal data placement scheme of the proposed algorithms. In Figure 6 a), the total cost only contains the storage cost and **EQ** can save approximately 18.6 % than **DC** when DAF is 0. It is worth noting that the polyline in Figure 6 a) is composed of several straight lines. It is because of the data placement scheme varying with DAF. For instance, the black polyline consists of four parts and the change points are 0.15, 0.30, and 0.55. This result can correspond to the change of the storage mode in Table 6 in Appendix.

We also explore the impact of data size on the total cost by varying it from 100 GB to 1 000 GB with the step size of 100 GB. As shown in Figure 6 b), the results of the proposed algorithms show the positive correlation between cost and data size. The reason why the result is a straight line is that the data placement scheme does not change as data size increases. The total cost of a resultant data placement scheme through **EQ** can save about \$ 2.8 comparing with that of **DC**.

6.3 Performance Comparison with Other Algorithms

There are many previous studies on data storage in multi-cloud environments. In this section, we compare the cost and availability performance of the proposed algorithms with two recently solutions **ACO** and **CHARM**.

The optimization objective of **ACO** contains the total cost and availability, which is the same as ours. We evaluate their performance, as shown in Figure 7. Figures 7 a) and 7 b) respectively depict the impact of DAF and data size on the availability of obtained data placement schemes. In Figure 7 a), it is obvious that



a) Availability b) Total cost

Figure 8. The availability and total cost of data placement scheme from our proposed algorithms and **CHARM** under the varying DAF

the result of **DC** is better than its two peers when DAF is less than 0.3. **EQ** can achieve higher availability than **DC** when DAF is grater than 0.55. The reason for this result is that data placement can tolerate the crash of more CSP at the same time than **DC**, which is shown in Table 6. **EQ** can tolerate the simultaneous crash of 2 CSPs, while **DC** can tolerate only one CSP's crash when DAF is greater than 0.55. Figure 7 b) shows the availability change under varying data size. It is evident that the performance of our proposed algorithms is superior to that of **ACO**. The results of the proposed algorithms are relatively more stable than those of **ACO**.

Data Size	**Availability**		**Cost**	
	CHARM	EQ	CHARM	EQ
100	0.9992856	0.9994588	3.38693	3.34
200	0.9992856	0.9994588	6.77027	6.68
300	0.9992856	0.9994588	10.1536	10.02
400	0.9992856	0.9994588	13.53693	13.36
500	0.9992856	0.9994588	16.92027	16.7
600	0.9992856	0.9994588	20.3036	20.04
700	0.9992856	0.9994588	23.68693	23.38
800	0.9992856	0.9994588	27.07027	26.72
900	0.9992856	0.9994588	30.4536	30.06
1 000	0.9992856	0.9994588	33.83693	33.4

Table 7. Availability and total cost ($) comparison of **CHARM** and **EQ** with the varying data size (GB)

We also explore the comparison of total cost by the varying DAF and data size. As shown in Figure 7 c), the proposed algorithms **EQ** and **DC** can approximately save $ 3.56 and $ 3.44, respectively, comparing to **ACO** when DAF equals 0.6. Figure 7 d) presents the total cost change by varying the data size from 100 GB to 1 000 GB with the interval of 100 GB. It is clear that the total cost of the data

placement schemes, obtained with the proposed algorithms is lower than **ACO**. Our proposed algorithms can save more than $ 10 than **ACO** when data size is 800 GB.

Finally, we compare the proposed algorithm **EQ** with **CHARM** through two scenarios including the varying DAF and data size. Since the optimization objective of **CHARM** is to minimize the total cost under the guaranteed availability, we select the data placement scheme from the Pareto solution of **EQ**, whose availability is not lesser than that of **CHARM**. Figure 8 a) depicts the availability of these two algorithms. It is obvious that the availability of **EQ** is larger than **CHARM**. Figure 8 b) presents the comparison of the total cost between them by varying DAF from 0.0 to 1.0. Our algorithm can clearly obtain the lower total cost than **CHARM**. Another scenario is to compare two algorithms through the varying data size, whose result is shown in Table 7. Our algorithm can save 1.29 % when data size is 1 000 GB. Although the advantages of our proposed algorithm are not obvious in total cost, its resulting availability of our algorithm is better than **CHARM**'s.

7 CONCLUSION

There are some risks such as vendor lock-in, low data availability and data privacy leakage if users put their data into a single cloud. Data hosting based on multi-cloud is becoming a new development trend. How to strike a trade-off between various factors and realize multi-objective optimization becomes one of the most important concerns in multi-cloud environments. So, in this paper, an architecture in multi-cloud storage is presented at first. Next, a multi-objective optimization problem is defined to minimize total cost and maximize data availability. Then, an approach based on NSGA-II is given with its goal to effectively solve a multi-objective optimization problem and obtain a set of non-dominated solutions (i.e., a list of cloud storage providers) and erasure coding parameters. Then, we use a method based on the entropy to recommend the most suitable solution for users who cannot choose one from the resulted Pareto-optimal set directly. Finally, the performance of this algorithm is examined through extensive experiments which are driven by real-world multiple cloud storage providers' information.

In the future, we intend to improve this work in two directions:

1. The SLA is important for users, and it directly affects the user experience. Thus, we will consider more SLAs in exploring a data hosting scheme, such as the overall security, latency and durability of cloud services [37].
2. Since the data placement varies with the change of user's DAF, it is necessary to propose solutions for dynamic data placement based on the varying DAF. Especially, for the absence of user's future DAF, we will predict it according to the historical data [38, 39, 40].
3. We will consider to optimize with the choice of cloud instance type [41, 42, 43].

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61602109, the DHU Distinguished Young Professor Program under Grant No. LZB2019003, the Shanghai Science and Technology Innovation Action Plan under Grant No. 19511101802, the Natural Science Foundation of Shanghai under Grant No. 19ZR1401900, and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Lv, Y.—Chen, Y.—Zhang, X.—Duan, Y.—Li, N.: Social Media Based Transportation Research: The State of the Work and the Networking. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 1, pp. 19–26, doi: 10.1109/JAS.2017.7510316.

[2] Wu, N. Q.—Li, Z. W.—Barkaoui, K.—Li, X. O.—Murata, T.—Zhou, M. C.: IoT-Based Smart and Complex Systems: A Guest Editorial Report. IEEE/CAA Journal of Automatica Sinica, Vol. 5, 2018, No. 1, pp. 69–73, doi: 10.1109/JAS.2017.7510748.

[3] Zhang, P.—Zhou, M.—Fortino, G.: Security and Trust Issues in Fog Computing: A Survey. Future Generation Computer Systems, Vol. 88, 2018, pp. 16–27, doi: 10.1016/j.future.2018.05.008.

[4] Mansouri, Y.—Buyya, R.: To Move or Not to Move: Cost Optimization in a Dual Cloud-Based Storage Architecture. Journal of Network and Computer Applications, Vol. 75, 2016, pp. 223–235, doi: 10.1016/j.jnca.2016.08.029.

[5] Ghahramani, M. H.—Zhou, M. C.—Hon, C. T.: Toward Cloud Computing QoS Architecture: Analysis of Cloud Systems and Cloud Services. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 1, pp. 6–18, doi: 10.1109/JAS.2017.7510313.

[6] Gao, Y.—Guan, H.—Qi, Z.—Hou, Y.—Liu, L.: A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing. Journal of Computer and System Sciences, Vol. 79, 2013, No. 8, pp. 1230–1242, doi: 10.1016/j.jcss.2013.02.004.

[7] Armbrust, M.—Fox, A.—Griffith, R.—Joseph, A. D.—Katz, R.—Konwinski, A.—Lee, G.—Patterson, D.—Rabkin, A.—Stoica, I.—Zaharia, M.: A View of Cloud Computing. Communications of the ACM, Vol. 53, 2010, No. 4, pp. 50–58, doi: 10.1145/1721654.1721672.

[8] Opara-Martins, J.—Sahandi, R.—Tian, F.: Critical Analysis of Vendor Lock-In and Its Impact on Cloud Computing Migration: A Business Perspective. Journal of Cloud Computing, Vol. 5, 2016, No. 1, pp. 4–22, doi: 10.1186/s13677-016-0054-z.

[9] Mansouri, Y.—Toosi, A. N.—Buyya, R.: Brokering Algorithms for Optimizing the Availability and Cost of Cloud Storage Services. Proceedings of 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, UK, 2013, pp. 581–589, doi: 10.1109/CloudCom.2013.83.

[10] Aldossary, S.—Allen, W.: Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions. International Journal of Ad-

vanced Computer Science and Applications, Vol. 7, 2016, No. 4, pp. 485–498, doi: 10.14569/IJACSA.2016.070464.

[11] Mansouri, Y.—Toosi, A. N.—Buyya, R.: Data Storage Management in Cloud Environments: Taxonomy, Survey, and Future Directions. ACM Computing Surveys (CSUR), Vol. 50, 2017, No. 6, Art. No. 91, 51 pp., doi: 10.1145/3136623.

[12] Cloudharmony, 2017. [Online] Available at: `http://www.cloudharmony.com`.

[13] Abu-Libdeh, H.—Princehouse, L.—Weatherspoon, H.: RACS: A Case for Cloud Storage Diversity. Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10), New York, NY, USA, 2010, pp. 229–240, doi: 10.1145/1807128.1807165.

[14] Papaioannou, T. G.—Bonvin, N.—Aberer, K.: Scalia: An Adaptive Scheme for Efficient Multi-Cloud Storage. Proceedings of the 2012 International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12), Utah, USA, 2012, 10 pp., doi: 10.1109/SC.2012.101.

[15] Hadji, M.: Scalable and Cost-Efficient Algorithms for Reliable and Distributed Cloud Storage. In: Helfert, M., Méndez Muñoz, V., Ferguson, D. (Eds.): Cloud Computing and Services Science (CLOSER 2015). Springer, Cham, Communications in Computer and Information Science, Vol. 581, 2015, pp. 15–37, doi: 10.1007/978-3-319-29582-4_2.

[16] Ma, Y.—Nandagopal, T.—Puttaswamy, K. P.—Banerjee, S.: An Ensemble of Replication and Erasure Codes for Cloud File Systems. Proceedings of 2013 INFOCOM, Turin, Italy, 2013, pp. 1276–1284, doi: 10.1109/INFCOM.2013.6566920.

[17] Zhang, Q.—Li, S.—Li, Z.—Xing, Y.—Yang, Z.—Dai, Y.: CHARM: A Cost-Efficient Multi-Cloud Data Hosting Scheme with High Availability. IEEE Transactions on Cloud Computing, Vol. 3, 2015, No. 3, pp. 372–386, doi: 10.1109/TCC.2015.2417534.

[18] Mansouri, Y.—Toosi, A. N.—Buyya, R.: Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers. IEEE Transactions on Cloud Computing, Vol. 7, 2019, No. 3, pp. 705–718, doi: 10.1109/TCC.2017.2659728.

[19] Wu, Z.—Butkiewicz, M.—Perkins, D.—Katz-Bassett, E.—Madhyastha, H. V.: SPANStore: Cost-Effective Geo-Replicated Storage Spanning Multiple Cloud Services. Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles (SOSP '13), 2013, pp. 292–308, doi: 10.1145/2517349.2522730.

[20] Liu, G.—Shen, H.—Wang, H.: An Economical and SLO-Guaranteed Cloud Storage Service Across Multiple Cloud Service Providers. IEEE Transactions on Parallel and Distributed Systems, Vol. 28, 2017, No. 9, pp. 2440–2453, doi: 10.1109/TPDS.2017.2675422.

[21] Weatherspoon, H.—Kubiatowicz, J.: Erasure Coding vs. Replication: A Quantitative Comparison. In: Druschel, P., Kaashoek, F., Rowstron, A. (Eds.): Peer-to-Peer Systems (IPTPS 2002). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2429, 2002, pp. 328–337, doi: 10.1007/3-540-45748-8_31.

[22] Rodrigues, R.—Liskov, B.: High Availability in DHTs: Erasure Coding vs. Replication. In: Castro, M., van Renesse, R. (Eds.): Peer-to-Peer Systems IV (IPTPS 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3640, 2005, pp. 226–239, doi: 10.1007/11558989_21.

[23] Wei, Q.—Veeravalli, B.—Gong, B.—Zeng, L.—Feng, D.: CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster. Proceedings of the 2010 IEEE International Conference on Cluster Computing (CLUSTER 2010), Heraklion, Greece, 2010, pp. 188–196, doi: 10.1109/CLUSTER.2010.24.

[24] Bessani, A.—Correia, M.—Quaresma, B.—André, F.—Sousa, P.: DepSky: Dependable and Secure Storage in a Cloud-of-Clouds. ACM Transactions on Storage (TOS), Vol. 9, 2013, No. 4, Art. No. 12, 33 pp., doi: 10.1145/2535929.

[25] Mu, S.—Chen, K.—Gao, P.—Ye, F.—Wu, Y.—Zheng, W.: μLibCloud: Providing High Available and Uniform Accessing to Multiple Cloud Storages. Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing, Beijing, China, 2012, pp. 201–208, doi: 10.1109/Grid.2012.28.

[26] Singh, Y.—Kandah, F.—Zhang, W.: A Secured Cost-Effective Multi-Cloud Storage in Cloud Computing. Proceedings of the 2011 IEEE Conference on Computer Communications Workshops, Shanghai, China, 2011, pp. 619–624, doi: 10.1109/INFCOMW.2011.5928887.

[27] Wang, P.—Zhao, C.—Zhang, Z.: An Ant Colony Algorithm-Based Approach for Cost-Effective Data Hosting with High Availability in Multi-Cloud Environments. Proceedings of 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018, 6 pp., doi: 10.1109/ICNSC.2018.8361288.

[28] Su, M.—Zhang, L.—Wu, Y.—Chen, K.—Li, K.: Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements. IEEE Transactions on Computers, Vol. 65, 2016, No. 6, pp. 1964–1977, doi: 10.1109/TC.2015.2462821.

[29] Ford, D.—Labelle, F.—Popovici, F. I.—Stokely, M.—Truong, V.-A.—Barroso, L.—Grimes, C.—Quinlan, S.: Availability in Globally Distributed Storage Systems. Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI '10), Vancouver, BC, Canada, 2010, pp. 61–74.

[30] Amazon S3, 2018. [Online] Available at: `https://aws.amazon.com/cn/s3/pricing/?nc=sn&loc=4`.

[31] Deb, K.—Pratap, A.—Agarwal, S.—Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, 2002, No. 2, pp. 182–197, doi: 10.1109/4235.996017.

[32] Ma, J.—Fan, Z. P.—Huang, L. H.: A Subjective and Objective Integrated Approach to Determine Attribute Weights. European Journal of Operational Research, Vol. 112, 1999, No. 2, pp. 397–404, doi: 10.1016/S0377-2217(98)00141-6.

[33] Shannon, C. E.: A Mathematical Theory of Communication. The Bell System Technical Journal, Vol. 27, 1948, No. 3, pp. 379–423, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[34] Ma, L. H.—Zhang, Y. P.—Zhao, Z. W.: Improved VIKOR Algorithm Based on AHP and Shannon Entropy in the Selection of Thermal Power Enterprise's Coal Suppliers. 2008 International Conference on Information Management, Innovation

Management and Industrial Engineering, Taipei, Taiwan, 2008, Vol. 2, pp. 129–133, doi: 10.1109/ICIII.2008.29.

[35] Wang, T.-C.—Lee, H.-D.: Developing a Fuzzy TOPSIS Approach Based on Subjective Weights and Objective Weights. Expert Systems with Applications, Vol. 36, 2009, No. 5, pp. 8980–8985, doi: 10.1016/j.eswa.2008.11.035.

[36] Shemshadi, A.—Shirazi, H.—Toreihi, M.—Tarokh, M. J.: A Fuzzy VIKOR Method for Supplier Selection Based on Entropy Measure for Objective Weighting. Expert Systems with Applications, Vol. 38, 2011, No. 10, pp. 12160–12167, doi: 10.1016/j.eswa.2011.03.027.

[37] Xia, Y.—Zhou, M.—Luo, X.—Zhu, Q.—Li, J.—Huang, Y.: Stochastic Modeling and Quality Evaluation of Infrastructure-as-a-Service Clouds. IEEE Transactions on Automation Science and Engineering, Vol. 12, 2015, No. 1, pp. 160–172, doi: 10.1109/TASE.2013.2276477.

[38] Gao, S.—Zhou, M.—Wang, Y.—Cheng, J.—Yachi, H.—Wang, J.: Dendritic Neuron Model with Effective Learning Algorithms for Classification, Approximation and Prediction. IEEE Transactions on Neural Networks and Learning Systems, Vol. 30, 2019, No. 2, pp. 601–614, doi: 10.1109/TNNLS.2018.2846646.

[39] Li, W.—Xia, Y.—Zhou, M.—Sun, X.—Zhu, Q.: Fluctuation-Aware and Predictive Workflow Scheduling in Cost-Effective Infrastructure-as-a-Service Clouds. IEEE Access, Vol. 6, 2018, pp. 61488–61502, doi: 10.1109/ACCESS.2018.2869827.

[40] Bi, J.—Zhang, L.—Yuan, H.—Zhou, M.: Hybrid Task Prediction Based on Wavelet Decomposition and ARIMA Model in Cloud Data Center. Proceedings of 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018, 6 pp., doi: 10.1109/ICNSC.2018.8361342.

[41] Wang, P.—Zhou, W.—Zhao, C.—Lei, Y.—Zhang, Z.: A Dynamic Programming-Based Approach for Cloud Instance Types Selection and Optimization. International Journal of Information Technology and Management, Vol. 19, 2020, No. 4, doi: 10.1504/IJITM.2020.10028804.

[42] Liu, W.—Wang, P.—Meng, Y.—Zou, G.—Zhang, Z.: A Novel Algorithm for Optimizing Selection of Cloud Instance Types in Multi-Cloud Environment. 25th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 167–170, doi: 10.1109/ICPADS47876.2019.00033.

[43] Liu, W.—Wang, P.—Meng, Y.—Zhao, Q.—Zhao, C.—Zhang, Z.: A Novel Model for Optimizing Selection of Cloud Instance Types. IEEE Access, Vol. 7, 2019, pp. 120508–120521, doi: 10.1109/ACCESS.2019.2937511.

Pengwei Wang received his B.Sc. and M.Sc. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2005 and 2008, respectively, and his Ph.D. degree from the Tongji University, Shanghai, China, in 2013, all in computer science. He finished his postdoctoral research work at the Department of Computer Science, University of Pisa, Italy, in 2015. He is currently serving as Associate Professor in the School of Computer Science and Technology, Donghua University, Shanghai. His research interests include cloud computing, data mining, and service computing.

Caihui Zhao received his B.Sc. degree in software engineering from the Shandong University of Science and Technology, Qingdao, China, in 2016. He is currently a student studying for his Master's degree in software engineering at the Donghua University in Shanghai, China. His research interests include cloud computing, and multi-cloud storage.

Wenqiang Liu received his B.Sc. degree in computer science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2017. He is currently a student studying for his Master's degree in computer science and technology at the Donghua University in Shanghai, China. His research interests include cloud computing, operations research and machine learning.

Zhen Chen received his B.Sc. degree in software engineering from the Donghua University, Shanghai, China, in 2018, where he is currently pursuing his Master's degree in computer science and technology. His current research interests include cloud computing, crowd intelligence, and machine learning.

Zhaohui Zhang received his B.Sc. degree in computer science from the Anhui Normal University, Wuhu, China, in 1994, his Master's degree from the University of Science and Technology of China, in 2000, and his Ph.D. degree in computer science from the Tongji University, Shanghai, China, in 2007. He became a teacher at the Anhui Normal University. He was Professor with the Anhui Normal University, in July 2015. He currently serves as Professor in the School of Computer Science and Technology, Donghua University, Shanghai. His research interests include network information services, service computing, and cloud computing.

Computing and Informatics, Vol. 39, 2020, 83–104, doi: 10.31577/cai_2020_1-2_83

VIRTUAL MACHINE DEPLOYMENT STRATEGY BASED ON IMPROVED PSO IN CLOUD COMPUTING

Shanchen Pang, Dekun Dong, Shuyu Wang

College of Computer Science and Technology
China University of Petroleum
Qingdao, 266580, China
e-mail: pangsc@upc.edu.cn, z18070068@s.upc.edu.cn

Abstract. Energy consumption is an important cost driven by growth of computing power, thereby energy conservation has become one of the major problems faced by cloud system. How to maximize the utilization of physical machines, reduce the number of virtual machine migrations, and maintain load balance under the constraints of physical machine resource thresholds that is the effective way to implement energy saving in data center. In the paper, we propose a multi-objective physical model for virtual machine deployment. Then the improved multi-objective particle swarm optimization (TPSO) is applied to virtual machine deployment. Compared to other algorithms, the algorithm has better ergodicity into the initial stage, improves the optimization precision and optimization efficiency of the particle swarm. The experimental results based on CloudSim simulation platform show that the algorithm is effective at improving physical machine resource utilization, reducing resource waste, and improving system load balance.

Keywords: Cloud computing, Pareto optimal solution, particle swarm optimization algorithm, resource reservation, virtual machine deployment

Mathematics Subject Classification 2010: 68-W99, 68-M01

1 INTRODUCTION

Cloud computing is a distributed computing model that provides available, convenient and on-demand network access to shared resource pools (such as facilities, applications, storage devices, etc.). Resources as a service is a primary form of cloud

computing, mainly divided into infrastructure, platform and software. Through cloud computing, consumers can use or uninstall resources anytime, anywhere, which improves service quality and reduces operation costs.

The cloud data center uses the virtualization technology to construct computing resources, storage resources and network resources into a dynamic virtual resource pool. It uses virtual resource management technology to realize automatic deployment, dynamic expansion and on-demand allocation of cloud computing resources. Generalized virtualization includes virtual memory, virtual machines, storage virtualization, etc. We focus on virtualization of physical machines in this paper. As the number of cloud users increases, more virtual machine requests are added, what increases the pressure on physical machines. Thus cloud data center requires an effective virtual machine deployment strategy. The deployment problem is proved to be a non-deterministic polynomial problem, which meant we cannot find a precise solution. But with the help of some intelligent algorithms, some sub-optimal solutions could be found [26].

Kennedy and Eberhart developed Particle Swarm Optimization (PSO), which is considered a new swarm intelligence and evolutionary algorithm [7]. It has the advantages of fast search speed, simple implementation, high efficiency and so on, which has attracted the attention of the academic community. It shows its ascendancy in solving practical problems. However, swarm intelligence algorithms, such as ant colony algorithm, differential evolutionary algorithm, genetic algorithm have the problem of local optimization.

Existing virtual machine deployment strategies only consider resource utilization or virtual machine migration, and ignore the impact of load balancing on system performance. In the paper, we proposed an improved multi-objective particle swarm optimization algorithm (TPSO). The algorithm introduces chaotic strategy into the initial stage of the iterative process, thus the particle swarm distribution has better ergodicity and uniformity. In the medium term, random grouping strategy is used as the core process of the algorithm. Clustering is applied later, which increases the fineness of the end of the search and ensures convergence normally. Multi-objective selection in the updating process of particle swarm algorithm is a representative problem. Most of them add weight to the objective function and convert it into a single objective function. We set three objective functions, and seek the optimal solution of the algorithm by Technique for Order Preference by Similarity to an Ideal Solution method (TOPSIS). To improve physical machine resource utilization and save cloud system energy consumption, we apply the improved algorithm to virtual machine deployment, take physical machine resource utilization, virtual machine migration and load balance as the objective functions.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives the resource waste model. Section 4 gives the improved algorithm design. The experiments and results are discussed in Section 5, and the Section 6 summarizes our work.

2 RELATED WORK

In recent years, PSO has become a focus of researchers due to its simple structure, few parameters and easy implementation of code, especially in image recognition, path planning and artificial intelligence. The theoretical research of this algorithm is mainly divided into two categories. The first category focuses on the topological structure, parameter optimization and population diversity of the algorithm. The second category mainly includes the combination of the algorithm and other intelligent algorithms.

Particle swarm optimization has performed advantages in solving practical problems and has a great development value and development space. However, due to the quick loss of diversity, it tends to fall into local optima. In order to enhance the performance of particle swarm, Wang et al. [23] proposed DNSPSO based on the enhanced diversity and neighborhood search, but the neighborhood radius may affect the effectiveness of the neighborhood search. For the traditional linear learning strategy, Zhao et al. [31] proposed a new position storage mechanism. The algorithm efficiently and quickly seeks high-quality solutions. To avoid premature convergence of the algorithm, Wang et al. [24] presented GOPSO by speeding up convergence and escaping local optimization, but generalized opposition-based learning performs badly on shifted and large scale problems. Cho et al. [1] presented a novel multimodal optimization algorithm, which used deterministic sampling to produce new particles during the optimization process. This paper did not address the case in which there are more local optima than could possibly be detected. In PSO, different inertia weight strategies can influence the performance of algorithm. Nickabadi et al. [13] summarized various inertial weight strategies and proposed a new adaptive inertia weight strategy based on the success rate of the particles. In order to improve the search efficiency in the complex problem spaces, Zhan et al. [29] proposed an orthogonal learning method (OL). Through orthogonal experimental design, the particle swarm can discover more useful information that exists in the best experience of history and the best experience of neighbors. Based on perturbing global best strategy, Zhao et al. [33] proposed an modified discrete immune optimization algorithm. Zhao et al. [32] guided the search strategy with multigroup information communication and sharing mechanism and multi-stage global disturbance in their paper, they considered the population diversity and selection pressure simultaneously. In the second category, classic intelligent algorithms include ant colony algorithm, genetic algorithm, and differential evolutionary algorithm [3, 4, 21, 16].

Virtual machine deployment is the process of assigning virtual machines to physical machines by the allocation strategy. Since the cloud infrastructure is completely virtual, the deployment of virtual machines becomes a core issue for cloud systems. To this end, a lot of research has been done on optimizing the deployment of virtual machines. Works [8, 28, 30, 2] focused on virtual machine deployment through particle swarm optimization, most of them focused on one or two optimization goals. Wilcox et al. [27] and Nguyen et al. [17] mainly took into account two goals based on

the genetic algorithm. Maurer et al. [12] proposed a VM deployment algorithm that used heuristic packing algorithms and forecasting techniques, they achieved the minimum number of physical machines and ensured a certain amount of service-level agreement. For reducing the unnecessary migrations, Sato et al. [18] used Auto Regressive Model to predict virtual machine usage, but it must be evaluated on the servers that the VMs are deployed on. For some performance bottlenecks with MapReduce, Shabeera and Madhu Kumar [20] and Li et al. [9] optimized MapReduce performance under the specified constraints. Wang et al. [25] balanced power consumption and quality of service by controlling the number of servers and virtual machine time sharing. Based on considering VM placement and data placement simultaneously, Shabeera et al. [19] proposed a meta-heuristic algorithm using ant colony algorithm. Song et al. [22] saved data center energy by effectively utilizing resource fragments, but they inevitably increased the number of VM migrations to further categorize the different intervals. Pang et al. [15] proposed a task-oriented resource allocation method based on ACO, it can reduce the power consumption of data center effectively on the premise of performance guarantee. Ma et al. [11] optimized time and power based on Genetic Algorithm, but the encoding process is complicated and the search speed is slow. Through estimation of distribution algorithm and genetic algorithm, Pang et al. [14] developed an EDA-GA hybrid scheduling algorithm, however, this paper does not consider the dynamics and uncertainty of the cloud computing environment. For example, the computing speed of virtual machines changes in real time.

3 RESOURCE WASTE MODEL

3.1 Problem Description

In the paper, we proposed a virtual machine deployment scheme based on particle swarm optimization. The scheme reduces the energy consumption of the cloud system by rationally deploying virtual machines. In the cloud system, virtual machines are divided into the deployed virtual machines and the newly requested virtual machines. The VM deployment framework is shown in Figure 1.

The virtual machine deployment solution proposed in this paper monitors the resource state of physical machines, including CPU, memory, and network bandwidth. Suppose we set n physical machines in the cloud environment. The threshold of the physical machine (PM) is represented by the triplet as $R_j = \{R_j^C, R_j^M, R_j^B\}$, where R_j^C represents the CPU threshold, R_j^M represents the memory threshold, R_j^B represents the bandwidth threshold. There are m virtual machines. Virtual machine (VM) requirements are represented by the triplet as $V_j = \{V_j^C, V_j^M, V_j^B\}$, where V_j^C represents the CPU requirement, V_j^M represents the memory requirement, V_j^B represents the bandwidth requirement. $m = m_r + m_s$, m_r and m_s indicate the number of newly requested virtual machines and deployed virtual machines, respectively. The matrix represents the deployment of the virtual machine. Each term in the matrix x_{ij} is 0 or 1. If $x_{ij} = 1$, representing the i^{th} VM is deployed on the j^{th} PM.

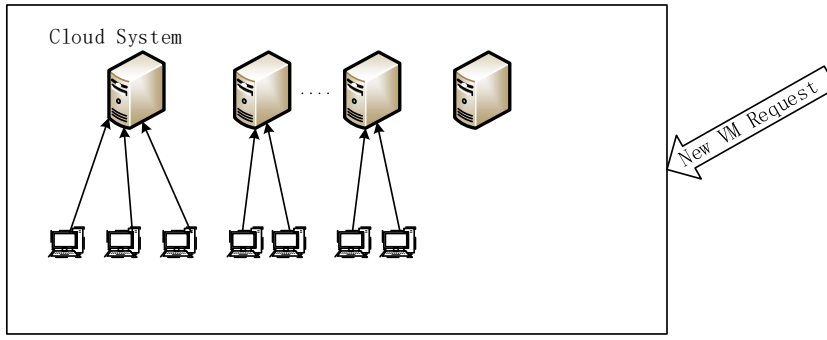


Figure 1. Deployment framework for VMs

If $x_{ij} = 0$, it is not deployed there. The matrix is expressed as follows:

$$\begin{bmatrix} x_{11} & x_{12} \dots & x_{1n} \\ x_{21} & x_{22} \dots & x_{2n} \\ \dots & & \\ x_{m1} & x_{m2} \dots & x_{mn} \end{bmatrix}. \tag{1}$$

3.2 Objective Functions and Constraint Functions

The objective functions are as follows:

$$\text{Max} f_1 = \left(\frac{\sum_{i=1}^{m} V_i^C}{\sum_{j=1}^{n} R_j^C} + \frac{\sum_{i=1}^{m} V_i^M}{\sum_{j=1}^{n} R_j^M} + \frac{\sum_{i=1}^{m} V_i^B}{\sum_{j=1}^{n} R_j^B} \right) /3, \tag{2}$$

$$\text{Min} f_2 = \sum_{i=1}^{m} m_i, \tag{3}$$

$$\text{Min} f_3 = \sqrt{\left(\sum_{j=1}^{n} (C_{juse} - C_{use})^2 + \sum_{j=1}^{n} (M_{juse} - M_{use})^2 + \sum_{j=1}^{n} (B_{juse} - B_{use})^2 \right) /n}. \tag{4}$$

The constraint functions are as follows:

$$\sum_{j=1}^{n} x_{ij} = 1, \tag{5}$$

$$\sum_{i=1}^{m} V_j^C \times x_{ij} < R_j^C, \tag{6}$$

$$\sum_{i=1}^{m} V_j^M \times x_{ij} < R_j^M, \tag{7}$$

$$\sum_{i=1}^{m} V_j^B \times x_{ij} < R_j^B. \tag{8}$$

Equation (2) represents virtual machine utilization. Equation (3) represents the number of migrations of the virtual machine. Equation (4) represents the load imbalance, where C_{use}, M_{use}, and B_{use} represent the average utilization of the system CPU, memory, and bandwidth, respectively. Correspondingly, C_{juse}, M_{juse} and B_{juse} respectively represent the CPU, memory and bandwidth utilization of the j^{th} physical machine. Equation (5) means that the same VM can only be deployed on one PM. Equations (6), (7), (8) indicate that the resource requirements (CPU, memory and bandwidth) of virtual machines on the same physical machine do not exceed its corresponding threshold.

3.3 Chaos Strategy

Chaos is a motion system synthesized when both the deterministic and random components of the system are clearly present, and it is a contradictory unity that exists objectively. Chaos optimization algorithm is a search algorithm that transforms variables from chaotic space to solution space. The algorithm has the advantages of global asymptotic convergence, easy to escape the local optimum, and fast convergence speed. To improve the ergodicity and uniformity, we use the chaos strategy to initialize the particle swarm.

3.4 Pareto Optimal Solution

Pareto optimality is an ideal state in the process of resource allocation. In some multi-objective models, Pareto optimality does not exist. We can only weigh each target to choose a solution from the Pareto front. In the data center, when we increase the utilization of physical machines, migrating virtual machine is inevitable, however, excessive virtual machine migration is bound to increased energy consumption. At the same time, load balance also interacts with them, which is obviously an optimal problem in Pareto. We use the TOPSIS method to optimize this problem in virtual machine dynamic deployment.

4 IMPROVED ALGORITHM DESIGN

4.1 Classic Particle Swarm Optimization

PSO belongs to one of swarm intelligence and evolutionary algorithms. The main idea of particle swarm optimization is that through multiple iterations, particles use their own experience and the experience of the group to gradually find the best solution to the problem. Unlike genetic algorithms, particle swarms do not use selection. Normally, all members of the group can survive from the beginning of the experiment to the end. Through multiple iterations and interactions between individuals, the optimal solution to the problem is obtained [5].

In each iteration, the best experience of each particle in the search space is called the individual extreme value pBest. The best experience of the population is the current global optimal solution of the entire particle swarm, called the global extremum gBest. All particles adjust themselves through individual optimal values and global optimal values. The formula is as follows:

$$V_{id}(t+1) = V_{id}(t) + r_1 \times c_1 \times (P_{id}(t) - X_{id}(t)) + r_2 \times c_2 \times (P_{gd}(t) - X_{id}(t)), \quad (9)$$

$$X_{id}(t+1) = V_{id}(t) + X_{id}(t). \quad (10)$$

The formula (9) is the particle velocity update formula, which contains three parts. The first part is the speed of the t^{th} generation, called the particle inertia speed part. The second part updates the speed through the individual optimal value, which is the individual cognitive part. The third part updates the speed through the group optimal value, which is the social cognitive part. Formula (10) is the particle position update formula.

4.2 Improved Multi-Objective Particle Swarm Optimization Algorithm Design

Kennedy claimed that particle swarms with large neighborhoods achieve better at solving simple problems, while particle swarms with small neighborhoods may implement better at solving complex problems [6]. In the paper, we improved the traditional PSO. Firstly, chaotic algorithm initializes the particle swarm. The first ninety percent of the iterative process uses a random grouping strategy, and then the last ten percent uses clustering convergence.

4.3 Chaos Mapping

By using the sensitivity and ergodicity of the initial value of chaotic mapping, a particle is randomly initialized, and then the initial value of multiple particles is obtained by chaotic mapping. Chaos mapping is used to expand the initial particle swarm, changes the extraction process of initial particle swarm. Tent mapping has simple structure and good traversal property. The mapping steps are as follows.

Tent mapping:

$$x_{n+1} = \begin{cases} 2x_n, & 0 \le x_n \le 0.5, \\ 2(1 - x_n), & 0.5 < x_n \le 1. \end{cases} \tag{11}$$

Step 1. Randomly initialize particle i, the speed is $V_{id} = (v_{i1}, v_{i2}, \ldots, v_{id})$ and the position is $X_{id} = (x_{i1}, x_{i2}, \ldots, x_{id})$.

Step 2. The particle i is iterated b times respectively according to Equation (11), that is, b initial particles' velocity vectors and position vectors are obtained.

Step 3. The b particles in Step 2 are still iterated c times according to Equation (11) to get the initial particle swarm p, $p = b \times c$, and get their velocity vectors and position vectors.

4.4 Inertia Weight

The value of the inertia weight affects the convergence speed and convergence accuracy. When w is large, the particle swarm tends to be globally optimized. On the contrast, if w is small, particle swarms tend to be locally optimized. According to the influence of w value on the search results, we use the linear decrement method to set the w value in this paper. As shown in the following formula:

$$w = w_{\max} - (w_{\max} - w_{\min}) \times \sqrt{t/\text{total\_}t} \tag{12}$$

where $w_{\max}$ is 0.95, $w_{\min}$ is 0.05, $total\_t$ represents the total number of iteration, and t represents the current number of iteration. The particle swarm search area is large in the early period. In the later period, the particle swarm switched from global search to local search as w slowly decreases, and the algorithm does not converge too fast.

4.5 Random Grouping and Clustering Convergence

Two particles are not sufficient to construct a good swarm, while five particles show better local search ability and reduce global search ability. Three particles achieve a balance between them, the swarms show better global search ability [10]. The iterative process of the particle swarm algorithm is first updated by a random grouping strategy. Three particles form a small group, and the particles are randomly reorganized every 5 generations. The steps for random grouping are shown in Algorithm 1.

In the later stage of the improved particle swarm optimization. The k-means clustering algorithm based on Euclidean distance is used to converge the particle swarms. The k values are successively reduced to 1, that is, they are merged into one particle swarm. After each cluster center is stabilized, the particle swarm is updated within each group. The cluster convergence steps are shown in Algorithm 2.

Algorithm 1 Random grouping iteration process

Require: m: Small group members; n: Small groups' number; D: Regrouping period; $total\_t$: The total number of iteration;
Ensure: swarm p

```
initial m = 3 and D = 5;
Initialize p = 3 × n particles by chaos strategy;
Grouping the population randomly;
for i = 1;0.9 × total_t do
    Update each small group by formula (9), (10);
    if mod(i, D) == 0 then
        Regroup the small group randomly;
    end if
end for
```

Algorithm 2 Cluster convergence iteration process

Require: e: Each swarm's population; D: Regrouping period;
Ensure: optimal particle

```
initial e = 6 and D = 5;
for i = 0.9 × total_t; total_t || k = 1;e = e + 6 do
    Select k = p/e particles as the initial centroid ;
    repeat
        Assign each point to the nearest centroid;
        Form k clusters;
        Recalculate the centroids of each cluster;
    until The center of mass does not change;
    Use the information in the group to update the velocity and position;
    if mod(i, D) == 0 then
        Clustering regroup the swarms randomly
    end if
end for
```

4.6 Pareto Optimal Solution Design

The TOPSIS method is an effective multi-index evaluation method. The basic principle is to sort by calculating the relative distance between each object and the best and worst solutions. In this paper, it is applied to particle selection with multi-objective optimal value in TPSO algorithm. The strategy is as follows.

The decision matrix is as follows:

$$v = \begin{bmatrix} v_{11}v_{12} & v_{13} \\ v_{21}v_{22} & v_{23} \\ \cdots & \\ v_{p1}v_{p2} & v_{p3} \end{bmatrix} \tag{13}$$

where v_{ij} represents the function value of the deployment strategy of i^{th} particle in the j^{th} objective function.

Step 1. Process the objective function to the same trend.

$$f_1^{'} = 1/f_1. \tag{14}$$

Step 2. Vector normalization normalizes the decision matrix V.

$$X_{ij} = v_{ij} / \sqrt{\sum_{i=1}^{p} v_{ij}^2} \quad (i = 1, 2, \ldots, p; j = 1, 2, 3). \tag{15}$$

Step 3. Determine the ideal and the negative ideal solution.

Ideal solution:

$$S^+ = \left\{ \left(\min_{1 \le i \le p} x_{ij} \right) \mid i = 1, 2, \ldots, p \right\} = \left\{ x_1^+, x_2^+, x_3^+ \right\}. \tag{16}$$

Negative ideal solution:

$$S^- = \left\{ \left(\max_{1 \le i \le p} x_{ij} \right) \mid i = 1, 2, \ldots, p \right\} = \left\{ x_1^-, x_2^-, x_3^- \right\}. \tag{17}$$

Step 4. Determine the distance of each solution to the actual solution.

$$S_i^+ = \sqrt{\sum_{j=1}^{3} \left(x_{ij} - x_i^+ \right)^2}, \tag{18}$$

$$S_i^- = \sqrt{\sum_{j=1}^{3} \left(x_{ij} - x_i^- \right)^2}. \tag{19}$$

Step 5. Calculate the relative closeness of each solution to the ideal solution.

$$C_i^* = S_i^- / (S_i^+ + S_i^-). \tag{20}$$

Step 6. Prioritize the scheme according to the size of C_i^*. The one with the highest relative closeness is the selected particle.

4.7 Virtual Machine Coding

The m virtual machines to be deployed are first coded from 1 to m. Then we obtain the correspondence between the virtual machines and the physical machines

through the TPSO algorithm. Finally, the virtual machines are deployed on the corresponding physical machines according to the mapping relationship. Thereby we achieve the optimization goal. For example, $X_{id} = (1, 5, 4, 2, 3, 6 \ldots)$ means: Virtual machines No. 1, 5, 4, 2, 3, and 6 are deployed on physical machines No. 1, 2, 3, 4, 5, and 6, respectively, and so on. The value of the particle position vector represents a deployment strategy of the virtual machine. The deployment scenarios for a VM resource is shown in Figure 2.

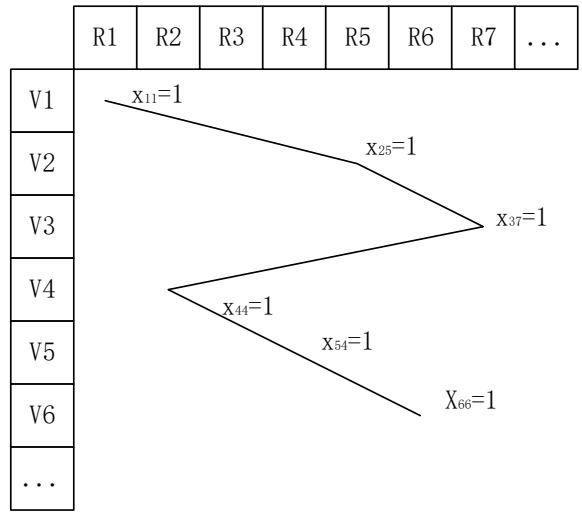


Figure 2. Deployment scenarios for a VM resource

4.8 Improved Algorithm Steps

Step 1. Set the number of particles, iterations, the learning factor, the inertia weight, the velocity threshold, and the position threshold.

Step 2. Initialize to generate p particles, and the chaotic mapping initializes the initial position and velocity of the particles in the population.

Step 3. Randomly group the particles. The adaptive values of the particles in each population are calculated in turn according to the objective function Formulas (2), (3), (4) under constraint conditions (5), (6), (7), (8).

Step 4. Record the individual extremum of every particle in each small group, calculate and record the global extremum of each small group by TOPSIS method.

Step 5. Update the particle's velocity with the individual extremum and the global extremum obtained by Step 4 and Equation (9). Update the particle's location by Equation (10).

Step 6. Regroup the swarms randomly every 5^{th} generation.

Step 7. Determine if the number of iterations reaches $0.9 \times total\_t$, and if it is reached, go to Step 8. Otherwise go to Step 3 to continue.

Step 8. Update particles position and speed through cluster convergence until the end of the iteration.

5 EXPERIMENT

CloudSim is a tool set for simulating cloud computing environment and evaluating resource scheduling algorithms. We select the CloudSim simulation platform for simulation experiments in this paper, and combine the physical model of cloud computing resource scheduling – the improved particle swarm algorithm with the resource model in CloudSim. Finally, we implement the TPSO algorithm with the inheritance classes in the basic CloudSim class.

The experimental environment of this paper was set as follows: compile environment JDK 1.8, compile software Eclipse 4.6, and simulate cloud computing environment CloudSim 3.0.

The experimental parameters were set as follows: 200 physical machines, 400 virtual machines. All physical machines were homogeneous: CPU with 10 processing units, 20 GB of memory, 100 M bandwidth. We divided 400 VMs into four types, and each type of request had 100 each. We set the request for 400 virtual machines to arrive randomly after each optimization. For different configuration of the algorithms, the requests arrived in the same order. CloudSim resource scheduling mechanism is shown in Figure 3.

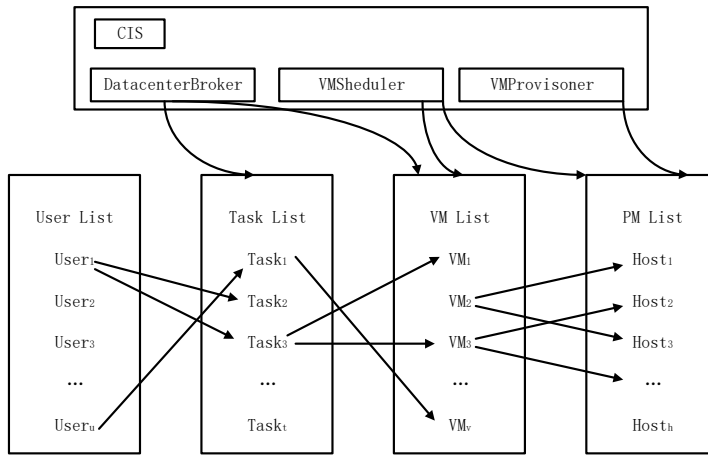


Figure 3. Resourse scheduling mechanism of CloudSim

We designed two experiments with CloudSim. The first experiment compares TPSO with single-objective PSO and DMS-PSO. The second experiment com-

pares the classic particle swarm optimization algorithm and the Round-Robin Algorithm (RR).

5.1 Experiment 1

In Experiment 1, we test the performance of the TPSO by six ten-dimensional benchmark functions. The equations are as follows:

1. Sphere Function
$$f(x) = \sum_{i=1}^{D} x_i^2$$
where $x \in [-5.12, 5.12]^D$.
2. Rosenbrock's Function
$$f(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$$
where $x \in [-2.048, 2.048]^D$.
3. Ackley's Function
$$f(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$$
where $x \in [-32.768, 32.768]^D$.
4. Griewank's Function
$$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4\,000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$
where $x \in [-600, 600]^D$.
5. Rastrigin's Function
$$f(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$$
where $x \in [-5.12, 5.12]^D$.
6. Weierstrass Function
$$f(x) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\ \max} \left[a^k \cos(2\pi b^k (x_i + 0.5))\right]\right) - D \sum_{k=0}^{k\ \max} \left[a^k \cos(2\pi b^k \cdot 0.5)\right]$$
where $a = 0.5$, $b = 3$, $k_{\max} = 20$, $x \in [-0.5, 0.5]^D$.

We use functions 1–6 to test the algorithm's optimization ability, global search ability, convergence speed, and whether it is liable to fall into a local optimum. The experimental results are as follows.

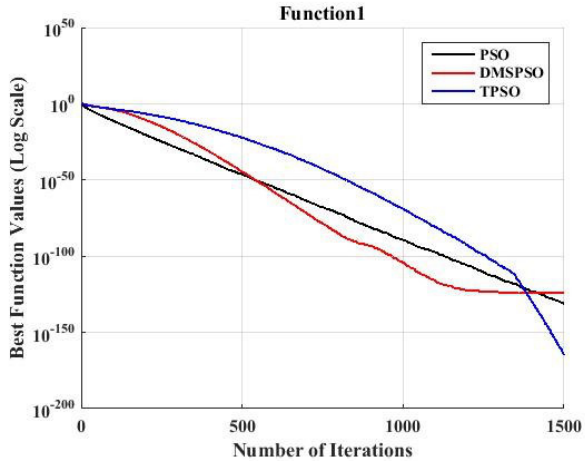


Figure 4. Results achieved under function1

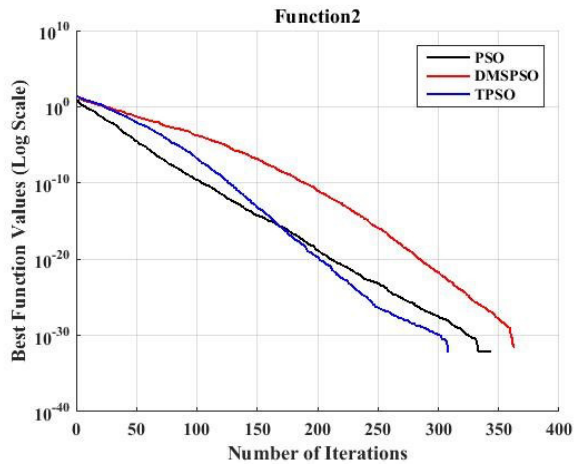


Figure 5. Results achieved under function2

As shown in Figures 4, 5, 6, 7, 8, 9, the six functions are based on experimental results under different iterations. The particle group by chaotic initialization has ergodicity and uniformity. In the early stage, groups constantly exchange information among small groups through random groupings. The late clustering convergence

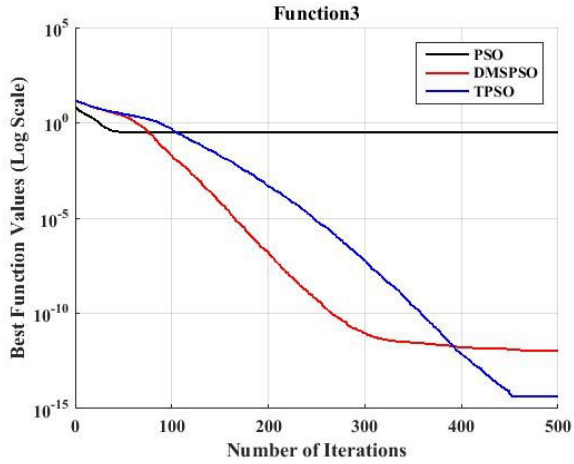


Figure 6. Results achieved under function3

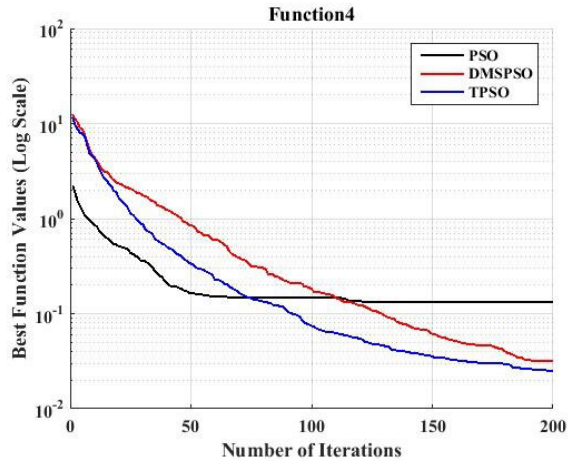


Figure 7. Results achieved under function4

makes the particle group more directional. Experiments show that the improved method TPSO is compared with two single-objective algorithms DMS-PSO and PSO, which is feasible and effective.

5.2 Experiment 2

In the second experiment, we tested and analyzed the performance of the multi-objective optimization algorithm TPSO for virtual machine resource deployment.

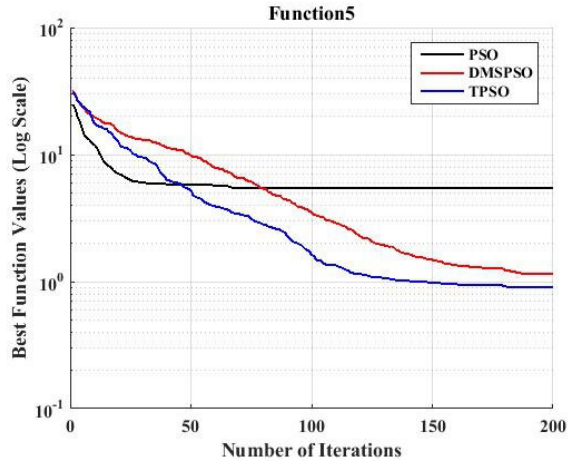


Figure 8. Results achieved under function5

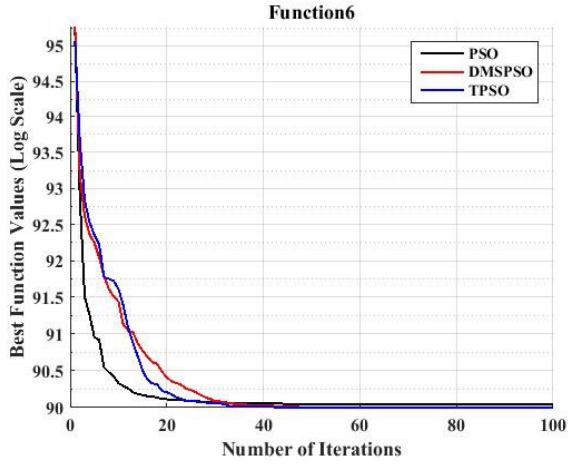


Figure 9. Results achieved under function6

We compared the multi-objective optimization algorithm TPSO with the traditional optimization algorithm PSO and the Round-Robin algorithm RR. The comparison was made from three aspects: resource utilization rate f1, virtual machine migration number f2 and load balance rate f3. Among them, the PSO had the same attention to the three goals, each accounting for 1/3. On the CloudSim platform, simulation experiments were carried out to record relevant data, and the experimental results were compared and analyzed. Finally, we obtained the experimental results distribution of the three algorithms.

Figure 10 shows the resource utilization of the three algorithms under different service requests. It proves that our proposed algorithm TPSO is effective and the resource utilization is higher than the other two algorithms. Figure 11 shows the times of virtual machine migrations. The virtual machine is migrated 220 times under PSO, and the virtual machine is migrated 160 times under TPSO. The proposed algorithm has fewer virtual machine migrations than PSO algorithm. Figure 12 shows the load imbalance for three algorithms at different iterations. It can be found from Figure 12 that the load imbalance of the TPSO algorithm is smaller than the other two algorithms, indicating that the property of the TPSO algorithm in load balance is better than other two algorithms. This is because the RR algorithm has poor dynamic adaptability, its efficiency is relatively low. The PSO algorithm lacks a resource selection mechanism and is liable to fall into local optimality. The population of TPSO algorithm has better ergodicity and uniformity.

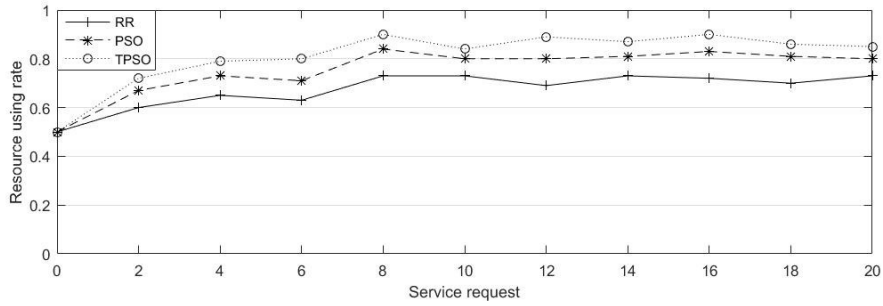


Figure 10. Resource using rate under different algorithms

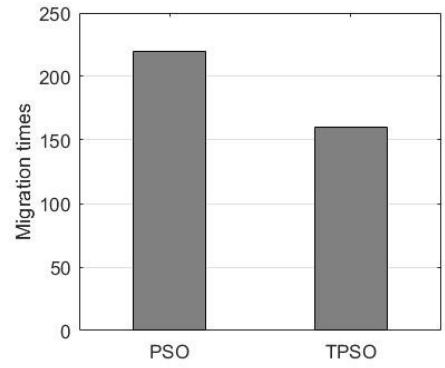


Figure 11. Migration times under different algorithms

The results of Experiment 2 show that the initial particle swarm distribution is more ergodic and uniform, maintaining the diversity of the population. Random

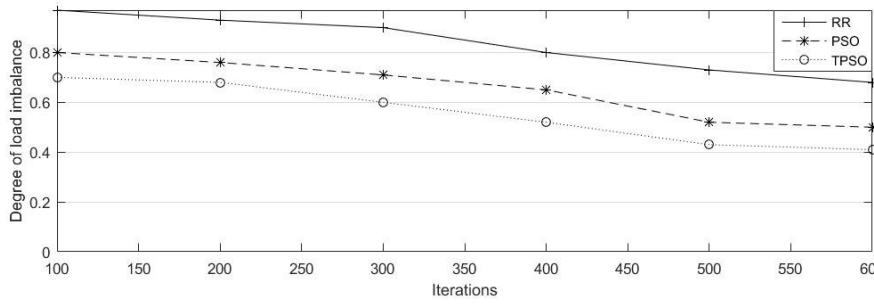


Figure 12. Load imbalance under different algorithms

grouping strategy is used in the medium term. These particles are frequently regrouped to exchange information among groups. Later clustering refined search. With keeping stability, the TOPSIS method selected the Pareto optimal solution. The Pareto optimal solution has better distribution and convergence. Compared to other two algorithms, TPSO has achieved better results in resource utilization, migration times and load balance.

6 CONCLUSIONS

Cloud computing improves the data center resource utilization through virtualization technology. As a key technology of cloud computing, virtual machine deployment algorithm has important research significance. The existing virtual machine deployment strategy only considers maximizing resource utilization and virtual machine migration, ignoring the impact of load balance on system performance. In the paper, we propose an improved multi-objective particle swarm optimization to balance the three goals (resource utilization, virtual machine migration, and load balance) to optimize data center performance.

The TPSO algorithm firstly initializes the particle swarm by chaos, making the population distribution have ergodicity and uniformity. We use the small group iterative update in the middle stage, these small groups are frequently regrouped to exchange information among groups, and later use clustering to converge the particle swarm algorithm, increasing the refinement of the late search. We designed two experiments: Experiment 1 compares the improved method TPSO with single-objective algorithms DMS-PSO and PSO, and verifies the effectiveness of the improved method. Experiment 2 compares TPSO with RR and PSO algorithm. The experimental results show that the algorithm balances the three objectives of resource utilization, virtual machine migration and load balance, and optimizes data center performance.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Nos. 61672033, 61873280, 61873281, 61972416, 61672248, 61902430), National Key Research and Development Project (No. 2018YFC1406204), Key Research and Development Program of Shandong Province (No. 2019GGX101067), Natural Science Foundation of Shandong Province (No. ZR2019MF012), Taishan Scholars Fund (No. ZX20190157), Independent Innovation Research Project (No. 18CX02152A), Fundamental Research Funds for the Central Universities (No. 19CX02028A).

REFERENCES

[1] Cho, H.—Kim, D.—Olivera, F.—Guikema, S. D.: Enhanced Speciation in Particle Swarm Optimization for Multi-Modal Problems. European Journal of Operational Research, Vol. 213, 2011, No. 1, pp. 15–23, doi: 10.1016/j.ejor.2011.02.026.

[2] Dashti, S. E.—Rahmani, A. M.: Dynamic VMS Placement for Energy Efficiency by PSO in Cloud Computing. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 28, 2016, No. 1-2, pp. 97–112, doi: 10.1080/0952813X.2015.1020519.

[3] Garg, H.: A Hybrid PSO-GA Algorithm for Constrained Optimization Problems. Applied Mathematics and Computation, Vol. 274, 2016, pp. 292–305, doi: 10.1016/j.amc.2015.11.001.

[4] Higashi, N.—Iba, H.: Particle Swarm Optimization with Gaussian Mutation. Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS '03), 2003, pp. 72–79, doi: 10.1109/SIS.2003.1202250.

[5] Kennedy, J.: Particle Swarm Optimization In: Sammut, C., Webb, G. I. (Eds.): Encyclopedia of Machine Learning. Springer, Boston, MA, 2011, pp. 760–766, doi: 10.1007/978-0-387-30164-8_630.

[6] Kennedy, J.: Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance. Proceedings of the 1999 Congress on Evolutionary Computation (CEC '99), IEEE, 1999, Vol. 3, pp. 1931–1938, doi: 10.1109/CEC.1999.785509.

[7] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of International Conference on Neural Networks (ICNN '95), IEEE, 1995, Vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[8] Kumar, D.—Raza, Z.: A PSO Based VM Resource Scheduling Model for Cloud Computing. 2015 IEEE International Conference on Computational Intelligence and Communication Technology, 2015, pp. 213–219, doi: 10.1109/CICT.2015.35.

[9] Li, M.—Subhraveti, D.—Butt, A. R.—Khasymski, A.—Sarkar, P.: CAM: A Topology Aware Minimum Cost Flow Based Resource Manager for MapReduce Applications in the Cloud. Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing (HPDC '12), 2012, pp. 211–222, doi: 10.1145/2287076.2287110.

[10] Liang, J. J.—Suganthan, P. N.: Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search. 2005 IEEE Congress on Evolutionary Computation (CEC 2005), 2005, Vol. 1, pp. 522–528, doi: 10.1109/CEC.2005.1554727.

[11] Ma, T.—Pang, S.—Zhang, W.—Hao, S.: Virtual Machine Based on Genetic Algorithm Used in Time and Power Oriented Cloud Computing Task Scheduling. Intelligent Automation and Soft Computing, Vol. 25, 2019, No. 3, pp. 605–613.

[12] Maurer, M.—Emeakaroha, V. C.—Brandic, I.—Altmann, J.: Cost-Benefit Analysis of an SLA Mapping Approach for Defining Standardized Cloud Computing Goods. Future Generation Computer Systems, Vol. 28, 2012, No. 1, pp. 39–47, doi: 10.1016/j.future.2011.05.023.

[13] Nickabadi, A.—Ebadzadeh, M. M.—Safabakhsh, R.: A Novel Particle Swarm Optimization Algorithm with Adaptive Inertia Weight. Applied Soft Computing, Vol. 11, 2011, No. 4, pp. 3658–3670, doi: 10.1016/j.asoc.2011.01.037.

[14] Pang, S.—Li, W.—He, H.—Shan, Z.—Wang, X.: An EDA-GA Hybrid Algorithm for Multi-Objective Task Scheduling in Cloud Computing. IEEE Access, Vol. 7, 2019, pp. 146379–146389, doi: 10.1109/ACCESS.2019.2946216.

[15] Pang, S.—Zhang, W.—Ma, T.—Gao, Q.: Ant Colony Optimization Algorithm to Dynamic Energy Management in Cloud Data Center. Mathematical Problems in Engineering, Vol. 2017, 2017, Art. No. 4810514, 10 pp., doi: 10.1155/2017/4810514.

[16] Pant, M.—Thangaraj, R.—Abraham, A.: DE-PSO: A New Hybrid Meta-Heuristic for Solving Global Optimization Problems. New Mathematics and Natural Computation, Vol. 7, 2011, No. 03, pp. 363–381, doi: 10.1142/S1793005711001986.

[17] Quang-Hung, N.—Nien, P. D.—Nam, N. H.—Tuong, N. H.—Thoai, N.: A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud. In: Mustofa, K., Neuhold, E. J., Tjoa, A. M., Weippl, E., You, I. (Eds.): Information and Communication Technology (ICT EurAsia 2013). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7804, 2013, pp. 183–191, doi: 10.1007/978-3-642-36818-9_19.

[18] Sato, K.—Samejima, M.—Komoda, N.: Dynamic Optimization of Virtual Machine Placement by Resource Usage Prediction. 2013 11th IEEE International Conference on Industrial Informatics (INDIN), 2013, pp. 86–91, doi: 10.1109/INDIN.2013.6622863.

[19] Shabeera, T. P.—Madhu Kumar, S. D.—Salam, S. M.—Krishnan, K. M.: Optimizing VM Allocation and Data Placement for Data-Intensive Applications in Cloud Using ACO Metaheuristic Algorithm. Engineering Science and Technology, an International Journal, Vol. 20, 2017, No. 2, pp. 616–628, doi: 10.1016/j.jestch.2016.11.006.

[20] Shabeera, T. P.—Madhu Kumar, S. D.: Optimising Virtual Machine Allocation in MapReduce Cloud for Improved Data Locality. International Journal of Big Data Intelligence, Vol. 2, 2015, No. 1, pp. 2–8, doi: 10.1504/IJBDI.2015.067563.

[21] Shelokar, P. S.—Siarry, P.—Jayaraman, V. K.—Kulkarni, B. D.: Particle Swarm and Ant Colony Algorithms Hybridized for Improved Continuous Optimization. Applied Mathematics and Computation, Vol. 188, 2007, No. 1, pp. 129–142, doi: 10.1016/j.amc.2006.09.098.

[22] Song, T.—Wang, Y.—Li, G.—Pang, S.: Server Consolidation Energy-Saving Algorithm Based on Resource Reservation and Resource Allocation Strategy. IEEE Access, Vol. 7, 2019, pp. 171452–171460, doi: 10.1109/ACCESS.2019.2954903.

[23] Wang, H.—Sun, H.—Li, C.—Rahnamayan, S.—Pan, J.-S.: Diversity Enhanced Particle Swarm Optimization with Neighborhood Search. Information Sciences, Vol. 223, 2013, pp. 119–135, doi: 10.1016/j.ins.2012.10.012.

[24] Wang, H.—Wu, Z.—Rahnamayan, S.—Liu, Y.—Ventresca, M.: Enhancing Particle Swarm Optimization Using Generalized Opposition-Based Learning. Information Sciences, Vol. 181, 2011, No. 20, pp. 4699–4714, doi: 10.1016/j.ins.2011.03.016.

[25] Wang, J.—Huang, C.—Liu, Q.—He, K.—Wang, J.—Li, P.—Jia, X.: An Optimization VM Deployment for Maximizing Energy Utility in Cloud Environment. In: Sun, X. et al. (Eds.): Algorithms and Architectures for Parallel Processing (ICA3PP 2014). Springer, Cham, Lecture Notes in Computer Science, Vol. 8630, 2014, pp. 400–414, doi: 10.1007/978-3-319-11197-1_31.

[26] Wigderson, A.: P, NP and Mathematics – A Computational Complexity Perspective. Proceedings of the 2006 International Congress of Mathematicians, Madrid, 2006. EMS Publishing House, Zurich, 2007, pp. 665–712.

[27] Wilcox, D.—McNabb, A.—Seppi, K.: Solving Virtual Machine Packing with a Reordering Grouping Genetic Algorithm. 2011 IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 362–369, doi: 10.1109/CEC.2011.5949641.

[28] Xu, B.—Peng, Z.—Xiao, F.—Gates, A. M.—Yu, J.-P.: Dynamic Deployment of Virtual Machines in Cloud Computing Using Multi-Objective Optimization. Soft Computing, Vol. 19, 2015, No. 8, pp. 2265–2273, doi: 10.1007/s00500-014-1406-6.

[29] Zhan, Z.-H.—Zhang, J.—Li, Y.—Shi, Y.-H.: Orthogonal Learning Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation, Vol. 15, 2010, No. 6, pp. 832–847, doi: 10.1109/TEVC.2010.2052054.

[30] Zhao, G.: Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment. International Journal of Grid and Distributed Computing, Vol. 7, 2014, No. 1, pp. 33–42, doi: 10.14257/ijgdc.2014.7.1.04.

[31] Zhao, X.—Lin, W.—Zhang, Q.: Enhanced Particle Swarm Optimization Based on Principal Component Analysis and Line Search. Applied Mathematics and Computation, Vol. 229, 2014, pp. 440–456, doi: 10.1016/j.amc.2013.12.068.

[32] Zhao, X.—Liu, Z.—Yang, X.: A Multi-Swarm Cooperative Multistage Perturbation Guiding Particle Swarm Optimizer. Applied Soft Computing, Vol. 22, 2014, pp. 77–93, doi: 10.1016/j.asoc.2014.04.042.

[33] Zhao, X.—Song, B.—Huang, P.—Wen, Z.—Weng, J.—Fan, Y.: An Improved Discrete Immune Optimization Algorithm Based on PSO for QoS-Driven Web Service Composition. Applied Soft Computing, Vol. 12, 2012, No. 8, pp. 2208–2216, doi: 10.1016/j.asoc.2012.03.040.

Shanchen Pang received his graduation degree from the Tongji University of Computer Software and Theory, Shanghai, China, in 2008. He is Professor in the China University of Petroleum, Qingdao, China. His current research interests include theory and application of Petri net, service computing, trusted computing.

Dekun Dong graduated from the Shandong University of Science and Technology and got his Bachelor degree in engineering and management. Currently, he is pursuing his Master's degree in the China University of Petroleum. His research area is cloud computing.

Shuyu Wang received her graduation degree from the Shandong Women's University, Jinan, China, in computer science and technology in 2018. Currently, she is pursuing her Master's degree in the China University of Petroleum, Qingdao, China. Her current research interests include cloud computing and workflow scheduling.

Computing and Informatics, Vol. 39, 2020, 105–140, doi: 10.31577/cai_2020_1-2_105

MULTI-DIMENSIONAL RECOMMENDATION SCHEME FOR SOCIAL NETWORKS CONSIDERING A USER RELATIONSHIP STRENGTH PERSPECTIVE

Bo Zhang

College of Information, Mechanical and Electrical Engineering
Shanghai Normal University, Shanghai 200234, China
&
Institute of Artificial Intelligence on Education
Shanghai Normal University Shanghai 200234, China

Ya Zhang, Yanhong Bai, Jie Lian, Meizi Li

College of Information, Mechanical and Electrical Engineering
Shanghai Normal University, Shanghai 200234, China
e-mail: {lianjie, limeizi}@shnu.edu.cn

Abstract. Developing a computational method based on user relationship strength for multi-dimensional recommendation is a significant challenge. The traditional recommendation methods have relatively low accuracy because they lack considering information from the perspective of user relationship strength into the recommendation algorithm. User relationship strength reflects the degree of closeness between two users, which can make the recommendation system more efficient between users in pairs. This paper proposes a multi-dimensional comprehensive recommendation method based on user relationship strength. We take three main factors into consideration, including the strength of user relationship, the similarity of entities, and the degree of user interest. First, we introduce a novel method to generate a user candidate set and an entity candidate set by calculating the relationship strength between two users and the similarity between two entities. Then, the algorithm will calculate the user interest degree of each user in the user candidate set to each entity in the entity candidate set, if the user interest degree is larger than or equal to a threshold, this particular entity will be recommended to this user. The performance of the proposed method was verified based on the real-world social network

dataset and the e-commerce website dataset, and the experimental result suggests that this method can improve the recommendation accuracy.

Keywords: Recommendation system, social network, user relationship strength, user interest, entity similarity

1 INTRODUCTION

Nowadays, as all kinds of social networks (such as Facebook, Twitter, MySpace, etc.) are developing rapidly, these websites have become the major platforms for people's life, work and entertainment. Moreover, recommendation systems have been widely used in many e-commerce websites that can recommend products to target users according to the recommendation algorithms. Generally, the traditional recommendation algorithms ignored the user relationship in social networks, but the fact is that friends tend to have similar shopping preferences, and consumers may purchase a product based on what their friends purchased as well. Therefore, the traditional recommendation algorithms have relatively low accuracy in practical applications.

In recent years, many studies have addressed how to connect social networks to e-commerce websites in recommendation algorithms. The traditional recommendation algorithms mainly include the collaborative filtering recommendation algorithm [1, 2, 3], the content-based recommendation algorithm [4, 5, 6] and the knowledge-based recommendation algorithm [7, 8, 9], etc. In the collaborative filtering recommendation model, when the collaborative filtering recommendation algorithm is based on similar users, its performance is bad. And when the algorithm is based on similar entities, it results in some problems, such as data sparseness and cold start. In the content-based recommendation model, the recommendation results show that it is not ideal for unstructured information. In the knowledge-based recommendation model, the additional information that needs to be provided manually is required, and the information is not only difficult to obtain but also expensive. Therefore, from the perspective of user relationship strength, we suggest that more recommendation factors should be comprehensively considered in the recommendation system in order to achieve consumer satisfaction and maximize business profits.

Based on this analysis, we propose a novel multi-dimensional comprehensive recommendation method based on the social network. First, we present three algorithms to calculate user tightness, user interest degree and entity similarity, respectively. These algorithms are developed according to the social network analysis, such as the interaction frequency between users, comment stability, and similar communities. Then, an entity candidate set is generated based on the entities in the e-commerce website, and a user candidate set is generated based on the users in the social network. After that, the novel recommendation algorithm will recommend

entities from the entity candidate set to the target user. The recommended entity must satisfy certain conditions by using the correlation algorithm to the users in the user candidate set. To the best of our knowledge, considering both the entity similarity dimension, friend's tightness dimension and user interest dimension in recommendation methods was rarely studied before. And since the social network characteristics will affect users' purchase motivation to some extent, considering social network factors in the recommendation method can improve the recommendation accuracy.

In general, our work aims at improving the recommendation performance by providing a novel recommendation algorithm that can take both the user relationship, entity similarity and user interest degree into consideration. The main contributions are summarized as follows:

1. Three methods were introduced to define and estimate the tightness between users, the similarity between entities, and the user interest degree, respectively, by considering the comments stability between users, friend reliability, interaction frequency, mutual neighbors and similar communities, and some entity attributes.
2. A novel multi-dimensional comprehensive recommendation method based on user tightness, entity similarity, and user interest degree was proposed to recommend entities to the target users from the perspective of user relationship strength.

The rest of the paper is organized as follows: Section 2 reviews the literature related to this study, Section 3 presents the problem definition, Section 4 introduces the multi-dimensional comprehensive recommendation method, Section 5 presents the experimental results, and Section 6 concludes this study and provides some future suggestions.

2 RELATED WORKS

Recommendation system first introduced by Resnick and Varian [10] can provide product suggestions for users when users do online shopping based on information retrieval and information filtering. In general, the recommendation system contains three elements, including entities, users, and recommendation algorithm. According to different algorithms, recommendation systems can be divided into four types, which contain content-based recommendation systems, collaborative filtering recommendation systems, knowledge-based recommendation systems, and hybrid recommendation systems [11, 12].

A content-based recommendation system needs to calculate user similarities based on their historical purchase records, and extract user characteristics by statistics and machine learning methods. This system has been applied in many areas. For example, Puglisi et al. [13] proposed a content-based recommendation method and user privacy technique in social-tagging systems. Musto et al. [14] proposed a rec-

ommendation system by learning word embeddings from Wikipedia. Gu et al. [15] introduced a method by learning global term weights to the content-based recommendation system. In general, the content-based recommendation system can be used to deal with structured information (news and articles) well. However, for the unstructured information, it has a relatively low performance.

Recently, the collaborative filtering recommendation method has become one of the most successful methods that can realize personalized services. This method needs to calculate the similarity between the target user and the other users. And users with a bigger similarity tend to purchase similar products. The collaborative filtering method has been applied in many systems, such as joke recommendation [16], news recommendation [17] and movie recommendation [18]. Additionally, Fang et al. [19] proposed a generalized cross-domain collaborative filtering framework that can integrate social network information seamlessly with cross-domain data. Du et al. [20] developed a method based on the trust network that can improve the system performance greatly. Although the collaborative filtering method has been widely used, there are still some problems that need to be solved, such as data sparsity and scalability.

The knowledge-based recommendation system usually needs to use additional information about the current user and effective entities based on knowledge. This kind of system is often applied to specific areas, such as e-learning recommendation [21], music recommendation [22], and e-commerce product recommendation [23]. The major advantage of the knowledge-based recommendation system is that it can avoid the cold start problem because it does not need to rely on user information to calculate the product entity scores [24, 25].

The hybrid recommendation system employs a new algorithm that can combine the above three recommendation algorithms. For example, Wang et al. [26] proposed a hybrid recommendation model that contains two key components: incremental update item-based collaborative filtering and latent semantic analysis based relative term frequency algorithms. Zhu et al. [27] proposed a hybrid model combining the collaborative filtering algorithm with the knowledge map to represent the learning method, which can improve the recommendation performance greatly.

However, the previous methods still have some problems. First, most of the recommendation methods took only one factor into consideration, such as only the similarity between users, the similarity between entities, and the user's interest to the entity. Second, the traditional recommendation methods cannot be applied to social networks, which will affect the recommendation accuracy. In order to solve the above problems, in this study, we propose a multi-dimensional comprehensive recommendation method based on user relationship strength in social networks. This method considers and quantifies the tightness between users, the user explicit interest to entities, and the entity similarity, which can improve the recommendation performance in accuracy, coverage, and the recommendation diversification compared with traditional recommendation systems.

3 PROBLEM DEFINITION

The social network is a graph model that can describe the relationship between users. The vertexes in the graph model represent users, and the edges between vertexes represent the user relationship. The binary relation of social network graph can describe the relationship between users, which is consistent with the social connection between people in real life. Since a user and his (her) friends are most likely to have similar interests, prediction of user's preferences based on his (her) friends' interests is often used in the recommendation system nowadays. The relevant definitions are described below.

Definition 1 (Community Model)**.** A community $C = \langle CV, CE \rangle$ is a sub-graph of the social network, and it is composed of users who have similar interests, where $CV \subseteq V$, $CE \subseteq E$.

Definition 2 (Tightness of Users)**.** The tightness of two users reflects the closeness degree between them. The frequent contacts between the source user and target user normally represent that they trust each other, which means the link between them is stable. Therefore, to compute the tightness of users, five aspects are considered in this paper, including the comments stability, the friend reliability, the interaction frequency, the mutual neighbors and the similar communities. The tightness of users is denoted by closeness(su, tu), where su represents the source user, and tu represents the target user.

Definition 3 (User Interest Degree)**.** User interest degree $I(v|\text{item})$ reflects the interested level that the user v is to the entity item. It is commonly used to predict the purchase probability of a user to a particular entity. The user interest can be divided into explicit interest and implicit interest. Explicit interest can be expressed directly by the users' behaviors, such as commenting, browsing time, forwarding, and approving. The implicit interest means that the user may purchase the products in the same category as the product he (she) purchased before.

Definition 4 (Entity Similarity)**.** The entity similarity $\text{sim}(\text{item}_k, \text{item}_j)$ describes the degree of consistency between two entities, where item_k denotes the entity k and item_j denotes the entity j. The entity similarity is calculated based on four attributes, including category, price, quality, and discount.

With the notations introduced above in Table 1, we define our recommendation problem as follows. Given an e-commerce website, let user $u_i \in U = \{u_1, u_2, \ldots, u_n\}$, where U denotes a set of users, the friends set of user u_i is denoted by $\text{friend}(u_i) = \{\text{friend}_1, \text{friend}_2, \ldots, \text{friend}_x\}$. Now, if a user u_i has already purchased an entity $\text{item}_j \in \mathit{Item} = \{\text{item}_1, \text{item}_2, \ldots, \text{item}_m\}$, then some entities from Item will be recommended to u_i's friends from the friends set $\text{friend}(u_i)$, based on the tightness of users closeness(su, tu), entity similarity $sim(\text{item}_k, \text{item}_j)$ and user interest degree $I(v|\text{item})$.

Algebraic Symbol	Description
C	The community is composed of users with similar interests
$C = \langle CV, CE \rangle$	A sub-graph of social network
$CV \subseteq V$	The users in the community belong to users in the social network
$CE \subseteq E$	The edges between users in the community belong to the edges in the social network
su	The source user
tu	The target user
closeness(su, tu)	The tightness strength between user su and user tu
item$_k$	Entity k
item$_j$	Entity j
sim(item$_k$, item$_j$)	The similarity of two entities
$I(v$|item$)$	The interest level of user v to an entity item

Table 1. Algebraic symbols corresponding to the description

The architecture of our proposed model is shown in Figure 1. The social network consists of a large number of users and user relationships, expressed by a graph model which is the basis of the multidimensional comprehensive recommendation algorithm based on user relationship strength proposed in this paper. We need to complete the calculation of the user relationship strength in this research on the basis of social networks, and it is crucial for the establishment of our recommendation model. The core recommendation algorithm module is comprised of three sub-modules M_1, M_2, and M_3. The sub-module M_1 represents the modeling and analysis of user relationship strength. In this sub-module, if the relationship strength between user u_i and u_i's friend friend$_y$, denoted by closeness(u_i, friend_y), is larger than or equal to a threshold γ, then the user friend$_y$ will be added to the user candidate set R_user. The sub-module M_2 represents the modeling and analysis of entity similarity between entities. In this sub-module, if the user u_i has purchased an entity item$_j$, then calculate the entity similarity sim(item$_k$, item$_j$) between the item$_j$ and item$_k \in$ Item. And if the similarity is larger than or equal to the threshold α, this particular entity item$_k$ will be added to the entity candidate set R_item. The sub-module M_3 represents the modeling and analysis of how the users in the user candidate set are interested in the entities in the entity candidate set. If the interest degree of the user $u_i \in$ R_user to the entity item$_k \in$ R_item is larger than or equal to the threshold β, then the entity item$_k$ will be recommended to the user u_i.

Obviously, the core modules of the multi-dimensional comprehensive recommendation method proposed in this study are the sub-models M_1, M_2, and M_3, which can calculate the user relationship strength, the entity similarity, and the user interest degree, respectively. Therefore, the three sub-modules will be introduced in detail in the next section.

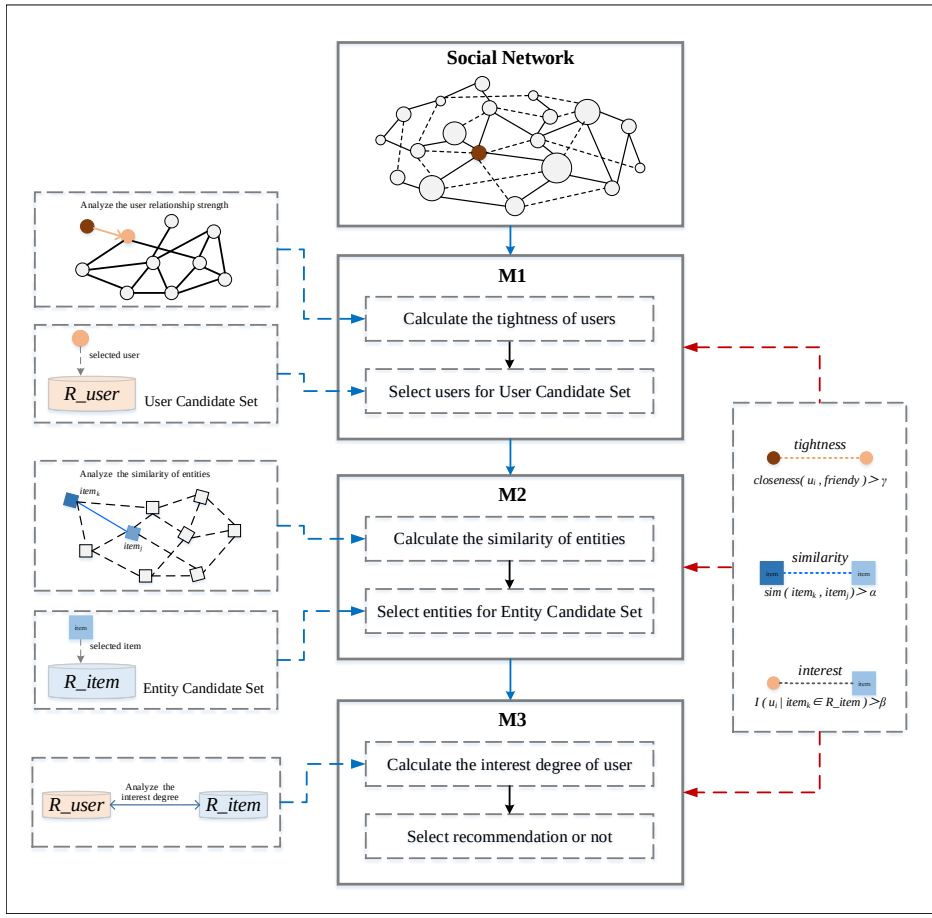


Figure 1. The architecture of the multi-dimensional comprehensive recommendation method

4 METHOD

4.1 Tightness of Users in Social Network

The connection between users contains direct connection and indirect connection in the social network graph model, whereas only the direct connection is considered in this study. The strength degree of a link reflects the closeness between the two users [28]. A bigger strength degree of a link indicates there is more frequent contact between the source user and the target user, which means they trust each other in a stable way [29]. Normally, four aspects between users are considered to calculate the user tightness, including the comments stability, the reliability of users, the

frequency of interaction, and mutual neighbors and similar communities [30]. It is shown in Figure 2, where the su represents the source user extracting from the social network, and tu represents the target user that is connected directly with the source user. For example, assume that there is a source user su whose target user tu is from the set of users directly connected to su. To calculate the user tightness between su and tu, firstly, the scores of COM_STA(su, tu), MUT_REL(su, tu), INT_FRE(su, tu) and C_nei-com(su, tu) need to be calculated separately, then the score of the user tightness is calculated based on these four scores.

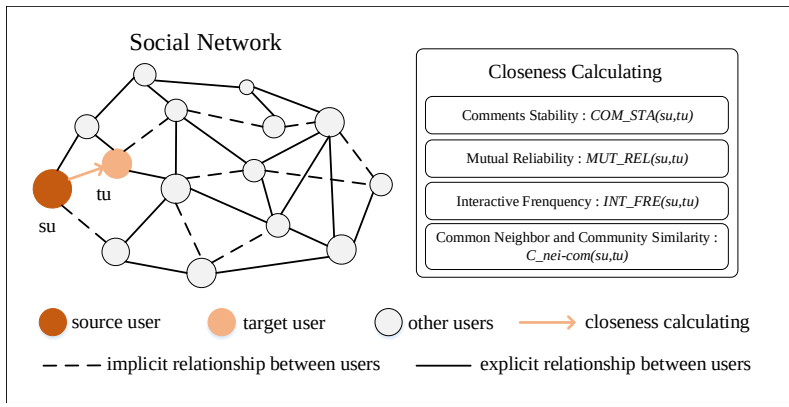


Figure 2. Four dimensions when calculating the user relationship strength

4.1.1 Comment Stability

The comment stability reflects the comment fluctuations from the source user to the target user. Many studies have found that higher comment stability indicates the more similar comments of the source user to the target user, and vice versa [31]. Therefore, the stability of the comments can reflect the tightness between users. The comments set between the user su and tu is represented by COM(su, tu), the total number of comments set is represented by $|\text{COM}(su, tu)|$, and the average number of comments is represented by $\overline{\text{com}}(su, tu)$. Therefore, the comments stability between the source user su and the target user tu can be calculated by the following equation:

$$\text{COM\_STA}(su, tu) = 1 - \sqrt{\frac{\sum_{i=1}^{|\text{COM}(su,tu)|} \left[\text{com}(tu, su)_i - \overline{\text{com}}(su, tu)\right]^2}{|\text{COM}(su, tu)|}}. \quad (1)$$

For example, a user a sends comments to a user b, and the comments set is $\text{COM}(a, b) = \{1, 1, 2, 3, 3\}$, the average number of comments is represented by $\overline{\text{com}}(a, b) = 2$. Assume that the user b gives comments to the user a, and the comments set is $\text{COM}(b, a) = \{2, 1, 3, 2, 2\}$, then the comments stability value between

the source user a and the target user b can be calculated as follows,

$$\begin{aligned}\text{COM\_STA}(a,b) &= 1-\sqrt{\frac{(2-2)^2+(1-2)^2+(3-2)^2+(2-2)^2+(2-2)^2}{5}}\\ &\approx 0.3675.\end{aligned}$$

4.1.2 Mutual Reliability Degree

The mutual reliability degree represents the reliability between two users, and it is reflected by three aspects: reliabilities of the comment, forwarding and approving. In this paper, given two users that are connected directly, the reliability degree considering all three factors can be denoted by

$$\begin{aligned}R\_C(u_1,u_2) = \{&v_1 = \text{reliability\_com}(u_1,u_2), v_2 = \text{relibility\_for}(u_1,u_2),\\ &v_3 = \text{reliability\_apr}(u_1,u_2)\}\end{aligned} \tag{2}$$

where v_1 represents the reliability of the comments, v_2 represents the forwarding reliability and v_3 represents the approving reliability. Then the mutual reliability (denoted by MUT_REL(su,tu)) can be calculated by Equation (3).

$$\text{MUT\_REL}(su,tu) = \frac{\sum_{i=1}^{3} R\_C(su,tu)\cdot v_i \times R\_C(tu,su)\cdot v_i}{\sqrt{\sum_{i=1}^{3}\left(R\_C(su,tu)\cdot v_i\right)^2}\times\sqrt{\sum_{i=1}^{3}\left(R\_C(tu,su)\cdot v_i\right)^2}}. \tag{3}$$

For instance, there are user a and user b, and the reliability degree $R\_C(a,b)$ from a to b is 0.7, reliability degree $R\_C(b,a)$ from b to a is 0.6. Assume that v_1, v_2 and v_3 of $R\_C(a,b)$ are 0.6, 0.7, 0.5, and the v_1, v_2 and v_3 of $R\_C(b,a)$ are 0.5, 0.6, 0.3, respectively. Then, the mutual reliability can be calculated as follows:

$$\begin{aligned}\text{MUT\_REL}(a,b) &= \frac{(0.7\times0.6\times0.6\times0.5)+(0.7\times0.7\times0.6\times0.6)+(0.7\times0.5\times0.6\times0.3)}{\sqrt{(0.7\times0.6)^2+(0.7\times0.7)^2+(0.7\times0.5)^2}\times\sqrt{(0.6\times0.5)^2+(0.6\times0.6)^2+(0.6\times0.3)^2}}\\ &\approx 0.6750.\end{aligned}$$

4.1.3 Interactive Frequency

Interaction frequency can signify the relationship between users in social networks. To retrieve the degree of interactive frequency, three indicators will be used, including the number of interactions in unit time, the average time length of interaction, and the average interaction time interval.

For the source user su, assume that the maximum number of interactions between this user and the user's friends is MAX$[\text{num}(su)]_{uk}$ in unit time u_k. Among all these interactions, the longest time length of a continuous interaction is denoted by MAX$[\text{len}(su)]_{uk}$, and the shortest time interval of these interactions is denoted by

$\text{MIN}[\text{int}(su)]_{uk}$. The numbers of the past interactions from the source user to the target user is denoted by $\text{NUM}(su, tu)_{uk}$ in unit time u_k. The average time length of the past interactions from the source user to the target user is denoted by $\text{LEN}(su, tu)_{uk}$, and the average time interval of the past interactions is denoted by $\text{INT}(su, tu)_{uk}$. Then, in this study, the interaction frequency denoted by $\text{INT\_FRE}(su, tu)$ can be calculated by Equation (4):

$$\text{INT\_FRE}(su, tu) = \frac{\sum_{k=1}^{|\text{Unit}|}(l_k)}{|\text{Unit}|} \tag{4}$$

where $|\text{Unit}|$ is the number of time units, and l_k is the value of interaction frequency factors in unit time u_k, which can be calculated by Equation (5):

$$l_k = \frac{1}{3}\left[\frac{\text{NUM}(su, tu)_{uk}}{\text{MAX}[\text{NUM}(su)]_{uk}} + \frac{\text{LEN}(su, tu)_{uk}}{\text{MAX}[LEN(su)]_{uk}} + \frac{\text{MIN}[\text{INT}(su)_{uk}]}{\text{INT}(su, tu)_{uk}}\right]. \tag{5}$$

Assume that there are user a and user b, and let the maximum number of interaction, the longest time length of continuous interaction, and the shortest time interval of the interaction be 8 minutes, 25 minutes, and 5 minutes in this example, respectively. If the number of the past interaction from a to b is 5, and let the average time length and the average time interval of the past interactions be 20 minutes and 15 minutes, respectively, then the l_k is $\frac{1}{3}(\frac{5}{7} + \frac{20}{25} + \frac{15}{5})$. Next, we assume the total number of user interaction time units $|\text{Unit}|$ is 5, and the values of interaction frequency factors l_k in these unit times are 1.3, 1.1, 0.6, 0.7, and 0.5, respectively. Then the interaction frequency can be calculated by

$$\text{INT\_FRE}(a, b) = \frac{1.3 + 1.1 + 0.6 + 0.7 + 0.5}{5} = 0.84.$$

4.1.4 Common Neighbor and Similar Community

Since community can promote interactions between users, common neighbors and similar communities are used to evaluate the link strength between users in this study. First, considering similar community, customer intimacy is decreasing as the growth of the community scale, which means the smaller size community will bring more contributions than the larger community. Second, the number of common neighbors also reflects the link intensity between two users. That is to say, two users who have more common neighbors will generate a stronger relationship between them.

The source user community is denoted by C_{su} and the target user community is denoted by C_{tu}. Then, the intersection between the source user community and the target user community is denoted by $\text{Same}C_i \in C_{su} \cap C_{tu}$, in which the source user and the target user have their neighbors set expressed by $N_{\text{Same}C_i}(su)$ and $N_{\text{Same}C_i}(tu)$, respectively. Here, we use $|\text{Same}C_i|$ and $|\text{Same}C_i(su)|$ to express the number of members in the community $SameC_i$, and the number of neighbors in

the set Same$C_i(su)$, respectively. Therefore, the common neighbors and similar community expressed as C_nei-com(su, tu) can be calculated by Equation (6) as follows:

$$\text{C\_nei-com}(su,tu) = \frac{\sum_{N_{\text{Same}}C_i \in C_{su} \cap C_{tu}} \left[\left(\frac{1}{\log_2 |\text{Same}C_i|} \right) \times \left| \frac{N_{\text{Same}C_i}(su) \cap N_{\text{Same}C_i}(tu)}{N_{\text{Same}C_i}(su) \cup N_{\text{Same}C_i}(tu)} \right| \right]}{\sum_{N_{\text{Same}}C_i \in C_{su} \cap C_{tu}} \left(\frac{1}{\log_2 |\text{Same}C_i|} \right)}. \tag{6}$$

For example, to make it easier to understand, we assume that the number of intersection communities between the source user and the target user is 2, i.e. $|\text{Same}C_i| = 2$. Meanwhile, we suppose that the number of the intersection between the source user's neighbors set and the target user's neighbors set in these two common communities are 12 and 16, and the number of the union between those neighbors set are 20 and 20, respectively. Then, the common neighbors and similar community can be calculated as follows:

$$\text{C\_nei-com}(su,tu) = \frac{\left[\left(\frac{1}{\log_2 2} \right) \times \left| \frac{12}{20} \right| \right] + \left[\left(\frac{1}{\log_2 2} \right) \times \left| \frac{16}{20} \right| \right]}{\frac{1}{\log_2 2} + \frac{1}{\log_2 2}} = 0.7.$$

At last, according to the community stability COM_STA(su, tu), the mutual reliability degree MUT_REL(su, tu), the interactive frequency INT_FRE(su, tu), and the common neighbors and similar community C_nei-com(su, tu), we can calculate the link strength between users, denoted by closeness(su, tu) in Equation (7):

$$\begin{aligned} \text{closeness}(su,tu) = \frac{1}{4}[&\text{COM\_STA}(su,tu) + \text{MUT\_REL}(su,tu) \\ &+ \text{INT\_FRE}(su,tu) + \text{C\_nei-com}(su,tu)]. \end{aligned} \tag{7}$$

4.2 Entity Similarity

Entity similarity is the similarity degree between two entities on the same attribute, which is an important consideration when designing a complete entity recommendation system. Entities contain various attributes, such as price, category, quality, discount, size, color, etc. The attributes of an entity can uniquely identify the corresponding entity. Normally, four attributes of an entity are considered to calculate the entity similarity, including entity category, price, quality, and discount [32]. In this section, we will introduce how to calculate the similarity degree of two entities, denoted by sim(item$_k$, item$_j$), where item$_k$ represents entity k and item$_j$ represents entity j. Both the entity k and entity j are from the same entity library, shown in Figure 3. The calculation of the similarity degree between the entity item$_k$ and the entity item$_j$ is based on the entity category, price, quality, and discount. The calculation equation is in below:

$$\text{sim}(\text{item}_k, \text{item}_j) = \frac{1}{4} \times (\text{sim\_type} + \text{sim\_price} + \text{sim\_quality} + \text{sim\_sale}) \tag{8}$$

where sim_type, sim_price, sim_quality and sim_sale represent the type similarity, the price similarity, the quality similarity and the sale similarity, respectively. If the similarity degree between the entity item_k and the entity item_j is larger than or equal to a threshold α, and the value of threshold α is mainly based on the experiment in Section 5 to obtain the optimal solution, then the entities should be put into the entity candidate set R_item. In the following sub-sections, we will introduce how to calculate sim_type, sim_price, sim_quality and sim_sale, respectively.

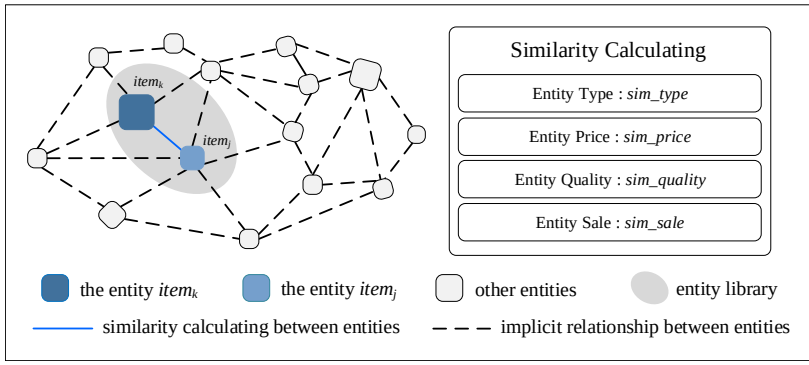


Figure 3. Four dimensions when calculating the entity similarity

4.2.1 The Calculation of Entity Type Similarity Degree

Normally, customers will search for products based on the category name, which makes the entity category become the primary consideration in the entity recommendation system. Therefore, in this paper, we propose a method based on tree structure to calculate the similarity degree of the entity type.

The tree structure with n nodes is a hierarchical data structure that is defined with branch relation. In any non-empty tree structure, only one specific node can become the root node. And if there is more than one node in the tree ($n > 1$), then the rest of nodes can be divided into m ($m > 0$) mutually disjoint finite sets, T_1, T_2, ..., T_m, and every set itself is a tree structure called subtree. For example, in Figure 4 a), there is a tree with only one node; in Figure 4 b), there is a tree with 13 nodes, and among which, A is the root node, the rest are divided into three mutually disjoint subtrees $T_1 = \{B, E, F, K, L\}$, $T_2 = \{C, G\}$ and $T_3 = \{D, H, I, J, M\}$. For the subtree T_1, the root node is B, and the rest four nodes are divided into two mutually disjoint subtrees again, which are $T_{11} = \{E, K, L\}$ and $T_{12} = \{F\}$.

A subtree is a child node of its root node, the root node is called the child node's parent. The child nodes with the same parent are brother nodes, such as node K and node L are brothers in Figure 4 b). The ancestors of a node are the nodes that traverse from the root to itself. For example, in Figure 4 b), the nodes A and C are ancestors of the node G. The level in the tree structure means that the root node

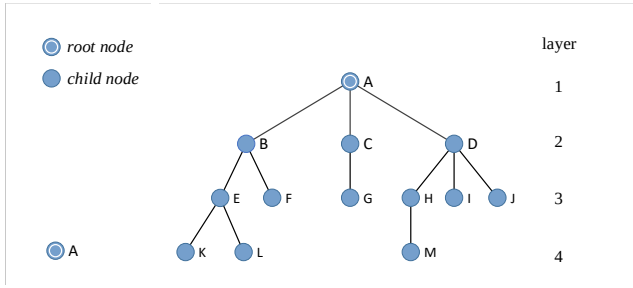


Figure 4. The tree structure: a) tree without child node, b) tree with multiple child nodes

is in the first layer, the children of the root node are in the second layer. And if one node is on the layer l, the subtrees of this node are on the layer $l+1$. If the parent nodes of two particular nodes are on the same layer, then these two nodes are cousins. For example, node E and node G are cousins, which is because their parent nodes B and C are on the same layer.

In this paper, we build a tree structure to calculate the type similarity degree of two entities. In this tree structure, entities within the same category will be in the same subtree. The concept of the layer is introduced to distinguish which layer the entity belongs, c_i means the category on the layer i, where $i \leq 4$, such that c_1 represents the first entity category. The $distanace(\text{item}_x, \text{item}_y)$ means the category distance between the category of entity item_x and entity item_y. For example, in Figure 5, the distance between the entity item_a and entity item_b is 2.

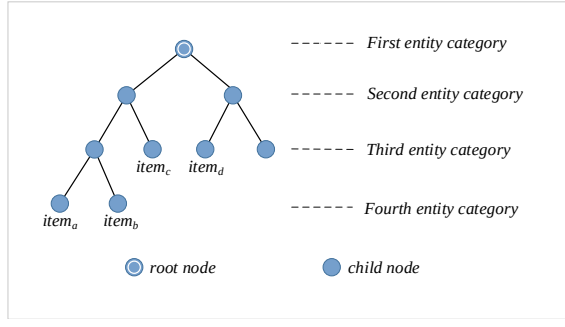


Figure 5. An example of the entity category

When calculating the similarity degree of the entity category (denoted by sim_type), it is required to consider if the categories of these two entities are on the same layer. If the categories of two entities belong to the same layer, then the

similarity degree calculation equation of the entity category is shown below.

$$\text{sim\_type} = \sqrt[c_i]{\frac{1}{1+\text{distance}(\text{item}_x, \text{item}_y)}} \tag{9}$$

where c_i represents the layer of the category, $\text{distance}(\text{item}_x, \text{item}_y)$ means the category distance between the category of entity item_x and entity item_y. For example, the distance between item_c and item_d is 4, c_i is 2 in Figure 5, so the sim_{type} is $\sqrt[2]{\frac{1}{5}}$.

If the categories of the two entities are not in the same layer, then the similarity degree calculation equation of the entity category is shown below.

$$\text{sim\_type} = \sqrt[\frac{c_x+c_y}{2}]{\frac{1}{1+\text{distance}(\text{item}_x, \text{item}_y)}}. \tag{10}$$

In Figure 5, the distance between item_b and item_c is 3, c_b is 3, c_c is 2, so the sim_type is $\sqrt[\frac{5}{2}]{\frac{1}{4}}$. And the value range of sim_type is from 0 to 1.

4.2.2 The Calculation of Entity Price Similarity Degree

Consumers will take price as an important consideration when purchasing products. Therefore, the price similarity between two entities will be one important factor of entity similarity. To calculate the price similarity, the entity price will be converted into the elasticity coefficient, then calculate the entity similarity according to the price range. For example, $39 = 0.39 \times 10^2$, if the price range is in $0.39 \times 10^2 \pm 0.39 \times 10^1$, it is the effective candidate entity; otherwise, it is not the effective candidate entity, assuming that the effective candidate entities are denoted by $R\_P\{\text{item}_1, \text{item}_2, \text{item}_3, \ldots\}$. The calculation of sim_price is shown in Equation (11).

$$\text{sim\_price} = 1 - \frac{|\text{price}_a - \text{price}_b|}{\text{price}_a} \tag{11}$$

where price_a is the price of the selected entity item_a, and price_b is the price of the candidate entity item_b, which is from the entity candidate set. The value range of sim_price is from 0 to 1 as well.

4.2.3 The Calculation of Entity Quality Similarity Degree

Entity quality is another important consideration to make entities recommendation. The entity quality similarity, denoted by sim_quality is calculated based on user evaluation according to our common sense. The comment score in total is denoted by max, and the comment score in average is denoted by ave. If the comment score of an entity is equal or greater than ave/max, then put this entity into the recommended candidate set. The equation to calculate the sim_quality is below.

$$\text{sim\_quality} = \frac{\text{score}_k}{\text{max}} \geq \frac{\text{ave}}{\text{max}} \tag{12}$$

where score_k is the comment score of the selected entity. The value range of sim_quality is from 0.8 to 1 according to the value of ave/max calculated based on the available data.

4.2.4 The Calculation of Entity Discount Similarity Degree

Since consumers will consider if an entity is on sale when making a purchase, the discount similarity between two entities will also be introduced as a factor in entities recommendation system. When calculating the discount similarity between two entities denoted by sim_sale, the discount degree of the target entity is written by sale, if the discount degree of the selected entity is greater than sale and sale is greater than zero, then put this particular entity into the recommended candidate set. The calculation equation of sim_sale is shown as follows:

$$\text{sim\_sale} = \frac{\text{sale\_k} - \text{sale}}{\text{sale\_k}} \tag{13}$$

where sale_k is the discount degree of the selected entity and the value range of sim_sale is from 0 to 1 as well.

4.3 The User Interest Degree

The user interest degree reflects how a user is interested in an entity, and it can be quantified by a value to predict the purchase probability of an entity [33]. The user interest can be divided into explicit interest and implicit interest. The explicit interest can be expressed directly by users' behaviors, such as commenting, browsing frequency, forwarding, and approving [34]. The implicit interest degree is mainly extracted and analyzed by user's relationship because the link relations between users can show possible implicit interest, which was introduced in Section 4.1. It is obvious that the explicit interest can be evaluated through the users' behavior directly, while the implicit interest needs to be extracted by the user relationship. Therefore, only the explicit interest is considered in this section.

In this study, the direct behaviors made by users are used as interest evidence to calculate the explicit interest degree, including forwarding, approving, following and comments. Then, to calculate the user interest degree by interest evidence, we need to consider two aspects. First, the explicit interest is measured by the level and weight of the interest evidence, and each interest evidence will have a different effect on the user interest degree. Second, the explicit interest is affected by the influence degree of the entities, which means users will be more interested in the entity that has a bigger influence.

Taking these two considerations into account, the calculation of the explicit interest degree of entities is introduced below. Assume that the interest evidence from the previous behaviors of the user v is denoted by $Die_j \in \{Die_1, Die_2, \ldots, Die_m\}$, and $P(Die_j)$ denotes the frequency of interest evidence Die_j. If the user does n interest evidences to an entity, then the interest evidence set can be expressed by

$IE = \{\varsigma_1, \varsigma_2, \ldots, \varsigma_i, \ldots\}$; and $P(\text{item}|Die_j)$ indicates the probability of Die_j in the interest evidence of an entity item. The weight of each interest evidence Die_j is denoted by $\text{right}(Die_j)$ ($\text{right}(Die_j) \in [0, 1]$). Then, the explicit interest value about the user node v to an entity is calculated by:

$$I(v|\text{item}) = \begin{cases} 0, & n = 0, \\ \left[\frac{1}{n} \times \sum_{i=1}^{n} \left(right_{\varsigma_i \in Die_j}(Die_j) \times P_v(Die_j|\text{item})\right)\right] \times \lambda^{impact}, & n \geq 1, \end{cases} \tag{14}$$

where λ^{impact} is introduced to represent the influence of explicit interest except implicit interests, and $P_v(Die_j|\text{item})$ denotes the occurrence probability of the interest evidence Die_j for the entity item, and it is calculated as follows:

$$P_v(Die_j|\text{item}) = \frac{p(Die_j) \times P_v(\text{item}|Die_j)}{\sum_{j=1}^{m} (P(Die_j) \times P_v(\text{item}|Die_j))}. \tag{15}$$

For instance, there are two evidences, i.e. $m = 2$, and the values of $P_v(Die_1)$, $P_v(Die_2)$, $P_v(\text{item}|Die_1)$ and $P_v(\text{item}|Die_2)$ are 0.3, 0.2, 0.6 and 0.7, respectively. Then, the occurrence probability of the interest evidence Die_1 for the entity item, i.e. $P_v(Die_1|\text{item})$ can be calculated by,

$$\begin{aligned} P_v(Die_1|\text{item}) &= \frac{p(Die_1) \times P_v(\text{item}|Die_1)}{P(Die_1) \times_v (\text{item}|Die_1) + P(Die_2) \times P_v(\text{item}|Die_2)} \\ &= \frac{0.3 \times 0.6}{0.3 \times 0.6 + 0.2 \times 0.7} = 0.5625. \end{aligned}$$

Next, based on Equation (14), the explicit interest value $I(v|\text{item})$ can be calculated according to the above calculation results. For example, firstly, when the condition $n = 0$ is true, the value of $I(v|\text{item})$ is zero. When the condition is true, to make the calculation easier, we assume that the influence of explicit interest λ^{impact} is 0.8, the value of $right_{\varsigma_i \in Die_1}(Die_1)$ is 0.6 and $n = 1$. Then the explicit interest value about the user node v to an entity item is calculated as follows:

$$\begin{aligned} I(v|\text{item}) &= \left[\frac{1}{1} \times (\text{right}_{\varsigma_i \in Die_1}(Die_1) \times P_v(Die_1|\text{item}))\right] \times \lambda^{\text{impact}} \\ &= \left[\frac{1}{1} \times (0.6 \times 0.5625)\right] \times 0.8 = 0.27. \end{aligned}$$

To calculate the concrete weight of $\text{right}(Die_j)$, the inherent relationships between the interest evidence need to be considered, that is because some interest evidence may appear continuously and simultaneously. For example, the interest evidence of long time browsing is likely to happen simultaneously with the interest evidence "approving" or "add to favorite list", and they may influence each other

and strengthen the contact. From this point of view, the basic principle of the weight calculation of interest evidence is similar to PageRank [35]. It means the more important interest evidence is likely to be associated with the other important interest evidence. In this study, the interest evidence y caused by another interest evidence x is denoted as: $x \to y$ for convenience. Therefore, the set that is associated with interest evidence Die_i can be expressed as:

$$L(Die_i) = \{Die_j | \exists (Die_j \to Die_i) \wedge (i \neq j)\}. \tag{16}$$

Then, the weight of the interest evidence is calculated as follows:

$$\text{right}(Die_i) = \frac{1 - p(Die_i)}{|ID|} + p(Die_i) \times \sum_{Die_j \in \text{Link}(Die_i)} \frac{\text{right}(Die_j)}{L(Die_j)} \tag{17}$$

where $p(Die_i)$ means the probability of interest evidence Die_j among the previous behaviors of the user, and $L(Die_j)$ means the number of interest evidence that link with interest evidence Die_j.

Since the user interest will change by time, we propose a dynamic prediction method based on the aging algorithm to describe the change of interest. Assume the explicit interest about the user v to the entity at timestamp t_{n-1} is $I(v|\text{item})_{n-1}$, the explicit interest about the user v to the entity at the next timestamp t_n is $I(v|\text{item})_n$. Then the predicted explicit interest at timestamp t_n is based on $I(v|\text{item})_{n-1}$ and $I(v|\text{item})_n$. The calculation method is in Equation (18).

$$I(v|\text{item})_n = \xi \times I(v|\text{item})_{n-1} + (1 - \xi) \times \text{Change\_I}_v(\text{item})_n. \tag{18}$$

At last, if the explicit interest degree of a user to an entity is greater than or equal to a given threshold β, where the value of β is based on the experiment in Section 5 to obtain the optimal solution, then the user will be interested in this particular entity.

4.4 Multi-Dimensional Comprehensive Recommendation Algorithm Based on Social Network

In this study, a trust-based multi-dimensional comprehensive recommendation algorithm on social network is proposed, which mainly contains four algorithm modules, including the user candidate set algorithm, the entity candidate set algorithm, the user interest degree algorithm, and the comprehensive module recommendation algorithm. The four algorithms will be introduced in the following sub-sections, respectively.

4.4.1 User Candidate Set Algorithm

The model of the proposed recommendation method is based on trust between users on social network, and the assessment of trust is mainly based on the strength of

Algorithm 1 Get user candidate set R_user

Input: Current user, cur_{user}; The friend set of current user, friend(cur_{user}); The number of users in the social network, n

Output: R_user

initial R_user = $\emptyset$;
for int $i_1 = 1$; $i_1 <$ length(friend(cur_{user})), $\text{friend}_{i_1} \in$ friend(cur_{user}); i_1++ **do**
  if closeness($\text{friend}_{i_1}, cur_{user}$) $\geq \gamma$ **then**
    $\text{friend}_{i_1} \rightarrow$ R_user; // Put friend_{i_1} into the user candidate set R_user
  else
    $\text{friend}_{i_1} \times$ R_user; // Do not put friend_{i_1} into the user candidate set R_user
  end if
end for

the user relationship. If a current user has a higher relationship strength with his (her) friend user, that means there is a greater similarity between them. Then, the purchased entities of the current user will be recommended to this particular friend user [36]. The ultimate goal of the user candidate set algorithm is to retrieve the user candidate set R_user for each current user cur_{user}. This method can join the user candidate set with the friends who have a strong relationship with the current user. This process will inevitably include some new users, who may also have the recommended entity set corresponding to them, which can solve the cold start problem to some extent. The method in detail is introduced in Algorithm 1. The time complexity of this algorithm is $O(n)$, where n represents the number of users in the social network.

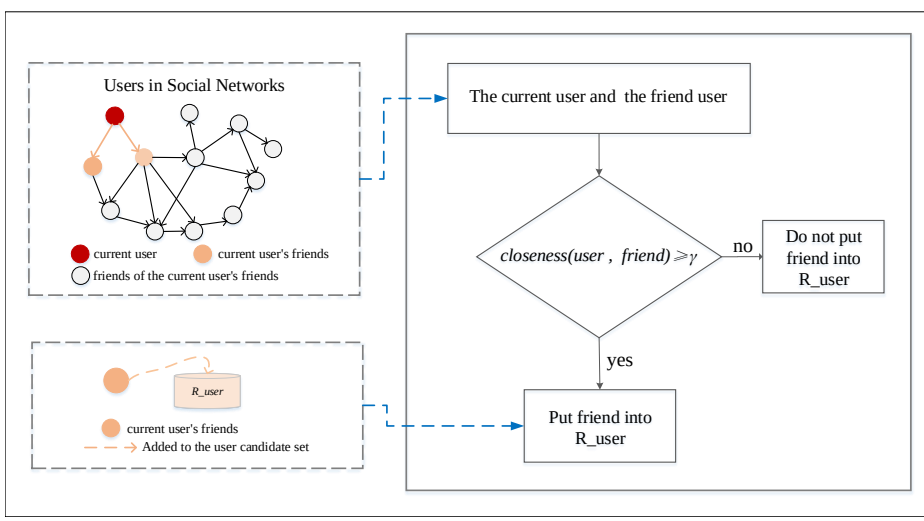


Figure 6. The flow of user candidate set algorithm

Figure 6 shows the general flow of Algorithm 1. On the left side of Figure 6, the red dot in social network circle represents the current user, and the orange dots that are directly connected with the current user represent the current user's friends, and the remaining gray dots represent the friends of the current user's friends. If the user relationship strength between the current user and the friend user is greater than or equal to the threshold γ, where γ is set by the experiment in Section 5 as well, then the friend user will be added to the user candidate set R_user; otherwise, he (she) will not be added.

4.4.2 Entity Candidate Set Algorithm

Since the ultimate goal of the recommendation system is to recommend the corresponding entities to users, and the e-commerce websites contain a large number of entities, the entities that are similar to the purchased entity need to be chosen as the recommendation entities. In another word, the current user has purchased an entity A, then the entity B that is similar to the entity A will be recommended to the friend users who have similar preferences with the current user. Therefore, the entity candidate set R_item that contains similar entities with a particular entity needs to be obtained. The method in detail is shown in Algorithm 2. The time complexity of this algorithm is $(m \log m)$, where m represents the number of entities in the initial entity candidate set.

Algorithm 2 Get entity candidate set R_item

Input: Current user purchased an entity, item_j; The similar entity set of item_j, R_item_0; The number of entities in R_item_0, m

Output: R_item

initial R_user $= \emptyset$,

$\text{friend}_{i_1} \in \text{friend}(cur_{\text{user}}) \cap \text{closeness}(\text{friend}_{i_1}, cur_{\text{user}}) \geq \gamma \cap \text{friend}_{i_1}$ did not purchase item_j;

for int $j_1 = 1$; $j_1 < \text{length}(\text{R\_item}_0)$, $\text{item}_{j_1} \in \text{R\_item}_0$; j_1++ **do**

  if $\text{sim}(\text{item}_{j_1}, \text{item}_j) \geq \alpha$ **then**

    $\text{item}_{j_1} \rightarrow$ R_item; // Put item_{j_1} into the entity candidate set R_item

  else

    $\text{item}_{j_1} \times$ R_item; // Do not put item_{j_1} into the user candidate set R_item

  end if

end for

Figure 7 shows the general flow of Algorithm 2. In this figure, if the current user has purchased an entity item_j, then each entity from the entity library will be selected to calculate the similarity with the entity item_j, if the similarity is bigger than or equal to the threshold α, then this entity item_{j_1} needs to be added to the entity candidate set R_item.

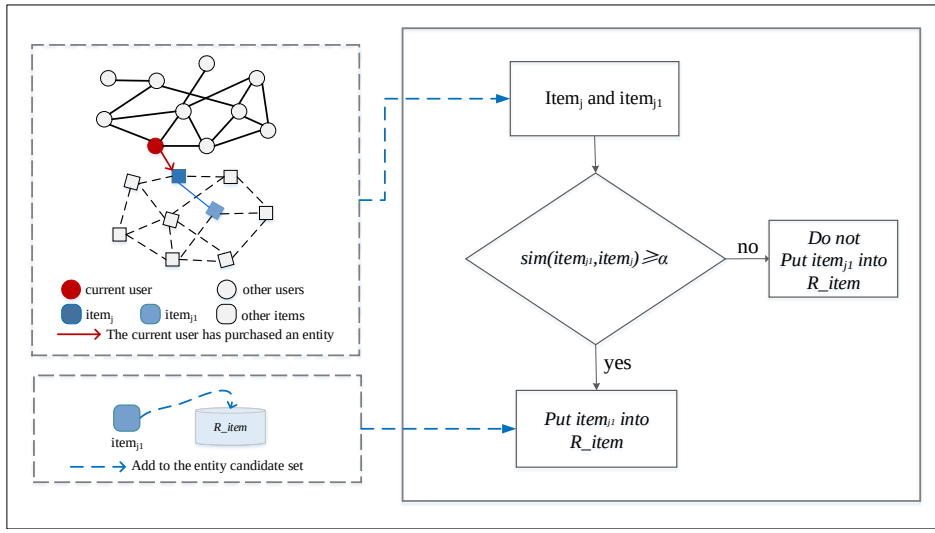


Figure 7. The flow of entity candidate set algorithm

4.4.3 User Interest Algorithm

The recommendation system concerns about if the user is really interested in the recommended entity, which can be measured by if a user purchased the recommended entity ultimately. In this study, we propose a user interest degree calculation method to decide whether to recommend an entity to the current user or not. The method is shown in Algorithm 3. The user interest $I(\text{friend}_{i_2}|\text{item}_{j_2}, \text{item}_{j_2} \in \text{R\_item})$ represents the interested degree of the user in the user candidate set R_user to the entity in the entity candidate set R_item. If the user interest degree is greater than or equal to the threshold β, then the entity item_{j_2} will be recommended to the current user friend_{i_2}. The time complexity of this algorithm is $(n * m \log m)$, where n represents the number of users in the user candidate set and m represents the number of entities in the entity candidate set R_item_0 of the current entity.

Figure 8 shows the general flow of Algorithm 3. In this figure, first, each friend user friend_{i_2} is selected from the user candidate set R_user of the current user, and each entity is selected from the entity candidate set R_item that contains the entities purchased by the current user. Then, calculate the user interest degree of the user friend_{i_2} to the entity item_{j_2}, denoted by $I(\text{friend}_{i_2}|\text{item}_{j_2})$, if the user interest degree is greater than or equal to the threshold β, then the entity item_{j_2} will be recommended to the friend user friend_{i_2}.

Algorithm 3 Calculate user interest

for int $i_2 = 1$; $i_2 <$ length(R_item), friend$_{i_2} \in$ R_user; i_2++ **do**
  for int $j_2 = 1$; $j_2 <$ length(R_item), item$_{j_2} \in$ R_item; j_2++ **do**
    if I(friend$_{i_2}$|item$_{j_2}$) $\geq \beta$ **then**
      item$_{j_2} \mapsto$ friend$_{i_2}$; // Recommend entity item$_{j_2}$ to user friend$_{i_2}$
    else
      item$_{j_2} \times$friend$_{i_2}$; // Do not put item$_{j_2}$ into the user candidate set friend$_{i_2}$
    end if
  end for
end for

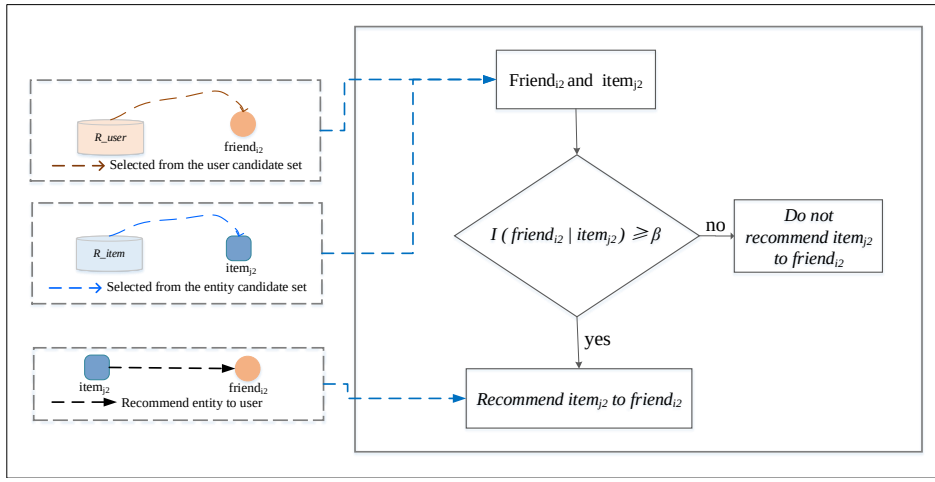


Figure 8. The flow of the user interest algorithm

4.4.4 The Comprehensive Module of the Recommendation Algorithm

The previous three sub-sections describe the user candidate set algorithm, the entity candidate set algorithm and the user interest degree algorithm, respectively. These algorithms are mainly used for estimating the corresponding recommendation factors. Based on these factors, we will introduce a comprehensive module recommendation algorithm in this section. The main purpose of the algorithm is to retrieve the appropriate entity in the entity candidate set R_item, and recommend it to the user in the user candidate set R_user. The method in detail is shown in Algorithm 4.

In this algorithm, the user relationship in social network is considered as a tree structure. The corresponding nodes in the tree structure can be considered as user nodes. First, based on the tree structure, calculate the user relationship strength between the user and the user's friend based on Algorithm 1. Second, calculate

the entity similarity between the purchased entity and other entities in the entity candidate set based on Algorithm 2. Then, calculate the user interest degree about the user in the user candidate set for the entity in the entity candidate set based on Algorithm 3. At last, recommend the entity in the entity candidate set that satisfies the conditions to the user in the corresponding user candidate set. The time complexity of this algorithm is $O(n * m \log m)$, where n represents the number of users in the user candidate set and m represents the number of entities in the entity candidate set R_item_0 of the current entity.

Algorithm 4 in detail is described in the following, which is the core of the multi-dimensional comprehensive recommendation method under the social network environment.

If a user a purchased an entity item_j, then retrieve the friend set $\text{friend}(a)$ of the user a. If a user b_i belongs to $\text{friend}(a)$ and the closeness between user b_i and user a is bigger than or equal to the threshold γ, at the same time, the user b_i has not purchased the entity item_j, then the following steps will be performed.

Step 1: Retrieve the candidate recommendation set R_item of the entity item_j (noting that R_item has already contained item_j) by calculating $\text{sim}(\text{item}_k, \text{item}_j)$. If the $\text{sim}(\text{item}_k, \text{item}_j)$ is bigger than or equal to the threshold α, then put the entity item_k into the set R_item.

Step 2: For each user b_i in $\text{friend}(a)$, calculate the user strength $\text{closeness}(b_i, a)$ between the user b_i and a, if $\text{closeness}(b_i, a)$ is greater than or equal to γ, then put the user b_i into the set R_user. After that, calculate the user interest degree $I(b_i|\text{item}_k \in \text{R\_item})$, if $I(b_i|\text{item}_k \in \text{R\_item})$ is greater than or equal to the threshold β, then recommend the entity item_k to the user b_i.

Step 3: The nodes of all the recommended entities are marked as C_p, regarding C_p as a. Then the recommended entity in the recommendation candidate set R_item from Step 1 is continued to be recommended to the user a's friends, then repeat Step 2. Finally, the recommendation algorithm will end until all the users have been traversed in the social network.

The recommendation algorithm process is shown in Figure 9. The left side of this figure shows the calculation process of the user candidate set and the entity candidate set in detail. The right side shows the user interest degree algorithm.

5 EXPERIMENTS

5.1 Experimental Dataset

To evaluate the performance of the multi-dimensional comprehensive recommendation algorithm based on social networks, we implemented some experiments using Douban reading dataset and Sina Weibo dataset. Douban reading is a website about reading books, which can recommend corresponding books to users. In our crawled

Algorithm 4 General recommendation process

Input: The set of all users User
Output: User interest between the user and the item

```
if User != ∅ then
    Vector ⟨int⟩ preorderTraversal (TreeNode * cur_user);
    vector ⟨int⟩ ret;
    if cur_user = NULL then
        return ret;
        stack⟨TreeNode*⟩st;
        st.push cur_user;
        while !st.empty() do
            TreeNode*tp = st.top();
            st.pop();
            ret.push_back(tp → val);
            Get user candidate set R_user;
            Get entity candidate set R_item;
            Calculate user interest;
            if tp → right != NULL then
                st.push(tp → right);
                Get user candidate set R_user;
                Get entity candidate set R_item;
                Calculate user interest;
            end if
            if tp → left != NULL then
                st.push(tp → left);
                Get user candidate set R_user;
                Get entity candidate set R_item;
                Calculate user interest;
            end if
        end while
    end if
else
    break;
return ret;
end if
```

Douban reading data, it contains 55 328 books in total, and each book contains corresponding information, including the serial number (a unique identifier corresponds to one book), book name, review score, price, category, retailer and user ID. Sina Weibo website is the largest micro-blogging site in China, which owns excellent social network features. In our crawled Sina Weibo data, it contains 63 641 user records, and each user record contains information including user ID, user nickname, user's province, user's city, user's gender, the number of fans of the user, the number of

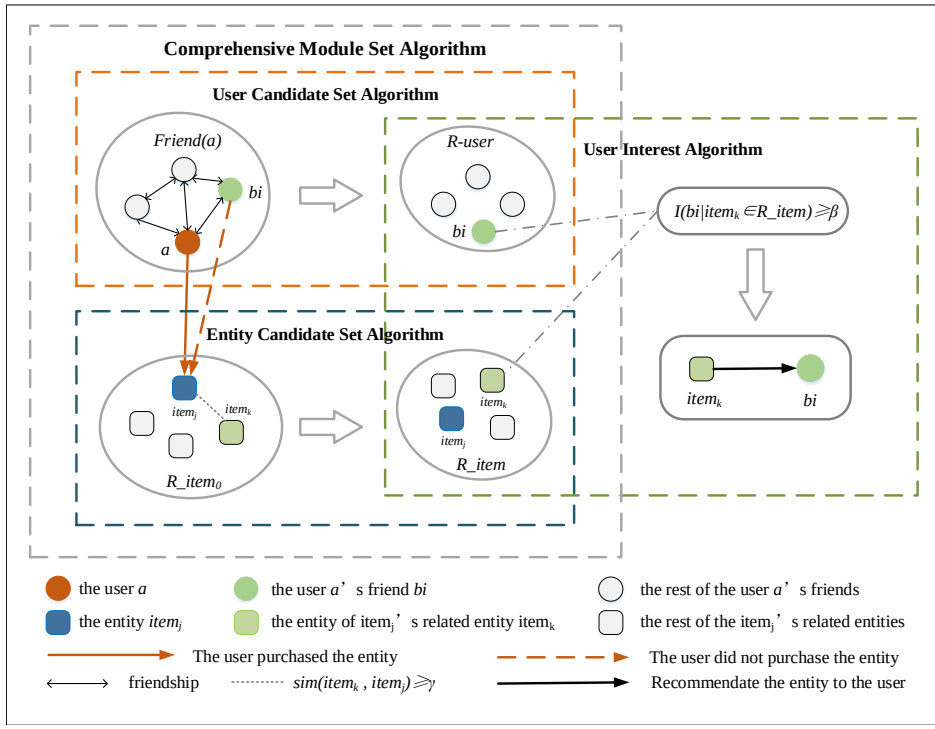


Figure 9. The overall process of the recommendation algorithm

friends of the user, the set of friends' ID of the users, the comments between friends and the number of interactions.

Since we proposed a multi-dimensional comprehensive recommendation system based on trust under the social network environment, the social network data and the e-commerce website data need to be linked. In this experiment, the fast login method, which means register an account in the commercial website by using the account in the social network, was chosen to solve this problem. Then the link between the user accounts in Sina Weibo and accounts in Douban reading can be established.

After retrieving user information and book information from Douban reading and Sina Weibo website, and establishing the user association between two datasets, we created a related table in Table 2. This table demonstrates the related data structure and description, which contains user account (userID), friend set account (friendsID), friend set number (friendsetNo), comment number (commentNo) and interaction frequency number (frequencyNo). To put it shortly, the friendsetNo represents the closeness between the current user and the current user's friends, the commentNo represents the comments between the current user and the current

user's friends, and the frequencyNo indicates the interaction frequency between the current user and the current user's friends.

Return Value Field	**Field Description**
userID	User account
friendID	Friend account
friendsetNo	Friend set number
commentNo	Comment number
frequencyNo	Interaction frequency number

Table 2. Related data structure and description

Then, the experiment is implemented by the following four steps. First, the user's behavior dataset is divided into 8 parts randomly, one part is used as the test set and the remaining seven parts are used as the training set. Second, train the user interest model using the training set, and get the weights γ_1, γ_2, γ_3 and γ_4, which are required for estimating the strength of the user relationship, get the weights α_1, α_2, α_3 and α_4, which are required for estimating the similarity of the entities, and get the recommended thresholds α, β and γ for each dimension based on the above results. Then, predict user behaviors on the test set using the thresholds retrieved from the previous steps, and define a triplet $cE = (\text{user}_a, \text{user}_b, \text{comm\_Entity})$ that represents the common set of entities purchased by the user user_a and the user user_b. At last, evaluate the prediction result on the test set by using some evaluation measurements that will be introduced in Section 5.2.

5.2 The Evaluation Metrics of the Recommendation Method

In the experiment, three evaluation metrics were used to validate the performance of the recommendation algorithm, including precision, recall, and F1-score. By comparing the recommended items with the user's true selection records, we can calculate the evaluation metrics. The equation of precision is shown as follows:

$$\text{precision} = \frac{\sum_{u \in U} |R(u) \cap B(u)|}{\sum_{u \in U} |R(u)|} \tag{19}$$

where $R(u)$ is the recommendation list of each user according to the user behaviors in the training set, $B(u)$ is the behavior list of each user in the testing set. And the recall is calculated as follows:

$$\text{recall} = \frac{\sum_{u \in U} |R(u) \cap B(u)|}{\sum_{u \in U} |B(u)|}. \tag{20}$$

The F1-score is calculated based on precision and recall, which is shown in Equation (21).

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{21}$$

5.3 Experimental Results and Analysis

In this section, we will show our experimental results and analyze the results from three aspects, which are social network analysis, weight setting and experimental results.

5.3.1 Social Network Analysis

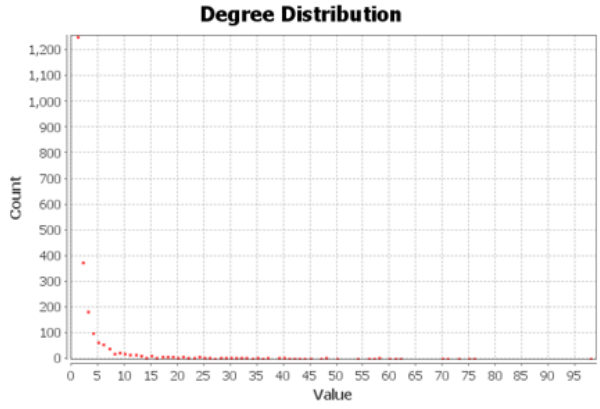
The proposed multi-dimensional comprehensive recommendation algorithm is based on user relationship strength in social network. In this section, we utilized Gephi tools [37] to draw the graph of user relationship, and then analyzed the relationship strength between users. The indegree of the user node represents the number of fans of the current user, and the outdegree of the user node represents the number of users that the current user followed.

In order to show the characteristics of the network topology, only a part of data is selected from the dataset for displaying in this experiment. Figure 10 a) shows that the approximate distribution of user degrees in Sina Weibo dataset (there were 2347 user nodes and 5001 edges). The distribution of power obeys the power law distribution, which shows that the Sina Weibo dataset is a network topology with no scale characteristics. Figure 10 b) shows the user relationship structure in social network, where each red dot represents a user. It is obvious that some users gathered to a cluster, which can indicate that these users belong to the same community. The average path length of the dataset selected in this study is 4.218, which means the network structure conforms to the features of the small-world network. In summary, the network model constructed by the Sina Weibo dataset owns the scale-free and small-world characteristics. Therefore, it is proved that the selected dataset is effective for analysis.

5.3.2 Weight Setting

In this section, first, the four weights γ_1, γ_2, γ_3 and γ_4 are trained through experiments, which will be used to calculate the strength of user relationship. Second, the four weights α_1, α_2, α_3 and α_4 are trained to calculate the entity similarity. Third, the recommendation thresholds α, β and γ are trained for each recommendation dimension. Finally, by using the thresholds and weights calculated above, we compare the performance of our proposed method with some traditional recommendation methods.

Firstly, since the source user's characteristic will affect the target user, the weights γ_1, γ_2, γ_3 and γ_4 that represent comment stability, mutual reliability, interactive frequency, and common neighbors and similar communities need to be considered comprehensively [38]. In this study, we make $\gamma_1 = \frac{1}{n}\sum_{i=1}^{n}\gamma_{1i}$, $\gamma_2 = \frac{1}{n}\sum_{i=1}^{n}\gamma_{2i}$, $\gamma_3 = \frac{1}{n}\sum_{i=1}^{n}\gamma_{3i}$, $\gamma_4 = \frac{1}{n}\sum_{i=1}^{n}\gamma_{4i}$, where n denotes the number of all relevant user pairs in the Sina Weibo dataset, γ_{1i}, γ_{2i}, γ_{3i} and γ_{4i} indicate the comment stability, mutual reliability, interactive frequency, and common neighbors and similar com-



a)

b)

Figure 10. a) The distribution of user degree, b) the structure diagram of social network

munities of the i^{th} user group, respectively. In this study, the values of γ_1, γ_2, γ_3 and γ_4 are set to 0.10, 0.40, 0.25 and 0.25, shown in Table 3, and the sum of γ_1, γ_2, γ_3 and γ_4 is 1.

Symbol	**Description**	**Value**
γ_1	the stability weights for comments between users	0.10
γ_2	the weight of mutual reliability between users	0.40
γ_3	the weight of interactive frequency between users	0.25
γ_4	the weight of common neighbors and similar community	0.25

Table 3. The corresponding weights setting of user relationship strength

Since the four weights will make different impacts on user relationship strength, the influence degree of each weight on the recommendation results needs to be examined. When the values of γ_1, γ_2, γ_3 and γ_4 are set to 0, 0.40, 0.25 and 0.25, respectively, the influence of the stability between users can be examined. When the values of γ_1, γ_2, γ_3 and γ_4 are set to 0.10, 0, 0.25 and 0.25, respectively, the influence of the mutual reliability between users can be examined. When the values of γ_1, γ_2, γ_3 and γ_4 are set to 0.10, 0.40, 0 and 0.25, respectively, the influence of the interactive frequency between users can be examined. And when the values of γ_1, γ_2, γ_3 and γ_4 are set to 0.10, 0.40, 0.25 and 0, respectively, the influence of the common neighbors and similar community can be examined.

Figure 11 shows the result of precision, recall, and F1-score by using different weights. The result of the original weights is shown in green color, which is clearly the best result. The precision, recall, and F1-score are the lowest when γ_2 is 0, γ_1 is 0.1, γ_3 is 0.25 and γ_4 is 0.25, which is shown in light brown color. The precision, recall and F1-score results are about 23.2 %, 23.3 % and 23.4 % lower than the original recommendation model. Therefore, compared with the other three factors, the mutual reliability has a greater impact on the strength of user relationship.

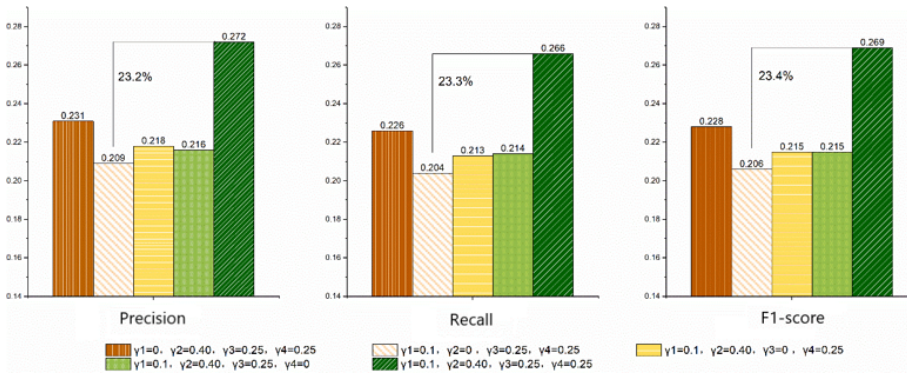


Figure 11. The results of using different γ_1, γ_2, γ_3 and γ_4

Secondly, the four weights α_1, α_2, α_3 and α_4 are required to calculate the entity similarity. Since both the category, price, comment, and sale of entities will affect the entity similarity to some extent, the values of α_1, α_2, α_3 and α_4 need to be considered comprehensively. In this study, we make $\alpha_1 = \frac{1}{m}\sum_{i=1}^{m}\alpha_{1i}$, $\alpha_2 = \frac{1}{m}\sum_{i=1}^{m}\alpha_{2i}$, $\alpha_3 = \frac{1}{m}\sum_{i=1}^{m}\alpha_{3i}$, $\alpha_4 = \frac{1}{m}\sum_{i=1}^{m}\alpha_{4i}$, where m denotes the number of all the relevant book pairs in Douban reading dataset, α_{1i}, α_{2i}, α_{3i} and α_{4i} indicate the category, price, comment, and sale of the i^{th} books group, respectively. In this study, the values of the α_1, α_2, α_3 and α_4 are set to 0.35, 0.28, 0.27 and 0.10, respectively, shown in Table 4, and the sum of α_1, α_2, α_3 and α_4 is 1.

Since the four weights will make different impacts on entity similarity, the influence degree of each weight on the recommendation results will be examined in this study as well. When the values of α_1, α_2, α_3 and α_4 are set to 0, 0.28, 0.27

Symbol	Description	Value
α_1	category weight	0.35
α_2	price weight	0.28
α_3	comment weight	0.27
α_4	sale weight	0.10

Table 4. The corresponding weights setting of entity similarity

and 0.10, respectively, the influence of the category weight can be examined. When the values of α_1, α_2, α_3 and α_4 are set to 0.35, 0, 0.27 and 0.10, respectively, the influence of the price weight can be examined. When the values of α_1, α_2, α_3 and α_4 are set to 0.35, 0.28, 0 and 0.10, respectively, the influence of the comment weight can be examined. And when the values of α_1, α_2, α_3 and α_4 are set to 0.35, 0.28, 0.27 and 0, respectively, the influence of the sale weight can be examined.

Figure 12 shows the result of precision, recall, and F1-score by using different weights of α_1, α_2, α_3 and α_4. The result shows that the original weight setting can achieve the best result, which is shown in dark green color. When α_2 is 0, α_1 is 0.35, α_3 is 0.27, and α_4 is 0.10, shown in dark brown color, the values of precision, recall, and F1-score are the lowest, which are approximately 21.7 %, 21.8 % and 21.9 % lower than using the original weight. Therefore, compared with the other three factors, the category has a greater impact on entity similarity.

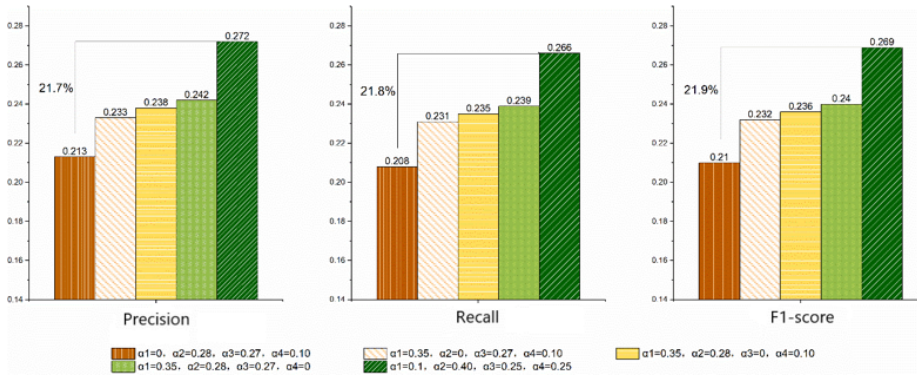


Figure 12. The results of using different α_1, α_2, α_3 and α_4

At last, the results of using different recommendation thresholds α, β and γ are shown in Table 5. The evaluation of the results is based on precision, recall and F1-score as well. When α is set to 0.3, β is set to 0.4 and γ is set to 0.3, emphasized in bold font, the proposed model can achieve the best performance.

(α, β, γ)	**Precision**	**Recall**	**F1-score**
(0.2, 0.6, 0.2)	0.246	0.239	0.242
(0.3, 0.5, 0.2)	0.249	0.245	0.247
(0.4, 0.4, 0.2)	0.253	0.249	0.251
(0.5, 0.3, 0.2)	0.258	0.256	0.257
(0.6, 0.2, 0.2)	0.264	0.262	0.263
(0.2, 0.5, 0.3)	0.267	0.262	0.264
(0.3, 0.4, 0.3)	**0.272**	**0.266**	**0.269**
(0.4, 0.3, 0.3)	0.270	0.267	0.268
(0.5, 0.2, 0.3)	0.263	0.261	0.262
(0.2, 0.4, 0.4)	0.261	0.259	0.260
(0.3, 0.3, 0.4)	0.259	0.255	0.257
(0.4, 0.2, 0.4)	0.257	0.252	0.254
(0.2, 0.3, 0.5)	0.256	0.252	0.254
(0.3, 0.2, 0.5)	0.253	0.249	0.251
(0.2, 0.2, 0.6)	0.241	0.237	0.239

Table 5. The values of Precision, Recall and F1-score for different (α, β, γ) pairs

5.3.3 Experimental Results

Finally, we compare our proposed method with some traditional recommendation methods using the same dataset. If our proposed method considers the entity similarity only, it will become the traditional content-based recommendation method; if it considers the strength of user relationship only, it will become the traditional social network-based recommendation method; if it considers the entity similarity and the user relationship strength only, it will become the knowledge-based recommendation method; and if the user relationship strength is not considered in our proposed method, it will become the traditional entity-based collaborative filtering recommendation method. In order to express conveniently, PNMCRS is used to represent our proposed method in this study, SNRS is used to represent the social network-based recommendation system method, KRS is used to represent the knowledge-based recommendation system method, CRS is used to represent the content-based recommendation system method, and CFRS is used to represent the collaborative filtering recommendation system method.

The comparison result is shown in Figure 13. It can be seen that our proposed comprehensive recommendation method has the highest accuracy, recall, and F1-score, which are 0.207, 0.218, and 0.212, respectively, following by the social network-based recommendation method, which are 0.203, 0.217 and 0.210, respectively. This is mainly because the traditional Pearson correlation coefficient (PCC) and Jaccard mean squared error (JMSD) are discarded when we calculate the user relationship strength in this study, however, the four aspects, which are comment stability, mutual reliability, interaction frequency, common neighbors and similar communities are taken into account. In general, the result proves that our proposed method can perform best when recommending entities to target users.

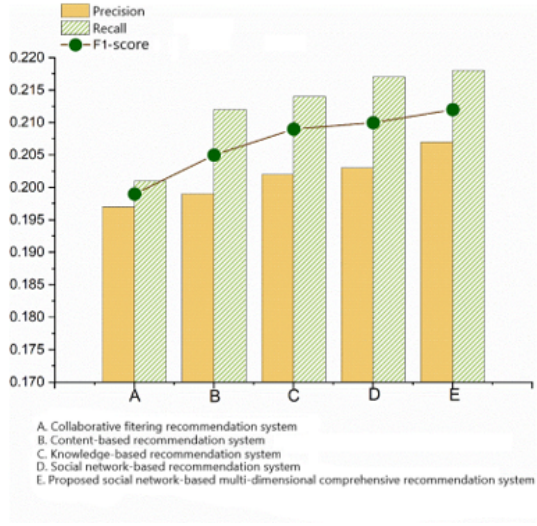


Figure 13. The three evaluation indicators of the proposed recommendation method and the traditional recommendation method

6 CONCLUSION AND FUTURE WORK

In social networks, mutual impact among users is common and inevitable. Improving recommendation performance from the perspective of user relationship strength is of great significance. In this paper, we propose a multi-dimensional recommendation algorithm from the perspective of user relationship strength in social network to improve the recommendation performance, which uses the user relationship, the similarity of entities and the degree of user interest information in three-level modeling comprehensively. In order to validate the effectiveness of our proposed model, we compared the performance of this novel model with some traditional recommendation models using the real-world dataset from Douban reading and Sina Weibo websites. The results of our experiments have demonstrated the excellent performance of our proposed model and its effectiveness on our existing dataset. The experimental results have been analyzed, which are consistent with the expected results. The proposed model can discover the interest degree of users and optimize the recommendation via multi-dimensional comprehensive recommendation factors based on user relationship strength in social network.

There are two major limitations in this study that could be addressed in future research. First, we ignore the behavior of some users who do not log in to the e-commerce platform through their social network accounts when implementing cross-platform data connection. In the future, we will retrieve more data to consider multiple login methods. Second, if the entity purchase time and information prop-

agation time in social network can be considered in the recommendation algorithm as well, the accuracy may be further improved.

Acknowledgement

This work is funded by the National Natural Science Foundation of China (No. 6180-2258, No. 61572326, No. 61702333), the Natural Science Foundation of Shanghai (No. 18ZR1428300), the Shanghai Sailing Program (No. 19YF1436900), the Shanghai Committee of Science and Technology (No. 17070502800).

REFERENCES

[1] Cao, J.—Li, W.: Sentimental Feature Based Collaborative Filtering Recommendation. 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), 2017, pp. 463–464, doi: 10.1109/BIGCOMP.2017.7881758.

[2] Li, S.—Luo, F.—Yang, J.—Ranzi, G.—Wen, J.: A Personalized Electricity Tariff Recommender System Based on Advanced Metering Infrastructure and Collaborative Filtering. International Journal of Electrical Power and Energy Systems, Vol. 113, 2019, pp. 403–410, doi: 10.1016/j.ijepes.2019.05.042.

[3] Liu, X.: A Collaborative Filtering Recommendation Algorithm Based on the Influence Sets of E-Learning Group's Behavior. Cluster Computing, Vol. 22, 2019, No. 2, pp. 2823–2833, doi: 10.1007/s10586-017-1560-6.

[4] Son, J.—Kim, S. B.: Content-Based Filtering for Recommendation Systems Using Multiattribute Networks. Expert Systems with Applications, Vol. 89, 2017, pp. 404–412, doi: 10.1016/j.eswa.2017.08.008.

[5] Shu, J.—Shen, X.—Liu, H.—Yi, B.—Zhang, Z.: A Content-Based Recommendation Algorithm for Learning Resources. Multimedia Systems, Vol. 24, 2018, No. 2, pp. 163–173, doi: 10.1007/s00530-017-0539-8.

[6] Suglia, A.—Greco, C.—Musto, C.—de Gemmis, M.—Lops, P.—Semeraro, G.: A Deep Architecture for Content-Based Recommendations Exploiting Recurrent Neural Networks. Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17), 2017, pp. 202–211, doi: 10.1145/3079628.3079684.

[7] Zhang, Y.—Saberi, M.—Chang, E.—Abbasi, A.: Solution and Reference Recommendation System Using Knowledge Fusion and Ranking. 2018 IEEE 15th International Conference on E-Business Engineering (ICEBE), 2018, pp. 31–38, doi: 10.1109/ICEBE.2018.00016.

[8] Guo, G.—Zhang, J.—Yorke-Smith, N.: Leveraging Multiviews of Trust and Similarity to Enhance Clustering-Based Recommender Systems. Knowledge-Based Systems, Vol. 74, 2015, pp. 14–27, doi: 10.1016/j.knosys.2014.10.016.

[9] Hong, Y.—Zeng, X.—Bruniaux, P.—Chen, Y.—Zhang, X.: Development of a New Knowledge-Based Fabric Recommendation System by Integrating the Collaborative Design Process and Multi-Criteria Decision Support. Textile Research Journal, Vol. 88, 2018, No. 23, pp. 2682–2698, doi: 10.1177/0040517517729383.

[10] Resnick, P.—Varian, H. R.: Recommender Systems. Communications of the ACM, Vol. 40, 1997, No. 3, pp. 56–58, doi: 10.1145/245108.245121.

[11] Lu, X.-H.—Huang, H.-H.—Wu, H.-Y.—Liu, W.-L.: A Hybrid Recommendation Model for Community Attributes of Social Networks Based on Association Rule Mining. 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE, 2018, pp. 420–424, doi: 10.1109/ICMCCE.2018.00094.

[12] Li, M.—Li, Y.—Lou, W.—Chen, L.: A Hybrid Recommendation System for Q & A Documents. Expert Systems with Applications, Vol. 144, 2020, Art. No. 113088, doi: 10.1016/j.eswa.2019.113088.

[13] Puglisi, S.—Parra-Arnau, J.—Forné, J.—Rebollo-Monedero, D.: On Content-Based Recommendation and User Privacy in Social-Tagging Systems. Computer Standards and Interfaces, Vol. 41, 2015, pp. 17–27, doi: 10.1016/j.csi.2015.01.004.

[14] Musto, C.—Semeraro, G.—de Gemmis, M.—Lops, P.: Learning Word Embeddings from Wikipedia for Content-Based Recommender Systems. In: Ferro, N. et al. (Eds.): Advances in Information Retrieval (ECIR 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9626, 2016, pp. 729–734, doi: 10.1007/978-3-319-30671-1_60.

[15] Gu, Y.—Zhao, B.—Hardtke, D.—Sun, Y.: Learning Global Term Weights for Content-Based Recommender Systems. Proceedings of the 25th International Conference on World Wide Web (WWW '16), 2016, pp. 391–400, doi: 10.1145/2872427.2883069.

[16] Huang, H.—Zhao, Q.: Social Book Recommendation Algorithm Based on Improved Collaborative Filtering. In: Liu, Q., Misir, M., Wang, X., Liu, W. (Eds.): The 8th International Conference on Computer Engineering and Networks (CENet2018). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 905, 2018, pp. 477–484, doi: 10.1007/978-3-030-14680-1_52.

[17] Liu, S.—Dong, Y.—Chai, J.: Research of Personalized News Recommendation System Based on Hybrid Collaborative Filtering Algorithm. 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 865–869, doi: 10.1109/CompComm.2016.7924826.

[18] Zhu, J.—Han, L.—Gou, Z.—Yuan, X.: A Fuzzy Clustering-Based Denoising Model for Evaluating Uncertainty in Collaborative Filtering Recommender Systems. Journal of the Association for Information Science and Technology, Vol. 69, 2018, No. 9, pp. 1109–1121, doi: 10.1002/asi.24036.

[19] Fang, Z.—Gao, S.—Li, B.—Li, J.—Liao, J.: Cross-Domain Recommendation via Tag Matrix Transfer. 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015, pp. 1235–1240, doi: 10.1109/ICDMW.2015.133.

[20] Du, Y.—Du, X.—Huang, L.: Improve the Collaborative Filtering Recommender System Performance by Trust Network Construction. Chinese Journal of Electronics, Vol. 25, 2016, No. 3, pp. 418–423, doi: 10.1049/cje.2016.05.005.

[21] Tarus, J. K.—Niu, Z.—Mustafa, G.: Knowledge-Based Recommendation: A Review of Ontology-Based Recommender Systems for E-Learning. Artificial Intelligence Review, Vol. 50, 2018, No. 1, pp. 21–48, doi: 10.1007/s10462-017-9539-5.

[22] Wang, D.—Xu, G.—Deng, S.: Music Recommendation via Heterogeneous Information Graph Embedding. 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 596–603, doi: 10.1109/IJCNN.2017.7965907.

[23] Marwade, A.—Kumar, N.—Mundada, S.—Aghav, J.: Augmenting E-Commerce Product Recommendations by Analyzing Customer Personality. 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN), IEEE, 2017, pp. 174–180, doi: 10.1109/CICN.2017.8319380.

[24] Kumar, S.—Varsha: Survey on Personalized Web Recommender System. International Journal of Information Engineering and Electronic Business (IJIEEB), Vol. 10, 2018, No. 4, pp. 33–40, doi: 10.5815/ijieeb.2018.04.05.

[25] Zhang, L.—Li, J.—Zhang, Q.—Meng, F.—Teng, W.: Domain Knowledge-Based Link Prediction in Customer-Product Bipartite Graph for Product Recommendation. International Journal of Information Technology and Decision Making, Vol. 18, 2019, No. 1, pp. 311–338, doi: 10.1142/S0219622018410031.

[26] Wang, H.—Zhang, P.—Lu, T.—Gu, H.—Gu, N.: Hybrid Recommendation Model Based on Incremental Collaborative Filtering and Content-Based Algorithms. 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2017, pp. 337–342, doi: 10.1109/CSCWD.2017.8066717.

[27] Zhu, M.—Zhen, D.—Tao, R.—Shi, Y.—Feng, X.—Wang, Q.: Top-N Collaborative Filtering Recommendation Algorithm Based on Knowledge Graph Embedding. In: Uden, L., Ting, I. H., Corchado, J. (Eds.): Knowledge Management in Organizations (KMO 2019). Springer, Cham, Communications in Computer and Information Science, Vol. 1027, 2019, pp. 122–134, doi: 10.1007/978-3-030-21451-7_11.

[28] Li, M.—Xiang, Y.—Zhang, B.—Huang, Z.—Zhang, J.: A Trust Evaluation Scheme for Complex Links in a Social Network: A Link Strength Perspective. Applied Intelligence, Vol. 44, 2016, No. 4, pp. 969–987, doi: 10.1007/s10489-015-0734-2.

[29] Kalaï, A.—Zayani, C. A.—Amous, I.—Sedès, F.: Expertise and Trust-Aware Social Web Service Recommendation. In: Sheng, Q., Stroulia, E., Tata, S., Bhiri, S. (Eds.): Service-Oriented Computing (ICSOC 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9936, 2016, pp. 517–533, doi: 10.1007/978-3-319-46295-0_32.

[30] Hamid, M. N.—Naser, M. A.—Hasan, M. K.—Mahmud, H.: A Cohesion-Based Friend-Recommendation System. Social Network Analysis and Mining, Vol. 4, 2014, No. 1, Art. No. 176, doi: 10.1007/s13278-014-0176-6.

[31] Beigi, G.—Liu, H.: Similar but Different: Exploiting Users' Congruity for Recommendation Systems. In: Thomson, R., Dancy, C., Hyder, A., Bisgin, H. (Eds.): Social, Cultural, and Behavioral Modeling (SBP-BRiMS 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10899, 2018, pp. 129–140, doi: 10.1007/978-3-319-93372-6_15.

[32] Singh, H.: Defining and Delivering Personalized Entity Recommendations. September 26, 2019, US Patent Application, 15/935, 579.

[33] Piao, G.—Breslin, J. G.: User Modeling on Twitter with WordNet Synsets and DBpedia Concepts for Personalized Recommendations. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16), 2016, pp. 2057–2060, doi: 10.1145/2983323.2983908.

[34] Li, M.—Xiang, Y.—Zhang, B.—Wei, F.—Song, Q.: A Novel Organizing Scheme of Single Topic User Group Based on Trust Chain Model in Social Network. International Journal of Communication Systems, Vol. 31, 2018, No. 1, Art. No. e3387, doi: 10.1002/dac.3387.

[35] Lodigiani, C.—Melchiori, M.: A PageRank-Based Reputation Model for VGI Data. Procedia Computer Science, Vol. 98, 2016, pp. 566–571, doi: 10.1016/j.procs.2016.09.088.

[36] Ma, X.—Ma, J.—Li, H.—Jiang, Q.—Gao, S.: ARMOR: A Trust-Based Privacy-Preserving Framework for Decentralized Friend Recommendation in Online Social Networks. Future Generation Computer Systems, Vol. 79, 2018, Part 1, pp. 82–94, doi: 10.1016/j.future.2017.09.060.

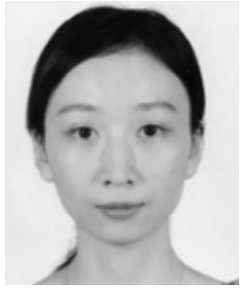
[37] Cherven, K.: Network Graph Analysis and Visualization with Gephi. Packt Publishing Ltd., 2013.

[38] Wang, Y.—Yin, G.—Cai, Z.—Dong, Y.—Dong, H.: A Trust-Based Probabilistic Recommendation Model for Social Networks. Journal of Network and Computer Applications, Vol. 55, 2015, pp. 59–67, doi: 10.1016/j.jnca.2015.04.007.

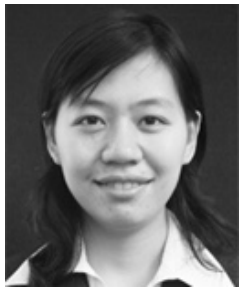
Bo Zhang received his Ph.D. degree in computer science from the College of Electronics and Information Engineering, Tongji University, in 2009. And he finished his PostDoc research work in the Tongji University in 2012. He is now Professor in the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. His current research interests include trust computation and social network analysis. He is now Director of user group analysis project in social network, which is funded by the National Nature Science Foundation of China.

Ya Zhang is Master's student in the College of Computer Science and Technology, Shanghai Normal University, China. She received her B.Sc. from the Shanghai Normal University in 2018. Her research interests are data mining, machine learning, and social network analysis.

Yanhong Bai is now Master's degree candidate in computer application in the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University. Her research interests include social network analysis, swarm intelligence, and group opinion dynamics in social networks.



Jie Lian is Assistant Professor in the Computer Science Department at the Shanghai Normal University where she has been a faculty member since 2017, and she obtained the "Sailing" Talent Program of China in 2019. She completed her doctoral degree at the Towson University in 2017. Her research interests are in the area of spatio-temporal data mining, deep learning and big data, ranging from the theory to design and implementation.



Meizi Li is now Associate Professor in the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. Her current research interests include social network analysis, trust and reputation computation.

Computing and Informatics, Vol. 39, 2020, 141–155, doi: 10.31577/cai_2020_1-2_141

TIME-SENSITIVE COLLABORATIVE FILTERING ALGORITHM WITH FEATURE STABILITY

Shanchen Pang, Shihang Yu, Guiling Li
Sibo Qiao, Min Wang

Shandong University of Science and Technology
No.579 Qianwangang Road
Qingdao, China
e-mail: qd-liguiling@163.com

Abstract. In the recommendation system, the collaborative filtering algorithm is widely used. However, there are lots of problems which need to be solved in recommendation field, such as low precision, the long tail of items. In this paper, we design an algorithm called FSTS for solving the low precision and the long tail. We adopt stability variables and time-sensitive factors to solve the problem of user's interest drift, and improve the accuracy of prediction. Experiments show that, compared with Item-CF, the precision, the recall, the coverage and the popularity have been significantly improved by FSTS algorithm. At the same time, it can mine long tail items and alleviate the phenomenon of the long tail.

Keywords: Collaborative filtering, recommendation algorithm, long tail, time-sensitive

1 INTRODUCTION

With the upsurge of the 5G era, edge computing has also exploded. Edge computing complements cloud computing, providing better real-time, faster data processing capabilities, lower processing costs, lower network bandwidth requirements, and better privacy protection for mobile edge devices [1, 2, 3]. Edge computing brings the functions of the cloud computing center closer to the user-side network, shortens the space distance for users to obtain services, and greatly reduces the delay with services [4]. Nowadays, there are many new situations on the Internet.

Mobile e-commerce is becoming more and more popular. Privacy protection is getting more and more attention in the public. Netizens are drowned in the flood of information and cannot get the information they interest themselves in. However, the recommendation system can recommend the information data according to the history of the user, actively recommend the information that may be of interest to the user, and discover the information valuable to the user in the information flow. The recommendation algorithm is deployed on the edge computing server, which can provide users with real-time recommendations, and can meet the needs of users to protect privacy, and can extract the information of the real-time needs of users. The accuracy of the recommendation system depends mainly on the performance of the recommendation algorithm.

Recommendation algorithms are mainly divided into the neighborhood-based model [5], the latent factor model [6] and the graph-based model [7, 8]. The neighborhood-based model is most widely used in e-commerce scene. This paper mainly aims to improve the neighborhood-based model. The neighborhood-based model is the simplest and easiest algorithm among recommendation algorithms, which is researched deeply in academia and applied most widely in industry. Neighborhood-based algorithms can be divided into two categories: user-based collaborative filtering (User-CF) [9, 10] and item-based collaborative filtering (Item-CF) [11]. The recommendation process of user-based collaborative filtering algorithm is that when user A needs to get some recommendations, firstly, this algorithm will seek out user group B whose interest is like user A's, and then recommend items to A that user group B likes but A has not bought. This method is mainly applicable to areas where the number of users is small, the timeliness is not strong, and the user's personalized interest is not conspicuous. While the recommendation process of item-based collaborative filtering algorithm is that when user A purchases item a, firstly, the algorithm will find item set b which is like item a, and recommend items in item set b to user A which has not been purchased by user A. This method is suitable for the areas where the number of items is obviously less than the number of users, long tail items are abundant, and users have strong personalized needs. Item-based collaborative filtering algorithm is more widely applied in industry circles than user-based collaborative filtering algorithm. The algorithm in this paper is an improvement of the item-based collaborative filtering algorithm.

2 RELATED WORK

Lots of scholars have done numerous studies on collaborative filtering based on items. Huang and other scholars combined the user theme model with the item theme model, and considered the tag information of the item and the users' behavior data [12]. They proposed a hybrid recommendation algorithm based on the item collaborative filtering model, which improved the diversity and accuracy of the recommendation. Nikolakopoulos et al. proposed a new random migration method, which overcomes the obstacle of fast gait convergence to the graph center by us-

ing the spectral characteristics of uncoupled Markov chains [13]. With this method, they prolonged the follow-up effect of users' previous preferences on walking, allowed pedestrians to explore more basic networks, and improved the recommendation quality. Chen et al. put forward the Attentive Collaborative Filtering (ACF) model, which used groupware-level attention module to extract network from content features, and used project-level attention module to mark preferences of projects [14]. The experimental results show that ACF is superior to CF. Wei et al. used deep neural network to collect item content features, and considered the temporal dynamics of users' preferences and temporal dynamics of item features [15]. They used collaborative filtering to build a model, which improved the prediction accuracy of recommendation system. Dong et al. put forward a hybrid model through learning the efficiency of feature extraction, which can effectively know the users' depth and the potential features of the project from the score matrix [16]. Li et al. combined the decomposition of probability matrix with the stackable automatic which has the feature of edge noise reduction [17]. With this method, the problem of insufficient feature extraction caused by data sparsity has been solved. Nilashi et al. came up with an incremental updating multi-criteria collaborative filtering method based on clustering and regression [18]. This method automatically detects and subdivides users by clustering, the learning preference model is subdivided for each user, and the preference model can be updated incrementally. Although the methods which are talked about above got good forecasting results, the model is too complex, and it is difficult to calculate the similarity of items. At the same time, the long tail [19] distribution of items has not been taken into consideration.

The FSTS algorithm proposed in this paper not only extracts the features of items, but also considers the stability of the features of items. Meanwhile, a time-sensitive factor is added to make the algorithm time-sensitive to combat the phenomenon of interest drift [20]. This algorithm can effectively improve the long tail pheonmenon of items while ensuring the prediction accuracy.

3 FSTS ALGORITHM MODEL

FSTS algorithm constructs feature vectors of items by analyzing the frequency of items purchased by users and their rating. At the same time, because the user's interest will change with the passing of time, which is called interest drift, this algorithm adds time influence factor to solve user's interest drift problem. The main steps of FSTS include data preprocessing, feature extraction of items, time-sensitive factor antagonism, rule base generation and item recommendation. The main process is shown in Figure 1.

1. Data preprocessing stage: This stage mainly handles invalid ratings and invalid users. There are some invalid ratings in the user-item scoring matrix, such as those higher than the maximum value and those with error codes caused by some factors, which will affect the final rule base and lead to the inaccuracy of the recommendation prediction. Invalid users refer to those who do not give

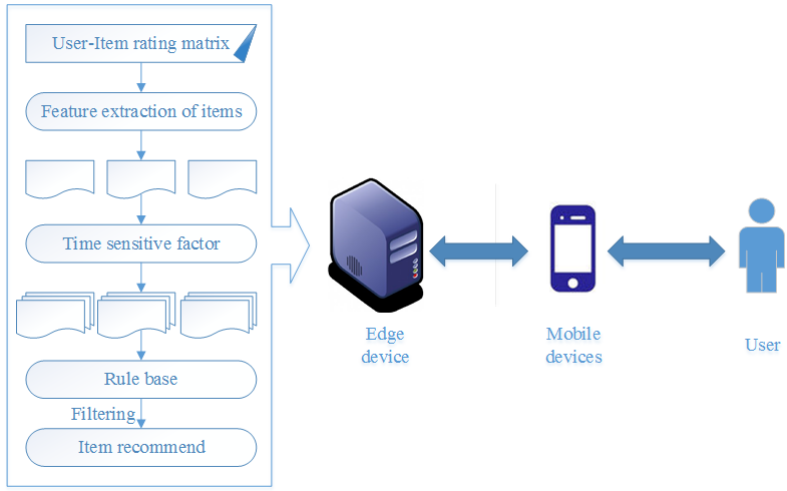


Figure 1. FSTS algorithm model

any marks on items or who give marks on all items. These users who are called zombie users are likely to attack the recommendation system. These users are meaningless to the final prediction, and even will affect the final prediction results, so they should be eliminated.

2. Feature extraction of items: The frequency of items being purchased and the ratings of items are regarded as the feature of items. The feature vectors of each item are calculated by using the user-item rating matrix. Then the feature vectors of all items are obtained, and the preliminary rule base is gained.
3. Adding time impact factors to confront: User's interest is changed with the passing of time, which is interest drift phenomenon. This algorithm adds time impact factor to solve interest drift problem, not only excavating users' long-term interest, but also discovering users' short-term interest.
4. Building the rule base. The algorithm is used to calculate the similarity model of items and build the recommendation rule base. When recommending items to users, the corresponding rules can be picked up directly from the rule base.
5. Item recommendation: When users browse, collect, add items into shopping carts and purchase items, some invalid rules are filtered out according to the rules related to the current items in the rule base. For example, Top-N, which is highly relevant to the current items but has been purchased by users, is recommended to users.

In the paper, we make the following assumptions:

1. In this recommendation system, we do not consider the cold start of systems.
2. In the paper, we do not consider the cold items and the cold users.

3. In the paper, we consider the single recommendation scene. And we do not consider the multiple recommendation scenes.

4 FSTS RECOMMENDATION ALGORITHMS

4.1 Problem Modeling

Suppose that there are a certain number of users and a certain number of items that were purchased by users. The set of these users is expressed as $U = \{u_1, u_2, \ldots, u_N\}$ and the set of items is expressed as $S = \{s_1, s_2, \ldots, s_M\}$. The user u_i scored the item s_i as a_{ij}, so the user's corresponding interest vector is expressed as $v_i = (a_{i1}, a_{i2}, \ldots, a_{ij})$. User-item rating matrix m whose size is $N \times M$, is composed of all users' interest vectors. For users u_i and items s_j with purchasing relationship, the corresponding rating is $a_{ij} > 0$, while for users and items without purchasing relationship, the corresponding rating is $a_{ij} = 0$.

4.2 The Description of the Algorithm

4.2.1 Features of Items

For recommendation system, if a user has not purchased any items or the user has purchased all items, it not only increases the difficulty for calculation, but also reduces the precision of prediction, so the existence of this user is meaningless. Then for a normal user, the more items he has purchased, the less influence he will have on the generation rules of associated rule base. Therefore, user's activity is defined as A, which is shown in Formula (1):

$$A_i = \log_2 (1 + n_i) . \tag{1}$$

In this formula, the activity of the user i is expressed as A_i. n_i represents the total number of items purchased by the user i.

In this paper, Formula (1) is used to perform feature extraction on users to distinguish different users, and thus to distinguish the different users and the different contributions of different users to the similarity of items.

Different users' rating criteria are also different. For example, if the full mark is ten points, some users will think that 9.5 is high enough. These users may not give 10 points for their favorite items, or give very little full marks, while other users could easily give 10 points for items they were satisfied with. Thus, it can be known that the rating is a process which is mixed with subjective factors. Users are satisfied with the same item, but the marks cannot objectively reflect the user's satisfaction. Therefore, this algorithm is designed to standardize the user's rating in order to eliminate the user's own bias on the item.

Directly related to the item's features is the user's rating to the item and the activity of the user who purchased the item. Therefore, the feature matrix of items

is defined as $F = (F_1, F_2, \ldots, F_M)^T$, the feature vector of the item j is defined as $F_j = (f_{j1}, f_{j2}, \ldots, f_{jN})$ and the feature value of the item j at the user i is f_{ji}, which is shown in Formula (2):

$$f_{ji} = \alpha \times \frac{R_{ji} - R_{imin}}{R_{imax} - R_{imin} + 1} + \beta \times \frac{A_i - A_{jmin}}{A_{jmax} - A_{jmin} + 1}. \tag{2}$$

In this formula, R represents the user's rating on the item, R_{ji} represents the user i's rating on the item j, R_{imin} means the minimum value of all the ratings given by the user i, and R_{imax} means the maximum value of all the ratings made by the user i, A represents the activity of users, A_i is the activity of user i, A_{jmin} represents the minimum activity of all users who have purchased item j, and A_{jmax} is the maximum activity of all users who have purchased item j. α, β are the formula parameters, and meet the formula requirements: $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$, $\alpha + \beta = 1$. In this experiments, $\alpha = 0.5$ and $\beta = 0.5$.

Different users have different ratings on the same item. Some items have been given higher ratings, some items have been given lower ratings, and some items have more differences in ratings. In this paper, the variance of ratings is used to describe the change of item' rating, that is, the stability of item rating. The lower the stability of the item rating, the more consistent the rating of all users is, which means the item is less significant for calculating its contribution to the similarity with other items. The higher the stability of item rating, the more inconsistent the rating of all users is. That is, the more inconsistent the user likes or not, which means the item is more useful for calculating its contribution to the similarity with other items. The description of the rating stability of item j is shown in Formula (3):

$$F\_S(j) = \frac{\sum_{i=1}^{N} (a_{ij} - \overline{a_{-j}})}{N}. \tag{3}$$

In this formula, $F\_S(j)$ represents the rating stability of the item j. a_{ij} represents the user i's rating on the item j and $\overline{a_{-j}}$ is the average value of all users' rating on the item j. N represents the total number of users.

4.2.2 Time-Sensitive Factors

As the time goes by, the user's interest is likely to change, that is called user's interest drift phenomenon. An item purchased by a user on the spot has a very low or even no correlation with the item he purchased long ago. While recommendation is timeliness, which means when the user decides to choose or purchase the current item, items related to or similar to the current item should be recommended immediately. Therefore, time-sensitive factor $e^{-\delta|t_{ij} - t_{ik}|}$ is added to this algorithm to solve interest drift caused by time passing. This factor is on a daily basis. t_{ij} represents the time when the user i purchases the item j, δ means the time-sensitive coefficient, which satisfies the requirement: $0 < \delta < 1$. In this expriments, $\delta = 1/7$. The closer the time for user i to purchase item j and item k, the greater the similarity between item j and item k.

4.2.3 FSTS Algorithm

The FSTS algorithm assigns linear weight to the stability of item rating and the time-sensitive factor, in order to influence the feature vector of the item, which is shown as Formula (4):

$$FSTS_{jk} = \frac{F_j \ldots F_k}{F_j F_k} \times \left(F\_S(j) \times F\_S(k) + \sum_{i \in U_j \cap U_k} e^{-\delta |t_{ij} - t_{ik}|} \right). \tag{4}$$

In this formula, $FSTS_{jk}$ represents the similarity between item j and item k. U_j represents a set of users who have all purchased item j and U_k is a set of users who have all purchased item k. This algorithm adds the feature stability of items and time-sensitive factor, which is called FSTS algorithm. The algorithm considers both the rating stability of items and time-sensitive factor, and it can flexibly adjust the influence weights of them for different application scenarios, which has achieved good prediction results.

5 EXPERIMENTAL ANALYSIS

5.1 Experimental Data Set

This experiment uses the public movie data set MovieLens provided by GroupLens, which is a specialized research laboratory of recommendation system. This experiment uses a 20 M MovieLens data set, which contains 138 000 users giving 20 000 000 marks to 27 000 movies. In this experiment, the data set is divided into test set and training set according to the ratio of 2 : 8.

5.2 Evaluation Criteria

There are four evaluation criteria in this experiment: Precision, Recall, Popularity and Coverage, which are used as performance indicators to evaluate the final recommendation results of the experiment. The four criteria are shown in Formulas (5), (6), (7), and (8):

$$\text{Precision} = \frac{\sum_U |R(u) \cap T(u)|}{\sum_U |R(u)|}, \tag{5}$$

$$\text{Recall} = \frac{\sum_U |R(u) \cap T(u)|}{\sum_U |T(u)|}, \tag{6}$$

$$\text{Popularity} = \sum_M \log_2 (1 + \text{sum}(m)), \tag{7}$$

$$\text{Coverage} = \frac{|U_{u \in U} R(u)|}{|I|}. \tag{8}$$

In these formulas, $R(u)$ represents the item collection recommended to users, $T(u)$ means the collection of items that users like in the test set; U represents the set of all users in the test set, I means the number of all users in the data set, M represents a collection of movies in the recommendation list, and $sum(m)$ is the times of the movie m appearing in the training set.

Precision and recall describe the prediction accuracy of recommendation results for users. The higher are the precision and recall, the better are the recommendation results. Popularity and coverage describe the composition of the items of the recommendation results. The lower is the popularity and the higher is the coverage, the better are the recommendation results.

5.3 Result Analysis

In this paper, FSTS is compared with Item-CF algorithm. The numbers of recommendation neighbors are respectively set to 5, 10, 15, 20, 25, 30, 35 and 40. The experimental results show that FSTS algorithm outperforms Item-CF algorithm in all aspects of prediction performance. Figure 2 shows that compared with Item-CF algorithm, the precision of FSTS algorithm is significantly better than that of Item-CF algorithm. At that time, the prediction accuracy of FSTS algorithm increased by 7.6 % when $K = 5$. As the value of K increases, the precision of the FSTS algorithm decreases as the similarity of the recommended items decreases. Figure 3 shows that the recall of FSTS algorithm is also better than that of Item-CF algorithm. At that time, the recall of FSTS algorithm increased by 6.1 % when $K = 25$. As the value of K increases, the similarity of the recommended items decreases, so the recall of FSTS algorithm increases which means the precision decreases.

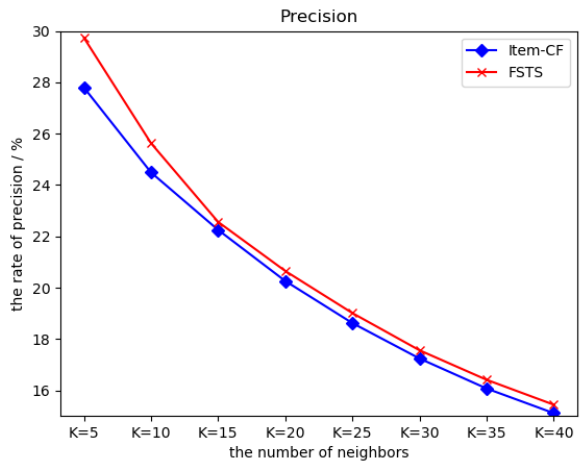


Figure 2. Comparison of precision between FSTS and item-CF

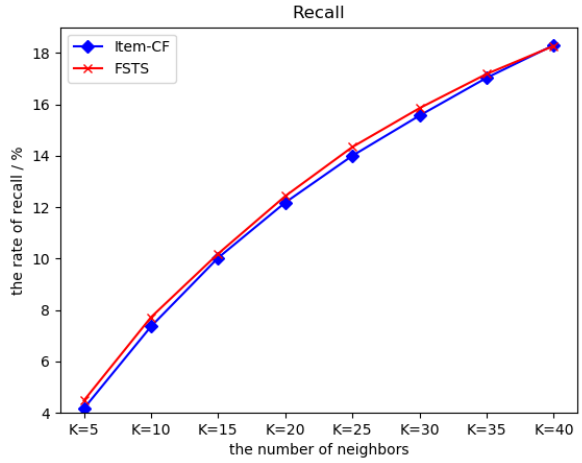


Figure 3. Comparison of recall between FSTS and item-CF

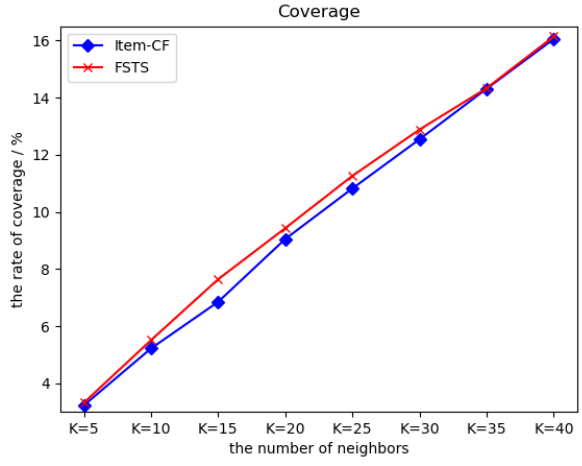


Figure 4. Comparison of coverage between FSTS and item-CF

Coverage and popularity reflect the overall quantity of items. Figure 4 shows that compared with Item-CF algorithm, the coverage of FSTS algorithm has been improved. When $K = 15$, the coverage of FSTS algorithm has increased by 15.1 %. As the value of K increases, the number of items recomended to users increases, so the coverage increases. Figure 5 shows that the proportion of popular items recommended by the FSTS algorithm is significantly lower than that recommended by the

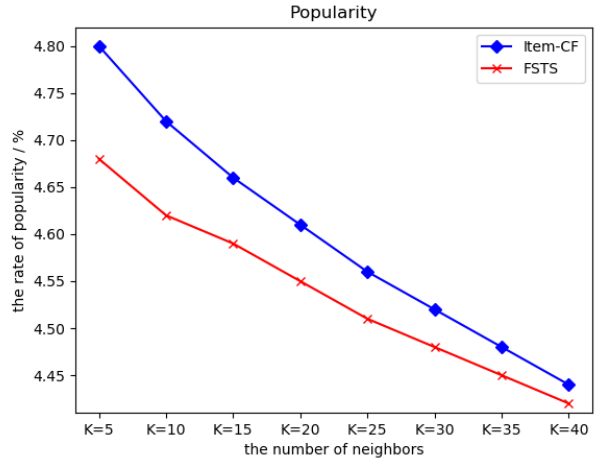


Figure 5. Comparison of popularity between FSTS and item-CF

Item-CF algorithm. At the same time, the popularity of items in the recommendation list of the FSTS algorithm decreased by 4.3 %, when $K = 5$. As the value of K increases, the popularity of items decreases, so the popularity of the FSTS algorithm decreases.

Long tail of items is also alleviated by this algorithm recommendation. Figure 6 shows the distribution of the movies watched, and Figure 7 shows the long

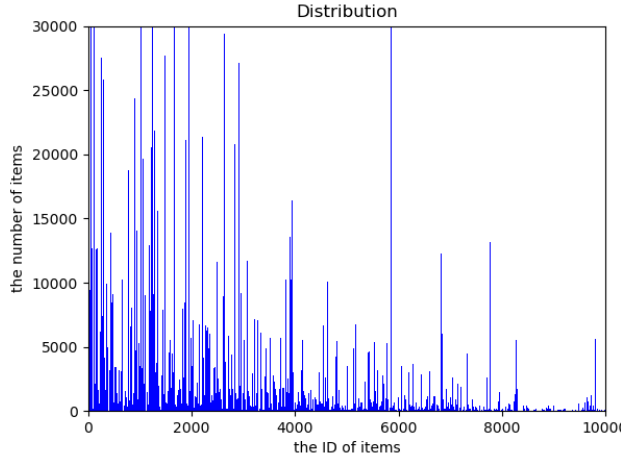


Figure 6. The distribution of item

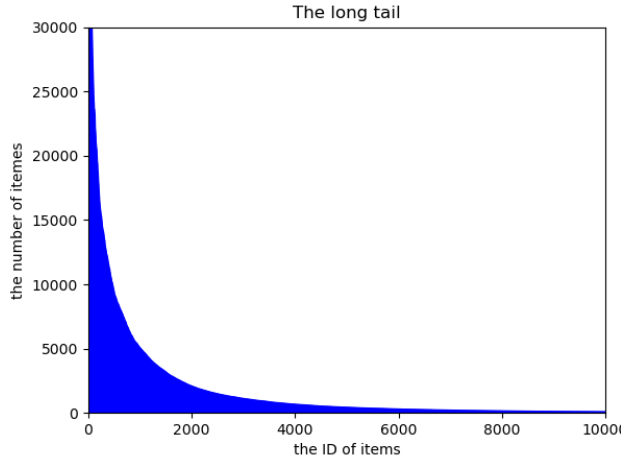


Figure 7. The long tail

tail of the items before the algorithm is used. It can be known that the data set is consistent with the long tail, and the long tail is obvious. Figure 8 shows that the long tail distribution of items has been significantly alleviated after multiple recommendation using FSTS algorithm, that is, items at the tail place have been more fully recommended and potential interests of users have been fully excavated.

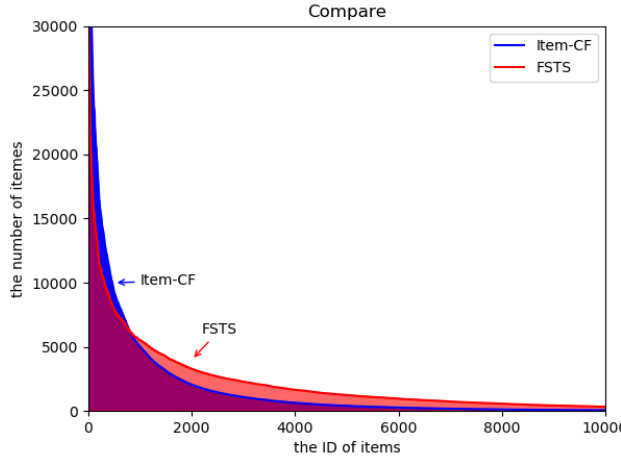


Figure 8. Comparison of long tail distribution between item-CF and FSTS

6 SUMMARY AND OUTLOOK

This paper is aimed at the characteristics of the edge devices and FSTS algorithm is designed in this paper. The algorithm takes into consideration both the stability of item features and users' interest drift. In the prediction process, the feature vector and time-sensitive factor are added to deeply extract items features and deal with the drift of user interest. Experiments show that FSTS algorithm improves both prediction performance and algorithm performance with low time complexity.

However, the dynamic transformation of the time-sensitive factor is weak. Next, we will strengthen the dynamic transformation of this factor, and deploy the algorithm to the recommendation system, and further optimize the algorithm through the feedback mechanism of the system, in order to obtain more accurate prediction data and higher efficiency. And in this paper, we do not consider the cold start scenes. So the FSTS algorithm does not fit the cold start scenes. Next, we will do the work about the cold start of system, the cold users and the cold items. Beside, we consider to use CNN net to accelerate the algorithm in the next step.

REFERENCES

[1] Pang, S.—Qiao, S.—Song, T.—Zhao, J.—Zheng, P.: An Improved Convolutional Network Architecture Based on Residual Modeling for Person Re-Identification in Edge Computing. IEEE Access, Vol. 7, 2019, pp. 106749–106760, doi: 10.1109/ACCESS.2019.2933364.

[2] Song, T.—Pang, S.—Hao, S.—Rodríguez-Patón, A.—Zheng, P.: A Parallel Image Skeletonizing Method Using Spiking Neural P Systems with Weights. Neural Processing Letters, Vol. 50, 2019, pp. 1485–1502, doi: 10.1007/s11063-018-9947-9.

[3] Pang, S.—Wang, M.—Qiao, S.—Wang, X.—Chen, H.: Fault Diagnosis for Service Composition by Spiking Neural P Systems with Colored Spikes. Chinese Journal of Electronics, Vol. 28, 2019, No. 5, pp. 1033–1040, doi: 10.1049/cje.2019.06.023.

[4] Pang, S.—Gao, Q.—Liu, T.—He, H.—Xu, G.—Liang, K.: A Behavior Based Trustworthy Service Composition Discovery Approach in Cloud Environment. IEEE Access, Vol. 7, 2019, pp. 56492–56503, doi: 10.1109/ACCESS.2019.2913432.

[5] Saelens, B. E.—Sallis, J. F.—Black, J. B.—Chen, D.: Neighborhood-Based Differences in Physical Activity: An Environment Scale Evaluation. American Journal of Public Health, Vol. 93, 2003, No. 9, pp. 1552–1558, doi: 10.2105/ajph.93.9.1552.

[6] Zhang, W.—Wang, J.—Feng, W.: Combining Latent Factor Model with Location Features for Event-Based Group Recommendation. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), ACM, 2013, pp. 910–918, doi: 10.1145/2487575.2487646.

[7] Pang, S.—Chen, H.—Liu, H.—Yao, J.—Wang, M.: A Deadlock Resolution Strategy Based on Spiking Neural P Systems. Journal of Ambient Intelligence and Humanized Computing, 2019, 12 pp., doi: 10.1007/s12652-019-01223-3.

[8] Fouss, F.—Pirotte, A.—Renders, J.-M.—Saerens, M.: Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, 2007, No. 3, pp. 355–369, doi: 10.1109/TKDE.2007.46.

[9] Zhao, Z.-D.—Shang, M.-S.: User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. 2010 Third International Conference on Knowledge Discovery and Data Mining (WKDD'10), IEEE, 2010, pp. 478–481, doi: 10.1109/WKDD.2010.54.

[10] Wang, S.—He, S.—Yuan, F.—Zhu, X.: Tagging SNP-Set Selection with Maximum Information Based on Linkage Disequilibrium Structure in Genome-Wide Association Studies. Bioinformatics, Vol. 33, 2017, No. 14, pp. 2078–2081, doi: 10.1093/bioinformatics/btx151.

[11] Linden, G.—Smith, B.—York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, Vol. 7, 2003, No. 1, pp. 76–80, doi: 10.1109/MIC.2003.1167344.

[12] Huang, L.—Lin, C.—He, J.—Liu, H.—Du, X.: Diversified Mobile App Recommendation Combining Topic Model and Collaborative Filtering. Journal of Software, Vol. 28, 2017, No. 3, pp. 708–720, doi: 10.13328/j.cnki.jos.005163 (in Chinese).

[13] Nikolakopoulos, A. N.—Karypis, G.: Boosting Item-Based Collaborative Filtering via Nearly Uncoupled Random Walks. arXiv preprint arXiv:1909.03579, 2019.

[14] Chen, J.—Zhang, H.—He, X.—Nie, L.—Liu, W.—Chua, T.-S.: Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17), ACM, 2017, pp. 335–344, doi: 10.1145/3077136.3080797.

[15] Wei, J.—He, J.—Chen, K.—Zhou, Y.—Tang, Z.: Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items. Expert Systems with Applications, Vol. 69, 2017, pp. 29–39, doi: 10.1016/j.eswa.2016.09.040.

[16] Dong, X.—Yu, L.—Wu, Z.—Sun, Y.—Yuan, L.—Zhang, F.: A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems. Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17), 2017, pp. 1309–1315.

[17] Li, S.—Kawale, J.—Fu, Y.: Deep Collaborative Filtering via Marginalized Denoising Auto-Encoder. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15), ACM, 2015, pp. 811–820, doi: 10.1145/2806416.2806527.

[18] Nilashi, M.—Jannach, D.—bin Ibrahim, O.—Ithnin, N.: Clustering- and Regression-Based Multi-Criteria Collaborative Filtering with Incremental Updates. Information Sciences, Vol. 293, 2015, pp. 235–250, doi: 10.1016/j.ins.2014.09.012.

[19] Oestreicher-Singer, G.—Sundararajan, A.: Recommendation Networks and the Long Tail of Electronic Commerce. MIS Quarterly, Vol. 36, 2012, No. 1, pp. 65–83, doi: 10.2307/41410406.

[20] Yin, H.—Zhou, X.—Cui, B.—Wang, H.—Zheng, K.—Nguyen, Q. V. H.: Adapting to User Interest Drift for POI Recommendation. IEEE Transactions on

Knowledge and Data Engineering, Vol. 28, 2016, No. 10, pp. 2566–2581, doi: 10.1109/TKDE.2016.2580511.

Shanchen Pang received his graduation degree from the Tongji University of Computer Software and Theory, Shanghai, China, in 2008. He is Professor in the China University of Petroleum, Qingdao, China. His current research interests include theory and application of Petri net, service computing, trusted computing.

Shihang Yu received his graduation degree in the Shandong University of Science and Technology, Qingdao, in 2017. He is graduate student of the China University of Petroleum, Qingdao, China. His current research interests include recommendation system, data mining.

Guiling Li received her engineering Ph.D. degree in control theory and control engineering from the Shandong University of Science and Technology, Qingdao, China, in 2009 and 2014. In 2005, she began working in the Academy of Mathematics and Systems Science at Shandong University of Science and Technology, Qingdao, China. Her research interests are in the area of stochastic control.

Sibo Qiao received his B.Sc. degree from the Shandong University of Science and Technology, Qingdao, in 2017. He is graduate student of the China University of Petroleum, Qingdao, China. His current research interests include deep learning, person re-identification, and object detection.

Min Wang received her M.Sc. degree from the China University of Petroleum, Qingdao, China, in 2019. She is Ph.D. student of control science and engineering in the China University of Petroleum, Qingdao, China. Her current research interests include theory and application of Petri net, trusted computing.

Computing and Informatics, Vol. 39, 2020, 156–173, doi: 10.31577/cai_2020_1-2_156

REVERSE INTERVENTION FOR DEALING WITH MALICIOUS INFORMATION IN ONLINE SOCIAL NETWORKS

Deyu YUAN, Haichun SUN*

College of Police Information Engineering and Cyber Security
People's Public Security University of China
Beijing 102623, China
&
Key Laboratory of Safety Precautions and Risk Assessment
Beijing 102623, China
e-mail: {yuandeyu, sunhaichun}@ppsuc.edu.cn

Abstract. Malicious information is often hidden in the massive data flow of online social networks. In "We Media" era, if the system is closed without intervention, malicious information may spread to the entire network quickly, which would cause severe economic and political losses. This paper adopts a reverse intervention strategy from the perspective of topology control, so that the spread of malicious information could be suppressed at a minimum cost. Noting that as the information spreads, social networks often present a community structure and multiple malicious information promoters may appear. Therefore, this paper adopts a divide and conquer strategy and proposes an intervention algorithm based on subgraph partitioning, in which we search for some influential nodes to block or release clarification. The main algorithm consists of two main phases. Firstly, a subgraph partitioning method based on community structure is given to quickly extract the community structure of the information dissemination network. Secondly, a node blocking and clarification publishing algorithm based on the Jordan Center is proposed in the obtained subgraphs. Experiments show that the proposed algorithm can effectively suppress the spread of malicious information with a low time complexity compared with the benchmark algorithms.

Keywords: Malicious information, social network, reverse intervention

* Corresponding author

1 INTRODUCTION

In recent years, social media has become an important platform for online users to participate in the Internet, such as Facebook, Twitter, Sina Weibo, WeChat, QQ, etc. Users on social media have formed OSN (Online Social Networks). The expansion of the social category from physical space to virtual space is a process from quantitative change to qualitative change. On the one hand, the deep integration of social media and politics, economy and culture releases positive energy, and the highly connected OSN provides infrastructure for the realization of "Internet +". On the other hand, malicious information such as rumors and fake news often hide in the massive social data, which brings unprecedented challenges for national security and social stability, making the high-speed diffusion of information in OSN a double-edged sword. The "information security" in online social networks has attracted more and more attention.

The control of malicious information in OSN is mostly studied from two aspects: credibility evaluation and information dissemination dynamics. In the perspective of credibility evaluation, classification or sorting methods are often used, and the text content of social media, supplemented by user information and communication characteristics could be analyzed. Kwon et al. [1] used the timing characteristics, combined with the structure and semantics of the message to identify rumors. Song et al. [2] analyzed the statistical characteristics of text content (such as number of repeated microblogs in the last 20 Weibo contents, number of external links, number of @ symbols, number of topic tags) and characteristics of user relationships (number of followers, reputation of each user) to identify malicious information on Twitter. In the perspective of information dynamics, Fang [3] used life cycle theory to divide the information fermentation process into four stages: gestation, diffusion, transformation and attenuation. Lan et al. [4] established a differential equation model based on the forming process and influencing factors to study the information evolution in the network, and the authors proposed three characteristic time points and four periods for public opinion diffusion.

The above studies provide the basis for reverse control (i.e. manual intervention) of malicious information, even though none of them mentioned the intervention of malicious information. In the analysis of the propagation of malicious information, OSN is regarded as a closed system. The attacker can choose a reasonable publishing strategy to make information spread quickly and achieve his purpose. However, in reality, the system is open. From a theoretical point of view, the multi-layered information dissemination process could be interfered by adding disturbance variables. From a practical point of view, it is possible to issue clarification or block rumor accounts, making malicious information and the clarification disturb each other, which could hinder the rapid spread of malicious information. Therefore, to timely and effectively disturb the evolution of malicious information is a challenging and important issue.

The current literatures of reverse control mostly compromise on effectiveness and efficiency. In this paper, we go further on reverse control in OSN and try

our best to suppress the spread of malicious information at a minimum cost. The main contributions of this paper can be summarized as follows. Firstly, we propose a novel community partitioning algorithm to reduce the complexity in large-scale networks. Secondly, we introduce a mechanism which incorporates both blocking and clarification publishing methods to impede the spread of malicious information. Thirdly, we utilize the Jordan Center to select key nodes for publishing clarifications. Finally, we verify the effectiveness of the model through the experiments.

The remainder of this paper is organized as follows. We review the related works in Section 2. In Section 3, we present the problem formulation. The reverse intervention algorithm based on subgraph partitioning is proposed in Section 4. Experimental results are presented in Section 5. Finally, we summarize this paper in Section 6.

2 RELATED WORK

The research of reverse control in OSN originated from the invulnerability of complex networks, in which different measurement and control indicators have been proposed and analyzed. For example, the authors of [5] studied the invulnerability of ad hoc network, in which "k-connectivity" and power control were used to protect the network against random failure. In [6], the critical removal ratio was used as the measurement for the networks with incomplete information, and the invulnerability of the network was analyzed based on characteristic spectrum. Albert et al. [7] used generating function to analyze the critical removal ratio under the random failure conditions. Cohen et al. [8] extended the problem to the generalized random graph. Callaway et al. [9] studied the percolation problem on graphs with completely general degree distribution and proposed some specific solutions for a variety of cases, including site percolation, bond percolation, and models in which occupation probabilities rely on vertex degree. In [10], highly optimized tolerance (HOT) theory and node preference attachment mechanism were used to build the invulnerable dynamic evolution model for the studied network.

The controllability and information diffusion were also analyzed in opportunistic social networks [11] and location-based social networks (LBSNs). For example, weight distribution between nodes and communities reconstitution were established in [12] to solve the problem of message delivery for social opportunistic networks. In [13], the authors proposed a routing algorithm called sensor communication area node extend (SCANE) to select relevance nodes and to recombine communication areas. In [14], a method for recommending points of interest (POIs) was proposed based on a collaborative tensor factorization (CTF) technique. Luan et al. [15] proposed a maximal-marginal-relevance-based personalized trip recommendation method that considers both relevance and diversity of trips in a trip planning. These literatures are inspiring and instructive to analyze the propagation of rumors.

In order to restrain the propagation of rumors, scientists have proposed many methods. The literatures can be roughly categorized as controlling influential users (links), and clarifying the rumors by spreading the truths under different diffusion models. For blocking strategies, the evaluation of important nodes and links plays an important role in the blocking strategies. The centrality indicators such as degree centrality, clustering coefficient, betweenness centrality, closeness centrality, k-shell decomposition [16], HITS algorithm [17], PageRank algorithm [18], network efficiency [19], Laplace centrality [20] , structural hole [21], minimum spanning tree index [22], mutual information method [23], and node contraction method [24] could be drew on in the proposed strategies. Fan et al. [25] explored the Least Cost Rumor Blocking (LCRB) problem to prevent rumors from spreading. The authors tried to minimize the number of people infected from the originate community to other communities by identifying a minimal bridge end set which diffuse the positive (protector) cascade. However, the authors assumed that the cascade of rumor and protector start at the same time, which was not in line with the real situation that the positive cascade was usually released after the rumor has been noticed. For clarifying the rumors, Wan et al. [26] proposed a novel model of competitive coupling to describe the complex process of information diffusion in online social networks and introduced the constrained intervention strategies. The analysis of coupling diffusion among different information is very inspiring when we introduce the clarifications. Wen et al. [27] numerically evaluated the two types of strategies used for restraining rumors in OSNs, including blocking rumors at important users and clarifying rumors by spreading truths, thus introduced a mathematical model to present the spread of rumors and truths. The authors found that the truth clarification method could eliminate more rumors in the long run while the blocking method based on degree could provide better performance in the early stage of the rumor spread.

Credibility analysis of posts and users was also used to control the rumor spreading. Bao et al. [28] proposed a novel immunization strategy called MST based on trust network. The authors established a weighted trust network based on the trust relationship between users, and determined the most important information diffusion paths to cut down. However, the trust weight of the links was hard to determine and the proposed algorithm was time consuming. Bao et al. also proposed a SPNR model in [29], in which the authors split the infected states into two separate states according to whether the user support or oppose the information. That is, the paper assumed the users in OSN could spontaneously oppose the rumor. However, only parameters' influence was analyzed and effective rumor control strategies need further discussion. Bhattacharya et al. [30] proposed a belief surveillance approach for specific propositions, which is inspired by studies on disease surveillance. The authors demonstrated that although factual statements garner a high degree of belief, some are still being questioned, and some fictional statements also garner a high degree of belief, which was instructive for the control of malicious information.

Inspired by the above literatures, we incorporate both blocking and clarification publishing methods to control the diffusion of malicious information. In this paper,

we break the assumption of closed systems and implement reverse interventions to impede the spread of malicious information.

3 PROBLEM FORMULATION

In this section, we give the definition of the Jordan Center and present information diffusion models and symbols used in this paper.

3.1 Diffusion Model

We use directed graph $G = (\mathbb{V}, \mathbb{E})$ to represent OSN, where $\mathbb{V}$ is the set of nodes, and $\mathbb{E}$ is the set of edges in the network. Two nodes connected by edges are called neighbors (e.g., there is a relationship of following). At some certain moment, an attacker in the network issues a malicious message m, and other nodes in the network will receive message m and forward it to its neighbor nodes.

Next, we describe the propagation model used in this paper. Existing information dissemination models can be roughly divided into two categories: epidemiological infection models such as SI (Susceptible-Infected), SIR (Susceptible-Infected-Recovery) and SIS (Susceptible-Infected-Susceptible) model and influence diffusion models such as IC (Independence Cascade) and LT (Linear Threshold) model. This paper focuses on the influence diffusion model, namely LT and IC model. These two models have received extensive attention since they were first proposed in the pioneering work of Kempe et al. [31].

IC model: An infected node v has only one chance to infect its susceptible neighbors, and each neighbor node $w \in N(v)$ ($N(v)$ represents the neighbor set of node v) can be infected with an independent probability $p_{v,w}$.

LT model: Each node in the network independently selects a threshold $\theta_v \in [0, 1]$ at the initial stage. Whether a susceptible node w adopts the information depends on the sum of all its neighbors' weights $p_{v,w}$, where $v \in N(w)$. When the sum of the weights for susceptible node w satisfies $\sum_{v \in N(w)} b_{v,w} \geq \theta_v$, the node w will be infected.

3.2 Problem Formulation

The reverse intervention of malicious information is closely related to the influence diffusion model of OSN. Malicious information spread together with other information in the network, and information holding the opposite opinion will compete with each other. In the real world, users who receive clarification usually should no longer accept the malicious information (rumors). In order to prevent people from being misled by malicious information, a natural way is to introduce clarifications to uninfected users as soon as possible, at least earlier than the arrival of malicious information. Once malicious information is detected, the network administrator (e.g.

police department) can generate a competitive positive cascading (clarification) to minimize the number of infected (propagating) users. In this paper, we assume that the clarification has higher priority than malicious information to activate nodes. Therefore, according to the IC and LT model discussed above, the problem that needs to be solved in this paper is described as the following optimization problem.

$$\min |\mathbb{S}| \tag{1}$$

s.t.

$$\mathbb{S} \subset \mathbb{V}, \tag{2}$$

$$\frac{|\mathbb{I}(G)|}{|\mathbb{V}|} \leq \beta. \tag{3}$$

That is, according to the propagation situation of malicious information m, we try to select a minimum set of nodes to block or publish clarification to control the spread of malicious information, so that the infection rate of the whole network after a time window T is less than β ($\mathbb{I}(G)$ in Equation 3 represents the set of infected nodes in the network).

3.3 Jordan Center

In this subsection, we give the definition of the Jordan Center according to the previous work [32, 33].

Definition 1 (Jordan Center)**.** Let $d(s, u)$ represent the distance between nodes s and u in graph G (i.e. length of the shortest path). $\mathbb{A}$ is a collection of randomly selected nodes in G, and $\bar{d}(s, \mathbb{A})$ is defined as the eccentricity of node s, i.e., the maximum distance between s and any selected node of $\mathbb{A}$, yielding:

$$\bar{d}(s, \mathbb{A}) = \max_{u \in \mathbb{A}} d(s, u). \tag{4}$$

Jordan Center of $\mathbb{A}$ is defined as the node with the smallest eccentricity in G.

4 REVERSE INTERVENTION ALGORITHM BASED ON SUBGRAPH PARTITIONING

In online social networks, algorithms based on community partitioning have been proven to be effective [34, 35]. In the actual network, we can usually observe the fragments of the propagation, take SIR model as an example, some nodes will change from infected state to recovery state. In this paper, we ignore the problem of incomplete observation. Considering the huge advantages of community partitioning, we propose a community-based heuristic method according to the network topology to solve the problem of reverse intervention for malicious information. Specifically, our approach consists of two main phases:

1. subgraph partitioning based on community structure to quickly reveal the community structure of the network;
2. node selection based on the Jordan Center to effectively control the spread of malicious information by means of high-influence nodes.

4.1 Subgraph Partitioning Algorithm Based on Community Structure

In networks with distinct community structures, information is more likely to spread within the community and then spread to other areas of the network. As shown in Figure 1, the network often presents a community structure. As the malicious information spreads, users will hold different opinions on the current event, thus malicious information and external disturbances will form a competition process. The red arrow in Figure 1 represents the opponent flow of malicious information, the blue arrow represents the supporter flow, and the two will form a hedge. By observing the current information dissemination, this paper uses the community structure of the network to distinguish the spread of malicious information, and then suppress the spread of malicious information by publishing clarification in each community.

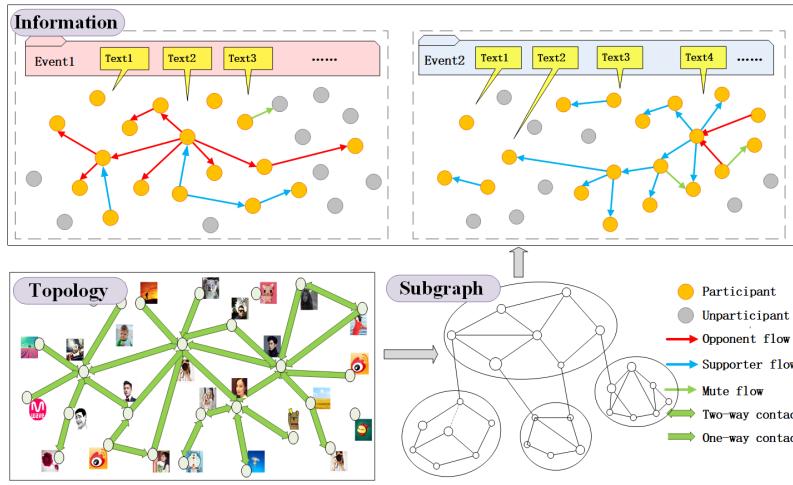


Figure 1. Reverse intervention of malicious information

Lots of measures of the strength of division of a network into communities have been proposed by experts in this field, such as conductance, normalized cut, cut ratio, triangle participation ratio (TPR), etc. [36, 37]. In this paper, we adopt the classic index of modularity proposed by Mark Newman [38] to measure the quality of the community partitioning algorithm, which compares the connection density between the original network and the reference network in the same community. The reference network is defined as a random network having the same degree sequence

as the original network. Suppose $\mathbb{A}$ is the adjacency matrix of a network, where $k_v(k_w)$ is the degree of node $v(w)$ and the total number of edges in the network is N. Then, the modularity is defined as follows:

$$Q = \frac{1}{2N} \sum_{v,w} \left[\mathbb{A}_{vw} - \frac{k_v k_w}{2N} \right] \delta(\mathbb{C}_v, \mathbb{C}_w) \tag{5}$$

where $\mathbb{C}_v$ is the community to which node v belongs. If node v and node w belong to the same community, i.e., $\mathbb{C}_v = \mathbb{C}_w$, then $\delta(\mathbb{C}_v, \mathbb{C}_w) = 1$; otherwise, $\delta(\mathbb{C}_v, \mathbb{C}_w) = 0$.

The higher the modularity, the better the community partitioning algorithm. As an important indicator to measure the quality of community division, modularity has been widely used [39].

In this paper, in order to evaluate the community characteristics of subgraphs, we use the definition of subgraph fitness function [40]:

$$Q(\mathbb{C}) = \frac{\sum in}{2N} - \left(\frac{\sum tot}{2N} \right)^2 \tag{6}$$

where $\sum in$ denotes the number of inner edges of a subgraph $\mathbb{C}$, and $\sum tot$ denotes the total number of edges connected to the nodes inside subgraph, including the edges inside the subgraph and the edges outside the subgraph. The subgraph fitness function measures the degree of "cohesion" of the edges in the subgraph. Obviously, if the community structure of $\mathbb{C}$ is more obvious, the value of Q is larger, and vice versa.

For a particular node, when its edges are mostly inside a subgraph, it is more likely to belong to the subgraph. When most of the edges point to the external nodes of the subgraph, it is unlikely to belong to the subgraph. Therefore, we define the evaluation function f of node n as follow:

$$f(n) = k_n^{\mathbb{C}} / k_n^G \tag{7}$$

where $k_n^{\mathbb{C}}$ is the degree of node n inside subgraph $\mathbb{C}$. k_n^G is the degree of node n in the entire network G.

Accordingly, we propose the subgraph partitioning algorithm based on community structure in this subsection. The basic idea of the algorithm is to randomly select nodes in the network, and then gradually expand the subgraph until the local subgraphs satisfying the given conditions are constructed. That is, the existing structure of the network is divided according to the local subgraph, and the specific process is shown in Algorithm 1.

The subgraph partitioning algorithm starts from a randomly selected set of nodes $\{V_1, V_2, \ldots, V_k\}$ and extends the subgraph along the edges. In order to ensure the community structure of the obtained subgraph, the nodes are first screened in the process of expansion. The algorithm selects the node with the highest evaluation function (most likely belongs to the subgraph) (Step 6), and judges whether adding the node to the current subgraph can increase the subgraph fitness function

Input: Online social network $G = (\mathbb{V}, \mathbb{E})$. The number of nodes initially selected (initial number of subgraphs) k, the maximum number of nodes m in each subgraph.
Output: Subgraphs $\mathbb{C}_i = \{\mathbb{C}_i, i = 1, 2, \dots\} \subset \mathbb{V}$.
Initialize: Randomly select k nodes V_i from the node set, let $\mathbb{C}_i = \{V_i\}, i = 1, 2, \dots, k$
1. **repeat**
2. **for each** $\mathbb{C}_i$ **do**
3. **if** $size(\mathbb{C}_i) < M$ **then**
4. $N_{\mathbb{C}_i} = neighbor(\mathbb{C}_i)$,$increase.\mathbb{C}_i = false$
5. **repeat**
6. $m = \arg\max_{m \in N_{\mathbb{C}_i}} f(m)$
7. **if** $Q(\mathbb{C}_i \cup \{m\}) > Q(\mathbb{C}_i)$ **then**
8. $\mathbb{C}_i = \mathbb{C}_i \cup m, increase.\mathbb{C}_i = true$
9. **end if**
10. $N_{\mathbb{C}_i} = N_{\mathbb{C}_i} - \{m\}$
11. **until** $size(N_{\mathbb{C}_i}) = 0$
12. **end if**
13. **end for**
14. **if** $\mathbb{C}_i \cap \mathbb{C}_j \neq \phi$ **and** $Q(\mathbb{C}_i \cup \mathbb{C}_j) > \max(Q(\mathbb{C}_i), Q(\mathbb{C}_j))$
15. **then** $\mathbb{C}_i = \mathbb{C}_i \cup \mathbb{C}_j, \mathbb{C}_j = \phi$
16. **end if**
17. **until** $size(\mathbb{C}_i) > M$ or $increase.\mathbb{C}_i = false$

Return: Subgraphs $\mathbb{C}_i, i = 1, 2, \dots$

Algorithm 1: Subgraph Partitioning Algorithm

(Steps 7–9). If yes, the node is added to the subgraph, otherwise the node is abandoned. Repeat the above steps until the subgraph reaches the specified scale m or the subgraph fitness stops growing (Step 17).

Due to the randomness of the initial node selection, subgraph initialized from different nodes may overlap. For overlapped subgraphs, the algorithm chooses to merge them according to whether the combined fitness function Q increases (Steps 14–16). Therefore, when selecting k, the subgraph merge situation that may occur should be considered. In order to suppress all possible sources of malicious information, this paper chooses k to be larger than the estimated number of sources in the network.

It is not necessary to divide the network into complete community structures to achieve perfect reverse intervention for malicious information. Therefore, in order to reduce the complexity of the algorithm, by setting the value of a reasonable subgraph size m, the algorithm stops when the subgraph has been extended to the expected size.

4.2 Reverse Intervention Algorithm Based on Jordan Center

Once the first phase is completed, we get a subgraph structure $\mathbb{C}_i = \{\mathbb{C}_i, i = 1, 2, \dots\} \subset \mathbb{V}$, where $\mathbb{C}_i, i = 1, 2, \dots$ is a disjoint subset, now we need to select nodes from these subgraphs to block or post clarification. For simplicity, we assume that the subgraphs $\mathbb{C}_i = \{\mathbb{C}_i, i = 1, 2, \dots\}$ are sorted in a non-incremental order with number of nodes (i.e., $|\mathbb{C}_1| \geq |\mathbb{C}_2| \geq |\mathbb{C}_3| \dots$). Since users exchange information more frequently with users in the same community, and nodes from different subgraphs typically have a small chance to spread malicious information (or clarification) to nodes in other subgraphs. Therefore, our problem is equivalent to finding nodes in each subgraph to control the infection rate of malicious information, so that the infection rate of the whole network can be lower than β.

This paper uses the Jordan Center to find the key node in each subgraph to control the propagation of malicious information. The specific process is summarized in Algorithm 2. The algorithm selects the most influential node in each subgraph according to the definition of Jordan center (Step 4), and determines if it is an infected node. If yes, we delete the node (block the account), otherwise we select it as the clarification publishing node (Steps 5–9). Repeat the above steps until the infection rate of the whole network is lower than β.

Input: Online social network $G = (\mathbb{V}, \mathbb{E})$, number of subgraph p, the infection rate of malicious information β
Output: Set $\mathbb{S} \subset \mathbb{V}$ makes $\frac{|\mathbb{I}(G)|}{|\mathbb{V}|} \leq \beta$
1. **Let** $\mathbb{S} = \phi$
2. **for** i **from** 1 to p **do** $\mathbb{S}_i = \phi$
3. **while** $\frac{|\mathbb{I}(\mathbb{C}_i)|}{|\mathbb{V}|} \leq \beta$ **do**
4. $v = \min_{s \in \mathbb{C}_i} \overline{d}(s, \mathbb{C}_i)$
5. **if** $v \in I(G)$ **then**
6. **block** and **delete** v;
7. **else**
8. $\mathbb{S}_i = \mathbb{S}_i \cup \{v\}$
9. **break**
10. **end if**
11. **end while**
12. **if** $\frac{|\mathbb{I}(G)|}{|\mathbb{V}|} \leq \beta$ **then**
13. **break**
14. **end if**
15. **end for**
Return $\mathbb{S}$

Algorithm 2: Reverse Intervention Algorithm

5 EXPERIMENT RESULTS

In this section, we used real large-scale networks to experimentally evaluate the performance of our proposed method in this paper. The datasets we used were downloaded from Stanford dataset collection (`http://snap.stanford.edu/data`). The first dataset is ego-Facebook, which contains 88 234 edges and 4 039 nodes, and the average clustering coefficient is 0.6055. The second dataset is cit-HepPh, which is a paper citation network, containing 34 546 nodes and 421 578 edges, and the average clustering coefficient is 0.2848. The experimental environment in which the algorithm ran is: processor Intel® Core™ i7-7500M @ 2.70 GHz, memory 8 GB, operating system Windows 10, programming language is Python.

We chose the following four benchmark methods to compare with our proposed algorithm:

1. Random: Randomly selected nodes in the network to block (or publishing clarification) until the infection rate met the requirements.
2. High-degree: The degree based heuristic algorithm, which selected nodes with the highest degree in the network to block (or publishing clarification) until the infection rate met the requirements.
3. Topcgo: A method proposed by Eftekhar et al. [41], which selected nodes with the greatest margin of information spread until the stopping criterion was met.
4. Greedy: The basic greedy algorithm proposed by Kempe et al. [31], which calculated the information dissemination range of each node under the IC model.

In all experiments, Monte Carlo simulation was implemented to estimate the effectiveness of the algorithms. That is, the results were averaged over 1 000 runs for consistency. We chose $p_{v,w}$ in the IC model as 0.25 for any node. And parameter β changed from 0.1 to 0.5. For each β, our proposed algorithm and the benchmark algorithms were independently implemented to get the number of required nodes to achieve the inhibitory effect.

We first consider the IC model using the two datasets. As depicted in Figures 2 and 3, the number of required nodes in our proposed method was highly competitive in comparison with those of others, especially in case that large number of nodes need to be immunized with the malicious information. In particular, when β was small ($\beta \in [0 \ldots 0.09]$), our proposed method did not performed as good as other methods. However, it became much better than other methods except Greedy algorithm as β gets larger. This is because the benchmark methods chose the candidate nodes within the whole network and our proposed method chose the candidate nodes based on community structure. When a small number of nodes were required, the Random, High-degree and Topcgo algorithm could easily select the influential nodes while our proposed algorithm must select nodes in each subgraph. In fact, influential nodes were often distributed in different subgraphs. As the number of required nodes increased, our proposed algorithm could effectively pick up the key nodes in each subgraph, which had influence to other nodes within the subgraph.

However, the benchmark methods had to choose these influential nodes in the whole network.

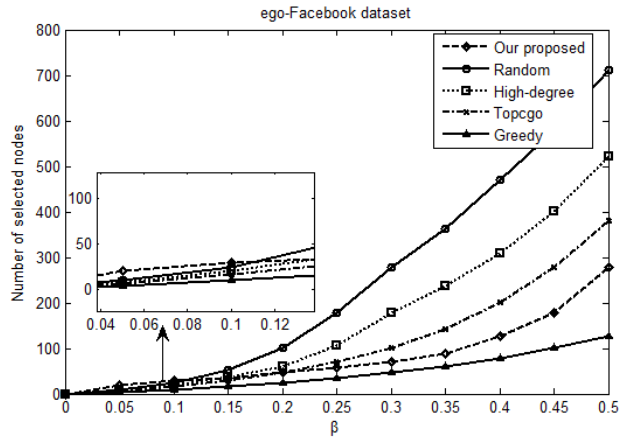


Figure 2. Nodes selected in different algorithms for ego-Facebook dataset

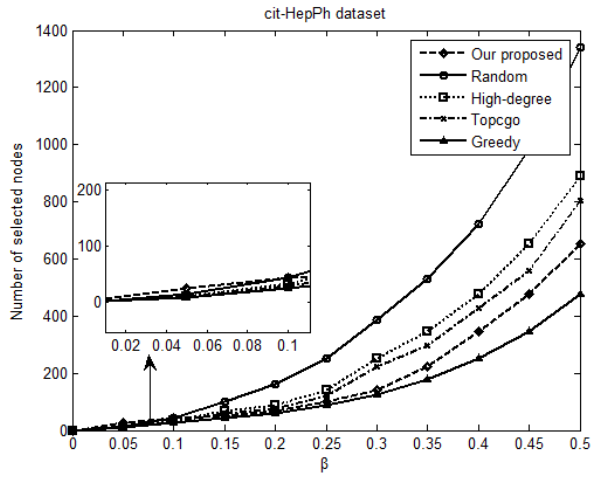


Figure 3. Nodes selected in different algorithms for cit-HepPh dataset

We next illustrate the difference when our proposed method is used under IC model and LT model. As shown in Figure 4, under the LT model, the proposed method could select a slightly fewer nodes to block or release clarification to achieve the desired effect than the IC model. This is because in Steps 5–9 of Algorithm 2, once the Jordan Center in the community was an infected node, we blocked it

and selected the node with the second largest influence (i.e., the second smallest eccentricity) to release clarification if it was not infected, and so on. In the IC model, the infected node had only one chance to affect its neighbor nodes. Whether to block this node or not had no effect to depress the propagation of the malicious information. But the infected nodes still had an impact under the LT model. It is worth noting that the time when to intervene was very important. This is beyond the scope of this paper and will be discussed in our future work.

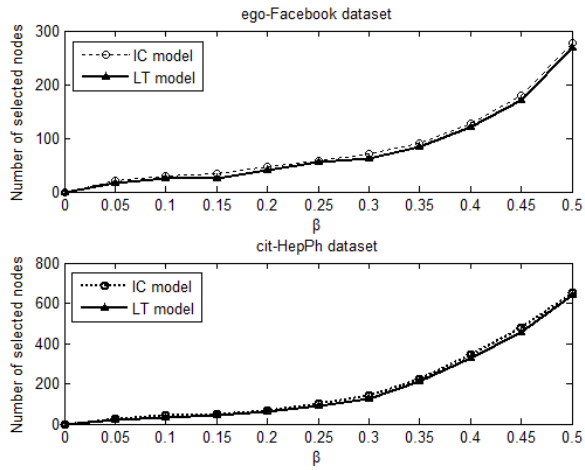


Figure 4. Comparison of the proposed algorithm under different propagation models

We finally evaluated the running time of our proposed algorithm and benchmark methods in Figure 5. Since the time consumption of the Random algorithm was very small, it is not depicted in this figure. As shown in the figure, although Greedy algorithm had the best performance (fewest nodes required to suppress the spread of malicious information), its time complexity was too high, especially on cit-HepPh dataset where it took more than 7500 seconds to meet the condition. Compared with other benchmark algorithms, our proposed algorithm had not only the advantage in intervention performance, but also had the advantage in time complexity. This is because we first divided the entire network into community structures, which could reduce much processing time during the influential node selection period. Therefore, our proposed algorithm could effectively impress the propagation of malicious information in a timely manner.

6 CONCLUSIONS

In this paper, we propose a reverse intervention algorithm based on subgraph partitioning, which impede the spread of malicious information from the perspective

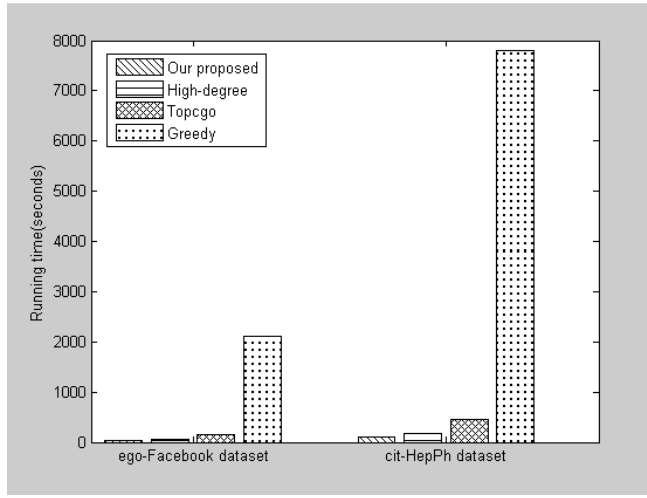


Figure 5. Time consumption of different algorithms

of network topology. Firstly, a subgraph partitioning method based on community structure is given. Secondly, a node blocking and clarification publishing algorithm based on the Jordan Center is proposed in the obtained subgraphs. Experiments on real-world networks including ego-Facebook and cit-HepPh show that the proposed algorithm can effectively suppress the spread of malicious information under a low time complexity.

Acknowlegement

The authors contributed equally to this study and share the first authorship. This work was supported by the National Key R & D Program of China (Grant No. 2017YFC0803700), the Beijing Natural Science Foundation Program (Grant No. 4184099), the National Natural Science Foundation of China (Grant No. 61771072), and the National Social Science Fund of China (Grant No. 17CXW014).

REFERENCES

[1] Kwon, S.—Cha, M.—Jung, K.—Chen, W.—Wang, Y.: Prominent Features of Rumor Propagation in Online Social Media. 2013 IEEE 13th International Conference on Data Mining (ICDM), IEEE, 2013, pp. 1103–1108, doi: 10.1109/ICDM.2013.61.

[2] Song, J.—Lee, S.—Kim, J.: Spam Filtering in Twitter Using Sender-Receiver Relationship. In: Sommer, R., Balzarotti, D., Maier, G. (Eds.): Recent Advances in Intrusion Detection (RAID 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6961, 2011, pp. 301–317, doi: 10.1007/978-3-642-23644-0_16.

[3] Fang, F.: Study on the Evolution of Public Opinion on Network of Unexpected Event. Ph.D. Dissertation, Huazhong University of Science and Technology, China, 2011.

[4] Lan, Y.—Deng, X.—Ma, M.: Construction of Public Opinion Security Evaluation Index System for Group Events. Information Exploration, Vol. 10, 2011, pp. 37–39.

[5] Hu, X.—Zhang, X.—Wu, J.—Deng, H.: Research for Invulnerability of Ad Hoc Network Topologies. Computer Technology and Development, Vol. 20, 2010, No. 1, pp. 185–188 (in Chinese).

[6] Tan, Y.—Wu, J.—Deng, H.: Progress in Invulnerability of Complex Networks. Journal of University of Shanghai for Science and Technology, Vol. 33, 2012, No. 6, pp. 653–668.

[7] Albert, R.—Jeong, H.—Barabási, A.-L.: Error and Attack Tolerance of Complex Networks. Nature, Vol. 406, 2000, No. 6794, pp. 378–382, doi: 10.1038/35019019.

[8] Cohen, R.—Erez, K.—ben-Avraham, D.—Havlin, S.: Resilience of the Internet to Random Breakdowns. Physical Review Letters, Vol. 85, 2000, No. 21, pp. 4626–4628, doi: 10.1103/PhysRevLett.85.4626.

[9] Callaway, D. S.—Newman, M. E. J.—Strogatz, S. H.—Watts, D. J.: Network Robustness and Fragility: Percolation on Random Graphs. Physical Review Letters, Vol. 85, 2000, No. 25, pp. 5468–5471, doi: 10.1103/PhysRevLett.85.5468.

[10] Liu, Y.: Network Invulnerable Dynamic Evolution Model Based on HOT Theory. Computer Engineering, Vol. 39, 2013, No. 1, pp. 97–101, doi: 10.3969/j.issn.1000-3428.2013.01.021 (in Chinese).

[11] Wu, J.—Chen, Z.—Zhao, M.: Information Cache Management and Data Transmission Algorithm in Opportunistic Social Networks. Wireless Networks, Vol. 25, 2019, No. 6, pp. 2977–2988, doi: 10.1007/s11276-018-1691-6.

[12] Wu, J.—Chen, Z.—Zhao, M.: Weight Distribution and Community Reconstitution Based on Communities Communications in Social Opportunistic Networks. Peer-to-Peer Networking and Applications, Vol. 12, 2019, No. 1, pp. 158–166, doi: 10.1007/s12083-018-0649-x.

[13] Wu, J.—Chen, Z.: Sensor Communication Area and Node Extend Routing Algorithm in Opportunistic Networks. Peer-to-Peer Networking and Applications, Vol. 11, 2018, No. 1, pp. 90–100, doi: 10.1007/s12083-016-0526-4.

[14] Luan, W.—Liu, G.—Jiang, C.—Qi, L.: Partition-Based Collaborative Tensor Factorization for POI Recommendation. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 3, pp. 437–446, doi: 10.1109/JAS.2017.7510538.

[15] Luan, W.—Liu, G.—Jiang, C.—Zhou, M.: MPTR: A Maximal-Marginal-Relevance-Based Personalized Trip Recommendation Method. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, 2018, No. 11, pp. 3461–3474, doi: 10.1109/TITS.2017.2781138.

[16] Carmi, S.—Havlin, S.—Kirkpatrick, S.—Shavitt, Y.—Shir, E.: A Model of Internet Topology Using K-Shell Decomposition. Proceedings of the National Academy of Sciences of the United States of America (PNAS), Vol. 104, 2007, No. 27, pp. 11150–11154, doi: 10.1073/pnas.0701175104.

[17] Kleinberg, J. M.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM (JACM), Vol. 46, 1999, No. 5, pp. 604–632, doi: 10.1145/324133.324140.

[18] Page, L.—Brin, S.—Motwani, R.—Winograd, T.: The Pagerank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford InfoLab, 1999. Available at: `http://ilpubs.stanford.edu:8090/422/`.

[19] Hu, J.—Wang, B.—Lee, D.: Evaluating Node Importance with Multi-Criteria. Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications and International Conference on Cyber, Physical and Social Computing, IEEE Computer Society, 2010, pp. 792–797, doi: 10.1109/GreenCom-CPSCom.2010.26.

[20] Qi, X.—Duval, R. D.—Christensen, K.—Fuller, E.—Spahiu, A.—Wu, Q.—Wu, Y.—Tang, W.—Zhang, C.: Terrorist Networks, Network Energy and Node Removal: A New Measure of Centrality Based on Laplacian Energy. Social Networking, Vol. 2, 2013, No. 1, pp. 19–31, doi: 10.4236/sn.2013.21003.

[21] Burt, R. S.: Structural Holes: The Social Structure of Competition. Harvard University Press, 2010, pp. 150–188.

[22] Chen, Y.—Hu, A.—Hu, X.: Evaluation Method for Node Importance in Communication Networks. Journal of China Institute of Communications, Vol. 25, 2004, No. 8, pp. 129–134 (in Chinese).

[23] Zhang, Y.—Liu, Y.—Xu, K. et al.: Evaluation Method for Node Importance Based on Mutual Information in Complex Networks. Computer Science, Vol. 38, 2011, No. 6, pp. 88–89 (in Chinese).

[24] Tan, Y.—Wu, J.—Deng, H.: Evaluation Method for Node Importance Based on Node Contraction in Complex Networks. System Engineering – Theory and Practice, No. 11, 2006, pp. 79–84 (in Chinese).

[25] Fan, L.—Lu, Z.—Wu, W.—Thuraisingham, B.—Ma, H.—Bi, Y.: Least Cost Rumor Blocking in Social Networks. 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS), IEEE, 2013, pp. 540–549, doi: 10.1109/ICDCS.2013.34.

[26] Wan, P.—Wang, X.—Wang, X.—Wang, L.—Lin, Y.—Zhao, W.: Intervening Coupling Diffusion of Competitive Information in Online Social Networks. IEEE Transactions on Knowledge and Data Engineering, 2019, doi: 10.1109/TKDE.2019.2954901.

[27] Wen, S.—Jiang, J.—Xiang, Y.—Yu, S.—Zhou, W.—Jia, W.: To Shut Them Up or to Clarify: Restraining the Spread of Rumors in Online Social Networks. IEEE Transactions on Parallel and Distributed Systems, Vol. 25, 2014, No. 12, pp. 3306–3316, doi: 10.1109/TPDS.2013.2297115.

[28] Bao, Y.—Niu, Y.—Yi, C.—Xue, Y.: Effective Immunization Strategy for Rumor Propagation Based on Maximum Spanning Tree. 2014 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2014, pp. 11–15, doi: 10.1109/ICCNC.2014.6785296.

[29] Bao, Y.—Yi, C.—Xue, Y.—Dong, Y.: A New Rumor Propagation Model and Control Strategy on Social Networks. Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), 2013, pp. 1472–1473, doi: 10.1109/ASONAM.2013.6785909.

[30] Bhattacharya, S.—Tran, H.—Srinivasan, P.—Suls, J.: Belief Surveillance with Twitter. Proceedings of the 4th Annual ACM Web Science Conference (WebSci '12), ACM, 2012, pp. 43–46, doi: 10.1145/2380718.2380724.

[31] Kempe, D.—Kleinberg, J.—Tardos, É.: Maximizing the Spread of Influence Through a Social Network. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03), 2003, pp. 137–146, doi: 10.1145/956750.956769.

[32] Hage, P.—Harary, F.: Eccentricity and Centrality in Networks. Social Networks, Vol. 17, 1995, No. 1, pp. 57–63, doi: 10.1016/0378-8733(94)00248-9.

[33] Dekker, A. H.: Centrality in Social Networks: Theoretical and Simulation Approaches. Proceedings of the Simulation Technology and Training Conference (SimTecT) 2008, Melbourne, Australia, 2008, 6 pp.

[34] Nguyen, N. P.—Dinh, T. N.—Xuan, Y.—Thai, M. T.: Adaptive Algorithms for Detecting Community Structure in Dynamic Social Networks. 2011 Proceedings IEEE INFOCOM, Shanghai, China, IEEE, 2011, pp. 2282–2290, doi: 10.1109/INFCOM.2011.5935045.

[35] Nguyen, N. P.—Dinh, T. N.—Tokala, S.—Thai, M. T.: Overlapping Communities in Dynamic Networks: Their Detection and Mobile Applications. Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom 2011), Las Vegas, USA, 2011, pp. 85–96, doi: 10.1145/2030613.2030624.

[36] Yang, J.—Leskovec, J.: Defining and Evaluating Network Communities Based on Ground-Truth. Knowledge and Information Systems, Vol. 42, 2015, No. 1, pp. 181–213, doi: 10.1007/s10115-013-0693-z.

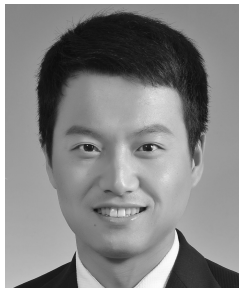
[37] Fortunato, S.: Community Detection in Graphs. Physics Reports, Vol. 486, 2010, No. 3–5, pp. 75–174, doi: 10.1016/j.physrep.2009.11.002.

[38] Newman, M. E. J.: Fast Algorithm for Detecting Community Structure in Networks. Physical Review E, Vol. 69, 2004, No. 6, Art. No. 066133, 5 pp., doi: 10.1103/PhysRevE.69.066133.

[39] Blondel, V. D.—Guillaume, J.-L.—Lambiotte, R.—Lefebvre, E.: Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics: Theory and Experiment, Vol. 2008, 2008, Art. No. P10008, 12 pp., doi: 10.1088/1742-5468/2008/10/P10008.

[40] Zhang, X.—Zhang, Y.—Lv, T.—Fu, S.—Zhang, B.: A Multi-Diffusion Source-Localization Method for Online Social Networks Based on Sub-Graph Extraction. Scientia Sinica Informationis, Vol. 46, 2016, No. 4, pp. 496–510, doi: 10.1360/N112015-00190.

[41] Eftekhar, M.—Ganjali, Y.—Koudas, N.: Information Cascade at Group Scale. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), ACM, 2013, pp. 401–409, doi: 10.1145/2487575.2487683.



Deyu Yuan received his Ph.D. degree from the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), China in 2015. He is now working as Lecturer at the College of Police Information Engineering and Cyber Security, People's Public Security University of China (PPSUC). His research interests include cyber security and complex networks. He has published over 20 papers in journals and conferences such as China Communications, HCC 2014.

Haichun Sun received her Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2015. She is currently Assistant Professor at the Police Information Engineering and Cyber Security, People's Public Security University of China, Beijing, China. She is a member of Professional Committee of Internet Information Service of the Chinese Association of Automation. Her current research interests include information service, Petri nets, and service-oriented computing. She has published over 10 papers in journals and conferences such as IEEE Transactions on Systems, Man, and Cybernetics, WISE 2014.

Computing and Informatics, Vol. 39, 2020, 174–192, doi: 10.31577/cai_2020_1-2_174

A METHOD FOR LEARNING A PETRI NET MODEL BASED ON REGION THEORY

Jiao Li, Ru Yang, Zhijun Ding

The MOE Key Laboratory of Embedded System and Service Computation
Tongji University
Shanghai, 201804, China
e-mail: {li_jiao, yangru}@tongji.edu.cn, zhijun_ding@outlook.com

Meiqin Pan*

School of Business and Management
Shanghai International Studies University
Shanghai, 200083, China
e-mail: panmqin@sina.com

Abstract. The deployment of robots in real life applications is growing. For better control and analysis of robots, modeling and learning are the hot topics in the field. This paper proposes a method for learning a Petri net model from the limited attempts of robots. The method can supplement the information getting from robot system and then derive an accurate Petri net based on region theory accordingly. We take the building block world as an example to illustrate the presented method and prove the rationality of the method by two theorems. Moreover, the method described in this paper has been implemented by a program and tested on a set of examples. The results of experiments show that our algorithm is feasible and effective.

Keywords: Petri net, robot model, robot learning, region theory, Petri net synthesis

Mathematics Subject Classification 2010: 93A30

* Corresponding author

1 INTRODUCTION

Robotics systems and techniques which appeared during the recent years have achieved astonishing development, and they not only facilitate humans' lives but also replace humans' work in some difficult situations. With the continuous expansion of the field of robot applications, higher requirements are needed for the safety, correctness and reliability of robots. In order to more effectively control a robot system and verify the system properties, it is necessary to build a model for the robot system.

In the field of robotics, many researchers have worked on modeling. Desai [1] proposed a new modeling method that can control multiple teams of mobile robots navigating in a terrain with obstacles, while maintaining a desired formation and changing formations when required. Wieber et al. [2] studied the modeling method of legged robots, and used the model to generate and control the dynamic motions, as well as analyze the stability of the robot. However, in some situations, deliberative planning or pre-programming to achieve tasks will not be always possible. Hence, there is a growing research interest in imbuing robots not only with the capability of perception and planning but also of learning [3]. According to current state of the art of robot learning, most of the successful results presented in the literature are applied by machine learning. There are many different implementation methods, such as reinforcement learning, artificial neural network and evolutionary techniques [4, 5, 6]. However, there are few studies on the modeling and learning of robots via formal methods.

Petri nets (PNs) [7] are a powerful formal modeling tool, which have advantages in the intuitiveness of its graphical modeling and the rigor of its analytical theory. Especially, they have the ability to describe the complex logical relationships between systems or process activities, such as concurrency, competition, synchronization, and order. Moreover, there are many PN modeling tools [8] which provide the functions of establishing, modifying, storing, and dynamic simulation that can be used to analyze and valid the properties of the PNs. Therefore, they have been widely applied in various fields, including the field of robotics. Lima et al. [9] introduced distinct Petri net types to model robotic tasks from different views of the robotic task model. Ziparo et al. [10] presented a language (Petri Net Plans) based on PNs, which allows for intuitive and effective robot and multi-robot behavior design. Chao et al. [11] developed a system for multimodal collaboration based on a timed Petri net representation, and implemented action interruptions in reciprocal interaction within the system. In these applications, PNs were mostly created manually rather than automatically. Chang et al. [12] proposed a learning method that automatically creates PNs from observation of human demonstrations to model the underlying structure of tasks. Different from most of the existing methods, this work enables PNs to be created automatically. But the operation sequences of imitation need to be designed artificially. It is our hope that the robot can learn a PN model in a limited number of task-oriented attempts without manual planning.

The attempts of the robot can be regarded as the system behavior. There are two major approaches to obtain a PN model from the information of system behavior. One approach is process mining technology [13, 14]. Although a PN model can always be gained by a process mining technology, the obtained model is not necessarily consistent with the actual model. The other way to transform the behaviors to a structure description model (PN) is related to the PN synthesis problem [15], whose method is mainly based on the region theory [16]. The original goal of PN synthesis is to construct an elementary PN according to a given transition system and test whether the reachability graph of the PN is isomorphic to the transition system. If it is isomorphic, such a PN is constructed. Nowadays, there are many extensions in this field, such as changing the transition system into formal languages and execution traces or changing the target from elementary PNs to Place/Transition nets. Several tools for synthesis have already existed, like petrify [17], genet [18], synet [19] and apt [20]. By this method, we can convert the effective attempts of a robot into an accurate PN model that describes the operation process of the robot in a compact form. At the same time, we can learn some rules from the limited attempts to enrich the known information so as to obtain a more complete model.

This paper proposes a PN model generation algorithm based on the region theory and gives two theorems as well as proofs to guarantee the rationality of the method. Also, to illustrate how to obtain a more complete PN model automatically within the robot's limited attempts, the problem of the robot in the building block world is taken as an example. The main contributions are as follows:

1. An automatic generation and learning scheme of robot model from the robot's limited attempts based on PN is given. It provides a new idea for robot model learning by using PN.
2. Two theorems are proved, which are the theoretical basis of this paper. One theorem guarantees the accuracy of the model generated from a transition system and the other reveals the rationality of adding information so that any transition system getting from a robot system can generate a PN model.
3. A PN model generation algorithm that provides an operational method for the scheme is recommended. It transforms the behaviors of the robot into a PN model and rationally supplements the information based on the region theory and two above theorems. By this means, the purpose of automatic model generation and learning is achieved.

The next section describes a classical problem in building block world and expounds the problem to be solved in this paper from the perspective of robot learning and control. Section 3 briefly reviews the basics of PN and some definitions related to the proposed approach. After proving two theorems, the PN model generation algorithm is given. In order to confirm the feasibility of the proposed method, some examples are shown in Section 4. Finally, conclusions and outlooks are presented in Section 5.

2 MOTIVATING EXAMPLE

In some situations, due to the uncertainty of the environment or difficulty of comprehensive analysis, it is hard to build accurate models manually. Taking the classical problem of building block world as an example, it is difficult to consider all the operation sequences of the robot and obtain a complete model manually. Thus, it requires a method which can generate a model automatically as well as learn some information from the existing information. There is no doubt that we can control robots better with a more complete model.

A building block world scene is as follows: a number of blocks on a table are placed and a robot is asked to change the initial state of the blocks to the target state. The robot has a mechanical arm (hand) and just can perform the specified actions which are shown in Table 1.

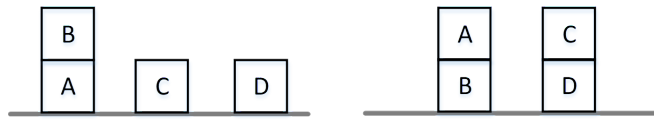
Action	**Explanation**	**Precondition**
unstack(A, B)	Pick up building block A from building block B.	Building block A is stacked on building block B; there is no other building block on building block A; the robot's hand is empty.
putdown(A)	Place building block A on the table.	Building block A is in robot's hand.
pickup(A)	Pick up building block A from the table.	Building block A is on the table; there is no other building block on building block A; the robot's hand is empty.
stack(A, B)	Place building block A on building block B.	There is no other building block on building block B; building block A is in robot's hand.

Table 1. The actions of robot

We define that if the robot executes the action "unstack(A, B)", it must put the building block A on the table before taking another action. That is to say, it is not allowed to place the building block A on another building block after executing the action "unstack(A, B)". So we can combine the action "unstack(A, B)" and the action "putdown(A)" into one operation, that is, "unstack(A, B)-putdown(A)" is an atomic operation. In addition, we suppose the robot can only pick up one building block at a time which means robot needs to put down block in hand before executing another action, so "pickup(A)-stack(A, B)" and "pickup(A)-putdown(A)" are also atomic operations. Because the atomic operation "pickup(A)-putdown(A)" doesn't change the state of the building block, we consider to ignore it. In summary, the operations that the robot can perform are "unstack(A, B)-putdown(A)" and "pickup(A)-stack(A, B)". For ease of writing, we will record the first operation as USPD(A, B) and the second one as PUS(A, B).

For states of this scene, we only consider the upper and lower relative positions of the blocks and the position of the blocks and the table, regardless of the left and

right relative positions of the blocks. For example, Figure 1 shows the initial state and target state of the building blocks that we assume. As in the target state, we only require the building block A is on the building block B, the building block C is on the building block D and the building blocks B and D are on the table, without concerning about the left and right relative positions of the building block AB and the building block CD. If multiple operations can be executed in a situation, robot can perform an operation at will until the state of all blocks is consistent with the target state.



a) b)

Figure 1. a) Initial state and b) target state of the building blocks

Then, based on the above scene assumptions, how can robots learn a PN model by limited attempts without human intervention? This question is answered in this paper.

3 METHODS

In this section, we first give the basic concepts of PN and PN synthesis, which are derived from [15, 21, 22, 23]. Next, we present two theorems which are the theoretical foundation of the proposed method. Finally, based on the definitions and theorems, we put forward the algorithm for generating PN models.

3.1 Preliminaries

Definition 1. A net is a quad $N = (P, T, F, W)$, where P is a finite set of places, T is a finite set of transitions such that $P \neq \varnothing$, $T \neq \varnothing$, $P \cap T = \varnothing$, $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, and W is a weight function such that $W(x, y) \in \mathbb{N}^+$ (here $\mathbb{N}^+ = \{1, 2, 3, \dots\}$) if $(x, y) \in F$ and $W(x, y) = 0$ if $(x, y) \notin F$.

The marking of a net is a function $M : P \to \mathbb{N}$ (here $\mathbb{N} = \{0, 1, 2, 3, \dots\}$). It is represented by a multiset expression or a $|P|$-vector $(M(p_1), \dots, M(p_{|P|}))^T$, where $M(p)$ is the number of tokens in place $p \in P$. A PN $PN = (N, M_0)$ is a net N with an initial marking M_0.

A transition $t \in T$ is said to be enabled at marking M, which is denoted as $M[t >$, if $\forall p \in P$, $M(p) \geq W(p, t)$. Firing an enabled transition t results in changing M into M', represented by $M[t > M'$, where $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$. A sequence of transitions $\sigma = t_1 t_2 \dots t_k$ is a firing sequence if there exists a sequence of markings such that $M[t_1 > M_1[t_2 > \dots M_{k-1}[k > M_k$, it

can be written as $M[\sigma > M_k$, and M_k is said to be reachable from M by firing σ. The reachability set $R(M)$ is a set of all markings reachable from M.

The reachability graph of PN is a directed graph with the set of vertices $R(M_0)$, and arcs $\{(M, t, M') | M, M' \in R(M_0) \wedge M[t > M'\}$.

Definition 2. A place $p \in P$ is said to be bounded or K-bounded if $\forall M \in R(M_0)$, $M(p) \leq K$, where $K \in \mathbb{N}^+$ (here $\mathbb{N}^+ = \{1, 2, 3, \dots\}$). A PN PN is said to be bounded if its every place is bounded. If $K(PN) = \max\{K(p) | p \in P\} = 1$, PN is a safe PN.

Here, we only consider the PN with an arc weight of 1, so the weight function W of the PN can be omitted, represented by $PN = (P, T, F, M_0)$. Unless otherwise stated, the PNs referred in this paper are all safe PNs.

Definition 3. A transition system (S, E, Δ) consists of a set of states S, a set of events E, and a set of transitions $\Delta \subset S \times E \times S$. An initialized transition system $TS = (S, E, \Delta, S_0)$ consists of a transition system (S, E, Δ) and an initial state $s_0 \in S$.

An event e is enabled in a state s, denoted by $s \xrightarrow{e}$, if there is a state s' such that $(s, e, s') \in \Delta$. This situation is written as $s \xrightarrow{e} s'$ and means that state s' is reachable from state s through the execution of event e. The definitions of enabledness and of the reachability relation are extended as usual to event sequences (or directed paths) $\sigma \in E^* : s \xrightarrow{\varepsilon}$ and $s \xrightarrow{\varepsilon} s'$ are always true; $s \xrightarrow{\sigma e}$ ($s \xrightarrow{\sigma e} s'$) iff there is a state s'' with $s \xrightarrow{\sigma} s''$ and $s'' \xrightarrow{e}$ ($s'' \xrightarrow{e} s'$, respectively). A state s' is reachable from a state s if there is an event sequence σ such that $s \xrightarrow{\sigma} s'$. A state s' is reachable if it is reachable from state s_0. By $s^{\rightarrow}$, we denote the set of states reachable from state s.

Definition 4. An initialized transition system $TS = (S, E, \Delta, s_0)$ is called finite if S and E (hence also Δ) are finite sets. It is deterministic if for any reachable state s, s', s'' and event e, $s \xrightarrow{e} s'$ and $s \xrightarrow{e} s''$ implies $s' = s''$ and it is totally reachable if $S = s_0^{\rightarrow}$ and $\forall e \in E : \exists s \in s_0^{\rightarrow} : s \xrightarrow{e}$.

A transition system characterizes the migration process of the system states, which can be either artificially designed or actually obtained. It should be noted that the transition systems involved in this paper are all gained by robot's actual attempts.

Definition 5. Two $TS_1 = (S_1, E, \Delta_1, s_{01})$ and $TS_2 = (S_2, E, \Delta_2, s_{02})$ over the same set of evens E are isomorphic if there is a bijection ζ: $S_1 \to S_2$ with $\zeta(s_{01}) = s_{02}$ and $(s, t, s') \in \Delta_1 \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \Delta_2$, for all $s, s' \in S_1$.

The reachability graph of a PN PN can be seen as an initialized transition system. If there is a PN PN whose reachability graph is isomorphic to a given initialized transition system TS, then we will say that PN solves TS [23].

Definition 6. A region of an initialized transition system $TS = (S, E, \Delta, s_0)$ is a triple $(\mathbb{R}, \mathbb{B}, \mathbb{F}) \in N^S \times N^E \times N^E$ such that the following holds:

$$\forall s \xrightarrow{e} s' \in \Delta : \mathbb{R}(s) \geqslant \mathbb{B}(e) \wedge \mathbb{R}(s') = \mathbb{R}(s) - \mathbb{B}(e) + \mathbb{F}(e).$$

In the above formula, the first condition states that no transition in the initialized transition system may be prevented, and the second condition enforces consistency between $\mathbb{R}$, $\mathbb{B}$, and $\mathbb{F}$. Intuitively, this describes a possible place in a PN generating TS where $\mathbb{B}(e)$ and $\mathbb{F}(e)$ describe the number of tokens consumed and produced, respectively, by a transition $e \in E$, and $\mathbb{R}(s)$ is the number of tokens on this place in state $s \in S$ [23].

For every region $(\mathbb{R}, \mathbb{B}, \mathbb{F})$ if $s \xrightarrow{\sigma} s'$ for some s, $s' \in S$ and $\sigma = e_{a1}e_{a2}\ldots e_{ak} \in E^*$, then $\mathbb{R}(s') = \mathbb{R}(s) + \sum_{i=1}^{k}(\mathbb{F}(e_{ai}) - \mathbb{B}(e_{ai}))$. Since we are assuming that the TS is totally reachable, $\mathbb{R}$ is thus fully determined by $\mathbb{R}(s_0)$ via $\mathbb{R}(s) = \mathbb{R}(s_0) + \sum_{i=1}^{n} \psi_s(e_i) \cdot (\mathbb{F}(e_i) - \mathbb{B}(e_i))$, where $\psi_s(e_i)$ is the number of times that e_i occurs in σ when $s_0 \xrightarrow{\sigma} s$. We identify a region $\rho = (\mathbb{R}, \mathbb{B}, \mathbb{F})$ with a vector $\rho \in N^{1+2n}$:

$$\rho = (\rho_0, \ldots, \rho_{2n}) = (\mathbb{R}(s_0), \mathbb{B}(e_1), \ldots, \mathbb{B}(e_n), \mathbb{F}(e_1), \ldots, \mathbb{F}(e_n)).$$

The function that reconstructs the value $\mathbb{R}(s)$ for a state $s \in S$ from such a vector is given by $\text{tokens}(\rho, s) = \rho_0 + \sum_{i=1}^{n} \psi_s(e_i) \cdot (\rho_{n+i} - \rho_i)$.

Definition 7. For a region set R of an initialized transition system $TS = (S, E, \Delta, s_0)$, the corresponding PN $PN = (P, T, F, M_0)$ has $P = R$, $T = E$ and for each $\rho = (\mathbb{R}_\rho, \mathbb{B}_\rho, \mathbb{F}_\rho) \in R$ defines $F(\rho, e) = \mathbb{B}_\rho(e)$, $F(e, \rho) = \mathbb{F}_\rho(e)$ and $M_0(\rho) = \mathbb{R}(s_0)$. If the reachability graph of the corresponding PN is isomorphic to the TS, i.e., TS is isomorphic to the reachability graph of the net system synthesized from R, we will say that the region set R solves TS.

For example, a region set $R = \{(1,0,1,0,1,0,0,0), (1,1,0,0,1,1,1,0)\}$ can be found in the transition system shown in Figure 2 a). For each region in R, we can define a place and the flow relationship between the place and transitions in the PN model. As shown in Figure 2 b), the region $\rho_1 = (1,0,1,0,1,0,0,0)$ corresponds to place p_1, and the region $\rho_2 = (1,1,0,0,1,1,1,0)$ corresponds to place p_2. Moreover, it can be seen that the reachability graph of PN_1 shown in Figure 2 c) is isomorphic to transition system TS_1, that is, the region set R solves TS_1.

Remark 1. It is a hope that the number of places in the PN is as small as possible, so it leads to the emergence of sink transitions. The sink transitions will not affect the normal behavior of the PN.

Definition 8. A state separation problem $\text{SSP}(s, s')$ is a set of two states $\{s, s'\} \subseteq S$ with $s \neq s'$ that must be distinguishable and it is solved by a region ρ with $\mathbb{R}_\rho(s) \neq \mathbb{R}_\rho(s')$. The corresponding predicate is $\text{SSP}(\rho, s, s') := (\text{tokens}(\rho, s) \neq \text{tokens}(\rho, s'))$. A counterexample is given in the Figure 3 a) which shows an initialized transition

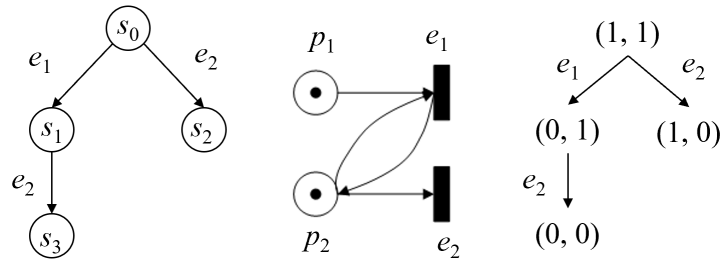


Figure 2. a) A transition system TS_1, b) the corresponding PN PN_1, and c) the reachability graph of the PN_1

system in which states s_3 and s_4 cannot be separated by any region. If all the state separation problems of a TS can be solved, we call this TS satisfies the state separation property.

An event/state separation problem $\mathrm{ESSP}(s, e)$ is a pair $(s, e) \in S \times E$ with $\neg(s \xrightarrow{e})$. This problem is solved by a region $(\mathbb{R}_\rho, \mathbb{B}_\rho, \mathbb{F}_\rho)$ iff $\mathbb{R}_\rho(s) < \mathbb{B}_\rho(e)$, which means that event e is prevented in state s. This is expressed by the predicate $\mathrm{ESSP}(\rho, s, e_i) := (\mathrm{tokens}(\rho, s) < \rho_i)$. One of its counterexamples is shown in the Figure 3 b). It demonstrates that event e_3 cannot be separated from state s_1 by any region in the initialized transition system. If all the event/state separation problems of a TS can be solved, we call this TS satisfies the event/state separation property.

The set of all separation problems of TS is called SP. For readability, given any kind of separation problem $pr \in \mathrm{SP}$, we define $\mathrm{SP}(\rho, pr)$:

$$\mathrm{SP}(\rho, pr) := \begin{cases} \mathrm{SSP}(\rho, s, s') = (\mathrm{tokens}(\rho, s) \neq tokens(\rho, s')), & \text{if } pr = \mathrm{SSP}(s, s'), \\ \mathrm{ESSP}(\rho, s, e_i) = (\mathrm{tokens}(\rho, s) < r_i), & \text{if } pr = \mathrm{ESSP}(s, e_i). \end{cases}$$

3.2 Relevant Theorems

According to the above definitions, we present the following two theorems which provide theoretical support for the subsequent algorithm. The first theorem states that the model generated by the algorithm is accurate. The second guarantees that any transition system can generate a PN model by supplementing the transitions if it failed, and the supplement of the information is reasonable.

Theorem 1. If there is a region set R of the initialized transition system $TS = (S, E, \Delta, s_0)$ that can solve all separation problems SP in the TS and satisfies $\forall p \in R$, $\mathrm{tokens}(\rho, s) \leq 1$, where $s \in S$, then the corresponding PN is safe and its reachability graph is isomorphic to the TS.

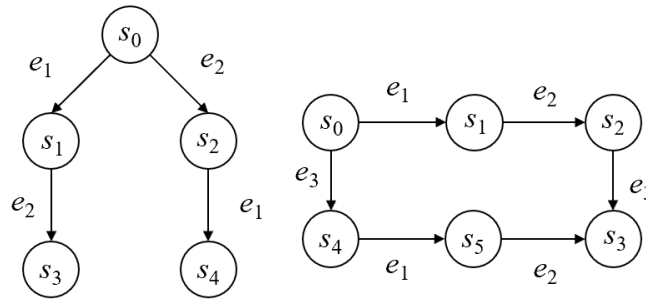


Figure 3. a) A transition system where state separation fails, and b) a transition system where event/state separation fails

Proof. Let $SN(TS)$ be the PN obtained by $TS = (S, E, \Delta, s_0)$, and its reachability graph is $RG(SN(TS))$, which is denoted as $TS_2 = (S_2, E_2, \Delta_2, s_0 2)$. Assuming that TS_2 is not isomorphic to TS, then either the event sets of TS_2 and TS are different or there is no bijection $\zeta : S \to S_2$ with $\zeta(s_0) = s_0 2$ and $(s, t, s') \in \Delta \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \Delta_2$, for all $s, s' \in S$. In view of synthesis method, the event sets of TS_2 and TS are the same. Then we consider the second assumption, that is, $\exists s, s' \in TS_2$, $\zeta(s_0) \neq s_0 2$ or $(s, t, s') \in \Delta_1 \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \Delta_2$ is not satisfied. Because the region set R can solve all the separation problems SP in TS, by Definition 6, the region in R considers all the constraints of $s \xrightarrow{e} s' \in \Delta$ and $\neg(s \xrightarrow{e})$ for all $s, s' \in S$ and $e \in E$, there must exist a bijection $\zeta : S \to S_2$ with $\zeta(s_0) = s_0 2$ and $(s, t, s') \in \Delta \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \Delta_2$, for all $s, s' \in S$, which opposites to the assumption. In addition, $\forall s \in S$, $\rho \in R$, $\text{tokens}(\rho, s) \leq 1$ conforms to the definition of safe PN. Hence, the proof is complete. □

This theorem is proposed based on the summary of the existing conclusions and here we give its proof. For a more detailed introduction of region theory, please refer to [16].

Theorem 2. Let $TS = (S, E, \Delta, s_0)$ be an initialized transition system and it satisfies the state separation property. For any event/state separation problem $\text{ESSP}(s, e)$ in the TS, if there is no region to solve the problem, then $s \xrightarrow{e}$ or $s \xrightarrow{e} s'$ (s' is a new state) can be added to the TS, and this supplement is reasonable.

Proof. For an event/state separation problem $\text{ESSP}(s, e)$ in TS, i.e. $\neg(s \xrightarrow{e})$, if the problem can be solved by a region ρ, then ρ satisfies $\mathbb{R}_\rho(s) < \mathbb{B}_\rho(e)$. But such region does not exist, that is to say, all regions ρ satisfy $\mathbb{R}_\rho(s) \geq \mathbb{B}_\rho(e)$, which means $s \xrightarrow{e}$ by Definition 6. Since $\mathbb{R}_\rho(s') = \mathbb{R}(s) - \mathbb{B}(e) + \mathbb{F}(e)$, state s' can be calculated. If s' does not exist in the state set S, it is necessary to add it into S. It can be seen from the above analysis that the event set is not changed and the supplemental state can

be obtained by occurring the event actually, what complies with the rules of the system. Hence, the supplement is reasonable. □

In Theorem 2, we prove that it is reasonable to add some transitions to the TS when facing the failure of PN generation. Corresponding to the real scene, the other information gained according to the known one by the robot is consistent with the actual. Obviously, this is a manifestation of learning.

3.3 Generation Algorithm for Petri Net Model

As stated above, we can construct a PN according to an initialized transition system TS based on the region theory. If the regions of the TS satisfy the certain conditions, a PN whose reachability graph is isomorphic to the initial transition can be generated. In a robot scene, an initialized transition system can be obtained by the records of execution sequences and state changes. Then, we can use it to produce a PN model automatically. If it failed to generate a PN model, this work gains some information from the known one and adds them to the initial transition system, which shows the robot has the ability to learn. Here we introduce the PN model generation algorithm.

Algorithm 1 Petri Net Model Generation Algorithm

Input: an initialized transition system $TS = (S, E, \Delta, s_0)$
Output: a PN Model PN
1) Let the region set Π and unresolved separation problem set Ξ be $\varnothing$(empty);
2) For each separation problem $pr \in$ SP in TS, do
3) If find a region ρ can solve pr and $\forall s \in S$, tokens$(\rho, s) \leq 1$, then
4) Put ρ into the set Π;
5) Else
6) Put pr into the set Ξ;
7) End if
8) End for
9) If unresolved separation problem set Ξ is not empty, then
10) For each separation problem ESSP(s_i, e_i) in Ξ, do
11) Calculate state s_i' which is reachable from s_i through the execution of e_i
12) If state s_i' is not in state set S, then
13) Add state s_i' to state set S;
14) End if
15) Add transition $s_i \xrightarrow{e_i} s_i'$ to transition set Δ;
16) End for
17) Return step 1; // Repeat steps, where TS has been changed.
18) Else
19) Synthesize a PN model PN by region set Π according to Definition 7;
20) End if
21) Output PN model PN;
End

In Algorithm 1, we first calculate solutions to all separation problems in TS by using a general PN synthesis algorithm [23] (steps 3–4). Then for the purpose of adding information to construct a PN whose reachability graph is isomorphic to the TS, we put the current unsolvable separation problem(s) into the set Ξ (step 6). If the set Ξ is empty, the PN model can be synthesized by the region set Π according to the Definition 7 (step 19); otherwise, some information needs to be added. As for an actual system, we require that any two states of the system should be distinguished, so the TS which comes from the reality satisfies the state separation property, that is, there are only event/state separation problems in the unresolved separation problem set Ξ. Thus, we can add some arcs (or arcs with states) by performing the steps 10 to 16. It can be seen from Theorem 2, the added information is reasonable. After that, return to step 1 and re-solve problems in the new TS until the region set of TS satisfies the conditions. Finally, the PN model PN can be output (step 21).

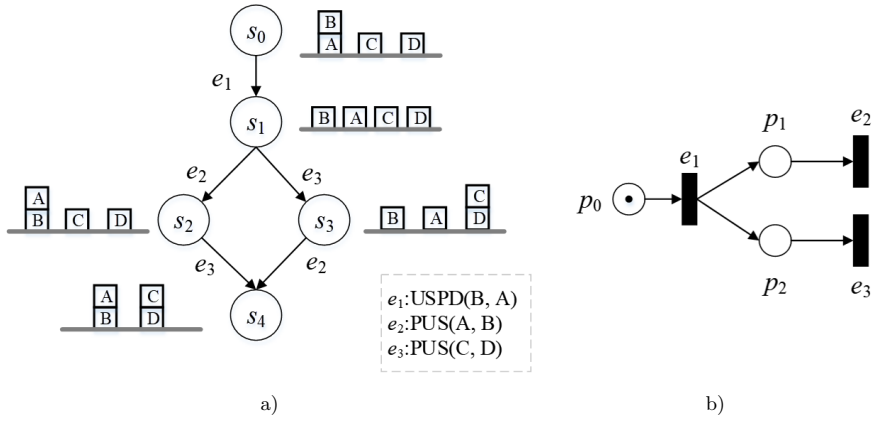


Figure 4. a) The initialized transition system TS_2, and b) the corresponding PN PN_2

Here we use the example shown in Figure 1 to explain the algorithm. It is assumed that a part of the initialized transition system TS can be obtained by the robot's autonomous attempts shown in Figure 4 a), where a state is represented by s_i $(0 \leq i \leq 4)$ and an event is represented by e_i $(1 \leq i \leq 2)$. For convenience, we show the position of building blocks next to the state and list the operations in the dashed box. Taking the TS as an input of Algorithm 1, the output can be obtained as shown in Figure 4 b). In Algorithm 1, the first operation is to solve each separation problem $pr \in \mathrm{SP}$ in TS. For example, in state s_2, the event e_2 is not enabled, so $pr = \mathrm{ESSP}(s_2, e_2)$ is an event/state separation problem. Because state s_2 is reached from state s_0 via event sequence e_1e_2, we can obtain $\mathrm{SP}(\rho, pr) = (\mathrm{tokens}(\rho, s_2) < \rho_2) = (\rho_0 + 1 \cdot (\rho_4 - \rho_1) + 1 \cdot (\rho_5 - \rho_2) + 0 \cdot (\rho_6 - \rho_3) < \rho_2)$ according to the Definition 6 and the Definition 8. In addition, the following inequalities can

be produced owing to the region constraints:

$$\begin{aligned}
&\rho_i \geq 0 \ (0 \leq i \leq 6)\\
&\wedge \rho_1 \leq \text{tokens}(\rho, s_0) = \rho_0\\
&\wedge \rho_2 \leq \text{tokens}(\rho, s_1) = \rho_0 + (\rho_4 - \rho_1)\\
&\wedge \rho_3 \leq \text{tokens}(\rho, s_1) = \rho_0 + (\rho_4 - \rho_1)\\
&\wedge \rho_3 \leq \text{tokens}(\rho, s_2) = \rho_0 + (\rho_4 - \rho_1) + (\rho_5 - \rho_2).
\end{aligned}$$

Considering the target PN is a safe PN, the constraint $\text{tokens}(\rho, s_i) \leq 1$ $(0 \leq i \leq 4)$ should also be satisfied. We can compute that the vector $\rho = (0, 0, 1, 0, 1, 0, 0)$ is a possible solution of the above constraints. In this example, there is no unsolvable separation problem, so the PN can be obtained according to the Definition 7 as shown in Figure 4 b), where the vector $\rho = (0, 0, 1, 0, 1, 0, 0)$ is corresponding to place p_1 in the PN.

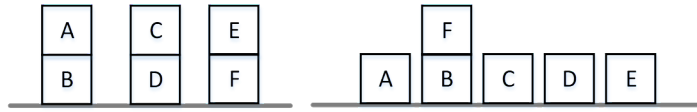
4 SIMULATION AND EXPERIMENTS

In Section 3, we give the algorithm of Petri net model generation and prove the corresponding theorems to ensure the rationality of the algorithm. In order to exhibit the effectiveness of the method better, in this section, we achieve a program to simulate the scene introduced in Section 2 and implement the algorithm. After experiments and comparisons, it is indicated that the method is reasonable and effective.

The program is coded in Java and requires input of the initial state and the target state of the building blocks. In the program, when a path from the initial state to the target is found, the Algorithm 1 is called to generate the PN model. In the process of simulation, there may be cases where the known information is insufficient and the PN cannot be obtained. At this time, some information needs to be added according to the unresolved separation problem set, that is, steps 9 to 17 of Algorithm 1 will be executed. Then an accurate PN model which is more complete can be obtained.

In the beginning, we introduce another example, as shown in Figure 5. Figures 5 a) and 5 b) show the initial state of the building block and the target one. When the initialized transition system TS_1 is produced by robot's attempts, as shown in Figure 6 a), it fails to generate a PN model. Therefore, the information can be supplemented according to the unresolved event/state separation problems. That is to say, the transitions s0 $\xrightarrow{e2}$ s10, s10 $\xrightarrow{e0}$ s5 and s10 $\xrightarrow{e1}$ s9 (s10 is a new state) should be added to TS_1 what results in TS_2, as shown in Figure 6 b). The initial state of the TS is marked in red labeled with s0 and the target state is marked in green labeled with s4. Besides, we mark the position of the building blocks in the state. For instance, "A-B" in s0 means that the building block A is on the building block B, and "@" separates the pile of building blocks. That is, "A-B" and "C-D" are two piles of building blocks. Then a PN model (Figure 7) can be automatically

generated by TS_2, where places are represented by circles, transitions are represented by rectangles and the red place (p2, p3, p5, p8) means the place containing one token. It's obvious that the operation USPD (E, F) can be performed in state s0 to reach the state s10. That is, this supplement is in line with reality, so as others. Consequently, the information is increased reasonably.



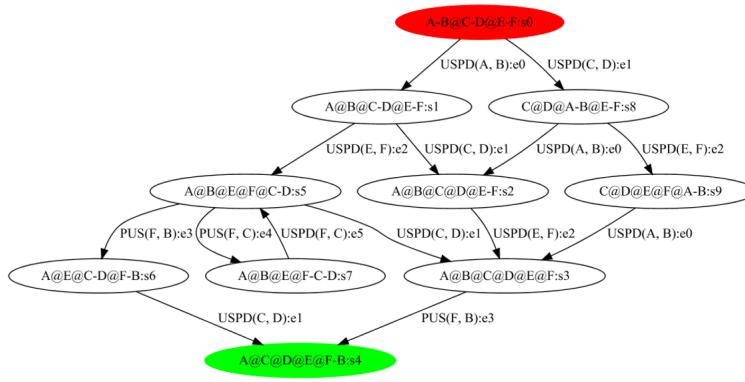
a) b)

Figure 5. a) Initial state and b) target state of the building blocks

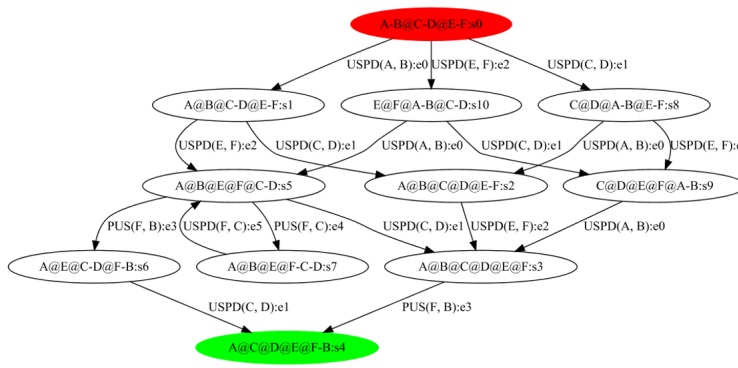
Then, in order to demonstrate the effectiveness of the algorithm better, we design three more complex examples and simulate them. The initial state and target state of the examples are shown in Table 2. In the simulation, we specify that the PN model is going to be generated when the program finds a path from the initial state to the target. For each example, we compare the results of the exhaustive generation and supplementary generation by program. The exhaustive generation means the generation of a PN model according to the known information directly and the supplementary generation means the generation of a PN model via using Algorithm 1 which includes the information added steps. The results are shown in Table 3. For exhaustive generation, the number of attempts to generate a PN model as well as the failure times is computed during the period of generating a complete TS. For supplementary generation, the number of attempts to generate a PN model by Algorithm 1, the states and arcs added to the TS during the whole process are calculated. The total number of states and arcs of TS are also listed in the table. Definitely, we can avoid the failures of model generation thanks to Algorithm 1, so there are no failure times in supplementary generation. It can be seen that the attempt times of supplementary generation are less than exhaustive generation for all examples listed in Table 3. Even more, supplementary generation can add some states and arcs to TS which lead to the reduction in attempt times. In other words, by using the proposed algorithm, we can obtain a complete PN model without traversing completely. Consequently, it is available to learn a PN model based on the method presented in this paper.

	Initial State	**Target State**
1	A-B@C-D@E-F	A@B-C@F@D-E
2	A-B-C@D-E	E-C-B@A-D
3	A@B@G-C@D@F-E	A-B@D-C@E@F@G

Table 2. Examples of simulations



a)



b)

Figure 6. a) The initialized transition system TS_3 which is produced by robot's attempts, b) the initialized transition system TS_4 after adding information to TS_3

5 CONCLUSIONS

With the development of intelligent technology, more and more researchers begin to join in the field of robotics and devote to the modeling and learning of the robot system. PN is an abstract formal modeling method, which can represent the sequence and concurrent events as well as the restrictions of various conditions. In view of the flexibility and effectiveness, PN can be applied to the robot field. Motivated by the scene of building block world, this paper introduced two theorems and a PN model generation algorithm based on region theory, which achieves a PN model generated automatically according to a transition system, as well as makes model more complete to some extent. Besides, the effectiveness of the method is demonstrated by a program which simulates the robot scene and applies the algorithm.

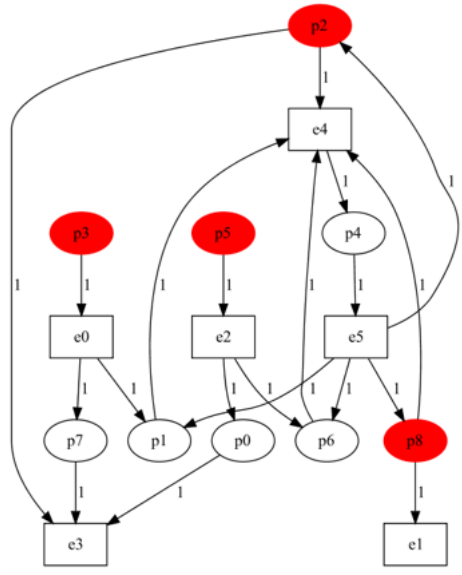


Figure 7. The corresponding PN of TS_4

	Exhaustive Generation		Supplementary Generation				
	Attempt Times	Failure Times	Attempt Times	Number of Added States	Number of Added Arcs	Total States	Total Arcs
1	90	33	75	6	24	40	90
2	243	106	183	28	95	98	249
3	175	85	128	15	65	62	171

Table 3. Results of simulations

In future work, we intend to improve the performance of the algorithm as well as studying the model analysis methods. Furthermore, we will focus on the extension of the method to multi-robot systems and other automated manufacturing systems.

Acknowledgement

This work is partially supported by the National Key Research and Development Program of China under Grant No. 2018YFB2100801 and by the National Natural Science Foundation of China under Grant No. 61672381, and in part by the Fundamental Research Funds for the Central Universities under Grant No. 22120180508.

REFERENCES

[1] Desai, J. P.: A Graph Theoretic Approach for Modeling Mobile Robot Team Formations. Journal of Robotic Systems, Vol. 19, 2002, No. 11, pp. 511–525, doi: 10.1002/rob.10057.

[2] Wieber, P.-B.—Tedrake, R.—Kuindersma, S.: Modeling and Control of Legged Robots. In: Siciliano, B., Khatib, O. (Eds.): Springer Handbook of Robotics. Springer Handbooks, Springer, Cham, 2016, pp. 1203–1234, doi: 10.1007/978-3-319-32552-1_48.

[3] Sim, S. K.—Ong, K. W.—Seet, G.: A Foundation for Robot Learning. 2003 4th International Conference on Control and Automation Proceedings, IEEE, 2003, pp. 649–653, doi: 10.1109/ICCA.2003.1595102.

[4] Norris, D. J.: Behavior-Based Robotics. Chapter 11. In: Norris, D. J.: Beginning Artificial Intelligence with the Raspberry Pi. Apress, Berkeley, CA, 2017, pp. 313–345, doi: 10.1007/978-1-4842-2743-5_11.

[5] Brooks, R. A.—Mataric, M. J.: Real Robots, Real Learning Problems. In: Connell, J. H., Mahadevan, S. (Eds.): Robot Learning. Springer, Boston, MA, The Springer International Series in Engineering and Computer Science (Knowledge Representation, Learning and Expert Systems), Vol. 233, 1993, pp. 193–213, doi: 10.1007/978-1-4615-3184-5_8.

[6] Demiris, J.—Birk, A.: Interdisciplinary Approaches to Robot Learning: Introduction. World Scientific Series in Robotics and Intelligent Systems, Vol. 24, 2000, pp. 1–7, doi: 10.1142/9789812792747_0001.

[7] Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, 1989, No. 4, pp. 541–580, doi: 10.1109/5.24143.

[8] Heiner, M.—Herajy, M.—Liu, F.—Rohr, C.—Schwarick, M.: Snoopy – A Unifying Petri Net Tool. In: Haddad, S., Pomello, L. (Eds.): Application and Theory of Petri Nets (PETRI NETS 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7347, 2012, pp. 398–407, doi: 10.1007/978-3-642-31131-4_22.

[9] Lima, P.—Gracio, H.—Veiga, V.—Karlsson, A.: Petri Nets for Modeling and Coordination of Robotic Tasks. 1998 IEEE International Conference on Systems, Man, and Cybernetics (SMC'98), San Diego, CA, USA, 1998, Vol. 1, pp. 190–195, doi: 10.1109/ICSMC.1998.725407.

[10] Ziparo, V. A.—Iocchi, L.: Petri Net Plans. Proceedings of Fourth International Workshop on Modelling of Objects, Components, and Agents (MOCA), June 2006, pp. 267–290.

[11] Chao, C.—Thomaz, A. L.: Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets. Journal of Human-Robot Interaction, Vol. 1, 2012, No. 1, pp. 4–25, doi: 10.5898/JHRI.1.1.Chao.

[12] Chang, G.—Kulić, D.: Robot Task Learning from Demonstration Using Petri Nets. 2013 IEEE International Workshop on Robot and Human Communication (ROMAN), Gyeongju, South Korea, 2013, pp. 31–36, doi: 10.1109/ROMAN.2013.6628527.

[13] van der Aalst, W.—Weijters, T.—Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, 2004, No. 9, pp. 1128–1142, doi: 10.1109/TKDE.2004.47.

[14] Roldán, J. J.—del Cerro, J.—Barrientos, A.: Using Process Mining to Model Multi-UAV Missions Through the Experience. IEEE Intelligent Systems, Vol. 32, 2017, No. 4, pp. 40–47, doi: 10.1109/MIS.2017.3121547.

[15] Badouel, E.—Bernardinello, L.—Darondeau, P.: Petri Net Synthesis. Springer, Berlin, Heidelberg, Texts in Theoretical Computer Science, 2015, doi: 10.1007/978-3-662-47967-4.

[16] Badouel, E.—Darondeau, P.: Theory of Regions. In: Reisig, W., Rozenberg, G. (Eds.): Lectures on Petri Nets I: Basic Models (ACPN 1996). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1491, 1996, pp. 529–586, doi: 10.1007/3-540-65306-6_22.

[17] Cortadella, J.—Kishinevsky, M.—Kondratyev, A.—Lavagno, L.—Yakovlev, A.: Petrify: A Tool for Manipulating Concurrent Specifications and Synthesis of Asynchronous Controllers. IEICE Transactions on Information and Systems, Vol. E80-D, 1997, No. 3, pp. 315–325.

[18] Carmona, J.—Cortadella, J.—Kishinevsky, M.: Genet: A Tool for the Synthesis and Mining of Petri Nets. 2009 Ninth International Conference on Application of Concurrency to System Design, Augsburg, Germany, July 2009, pp. 181–185, doi: 10.1109/ACSD.2009.6.

[19] Badouel, E.—Caillaud, B.—Darondeau, P.: Distributing Finite Automata Through Petri Net Synthesis. Formal Aspects of Computing, Vol. 13, 2002, No. 6, pp. 447–470, doi: 10.1007/s001650200022.

[20] Best, E.—Schlachter, U.: Analysis of Petri Nets and Transition Systems. Proceedings of the 8th Interaction and Concurrency Experience Workshop (ICE 2015). Electronic Proceedings in Theoretical Computer Science (EPTCS), Vol. 189, 2015, pp. 53–67, doi: 10.4204/EPTCS.189.6.

[21] Wu, Z. H.: Introduction to Petri Nets. China Machine Press, Beijing, 2006 (in Chinese).

[22] Best, E.—Devillers, R.—Schlachter, U.: Bounded Choice-Free Petri Net Synthesis: Algorithmic Issues. Acta Informatica, Vol. 55, 2018, No. 7, pp. 575–611, doi: 10.1007/s00236-017-0310-9.

[23] Schlachter, U.: Petri Net Synthesis for Restricted Classes of Nets. In: Kordon, F., Moldt, D. (Eds.): Application and Theory of Petri Nets and Concurrency (PETRI NETS 2016). Springer, Cham, Lecture Notes in Computer Science, Vol. 9698, 2016, pp. 79–97, doi: 10.1007/978-3-319-39086-4_6.

[24] Wolf, K.: Petri Net Synthesis with Union/Find. In: Khomenko, V., Roux, O. (Eds.): Applications and Theory of Petri Nets and Concurrency (PETRI NETS 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10877, 2018, pp. 60–81, doi: 10.1007/978-3-319-91268-4_4.

[25] Tredup, R.: Hardness Results for the Synthesis of b-Bounded Petri Nets. In: Donatelli, S., Haar, S. (Eds.): Applications and Theory of Petri Nets and Concurrency (PETRI NETS 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11522, 2019, pp. 127–147, doi: 10.1007/978-3-030-21571-2_9.

Jiao Li received her B.Sc. degree in computing science and technology from the Jiangsu University, Zhenjiang, China, in 2017. She is currently pursuing her M.Sc. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Her current research interests include Petri nets and formal engineering.

Ru Yang received her B.Sc. degree from the Shandong University of Science and Technology, Qingdao, China, in 2013. She is currently pursuing her Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Her current research interests include Petri nets and formal engineering.

Zhijun Ding received his M.Sc. degree from the Shandong University of Science and Technology, Tai'an, China, in 2001, and the Ph.D. degree from Tongji University, Shanghai, China, in 2007. He serves currently as Professor with the Department of Computer Science and Technology, Tongji University. He has published over 100 papers in domestic and international academic journals and conference proceedings. His research interests are in formal engineering, Petri nets, services computing, and mobile internet.

Meiqin Pan received her Ph.D. degree from Shandong University of Science and Technology, Qingdao, China, in 2008. Now she is Associate Professor of the School of Business and Management, Shanghai International Studies University. Her research interests are in information systems, data mining and technology optimization methods. She has published more than 20 papers in domestic and international academic journals and conference proceedings.

Computing and Informatics, Vol. 39, 2020, 193–212, doi: 10.31577/cai_2020_1-2_193

CHECKING DATA-FLOW ERRORS BASED ON THE GUARD-DRIVEN REACHABILITY GRAPH OF WFD-NET

Dongming Xiang

School of Information Science and Technology
Zhejiang Sci-Tech University
310018 Hangzhou, China
e-mail: flysky_xdm@163.com

Guanjun Liu

Department of Computer Science
Key Laboratory of Embedded System and Service Computing (MOE)
Tongji University
201804 Shanghai, China
e-mail: liuguanjun@tongji.edu.cn

Abstract. In order to guarantee the correctness of workflow systems, it is necessary to check their data-flow errors, e.g., missing data, inconsistent data, lost data and redundant data. The traditional Petri-net-based methods are usually based on the reachability graph. However, these methods have two flaws, i.e., the state space explosion and pseudo states. In order to solve these problems, we use WFD-nets to model workflow systems, and propose an algorithm for checking data-flow errors based on the guard-driven reachability graph (GRG) of WFD-net. Furthermore, a case study and some experiments are given to show the effectiveness and advantage of our method.

Keywords: Petri net, workflow system, data-flow errors, reachability graph

Mathematics Subject Classification 2010: 68-Q60

1 INTRODUCTION

Nowadays, workflow systems have been widely applied to our daily life, e.g., office automation (OA), medical treatment and electronic commerce, etc. In order to guarantee the correctness of workflow systems, we not only need to verify some properties and detect errors in the control-flows, but also model and analyze their data-flows. As we know, the control-flows focus on the partial orders of business activities, while the data-flows mostly include data elements, data operations (i.e., read, write and delete) and data conditions. The existing modeling and analysis methods of workflow systems are mainly concerned with the error detection of control-flows. In fact, data-flows are also greatly important in the design of workflow system. Once its activities conduct an improper operation on data-flows in business processes, some data-flow errors [15, 19, 28] easily take place, e.g., missing data, inconsistent data, lost data, redundant data and unsoundness. These errors can lead to some abnormal results, degrade the execution performance, and increase the maintenance cost, or even result in some insecurity problems, e.g., privacy disclosure, illegal user access, and fund loss.

There have been many studies on data-flows of workflow systems. Sadiq et al. [19] first proposed seven kinds of data-flow anomalies, but did not provide any detection methods. Sharma et al. [21] used BPMN (Business Process Modeling Notation) to model business processes and detected their data-flow errors. Guo et al. [8] solved the data exchange problems in the inter-organizational workflows. Sun et al. [24] calculated the dependence relationship of business processes in a UML (Unified Modeling Language) diagram, and detected errors in each process instance according to its data association. This work was further generalized in [15], where a systematic graph traversal approach was proposed to detect data-flow errors.

Some Petri-net-based methods are also proposed to detect data-flow errors. A Dual Flow Net (DFN) [27] was used to model the control- and data-flows in an embedded system. Based on the work in [21], Awad et al. [2] mapped BPMN into Petri net, and then detected and repaired its errors. In order to model the concurrent read operation, contextual net [3, 16] was proposed, and its unfolding technique was utilized to generate the minimal test suites for multi-threaded programs [13, 12]. Based on the contextual net, PN-DO (Petri net with data operation) [31] was given to detect data-flow errors of workflow systems. All of these methods have an advantage of a great capability to specify parallelism, concurrency and synchronization [11, 17]. However, the explicitly modeling of read/write arcs can increase the scales and complexity of Petri nets. By comparison, WFD-net is a workflow net [18] (a special Petri net) extended with conceptual data operations. Its transitions are labeled by *read*, *write*, *delete* or *guard*[1] functions [10, 22]. Naturally, the scale of WFD-net is much more smaller than the Petri nets with data operation arcs, e.g., contextual net and PN-DO.

[1] A guard is a Boolean expression which is formed by some data elements and predicates.

WFD-net has been widely used to check soundness [22], completion requirements [25] and data consistencies [34]. These verification/analysis methods are usually based on the classical reachability graphs [22, 32] of WFD-nets. However, they easily suffer from the state space explosion and pseudo states. On the one hand, a state may have an exponential number of successor states in a reachability graph since every possible value of a guard is considered. On the other hand, the logical relation (e.g., exclusion property) between guards likely generate some pseudo states. In order to solve these problems, we proposed the guard-driven reachability graph (GRG) of a WFD-net in the previous work [30].

In this paper, we use GRG to check data-flow errors in a workflow system, including missing data, inconsistent data, redundant data and lost data. We first define these data-flow errors in a WFD-net, and propose an algorithm for checking them. Furthermore, a case study and some experiments are given to illustrate the effectiveness and advantage of our algorithm.

The rest of this paper is organized as follows. Section 2 presents some basic notations. Section 3 proposes an algorithm to check data-flow errors based on the GRG of WFD-nets. Section 4 gives a case study. Section 5 conducts a group of experiments. The last section sums up the whole work.

2 BASIC NOTATIONS

2.1 WF-Net

A *net* is a triple $N = (P, T, F)$, where P and T are two disjoint and finite sets that are respectively called *place set* and *transition set*, and $F \subseteq (P \times T) \cup (T \times P)$ is a *flow* relation. A net N with an initial marking m_0 is called a *Petri net* or *net system* [14, 33], and denoted as $\Sigma = (N, m_0)$. For each node $x \in P \cup T$, its *preset* and *postset* are denoted by ${}^\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$, respectively.

Given a net $N = (P, T, F)$, a transition $t \in T$ is *enabled* at a marking m if $\forall p \in P : p \in {}^\bullet t \Rightarrow m(t) \geq 1$, which is denoted by $m[t\rangle$. After *firing* an enabled transition t at m, a new marking m' is generated, which is denoted as $m[t\rangle m'$, where $\forall\ p \in P$:

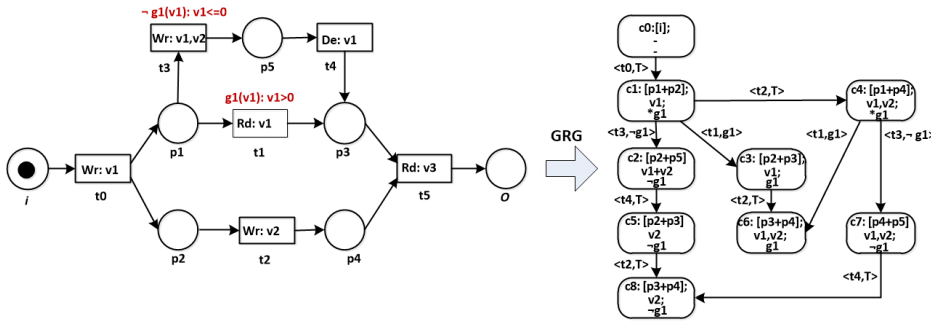
$$m'(p) = \begin{cases} m(p) - 1, & \text{if } p \in {}^\bullet t - t^\bullet, \\ m(p) + 1, & \text{if } p \in t^\bullet - {}^\bullet t, \\ m(p), & \text{otherwise.} \end{cases} \tag{1}$$

Definition 1 (Workflow net [1])**.** A net $N = (P, T, F)$ is a workflow net (WF-net) if

1. there is one source place i and one sink place o in P such that ${}^\bullet i = \emptyset \wedge o^\bullet = \emptyset$; and
2. $\forall x \in P \cup T$: $(i, x) \in F^*$ and $(x, o) \in F^*$ where F^* is the reflexive-transitive closure of F.

As a particular class of Petri net, WF-net has been widely used to model and verify workflow systems [1].

2.2 WFD-Net



a) b)

Figure 1. WFD-net and data-flow errors

Based on WF-net, *workflow net with data* (WFD-net) [25, 26] is proposed to model the control-flows and data-flows of a workflow system. That is, some data elements, data operations and guards are added into WF-net.

In a WFD-net, D is a finite set of data elements, and G is a set of *guards* over D. $\mathrm{Var}(g)$ denotes the set of variables in the guard $g \in G$. We assume that $\forall g \in G : d \in \mathrm{Var}(g) \Rightarrow d \in D$. Two different guards g_1 and g_2 are *exclusive* if $g_1 = \neg g_2$ and $g_2 = \neg g_1$, which is denoted by $g_1 \otimes g_2$. In other words, g_1 is TRUE iff g_2 is FALSE, and vice versa. For example, given two guards g_1 and g_2, if $g_1(v_1) : v_1 > 0$ and $g_2(v_1) : v_1 \leq 0$, then they are exclusive and denoted by $g_1 \otimes g_2$.

Definition 2 (Workflow net with data [22])**.** A 9-tuple $N = (P, T, F, D, G, Rd, Wr, De, \mathrm{Guard})$ is a WFD-net if

1. (P, T, F) is a WF-net;
2. D is a finite set of data elements;
3. G is a finite set of guards;
4. Rd: $T \to 2^D$ is a label function of reading data;
5. Wr: $T \to 2^D$ is a label function of writing data;

6. De: $T \to 2^D$ is a label function of deleting data; and
7. $Guard$: $T \to G$ is a label function of assigning a guard in G to each transition.

For example, Figure 1 a) is a WFD-net, where $D = \{v_1, v_2, v_3\}$, $G = \{g_1(v_1), \neg g_1(v_1)\}$, $\text{Guard}(t_1) = g_1(v_1)$, $\text{Guard}(t_3) = \neg g_1(v_1)$, $Rd(t_1) = \{v_1\}$, $Wr(t_2) = \{v_2\}$ and $De(t_4) = \{v_1\}$. Moreover, $g_1 \otimes \neg g_1$ and $\text{Var}(g_1) = \text{Var}(\neg g_1) = \{v_1\}$.

As for a WFD-net, a state is generally called a *weak configuration*, and it includes a marking and the evaluations of data and guards.

Definition 3 (Weak configuration)**.** Let $N = (P, T, F, D, G, Rd, Wr, De, \text{Guard})$ be a WFD-net. $c = \langle m, \sigma, \eta \rangle$ is a weak configuration, where

1. m is a marking of (P, T, F);
2. $\sigma : D \to \{\top, \bot\}$ assigns a defined value ($\top$) or an undefined value ($\bot$) to each data element; and
3. $\eta : G \to \{\text{TRUE}, \text{FALSE}, \bot, \top\}$ assigns TRUE, FALSE, an undefined value ($\bot$) or a defined value ($\top$) to each guard.

In a weak configuration, (σ, η) represents a data state. Besides, we use the guards labeled with $*$ to represent their defined values ($\top$). As shown in Figure 1 b), $c_1 = \langle m_1, \sigma_1, \eta_1 \rangle = \langle [p_1 + p_2], \{v_1\}, \{*g_1\} \rangle$ is a weak configuration of the WFD-net in Figure 1 a), where $\sigma_1(v_1) = \top$ and $*g_1$ represents that $\eta_1(g_1) = \top$.

Given the definition of WFD-net, we discuss its *weak firing* rules of an enabled transition at a weak configuration.

Definition 4 (Weak enabling/firing rules)**.** Let $N = (P, T, F, D, G, Rd, Wr, De, \text{Guard})$ be a WFD-net. $t \in T$ is enabled at a weak configuration $c = \langle m, \sigma, \eta \rangle$, which is denoted by $c[t\rangle$, if

1. $m[t\rangle$;
2. $\forall d \in Rd(t) : \sigma(d) = \top$; and
3. $\forall d \in \text{Var}(\text{Guard}(t))$: $\sigma(d) \neq \bot$ and $\eta(\text{Guard}(t)) \in \{\text{TRUE}, \top\}$.

After firing an enabled transition t at c, a weak configuration $c' = \langle m', \sigma', \eta' \rangle$ is generated, where

1. $m[t\rangle m'$;
2. $\forall d \in De(t) : \sigma'(d) = \bot$;
3. $\forall d \in Wr(t) \setminus De(t) : \sigma'(d) = \top$;
4. $\forall d \in D \setminus (De(t) \cup Wr(t)) : \sigma'(d) = \sigma(d)$;
5. $\exists g \in \text{Guard}(t) : \text{Var}(g) \cap Wr(t) = \emptyset \Rightarrow \eta'(g) = \text{TRUE}$; and
6. $\forall g \in G, \forall d \in \text{Var}(g) : (\sigma'(d) = \top \Rightarrow \eta'(g)\top) \wedge ((g \notin \text{Guard}(t) \wedge \text{Var}(g) \cap Wr(t) = \emptyset) \Rightarrow \eta'(g) = \eta(g))$.

It is denoted as $c[t\rangle c'$.

For example, the transition t_0 in Figure 1 a) is enabled at the *initial weak configuration* c_0 and $c_0[t_0\rangle c_1$, where $c_0 = \langle [i], -, - \rangle$ and $c_1 = \langle [p_1 + p_2], \{v_1\}, \{*g_1\} \rangle$. After firing the transition t_0 and writing a new value into the data v_1, the evaluations of g_1 is not definite because v_1 is associated with this guard. We assign a defined value ($\top$) to this guard in c_1. Thus, firing t_0 generates one unique weak configuration in Figure 1 b).

According to the enabling and firing rule of transitions, the may-reachability of WFD-net is defined as follows.

Definition 5 (May-reachability [22])**.** Let $N = (P, T, F, D, G, Rd, Wr, De, \text{Guard})$ be a WFD-net. c_1 and c_2 are two configurations.

1. There is a may-step from c_1 to c_2, denoted by $c_1 \rightarrow_{may} c_2$, if there is a transition $t \in T$ and a set of configurations C such that: $c_1[t\rangle C \wedge c_2 \in C$.
2. c_2 is may-reachable from c_1 if there exists a sequence of configurations $c^{(1)}, \ldots, c^{(n)}$ such that $c_1 \rightarrow_{may} c^{(1)} \rightarrow_{may} \cdots \rightarrow_{may} c^{(n)} \rightarrow_{may} c_2$. It is denoted as $c_1 \rightarrow^*_{may} c_2$.

The set of may-reachable configurations from c is denoted by $R(c)$. For example, there is a may-step from c_0 to c_1 in Figure 1 b), and the configuration $c_6 = \langle [p_3 + p_4], \{v_1, v_2\}, \{g_1\} \rangle$ is may-reachable from c_0.

2.3 Guard-Driven Reachability Graph

Although the classical reachability graph [22] is a fundamental method of analyzing and verifying a WFD-net, it easily suffers from the problems of state space explosion and pseudo configurations due to its guard evaluations and their exclusive relations [30]. Hence, we propose the *guard-driven reachability graph* (GRG) based on the weak configurations and the weak firing rules.

Definition 6 (Guard-driven reachability graph, GRG)**.** Let $N = (P, T, F, D, G, Rd, Wr, De, \text{Guard})$ be a WFD-net and c_0 be its initial weak configuration. $GRG(N) = (V^+, E^+, \ell^+)$ is the guard-driven reachability graph of N, where

1. $V^+ = R(c_0)$;
2. $E^+ = \{(c, c') \mid c, c' \in R(c_0) \wedge \exists t \in T : c[t\rangle c'\}$; and
3. $\ell^+ : E^+ \rightarrow T \times G$ such that $\ell^+(c, c') = \langle t, \text{Guard}(t) \rangle$ if $(c, c') \in E^+$ and $c[t\rangle c'$.

For example, Figure 1 b) is the GRG of the WFD-net in Figure 1 a), where $c_0, c_1 \in V^+$, $e_0 = (c_0, c_1) \in E^+$ and $\ell^+(e_0) = \langle t_0, \text{TRUE} \rangle$.[2]

In the guard-driven reachability graph of a WFD-net, the guard as the condition of enabling a transition determines the unique successor state when firing the

[2] If $\text{Guard}(t_0) = \emptyset$, we use $\langle t_0, \text{TRUE} \rangle$ to represent this case.

transition. Therefore, the idea of guard-driven reachability graph is to show the execution of a WFD-net by the evaluations of guards.

3 DATA-FLOW ERRORS DETECTION METHOD BASED ON THE GRG OF WFD-NET

3.1 Data-Flow Errors

Data-flow errors are caused by improper data operations in workflow systems, which mainly include missing data, redundant data, lost data, and inconsistent data. We first define these data-flow errors in a WFD-net.

1. Missing Data

Missing data occurs when a business process of workflow systems is reading or deleting some data, but this data is not existing at this time.

Definition 7 (Missing Data)**.** A WFD-net N with an initial weak-configuration c_0 has an error of missing data if $\exists t \in T, \exists c \in R(c_0) : c = \langle m, \sigma, \eta \rangle \wedge \neg c[t \wedge m[t\rangle \wedge \exists d \in Rd(t) \cup De(t) : \sigma(d) = \perp$.

For example, the transition t_5 in Figure 1 is not enabled at the reachable weak-configuration $[p_3 + p_4; v_2; g_1]$ because it cannot read the data from v_3 since this data has never been written into any values. At this time, missing data occurs.

2. Inconsistent Data

The error of inconsistent data usually occurs in a concurrent workflow system when one business process is reading or writing or deleting some data, but another process is concurrently writing or deleting this data. Notice that two transitions t_1 and t_2 in a bounded WFD-net are *concurrent* at a weak-configuration $c = \langle m, \sigma, \eta \rangle$, if they satisfy that $c[t_1\rangle \wedge c[t_2\rangle \wedge ({}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset \vee \forall p \in {}^\bullet t_1 \cap {}^\bullet t_2 : m(p) \geq 2)$ [29]. This is denoted by $t_1 \|_c t_2$.

Definition 8 (Inconsistent Data)**.** The error of inconsistent data takes place in a WFD-net if two concurrent transitions t_1 and t_2 satisfy $\exists c \in R(c_0) : (Rd(t_1) \cup Wr(t_1) \cup De(t_1)) \cap (Wr(t_2) \cup De(t_2)) \neq \emptyset$.

For example, two transitions t_2 and t_4 in Figure 1 are concurrently writing into the data v_2 at the reachable weak-configuration $[p_1 + p_2; v_1; *g_1]$. At this time, inconsistent data occurs.

3. Redundant Data

Redundant data occurs if a data is never read before it is deleted or the business process terminates.

Definition 9 (Redundant Data)**.** Σ has an error of redundant data if one of the following two conditions holds:

1. $\exists c_1, c_2 \in R(c_0)$, $\exists t_1 \in T$, $\exists v \in D : c_1[t_1\rangle c_2 \wedge v \in Wr(t_1) \wedge (\forall c_3 \in R(c_2), \forall t_2 \in T : c_3[t_2\rangle \rightarrow v \notin Rd(t_2))$;
2. $\exists c_1, c_2 \in R(c_0)$, $\exists t_1, t_2 \in T$, $\exists \sigma \in T^*$, $\exists v \in D : c_1[t_1\sigma\rangle c_2[t_2\rangle \wedge v \in Wr(t_1) \wedge v \in De(t_2) \wedge (\forall t_3 \in \sigma : v \notin Rd(t_3))$.

For example, the transition t_2 in Figure 1 is to overwrite the data v_2 at the reachable weak-configuration $[p_1 + p_2; v_1; *g_1]$. But the data has never been read until the business process terminates. Therefore, there is an error of redundant data.

4. Lost Data

Lost data means that once a data element is written into a value by a transition, it will never be read before it is written again by some follow-up transitions. In other word, the first value of this data element cannot be referenced again by other activities.

Definition 10 (Lost Data)**.** Σ has an error of lost data if $\exists c_1, c_2 \in R(c_0)$, $\exists t_1, t_2 \in T$, $\exists \sigma \in T^*$, $\exists v \in V : c_1[t_1\sigma\rangle c_2[t_2\rangle \wedge v \in Wr(t_1) \cap Wr(t_2) \wedge (\forall t_3 \in \sigma : v \notin Rd(t_3))$.

For example, the transition t_0 in Figure 1 writes a data value into v_1 at the initial weak-configuration. But this data is never be read before it is overwritten by t_3. Therefore, this is an error of lost data.

3.2 The Algorithm for Checking Data-Flow Errors Based on GRG

In order to check the above data-flow errors in a workflow system, we propose an algorithm based on GRG, which is shown in Algorithm 1.

- According to the definition of missing data, we can easily check this data-flow error by traversing each weak-configuration.
- At a weak-configuration, if two concurrent transitions are concurrently conducting some data operations on a data element, we can find out an error of inconsistent data.
- If an enabled transition is to write some value into a data element at a weak-configuration, we can traverse all successors of this weak-configuration. Then, the weak-configurations related to the operations on this data are obtained by the function

$$\text{FindRWDConfigs}(v, c', GRG(\Sigma), CT).$$

That is, we traverse all weak-configurations reachable from c', and obtain three reachable weak-configuration sets c_r, c_w and c_d, where c_r (resp. c_w, c_d) is the set

Procedure_1 *FindRWDConfigs*$(v, c', GRG(\Sigma), CT)$

```
1: if c' ∉ CT then
2:   CT.add(c');
3:   Get all edges from c', i.e., E2 = {(c', c'') | (c', c'') ∈ E};
4:   if E2 ≠ ∅ then
5:     for each (c', c'') ∈ E2 do
6:       if c'[t'⟩c'' or c'[ct'⟩c'' then
7:         if v ∈ Rd(t') then
8:           cr.add(c');
9:         end if
10:        if v ∈ Wr(t') then
11:          cw.add(c');
12:        end if
13:        if v ∈ De(t') then
14:          cd.add(c');
15:        end if
16:        if v ∉ Rd(t') ∪ Er(t') ∪ De(t') then
17:          FindRWD(v, c'', GRG(Σ), CT);
18:        end if
19:      end if
20:    end for
21:  end if
22: end if
    end Procedure_1
```

of reachable weak-configurations at which there is a read (resp. write, delete) operation. Finally, according to these weak-configurations, we determine whether there is an error of redundant data or lost data.

It is noted that GRG can effectively reduce the state space and avoid pseudo states since it fully considers the characters of guard functions. As a result, Algorithm 1 only needs to traverse a smaller state space to detect data-flow errors in comparison with the classical reachability graph in [22]. Moreover, it also prevents from some negative influence by pseudo states.

4 CASE STUDY

Our checking method for data-flow errors can be applied in the static program analysis. Figure 2 is a multi-thread program, and it is used to detect the errors of data inconsistency in the related work [9]. As our case study in this paper, we utilize it to check data-flow errors.

We first use a WFD-net to model this program, which is shown in Figure 3 a). Tables 1 and 2 list the related transitions and guards. Meanwhile, if we respec-

Algorithm 1 Data-flow error detection algorithm

Require: A WFD-net N

Ensure: All data-flow errors.

1: Initialize $C^\sharp = \emptyset$; /* *The detected weak-configurations.* */
2: Construct a GRG of N, i.e., $GRG(N) = (V^+, E^+, \ell^+)$.
3: **for each** $c \in R(c_0)$ such that $c \notin C^\sharp$ **do**
4: $C^\sharp$.add(c);
5: **if** $\exists t \in T : \neg c[t \wedge m[t\rangle \wedge \exists d \in Rd(t) \cup De(t) : \sigma(d) = \perp$ **then**
6: **print** *Missing data*;
7: **end if**
8: −
9: **if** $\exists t_1, t_2 \in T : t_1 || t_2 \wedge (Rd(t_1) \cup Wr(t_1) \cup De(t_1)) \cap (Wr(t_2) \cup De(t_2)) \neq \emptyset$ **then**
10: **print** *Inconsistent Data* between t_1 and t_2;
11: **end if**
12: −
13: Get all edges from c, i.e., $E_1 = \{(c, c') \mid (c, c') \in E^+\}$;
14: **for each** $(c, c') \in E_1$ **do**
15: **if** $c[t\rangle c'$ and $Wr(t) \neq \emptyset$ **then**
16: **for each** $v \in Wr(t)$ **do**
17: $CT = \emptyset$; /* *The traversed weak-configurations* */
18: Set $c_r = c_w = c_d = \emptyset$;
19: FindRWDConfigs$(v, c', GRG(\Sigma), CT)$;
/* *As shown in Procedure_1, it is used to compute* c_r, c_w and c_d */
20: **if** $c_w \neq \emptyset$ **then**
21: **print** *Lost Data*;
22: **end if**
23: −
24: **if** $c_d \neq \emptyset$ **then**
25: **print** *Redundant Data*;
26: **end if**
27: −
28: **if** $c_r = c_w = c_d = \emptyset$ **then**
29: **print** *Redundant Data*;
30: **end if**
31: −
32: **end for**
33: **end if**
34: **end for**
35: **end for**

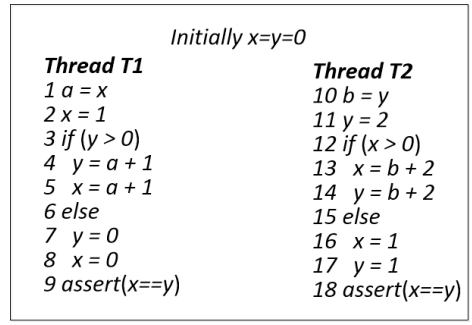


Figure 2. Pseudo-codes of a multi-thread program

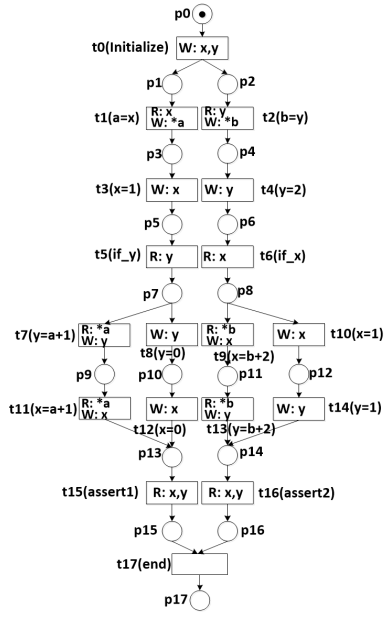
Transition ID	Codes	Read Data	Write Data
t_0	$x = y = 0$	–	x, y
t_1	$a = x$	x	a
t_2	$b = y$	y	b
t_3	$x = 1$	–	x
t_4	$y = 2$	–	y
t_5	if$(y > 0)$	y	–
t_6	if$(x > 0)$	x	–
t_7	$y = a + 1$	a	y
t_8	$y = 0$	–	y
t_9	$x = b + 2$	b	x
t_{10}	$x = 1$	–	x
t_{11}	$x = a + 1$	a	x
t_{12}	$x = 0$	–	x
t_{13}	$y = b + 2$	b	y
t_{14}	$y = 1$	–	y
t_{15}	assert1$(x == y)$	x, y	–
t_{16}	assert2$(x == y)$	x, y	–
t_{17}	end	–	–

Table 1. Program codes and data operations

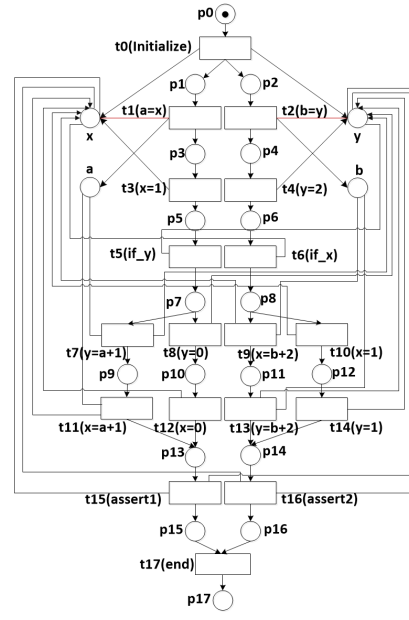
tively use a PN-DO, a contextual net and a Petri net without read/write arcs to model this program, we can get a comparison between them, which are shown in Figure 3 and Table 3. It is clear that WFD-net needs a smaller space than others.

ID	Guard	ID	Guard
t_7	$g_1(y) : y > 0$	t_8	$\neg g_1(y) : y <= 0$
t_9	$g_2(x) : x > 0$	t_{10}	$\neg g_2(x) : x <= 0$

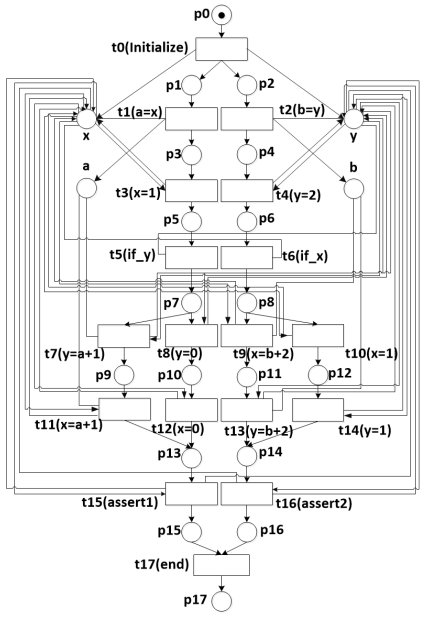
Table 2. Guards over transitions



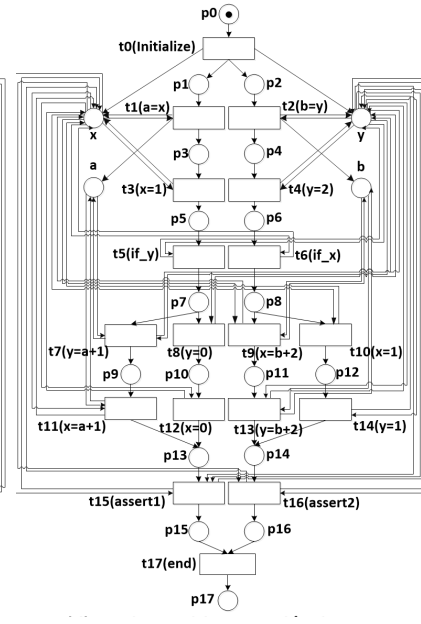
a) WFD-net



b) PN-DO



c) Contextual net

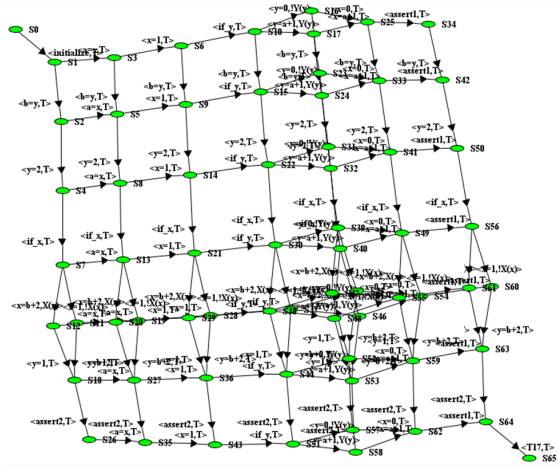


d) Petri net without read/write arcs

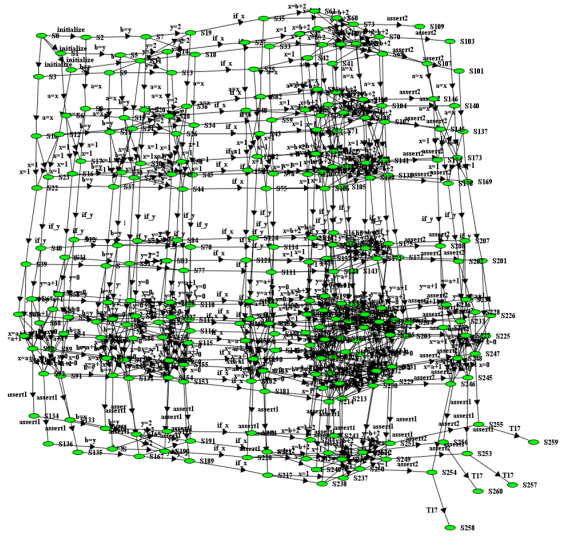
Figure 3. Different Petri nets for modeling the multi-thread program in Figure 2

	Nodes (Place & Transition)	Arcs
WFD-net [22, 26, 30]	36	38
PN-DO [31]	40	64
Contextual net [12]	40	74
Petri net without read/write arcs [2]	40	86

Table 3. The comparison between WFD-net and another Petri nets



a)



b)

Figure 4. a) GRG graph; b) the classical reachability graph

	(Weak) Configurations	Arcs	Time [ms]
RG	261	712	1 001
GRG	66	133	266

Table 4. The result comparison between RG and GRG

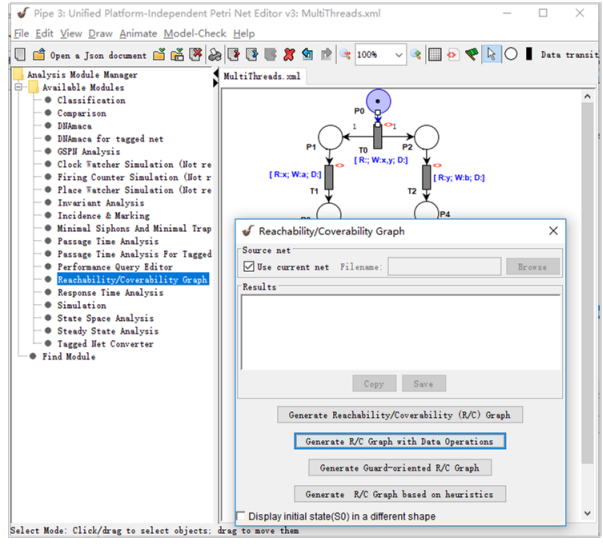


Figure 5. Our developed tool

As for the WFD-net in Figure 3 a), we utilize our tool [29] (see Figure 5) to construct its guard-driven reachability graph (GRG) and classical reachability graph (CRG), which are respectively shown in Figures 4 a) and 4 b). Table 4 gives a comparison between them in terms of state space and running time. Obviously, the former has an advantage over the latter.

Based on the GRG in Figure 4 a), we check the data-flow errors according to Algorithm 1. Table 5 lists these results. From these results, we can see that there exist some errors of inconsistent data because t_3 and t_9 (resp. t_{10}) are concurrent at the weak-configuration S_{13} and they satisfy $Wr(t_3) \cap Wr(t_9) = \{x\}$ (resp. $Wr(t_3) \cap Wr(t_{10}) = \{x\}$). The transitions t_4 and t_7 (resp. t_8) also suffer from the errors of inconsistent data. Moreover, t_{11} (resp. t_{12}) overwrites some data into x at weak-configurations S_1^+ after firing t_3, and t_{13} (resp. t_{14}) overwrites some data into y at weak-configurations S_2^+ after firing t_4. Therefore, there are some errors of lost data.

Data-Flow Errors	Weak-Configurations	Illustration
Missing Data	–	–
Inconsistent Data	S_{13}	t_3 and t_9 (or t_{10}) concurrently write some data into x
	S_{15}	t_4 and t_7 (or t_8) concurrently write some data into y
Redundant Data	–	–
Lost Data	$S_1^+ : \{S_{16}, S_{17}, S_{23}, S_{24}, S_{31}, S_{32}, S_{39}, S_{40}, S_{45}$–$S_{48}, S_{52}, S_{53}, S_{57}, S_{58}\}$	t_{11} (resp. t_{12}) overwrites some data into x
	$S_2^+ : \{S_{11}, S_{12}, S_{19}, S_{20}, S_{28}, S_{29}, S_{37}, S_{38}, S_{45}$–$S_{48}, S_{54}, S_{55}, S_{60}, S_{61}\}$	t_{13} (resp. t_{14}) overwrites some data into y

Table 5. The result of checking data-flow errors

5 EXPERIMENTS

We do a set of experiments in order to compare RG- and GRG-based methods for checking data-flow errors in terms of state space and runtime.

In our experiments, a tool is developed to generate RGs and GRGs of any bounded WFD-nets. It is based on PIPE (Platform Independent Petri Net Editor) [6], which is an open source tool of Petri net. Our tool can draw, edit, import and export a WFD-net.

Our experimental benchmarks are listed as follows:

- *KIT*[3] is a data set of BPMN 2.0 process models that describes 11 scenarios (including the business processes $BP_1 \sim BP_{11}$) with data specifications.
- *SystemC*, *blanc2010race* illustrates a SystemC (a modeling language) module.
- *AddGlobal*, *Sinha2010Staged* is an example of concurrency bugs when multi-threads access shared variables.

We utilize a PC with Intel Core i5-2400 CPU (3.10 GHz) and 4.0 GB memory to do experiments. We first use WFD-nets to model these benchmarks in our tool, and then respectively obtain their RGs and GRGs.

Based on GRGs, we can check data-flow errors. Table 6 presents the results of our experiments for all benchmarks. Obviously, the scale of GRG is much smaller than RG. Meanwhile, our GRG-based method spends less time to produce a GRG than the RG-based method. Naturally, the former has an advantage over the latter in terms of checking data-flow errors.

[3] `http://dbis.ipd.kit.edu/2134.php`, von2014detecting

Benchmark	RG			GRG			Errors
	Nos. of States	Nos. of Arcs	Time of RG	Nos. of States	Nos. of Arcs	Time of GRG	
BP_1	13	13	75.2	12	12	60.8	R
BP_2	17	18	70.6	16	16	65.8	R
BP_3	21	28	85.2	17	21	73.1	–
BP_4	15	14	70.3	15	14	69.4	–
BP_5	10	11	67.7	9	9	59.3	R
BP_6	23	43	73.3	18	30	62.7	R
BP_7	12	13	51.4	11	11	42.6	R
BP_8	103	103	318.5	29	28	71.8	–
BP_9	16	15	55.5	14	13	49.7	R
BP_{10}	36	40	73.3	30	32	66.0	–
BP_{11}	111	218	1 042.3	29	34	73.2	R
SystemC	33	62	76.6	25	39	62.5	I, L
AddGlobal	50	101	125.1	30	37	72.8	I, L

[1] Time: (ms).
[2] Errors: "I" denotes inconsistent data, "L" represents lost data, and "R" means redundant data.

Table 6. Experimental results

6 CONCLUSION

Petri net is widely used to check data-flow errors in workflow systems. As a special kind of Petri net, WFD-net is prominent in the modeling of control- and data-flows of business processes. Hence, we use a WFD-net to model workflow systems and its reachability graph to check data-flow errors in this paper. However, the classical reachability graphs of WFD-nets easily suffer from the problems of state space explosion and pseudo states. In order to avoid these problems, we propose a GRG-based method for checking data-flow errors. On one hand, our modeling method of WFD-net takes a smaller space than contextual net and PN-DO. On the other hand, our GRG-based method can effectively reduce the state space and avoid pseudo states in comparison with the classical reachability graph.

In the future work, we plan to do the following studies:

1. we utilize some existing techniques to reduce the scale of GRG, e.g., binary decision diagram (BDD) [5], abstraction [20] and partial order reduction [7]; and
2. we explore the unfolding-based technique of WFD-net [30] to check data-flow errors.

Acknowledgements

This paper was supported by Zhejiang Provincial Natural Science Foundation of China (Grant No. LQ20F020002), and in part by the Key Laboratory of Embedded

System and Service Computing (Ministry of Education) (Grant No. ESSCKF 2019-02).

REFERENCES

[1] van der Aalst, W. M. P.—van Hee, K. M.—ter Hofstede, A. H. M.—Sidorova, N.—Verbeek, H. M. W.—Voorhoeve, M.—Wynn, M. T.: Soundness of Workflow Nets: Classification, Decidability, and Analysis. Formal Aspects of Computing, Vol. 23, 2011, No. 3, pp. 333–363, doi: 10.1007/s00165-010-0161-4.

[2] Awad, A.—Decker, G.—Lohmann, N.: Diagnosing and Repairing Data Anomalies in Process Models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (Eds.): Business Process Management Workshops (BPM 2009). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 43, 2009, pp. 5–16, doi: 10.1007/978-3-642-12186-9_2.

[3] Baldan, P.—Bruni, A.—Corradini, A.—König, B.—Rodríguez, C.—Schwoon, S.: Efficient Unfolding of Contextual Petri Nets. Theoretical Computer Science, Vol. 449, 2012, pp. 2–22, doi: 10.1016/j.tcs.2012.04.046.

[4] Blanc, N.—Kroening, D.: Race Analysis for SystemC Using Model Checking. ACM Transactions on Design Automation of Electronic Systems (TODAES), Vol. 15, 2010, No. 3, Art. No. 21, doi: 10.1145/1754405.1754406.

[5] Butler, J. T.—Sasao, T.—Matsuura, M.: Average Path Length of Binary Decision Diagrams. IEEE Transactions on Computers, Vol. 54, 2005, No. 9, pp. 1041–1053, doi: 10.1109/TC.2005.137.

[6] Dingle, N. J.—Knottenbelt, W. J.—Suto, T.: PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets. ACM SIGMETRICS Performance Evaluation Review, Vol. 36, 2009, No. 4, pp. 34–39, doi: 10.1145/1530873.1530881.

[7] Bokor, P.—Kinder, J.—Serafini, M.—Suri, N.: Supporting Domain-Specific State Space Reductions Through Local Partial-Order Reduction. 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). IEEE Computer Society, 2011, pp. 113–122, doi: 10.1109/ASE.2011.6100044.

[8] Guo, X.—Sun, S. X.—Vogel, D.: A Dataflow Perspective for Business Process Integration. ACM Transactions on Management Information Systems, Vol. 5, 2014, No. 4, Art. No. 22, 33 pp., doi: 10.1145/2629450.

[9] Huang, J.—Zhang, C.—Dolby, J.: CLAP: Recording Local Executions to Reproduce Concurrency Failures. ACM SIGPLAN Notices, Vol. 48, 2013, No. 6, pp. 141–152, doi: 10.1145/2499370.2462167.

[10] Jensen, K.—Kristensen, L. M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer Science and Business Media, 2009, doi: 10.1007/b95112.

[11] Jiang, F.-C.—Hsu, C.-H.—Wang, S.: Logistic Support Architecture with Petri Net Design in Cloud Environment for Services and Profit Optimization. IEEE Transactions on Services Computing, Vol. 10, 2017, No. 6, pp. 879–888, doi: 10.1109/TSC.2016.2514506.

[12] Kähkönen, K.—Heljanko, K.: Testing Programs with Contextual Unfoldings. ACM Transactions on Embedded Computing Systems (TECS), Vol. 17, 2018, No. 1, Art. No. 23, doi: 10.1145/2810000.

[13] Kähkönen, K.—Saarikivi, O.—Heljanko, K.: Unfolding Based Automated Testing of Multithreaded Programs. Automated Software Engineering, Vol. 22, 2015, No. 4, pp. 475–515, doi: 10.1007/s10515-014-0150-6.

[14] Luan, W.—Qi, L.—Zhao, Z.—Liu, J.—Du, Y.: Logic Petri Net Synthesis for Cooperative Systems. IEEE Access, Vol. 7, 2019, pp. 161937-161948, doi: 10.1109/ACCESS.2019.2950971.

[15] Meda, H. S.—Sen, A. K.—Bagchi, A.: On Detecting Data Flow Errors in Workflows. ACM Journal of Data and Information Quality, Vol. 2, 2010, No. 1, Art. No. 4, 31 pp., doi: 10.1145/1805286.1805290.

[16] Montanari, U.—Rossi, F.: Contextual Nets. Acta Informatica, Vol. 32, 1995, No. 6, pp. 545–596, doi: 10.1007/BF01178907.

[17] Moutinho, F.—Gomes, L.: Asynchronous-Channels within Petri Net-Based GALS Distributed Embedded Systems Modeling. IEEE Transactions on Industrial Informatics, Vol. 10, 2014, No. 4, pp. 2024–2033, doi: 10.1109/TII.2014.2341933.

[18] Pradhan, A.—Joshi, R. K.: A Taxonomy of Consistency Models in Dynamic Migration of Business Processes. IEEE Transactions on Services Computing, Vol. 11, 2018, No. 3, pp. 562–579, doi: 10.1109/TSC.2017.2735413.

[19] Sadiq, S.—Orlowska, M.—Sadiq, W.—Foulger, C.: Data Flow and Validation in Workflow Modelling. Proceedings of the 15th Australasian Database Conference (ADC '04), Vol. 27, Australian Computer Society, Inc., 2004, pp. 207–214.

[20] Schlich, B.: Model Checking of Software for Microcontrollers. ACM Transactions on Embedded Computing Systems, Vol. 9, 2010, No. 4, Art. No. 36, 27 pp., doi: 10.1145/1721695.1721702.

[21] Sharma, D.—Pinjala, S.—Sen, A. K.: Correction of Data-Flow Errors in Workflows. Proceedings of the 25th Australasian Conference on Information Systems (ACIS), 2014, 10 pp.

[22] Sidorova, N.—Stahl, C.—Trčka, N.: Soundness Verification for Conceptual Workflow Nets with Data: Early Detection of Errors with the Most Precision Possible. Information Systems, Vol. 36, 2011, No. 7, pp. 1026–1043, doi: 10.1016/j.is.2011.04.004.

[23] Sinha, N.—Wang, C.: Staged Concurrent Program Analysis. Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE '10), 2010, pp. 47–56, doi: 10.1145/1882291.1882301.

[24] Sun, S. X.—Zhao, J. L.—Nunamaker, J. F.—Sheng, O. R. L.: Formulating the Data-Flow Perspective for Business Process Management. Information Systems Research, Vol. 17, 2006, No. 4, pp. 374–391, doi: 10.1287/isre.1060.0105.

[25] Trcka, N.—van der Aalst, W.—Sidorova, N.: Workflow Completion Patterns. 2009 IEEE International Conference on Automation Science and Engineering (CASE 2009), 2009, pp. 7–12, doi: 10.1109/COASE.2009.5234170.

[26] Trčka, N.—van der Aalst, W. M. P.—Sidorova, N.: Data-Flow Anti-Patterns: Discovering Data-Flow Errors in Workflows. In: van Eck, P., Gordijn, J.,

Wieringa, R. (Eds.): Advanced Information Systems Engineering (CAiSE 2009). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5565, 2009, pp. 425–439, doi: 10.1007/978-3-642-02144-2_34.

[27] Varea, M.—Al-Hashimi, B. M.—Cortés, L. A.—Eles, P.—Peng, Z.: Dual Flow Nets: Modeling the Control/Data-Flow Relation in Embedded Systems. ACM Transactions on Embedded Computing Systems, Vol. 5, 2006, No. 1, pp. 54–81, doi: 10.1145/1132357.1132360.

[28] Von Stackelberg, S.—Putze, S.—Mülle, J.—Böhm, K.: Detecting Data-Flow Errors in BPMN 2.0. Open Journal of Information Systems, Vol. 1, 2014, No. 2, pp. 1–19.

[29] Xiang, D.—Liu, G.—Yan, C.—Jiang, C.: Detecting Data Inconsistency Based on the Unfolding Technique of Petri Nets. IEEE Transactions on Industrial Informatics, Vol. 13, 2017, No. 6, pp. 2995–3005, doi: 10.1109/TII.2017.2698640.

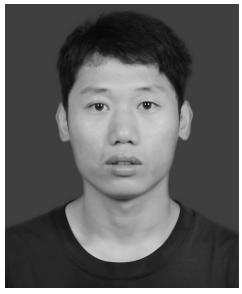
[30] Xiang, D.—Liu, G.—Yan, C.—Jiang, C.: A Guard-Driven Analysis Approach of Workflow Net with Data. IEEE Transactions on Services Computing, 2019, doi: 10.1109/TSC.2019.2899086.

[31] Xiang, D.—Liu, G.—Yan, C.—Jiang, C.: Detecting Data-Flow Errors Based on Petri Nets with Data Operations. IEEE/CAA Journal of Automatica Sinica, Vol. 5, 2018, No. 1, pp. 251–260, doi: 10.1109/JAS.2017.7510766.

[32] Yang, B.—Liu, G.—Xiang, D.—Yan, C.—Jiang, C.: A Heuristic Method of Detecting Data Inconsistency Based on Petri Nets. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 202–208, doi: 10.1109/SMC.2018.00045.

[33] You, D.—Wang, S.—Seatzu, C.: Verification of Fault-Predictability in Labeled Petri Nets Using Predictor Graphs. IEEE Transactions on Automatic Control, Vol. 64, 2019, No. 10, pp. 4353–4360, doi: 10.1109/TAC.2019.2897272.

[34] Zou, J.—Liu, X.—Sun, H.—Zeng, J.: Live Instance Migration with Data Consistency in Composite Service Evolution. 2010 6th World Congress on Services, IEEE, 2010, pp. 653–656, doi: 10.1109/SERVICES.2010.76.



Dongming Xiang received his Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2018. He is currently Lecturer with the Department of Computer Science and Technology, Zhejiang Sci-Tech University. His research interests include model checking, Petri nets, business process management, and service computing.

Guanjun Liu received his Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011. He was Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013. He was Post-Doctoral Research Fellow with the Humboldt University zu Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently Professor with the Department of Computer Science and Technology, Tongji University. He has authored over 80 papers including 15 papers in IEEE/ACM Transactions and one book entitled Liveness of Petri Nets and Its Application (Tongji University Press, 2017). His research interests include Petri net theory, model checking, Web service, workflow, discrete event systems, and information security.

Computing and Informatics, Vol. 39, 2020, 213–245, doi: 10.31577/cai_2020_1-2_213

ANALYSIS AND APPLICATION OF MIN-COST TRANSITION SYSTEMS TO BUSINESS PROCESS MANAGEMENT

Xiwen Feng, Dong Han

College of Energy and Mining Engineering
State Key Laboratory of Mining Disaster Prevention and Control
Co-founded by Shandong Province and the Ministry of Science and Technology
National Demonstration Center for Experimental Mining Engineering Education
Shandong University of Science and Technology
Qingdao 266590, China

Yinhua Tian*

Department of Information Engineering
Shandong University of Science and Technology
Taian 271000, China
e-mail: skdxxtyh@163.com

Abstract. To improve the efficiency of conformance checking in process mining, new alignment approaches are presented between event logs and process models based on the min-cost transition systems of Petri nets. An algorithm is presented to obtain the transition system with the minimum cost based on the product of the event net and process net. The min-cost transition system is a directed acyclic graph, where the paths from the initial node to the final node include all optimal alignments between the trace and the process model based on the given cost function. Two algorithms are proposed to calculate an optimal alignment and all optimal alignments, respectively. All algorithms are implemented in ProM platform. After a series of the simulation experiments, the feasibility and effectiveness of the proposed approaches are illustrated.

* Corresponding author

Keywords: Petri nets, event logs, process models, transition systems, business process management

Mathematics Subject Classification 2010: 68-Q05

1 INTRODUCTION

Business process management (BPM) aims to provide the unified modeling, running and monitoring environment for business processes from information technology and management technology [1]. In order to manage business processes better, the increasing enterprises and organizations utilize models to describe business processes. So, they can automatically implement processes, interact with participants and evaluate business processes [2]. Nowadays, most of the enterprises and organizations have established information management systems. With the continuous implementation of business processes, information systems will generate a large number of files on event logs. These files record massive data related to the execution processes, which are used to further analyze the performance of business processes in order to operate enterprises better [3].

Along with the increasing demand for business intelligence automatically extracted from event logs, process mining plays still more and more important role in business process management [4, 5, 6]. In enterprises, complete information management systems require high fitness between process models and event logs. However, there are always some deviations between the event logs recorded in the information system and the business process based models. Due to the deviations, event logs cannot be correctly replayed by the models. Because models are effective tools to identify and simulate the information systems, conformance checking becomes the necessary means to measure the compliance of process models and event logs.

At present, there are many conformance checking technologies between given models and event logs [7, 8, 9, 10, 11, 12, 13, 14, 15]. Alignment is one of the most advanced approaches. The main idea of alignment is to locate the deviations between process models and event logs. In general, the alignment results with the minimum deviations are considered to be the optimal alignments.

Through the analysis of various alignment approaches [16, 17, 18, 19, 20, 21, 22, 23], we find the existing problems of the current ones, mainly including: too large search space; high complexity of the search algorithms; unable to find the required and accurate optimal alignments; unable to find all the optimal alignments, and so on.

In our opinion, the alignment approaches can be divided into two steps:

1. generate a search space containing the optimal alignments according to traces and process models;
2. search for the optimal alignments in the search space based on the given cost function [24, 25].

The main framework of our approaches is shown in Figure 1.

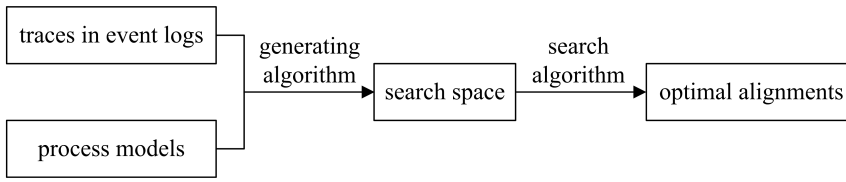


Figure 1. Framework of the alignment approaches

In the framework, the alignment approaches take the generating algorithm as preprocessing. When evaluating the performance of the alignment approaches, only the efficiency of the search algorithm is considered, including the mean computation time and the mean queued states. The search algorithms are widely used, well established and relatively easy to understand and implement. It is feasible to find and use an efficient search algorithm. Hence, to improve the efficiency of alignment approaches, the main means is to reduce the search space. In addition, if the search space is generated based on the trace and the process model but ignoring the cost function and other factors, the search space will include some redundant nodes which cannot reach to the optimal alignment. So, reducing the search space is the most effective approach to improve the efficiency of calculating optimal alignments.

In this paper, new alignment approaches are proposed, which can obtain the minimum space containing all the optimal alignments. The greatest advantage of our approaches is that the space contains only the useful nodes that can reach the optimal alignments, but no other redundant nodes. The main research objects of our approaches are traces in event logs and Petri nets-based models. Our approaches can generate a graph, in which a path from the initial node to the final node corresponds to an optimal alignment between traces and process models. The graph is called the min-cost transition system. By the means of the min-cost transition system, our approaches not only simplify the calculation procedures of optimal alignments, but also save the memory occupied by the search space.

The rest of this paper is organized as follows. Section 2 recalls some basic notions of Petri nets, event logs, alignment, and so on. The generating algorithm of the min-cost transition system is presented in Section 3. Section 4 proposes the approaches how to search for an optimal alignment and all optimal alignments in the min-cost transition system, respectively. Simulation experiments are done to illustrate the feasibility and effectiveness of our approaches in Section 5. Section 6 draws the conclusion and the future work.

2 PRELIMINARIES

In this section, we introduce the basic notations for multi-set, trace, event log, Petri nets, and so on.

A multi-set is a special set that allows multiple occurrences of the same element [26]. In a multi-set, only the number of occurrences of each element is concerned, and the order of occurrence of the elements is neglected.

Definition 1 (Multi-set)**.** Let S be a set. A *multi-set* S' over S is a mapping function $S' : S \to N^{0+}$.

N^{0+} refers to a set of zero and positive integers. Symbol $\emptyset$ denotes empty multi-set, and $\in$ denotes the inclusion relationship between elements and multi-sets. $\mathbb{B}(S)$ denotes the set of all multi-sets over a finite set S. $|S|$ is defined as the size of multi-set S.

Sequence is one of the most natural and appropriate ways to present traces in event logs [26].

Definition 2 (Sequence)**.** Let S be a set. σ is a finite *sequence* over S, written as $\sigma = \langle \sigma[1], \sigma[2], \sigma[3], \ldots, \sigma[n] \rangle$. σ is represented by listing its elements $\sigma[1]$, $\sigma[2]$, $\sigma[3]$, $\ldots$, $\sigma[n]$, where $\sigma[i] \in S (1 \leq i \leq n)$.

S^* denotes the set of all finite sequences over set S. $\langle\rangle$ denotes an empty sequence. Supposed that σ is a sequence over S, $\sigma[i]$ refers to the i^{th} element of σ. $\sigma[i] \in \sigma$ denotes the inclusion relationship, and $|\sigma|$ denotes the length of σ.

Let $x \in (S \times S)$ be a tuple of 2 elements(i.e., pair), $\pi_i(x)$ refers to the i^{th} element of x. For all $\sigma \in (S \times S)^*$, $\pi_i(\sigma) = \langle \pi_i(\sigma[1]), \pi_i(\sigma[2]), \pi_i(\sigma[3]), \ldots, \pi_i(\sigma[|\sigma|]) \rangle$. For all $Q \subseteq S$, $\sigma_{\downarrow Q}$ refers to the projection of $\sigma \in S^*$ on Q.

For any sequence σ over S, $\partial_{\text{set}}(\sigma) = \{\sigma[1], \sigma[2], \sigma[3], \ldots, \sigma[n]\}$, $\partial_{\text{multiset}}(\sigma) = [\sigma[1], \sigma[2], \sigma[3], \ldots, \sigma[n]]$. ∂_{set} converts a sequence into a set and $\partial_{\text{multiset}}$ converts a sequence into a multi-set. These conversions allow us to treat sequences as sets and multi-sets when needed.

A large number of events are recorded in the current information system and stored in the logs. An event log consists of cases and cases consist of events. The events for a case are represented in the form of a trace [15].

Definition 3 (Trace, Event log)**.** Let A be a set of activities. A *trace* $\sigma \in A^*$ is a process instance, i.e., a sequence of activities. An *event log* $L \in \mathbb{B}(A^*)$ is a multi-set of traces.

Petri nets are the most frequently used process modeling languages allowing for the modeling of concurrency [27]. The state of a Petri net is indicated by the distribution of tokens over places, and it is referred to as marking [28, 29, 30, 31, 32].

Transitions of Petri nets can be labeled with activities. Once the mapping relationship between transitions and activities is established, the transitions are related to the activities in the actual business process [15, 26].

Definition 4 (Labeled Petri net System)**.** Let A be a set of activities. A *labeled Petri net system* over A is a tuple $N = (P, T; F, \alpha, m_{\text{i}}, m_{\text{f}})$, where

1. P is the set of places;

2. T is the set of transitions, and $P \cup T \neq \emptyset$, $P \cap T = \emptyset$;
3. $F \subseteq (P \times T) \cup (T \times P)$ is an arc set between transitions and places, i.e., a flow relation;
4. $\alpha : T \to A^{\tau}$ is a function that maps transitions to labels, and τ denotes the invisible transition, $A^{\tau} = A \cup \{\tau\}$;
5. m_{i} and m_{f} are the initial marking and final marking, respectively.

For convenience, in the remainder of this paper, labeled Petri net system is abbreviated as Petri net.

Definition 5 (Per-set, Post-set)**.** Let $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ be a Petri net. For $\forall x \in P \cup T$,

$$^{\bullet}x = \{y \mid y \in P \cup T \wedge (y, x) \in F\}$$

$$x^{\bullet} = \{y \mid y \in P \cup T \wedge (x, y) \in F\}$$

where $^{\bullet}x$ represents the pre-set of x, $x^{\bullet}$ represents the post-set of x.

We describe the transition firing rules by using the multi-sets of places. For any reachable state $m \in \mathbb{B}(P)$, the transition firing rules of Petri net $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ are as follows:

1. For transition $t \in T$, if $^{\bullet}t \in m$, t is enabled denoted by $m[t >$; and
2. If $m[t >$, it means that the transition t can occur under the marking m, and after the transition t is fired, a new marking m' is generated, denoted by $m[t > m'$, where $m' = m \biguplus t^{\bullet} -^{\bullet} t$.

The event net of a trace is a Petri net with a linear structure, such that each transition in the net represents a unique activity occurrence in the trace. After traces are modeled as event nets, all possible movements are explicitly modeled by taking the product of two Petri nets, which are the event net and process net. The product of two Petri nets is the union of both nets with extra synchronous transitions, which are constructed by pairing transitions in event net with transitions in process net which have the same labels [15]. In the product of two Petri nets, all the places, transitions and arcs of the event net and process net are preserved in the product of two Petri nets.

An alignment between the process model and the trace is a movement sequence, and the move relates an event in the trace to an activity in the process model [15]. The synchronous move means that an event recorded in the trace is allowed according to the modeled behavior. The log move means that a recorded event is not allowed by the modeled behavior of process model. The model move means that an event which should have been recorded according to the modeled behavior is missed in the trace. The log moves and model moves indicate the deviations between traces and process models.

The symbol $\Gamma_{\sigma,N}$ denotes the set of all alignments between σ and N.

Given a trace and a Petri net model, there may be several different alignments that can be constructed. In order to get the most suitable alignments, a cost function $c((a,t))$ should assign a certain value to each move. According to the assigned cost function, the alignments with the least total cost are called optimal alignments.

In this paper, the standard likelihood cost function $lc()$ is used to assign the cost to the moves, i.e., the cost value of a synchronous move, log move and model move is 0, 1 and 1, respectively [15].

The symbol $\Gamma^{o}_{\sigma,N,lc}$ denotes the set of all optimal alignments between σ and N based on the function $lc()$.

3 GENERATION OF MIN-COST TRANSITION SYSTEMS

Alignments indicate the deviations between the process model and the trace in the event log. To express the idea of the approaches presented in this paper more clearly, the given process model and trace are taken as examples to illustrate.

3.1 Log Model and Process Model

Let $A = \{a, b, c, d, e\}$ be a set of activities. Given an event log $L = [\sigma^{10}]$, where $\sigma = \langle a, e, d\rangle$. The event net is shown in Figure 2, which is built according to the definition of event net [15]. We call the event net as log model, denoted as $N_{\mathrm{lm}} = (P_{\mathrm{lm}}, T_{\mathrm{lm}}; F_{\mathrm{lm}}, \alpha_{\mathrm{lm}}, m_{\mathrm{i,lm}}, m_{\mathrm{f,lm}})$.

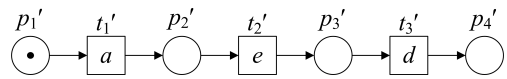


Figure 2. Log model N_{lm}

Given process model $N_{\mathrm{pm}} = (P_{\mathrm{pm}}, T_{\mathrm{pm}}; F_{\mathrm{pm}}, \alpha_{\mathrm{pm}}, m_{\mathrm{i,pm}}, m_{\mathrm{f,pm}})$, as shown in Figure 3.

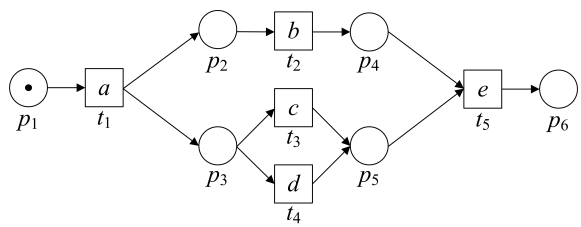


Figure 3. Process model N_{pm}

3.2 Product of Log Model and Process Model

The product model between the log model and the process model can be obtained. The product model consists of the log model, process model and synchronous transitions. The places, initial marking and final marking of the product model are the unions of the corresponding sets of the log model and the process model, respectively. Assuming that the log model $N_1 = (P_1, T_1; F_1, \alpha_1, m_{\mathrm{i},1}, m_{\mathrm{f},1})$ and the process model $N_2 = (P_2, T_2; F_2, \alpha_2, m_{\mathrm{i},2}, m_{\mathrm{f},2})$, the related information of the transitions in the product model is shown in Table 1. The arc relations can be established according to the pre-sets and the post-sets of transitions.

Transition	Type	Activity	Resource	Pre-Set	Post-Set
$(t_1, >>)$	log transition	$\alpha_1(t_1)$	$\{T_1\}$	${}^\bullet t_1$	$t_1^\bullet$
$(>>, t_2)$	model transition	$\alpha_2(t_2)$	$\{T_2 \mid \alpha(t_2) \neq \tau\}$	${}^\bullet t_2$	$t_2^\bullet$
(t_1, t_2)	synchronous transition	$\alpha_1(t_1) \setminus \alpha_2(t_2)$	$\{T_1 \times T_2 \mid \alpha(t_1) = \alpha(t_2) \neq \tau\}$	${}^\bullet t_1 \cup {}^\bullet t_2$	$t_1^\bullet \cup t_2^\bullet$
$(>>, t_2)$	invisible transition	$\alpha_2(t_2)$	$\{T_2 \mid \alpha(t_2) = \tau\}$	${}^\bullet t_2$	$t_2^\bullet$

Table 1. Transitions of the product of two Petri nets

Taking the log model N_{lm} and the process model N_{pm}, according to the conception of product of two Petri nets, the product model $N_{\mathrm{lm}*\mathrm{pm}} = (P_{\mathrm{lm}*\mathrm{pm}}, T_{\mathrm{lm}*\mathrm{pm}}; F_{\mathrm{lm}*\mathrm{pm}}, \alpha_{\mathrm{lm}*\mathrm{pm}}, m_{\mathrm{i,lm}*\mathrm{pm}}, m_{\mathrm{f,lm}*\mathrm{pm}})$ is built, as shown in Figure 4. According to Definition 4, the product model is also a Petri net.

3.3 Min-Cost Transition System of the Product Model

The transitions in the product model can be divided into four types: log transitions, model transitions, synchronous transitions and invisible transitions. Each transition is mapped to an activity, so the sort of transition also determines the sort of the activity. According to the standard likelihood cost function [15], we assign different weights to four types of transitions, and their corresponding relations are shown in Table 2.

Transition Type	Move Type	Weight Value
log transition	log move	1
model transition	model move	1
synchronous transition	synchronous move	0
invisible transition	invisible move	0

Table 2. Allocation of the weight value on transitions

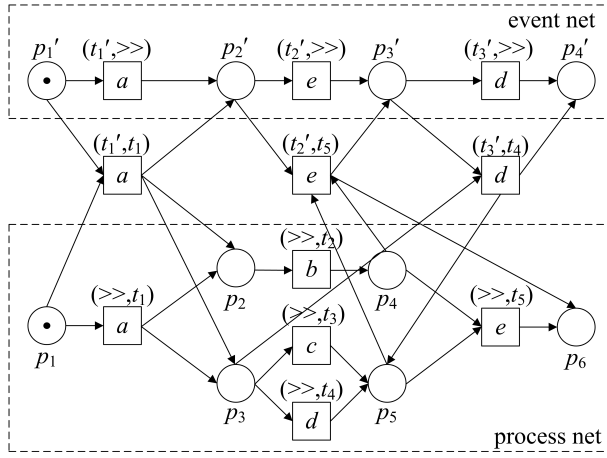


Figure 4. Product model $N_{\mathrm{lm}*\mathrm{pm}}$ between log model N_{lm} and process model N_{pm}

The product model is a Petri net, which can be performed for its reachable state graph. When calculating the reachable state graph, the weights of the transitions in the firing sequence are accumulated as the cost of the current reachable state. The min-cost reachable state graph can be obtained by counting the initial state, the minimum cost final state and all reachable states between them, which is called min-cost transition system. In the min-cost transition system, each node contains not only the current reachable marking but also its minimum cost.

Taking Petri net $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ as an example, we illustrate the generation process of its min-cost transition system and how the cost of a transition system is minimized. The main idea to generate the min-cost transition system for Petri net N is as follows:

Step 1: Suppose the minimum cost of the transition system is $mincost = +\infty$, and the state queue is $\emptyset$. Consider the state $(m_{\mathrm{i}}, 0)$ as the initial state and be enqueued, where m_{i} is the initial marking of Petri net N and the value 0 is the current cost because there has not any transition fired.

Step 2: Choose the state (m_x, c_x) as the current state, where $\nexists(m_x', c_x') \to (c_x > c_x')$. For all $t_x \in T$ that $m_x[t_x >$, generate the new state (m_y, c_y), where $m_x[t_x > m_y$ and $c_y = c_x + lc(t_x)$ ($lc(t_x)$ is the weight value of transition t_x).

Step 3: Examine the new generated state (m_y, c_y):

Step 3.1: If $m_y = m_{\mathrm{f}}$ and $mincost > c_y$, then $mincost = c_y$.

Step 3.2: If there is the existing state (m_y', c_y'), $m_y' = m_y$ and $c_y' = c_y$, then share the existing state; $m_y' = m_y$ and $c_y' < c_y$, then discard the generated state; $m_y' = m_y$ and $c_y' > c_y$, then delete the state (m_y', c_y') and enqueue the state (m_y, c_y).

Step 3.3: If $c_y > mincost$, then discard the generated state.

Step 4: Examine all the visited states, and delete the states without children.

Step 5: Continue to execute Step 4, until all the visited states have children.

Step 6: If there are unvisited states in the queue, jump to Step 2; else, the min-cost transition system is generated.

In the procedure mentioned above, Step 3.1 ensures that the min-cost transition system has a minimum cost. The last remained state whose marking is m_{f} is considered as the final state of the min-cost transition system, and its cost is the minimum cost. Step 3.2 ensures that there are not two states with the same markings and different costs. Step 3.3 guarantees that there are no states with the greater cost than that of the final state. Steps 4 and 5 guarantee that there are no states that cannot arrive at the final state. Hence, the procedure can guarantee that the cost of a transition system is minimized.

Min-cost transition system $G_{\mathrm{lm}*\mathrm{pm}}$ of product model $N_{\mathrm{lm}*\mathrm{pm}}$ can be obtained by preserving the valid states and the connecting edges between them. The specific transition system $G_{\mathrm{lm}*\mathrm{pm}}$ is shown in Figure 5.

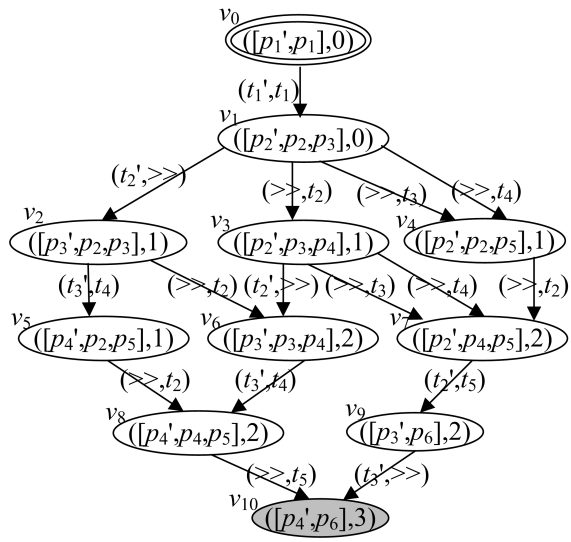


Figure 5. Min-cost transition system $G_{\mathrm{lm}*\mathrm{pm}}$

In the classic Petri nets, the marking represents the distribution of tokens in the places. However, in this paper, the state contains the marking as well as the cost in the min-cost transition system.

For arbitrary Petri nets, their structures may be very complicated and diverse. Here, we discuss a special structure for the Petri net and its influence on the min-cost

transition system. The Petri net contains cycles in which the cost of the transitions is 0. As a result, the min-cost transition system may contain cycles with cost 0, which results in countless paths between the initial node and the final node. In the context of practical application of our paper, the transitions with cost 0 are the invisible transitions in the Petri nets. The invisible transitions represent the activities that can never be observed, so the cycles containing only the invisible transitions have little meanings to the alignment results. In order to reach the final node from the initial node in a limited number of steps, we delete this kind of cycles from the min-cost transition system.

As shown in Figure 5, the min-cost transition system is a directed acyclic graph, which can also be called the min-cost reachability graph. In the graph, each node contains two attributes: one is the current marking of the product model that is represented by the multi-set of places; the other is the minimum cost of the state that is represented by a non-negative integer. The two attributes on the nodes are defined as the min-cost reachable state. In the graph, the label on the edge is the transition of the product model, which can be mapped to the move. There is one and only one node whose first attribute is the initial marking of the product net, which is called the min-cost initial state. There is one and only one node whose first attribute is the final marking of the product net, which is called the min-cost final state. Any node is on a path between the initial state and the final state.

3.4 Definitions of Min-Cost Transition System

Next, we discuss some special states in the transition system of the product model. And then, the definition of min-cost transition system is given and its basic properties are proved. For convenience, in this section, we agree as follows:

1. Petri nets are products of two Petri nets, so the concrete expression of transition is a tuple. However, in order to simplify the description, the symbol t is still used to represent the transition when the specific Petri net is not involved.
2. Transitions in the product nets can be mapped to moves, while moves can be mapped to real set by the standard likelihood cost function.

Hence, we can map the transitions to real set. Based on the weight assignment on the transitions in Table 2, the standard likelihood cost function $lc()$ is directly applied to the transitions, and its function value remains unchanged, which is called transition cost function.

Definition 6 (Reachable state with cost)**.** Let A be a set of activities. $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ is a Petri net over A. $lc()$ is a transition cost function. Supposing there is a transition firing sequence $t_1t_2t_3 \ldots t_n$ that makes $m_{\mathrm{i}}[t_1t_2t_3 \ldots t_n > m$, it is said that $(m, \sum_{\mathrm{i}=1}^{n} lc(t_i))$ is a *reachable state with cost*, denoted by m^c.

M^c is a set that includes all of the reachable states with cost, i.e., $m^c \in M^c$. Definition 6 shows that m^c is a 2-tuple, in which the first element is the reachable

marking of the Petri net, and the second one is the sum of the cost for each transition in the firing sequence that causes the Petri net from the initial state to the current state.

Definition 7 (Min-cost reachable state)**.** Let A be a set of activities. $N = (P, T; F, \alpha, m_\mathrm{i}, m_\mathrm{f})$ is a Petri net over A. $lc()$ is a transition cost function. Supposing there is a transition firing sequence $t_1 t_2 t_3 \dots t_n$ that makes $m_\mathrm{i}[t_1 t_2 t_3 \dots t_n > m$. $(m, \sum_{\mathrm{i}=1}^{n} lc(t_i))$ is a *min-cost reachable state* if and only if there is no transition firing sequence $t'_1 t'_2 t'_3 \dots t'_k$ that makes $m_\mathrm{i}[t'_1 t'_2 t'_3 \dots t'_k > m$ and $\sum_{\mathrm{i}=1}^{n} lc(t_i) > \sum_{j=1}^{k} lc(t'_j)$, denoted by $m^{\copyright}$.

$M^{\copyright}$ is a set that includes all of the min-cost reachable states, i.e., $m^{\copyright} \in M^{\copyright}$. In Petri nets, different transition sequences may reach the same reachable state. Each reachable state and its minimum cost constitute the min-cost reachable state. Obviously, $M^{\copyright} \subseteq M^{c}$.

For convenience, in the remainder of this paper, we abbreviate the min-cost reachable state as the reachable state.

Definition 8 (Min-cost initial state)**.** Let A be a set of activities. $N = (P, T; F, \alpha, m_\mathrm{i}, m_\mathrm{f})$ is a Petri net over A. $lc()$ is a transition cost function. Given $m^{\copyright}$ is a min-cost reachable state, $m^{\copyright}$ is called as the *min-cost initial state* if $\pi_1(m^{\copyright}) = m_\mathrm{i}$, denoted by $m_\mathrm{i}^{\copyright}$.

We abbreviate the min-cost initial state as the initial state. According to Definition 8, the first element of the initial state is the initial marking of the Petri net. Since no transition has been fired at present, the second element of the initial state is 0. Hence, $m_\mathrm{i}^{\copyright} = (m_\mathrm{i}, 0)$.

Definition 9 (Min-cost final state)**.** Let A be a set of activities. $N = (P, T; F, \alpha, m_\mathrm{i}, m_\mathrm{f})$ is a Petri net over A. $lc()$ is a transition cost function. Given $m^{\copyright}$ is a min-cost reachable state, $m^{\copyright}$ is called as the *min-cost final state* if $\pi_1(m^{\copyright}) = m_\mathrm{f}$, denoted by $m_\mathrm{f}^{\copyright}$.

We abbreviate the min-cost final state as the final state. According to Definition 9, the first element of the final state is the final marking of the Petri net. The second element of the final state is the minimum cost from the initial marking to the final marking for the Petri net. Hence, $m_\mathrm{f}^{\copyright} = (m_\mathrm{f}, \min(\{\sum_{\mathrm{i}=1}^{n} lc(t_i) \mid \forall t_1 t_2 t_3 \dots t_n \rightarrow m_\mathrm{i}[t_1 t_2 t_3 \dots t_n > m_\mathrm{f}\}))$, where $\min(S)$ is a function to find the minimum cost in the set S.

Definition 10 (Min-cost transition system)**.** Let A be a set of activities. $N = (P, T; F, \alpha, m_\mathrm{i}, m_\mathrm{f})$ is a Petri net over A. $lc()$ is a transition cost function. *Min-cost transition system* $G = (V, E)$ is a directed acyclic graph, where V is a finite node set and $E \subseteq (V \times V)$ is a finite set of directed edges between nodes. The graph satisfies the following conditions:

1. $V \subseteq M^{\copyright}$;

2. $\exists! v_{\mathrm{i}} \in V : (\forall v \in V : (v, v_{\mathrm{i}}) \notin E) \Rightarrow \left(v_{\mathrm{i}} = m_{\mathrm{i}}^{\copyright}\right)$;
3. $\exists! v_{\mathrm{f}} \in V : (\forall v \in V : (v_{\mathrm{f}}, v) \notin E) \Rightarrow \left(v_{\mathrm{f}} = m_{\mathrm{f}}^{\copyright}\right)$;
4. $\forall v \in V$: v is on the path from v_{i} to v_{f};
5. $\forall e \in E, w(e) : w(e) \in T$, where $w(e)$ is the weight of edge e.

Again, we abbreviate the min-cost transition system as the transition system. According to Definition 10, the transition system has such characteristics as follows:

1. Each node is labeled by the min-cost reachable state.
2. There is only one node that is the min-cost initial state in the graph, which is called as the initial node.
3. There is only one node that is the min-cost final state in the graph, which is called as the final node.
4. Any node in the graph is on the path from the initial node to the final node.
5. The weight of the edge in the graph is the name of the transition.

Next, we present Theorem 1 and Theorem 2 to illustrate the rationality of the min-cost transition system.

Theorem 1. Let $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ be a Petri net and its min-cost transition system be $G = (V, E)$. Given $m_1^{\copyright} \in V$ and $m_2^{\copyright} \in V$, if $m_1^{\copyright} \neq m_2^{\copyright}$, $\pi_1\left(m_1^{\copyright}\right) \neq \pi_1\left(m_2^{\copyright}\right)$.

Proof. For $\forall m_1^{\copyright} \in V$ and $\forall m_2^{\copyright} \in V$, if $m_1^{\copyright} \neq m_2^{\copyright}$, one of the following cases holds:

1. $\pi_1\left(m_1^{\copyright}\right) \neq \pi_1\left(m_2^{\copyright}\right)$ and $\pi_2\left(m_1^{\copyright}\right) \neq \pi_2\left(m_2^{\copyright}\right)$;
2. $\pi_1\left(m_1^{\copyright}\right) \neq \pi_1\left(m_2^{\copyright}\right)$ and $\pi_2\left(m_1^{\copyright}\right) = \pi_2\left(m_2^{\copyright}\right)$;
3. $\pi_1\left(m_1^{\copyright}\right) = \pi_1\left(m_2^{\copyright}\right)$ and $\pi_2\left(m_1^{\copyright}\right) \neq \pi_2\left(m_2^{\copyright}\right)$.

If case 1 or case 2 holds, the conclusion is found. Under case 3, supposed $\pi_2\left(m_1^{\copyright}\right) > \pi_2\left(m_2^{\copyright}\right)$, according to Definition 7, it is impossible that $m_1^{\copyright}$ is the min-cost reachable state; vice versa, so case 3 will never happen.

Hence, if $m_1^{\copyright} \neq m_2^{\copyright}$, $\pi_1\left(m_1^{\copyright}\right) \neq \pi_1\left(m_2^{\copyright}\right)$. □

Theorem 1 shows that the reachable markings of any two reachable states are different in the transition system. Hence, according to Theorem 1, both the initial node and the final node are unique.

Theorem 2. Let $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ be a Petri net and its min-cost transition system be $G = (V, E)$. For $\forall v \in V$, there must be a transition firing sequence $t_1 t_2 t_3 \ldots t_k$ that makes $m_{\mathrm{i}}[t_1 t_2 t_3 \ldots t_k > \pi_1(v)$. Similarly, there must be $t_{k+1} t_{k+2} t_{k+3} \ldots t_n$ that makes $\pi_1(v)[t_{k+1} t_{k+2} t_{k+3} \ldots t_n > m_{\mathrm{f}}$.

Proof. According to Definition 10, $V \subseteq M^{\textcircled{c}}$. For $\forall v \in V$, $v \in M^{\textcircled{c}}$. According to Definition 7, there must be a transition firing sequence $t_1t_2t_3 \ldots t_k$ that makes $m_{\mathrm{i}}[t_1t_2t_3 \ldots t_k > \pi_1(v)$.

We suppose that there is no transition firing sequence $t_{k+1}t_{k+2}t_{k+3} \ldots t_n$ that makes $\pi_1(v)[t_{k+1}t_{k+2}t_{k+3} \ldots t_n > m_{\mathrm{f}}$. If $\pi_2(v) > \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$, for $\forall t_i \in T$, $lc(t_i) \geq 0$, then it will never reach $m_{\mathrm{f}}^{\textcircled{c}}$ from v, so v can be deleted directly. This shows that $\pi_2(v) > \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$ is not founded. If $\pi_2(v) \leq \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$, we suppose that t_x can be fired under the reachable marking $\pi_1(v)$, and then $\pi_1(v)[t_x > v_x$. We consider all three cases:

1. Assume $v_x = m_{\mathrm{f}}^{\textcircled{c}}$, $\pi_1(v)[t_x > m_{\mathrm{f}}$ shows that the conclusion is rational.
2. Assume $\pi_2(v_x) > \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$, v_x will be deleted, and if v has no other child, v will also be deleted.
3. Assume $\pi_2(v_x) \leq \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$.

We consider v_x as v to continue the comparison process until there is no node v_n that makes $\pi_2(v_n) \leq \pi_2\left(m_{\mathrm{f}}^{\textcircled{c}}\right)$. Through the analysis, we can infer that for $\forall v \in V$, either v will be deleted because of the failure to reach the final node, or there will be a transition firing sequence $t_{k+1}t_{k+2}t_{k+3} \ldots t_n$ that makes $\pi_1(v)[t_{k+1}t_{k+2}t_{k+3} \ldots t_n > m_{\mathrm{f}}$.

Hence, $\forall v \in V : (\exists t_1t_2t_3 \ldots t_k \Rightarrow m_{\mathrm{i}}[t_1t_2t_3 \ldots t_k > \pi_1(v)) \wedge (\exists t_{k+1}t_{k+2}t_{k+3} \ldots t_n \Rightarrow \pi_1(v)[t_{k+1}t_{k+2}t_{k+3} \ldots t_n > m_{\mathrm{f}})$. □

Theorem 2 shows that any node is on the path from the initial node to the final node in the transition system, that is, any node is connected with the initial node and the final node.

3.5 Calculation of Min-Cost Transition Systems

After describing the generation process of the transition system through an example and presenting the definition of the transition system, a specific algorithm to realize the calculation of the transition system in this subsection, seen Algorithm 1.

Before giving the specific algorithm, in order to facilitate the explanation of the algorithm, the variables and functions used in the algorithm are introduced, as shown in Table 3 and Table 4, respectively.

Algorithm 1 The generation algorithm of min-cost transition systems (reachability graphs) of Petri nets according to the transition cost function

Input: Petri net model $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$, transition cost function $lc()$.
Output: Min-cost transition system $G = (V, E)$.
Initialize: $unvisitedSet \leftarrow \emptyset$, $cost \leftarrow +\infty$, $V \leftarrow \left\{m_{\mathrm{i}}^{\textcircled{c}}\right\}$, $E \leftarrow \emptyset$.

1: $unvisitedSet \leftarrow \left\{m_{\mathrm{i}}^{\textcircled{c}}\right\}$;

Variable	Data Type	Function Introduction
curnode	m^c	current node
newnode	m^c	new node
foundnode	m^c	existing node
cost	value	current minimum cost
unvisitedSet	set	to store the unvisited nodes
V	set	to store the nodes
E	set	to store the edges

Table 3. Variable declaration in Algorithm 1

Function	Parameter Type	Return Type	Function Introduction
Father(*node*)	*node* : m^c	m^c	to return the parent of *node*
Delete(*node*)	*node* : m^c	null	to delete the node without child, and check its ancestors recursively
AddNode(*node*)	*node* : m^c	null	to add the node *node*
AddEdge(*fathernode*, *node*)	*fathernode* : m^c *node* : m^c	null	to add the edge between node and its parent

Table 4. Function declaration in Algorithm 1

2: **while** ($unvisitedSet \neq \emptyset$) **do**
3: Choose the minimum cost node from *unvisitedSet* as current node *curnode*;
4: $unvisitedSet \leftarrow unvisitedSet - \{curnode\}$;
5: **for** (all ($t_i \in T$ **and** $\pi_1(curnode)[t_i >)$) **do**
6: $newnode \leftarrow (\pi_1(curnode)[t_i >, \pi_2(curnode) + lc(t_i))$;
7: **if** ($\pi_1(newnode) = m_\mathrm{f}$) **then**
8: **if** ($\pi_2(newnode) < cost$) **then**
9: **if** ($cost \neq +\infty$) **then**
10: Find the node $(m_\mathrm{f}, cost)$;
11: $Delete((m_\mathrm{f}, cost))$;
12: **end if**
13: $cost \leftarrow \pi_2(newnode)$;
14: $V \leftarrow V \cup \{newnode\}$;
15: $AddEdge(curnode, newnode)$;
16: **else**
17: **if** ($\pi_2(newnode) = cost$) **then**
18: Find the previous node $foundnode$ that is the same as *newnode*;
19: $AddEdge(curnode, foundnode)$;
20: **end if**
21: **end if**
22: **else**

```
23:         if (π2(newnode) > cost) then
24:           if (all transitions have been fired under π1(curnode) and curnode has
              no child) then
25:             Delete(curnode);
26:           end if
27:         else
28:           if (foundnode ∈ V and π1(foundnode) = π1(newnode)) then
29:             if (π2(foundnode) = π2(newnode)) then
30:               E ← E ∪ {(curnode, foundnode))};
31:             else
32:               if (π2(foundnode) < π2(newnode)) then
33:                 if (curnode has no child and all transitions have been fired under
                    π1(curnode)) then
34:                   Delete(curnode);
35:                 end if
36:               else
37:                 Delete(foundnode);
38:                 AddNode(newnode);
39:                 AddEdge(curnode, newnode);
40:               end if
41:             end if
42:           else
43:             AddNode(newnode);
44:             AddEdge(curnode, newnode);
45:           end if
46:         end if
47:       end if
48:     end for
49: end while
50: for (all cycles in G) do
51:     delete all the edges with cost 0;
52:     for (all nodes in the cycle ) do
53:       if (node has no out edge) then
54:         Delete(node);
55:       end if
56:     end for
57: end for
58: return  G = (V, E);
```

The computation complexity of the min-cost transition system of the Petri net is related to the number of the reachable states and that of the transitions fired by the Petri nets, which is a NP-hard problem. Although the min-cost transition system computed by this algorithm is a subgraph of the traditional reachable marking graph, its complexity is also very high. Especially when there are many transitions

with the concurrent relations in Petri nets, the number of reachable states increases exponentially, which even causes state space to explode.

Let $N = (P, T; F, \alpha, m_{\mathrm{i}}, m_{\mathrm{f}})$ be a Petri net. In this paper, N is considered to be sound if and only if $m_{\mathrm{f}} \in R(m_{\mathrm{i}})$, where $R(m_{\mathrm{i}})$ is the set which includes all the reachable markings from m_{i}.

The influence of the concurrent structures in the Petri net to its min-cost transition system is similar to the effect on its reachable marking graph. Due to the cost of the transitions, the transitions with the less cost will be fired in the choice structures and loop structures when generating the min-cost transition system. In this case, the scale of the min-cost transition system is mostly smaller than that of the reachable marking graph. However, in the sequence structures and concurrent structures, all the transitions should be fired, the min-cost transition system of the Petri net will be isomorphic to its reachable marking graph. Hence, if the Petri net is with the completely concurrent structures, the number of the state in the min-cost transition system will increase exponentially with the linear increase of the concurrent branches just as the reachable marking graph.

$m_{\mathrm{f}} \in R(m_{\mathrm{i}})$ is essential for Algorithm 1 to execute correctly. Too much concurrent branches in the Petri net maybe lead to state space explosion. Hence, in order to improve the availability of the algorithm, we only study the sound Petri nets with less concurrent transitions in this paper.

4 SEARCH ALGORITHM OF OPTIMAL ALIGNMENTS

The min-cost transition system of the product model can be obtained by Algorithm 1. In the transition system, the sequence of weights labeled on the directed edges of any path from the initial node to the final node corresponds to an optimal alignment between the trace and the model. Based on the min-cost transition system, two algorithms are presented in this section to calculate an optimal alignment and all optimal alignments, respectively.

4.1 Search Algorithm of an Optimal Alignment

In this subsection, we search for an optimal alignment in the min-cost transition system. In the generation process of the transition system, the states that cannot reach the final node are pruned, so all the nodes in the graph are valid.

In the transition system, as shown in Figure 5, the prefix alignment between the trace and the process model can be calculated according to the path from the initial node to any other node. Similarly, the optimal alignment between the trace and the process model can be inferred based on the path from the initial node to the final node in the graph.

In this paper, the path from the initial node to the final node is defined as the complete path, and the corresponding relationship between the complete path and the optimal alignment is proved.

Definition 11 (Complete path)**.** Let $G = (V, E)$ be a min-cost transition system. $m_{\mathrm{i}}^{\copyright}$ is the min-cost initial state and $m_{\mathrm{f}}^{\copyright}$ is the min-cost final state. A *complete path* is a sequence $\left\langle \left(m_{\mathrm{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{\mathrm{f}}^{\copyright}\right)\right\rangle$ of the min-cost reachable states, i.e., a path from $m_{\mathrm{i}}^{\copyright}$ to $m_{\mathrm{f}}^{\copyright}$, denoted by $m_{\mathrm{i}}^{\copyright} \Rightarrow m_{\mathrm{f}}^{\copyright}$.

Given a complete path, a complete movement sequence can be obtained through outputting all the weights labeled on the edges of the path and converting the weights into moves, referred to Definition 12.

Definition 12 (Complete movement sequence)**.** Let $G = (V, E)$ be a min-cost transition system. $\left\langle \left(m_{\mathrm{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{\mathrm{f}}^{\copyright}\right)\right\rangle$ is a complete path in G. A *complete movement sequence* is a sequence of successive moves corresponding to the weights on the edges of the complete path, denoted by λ.

Given a complete path, its corresponding movement sequence can be calculated, as detailed in Algorithm 2.

Algorithm 2 The algorithm to compute complete movement sequence λ of complete path $\left\langle \left(m_{\mathrm{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{\mathrm{f}}^{\copyright}\right)\right\rangle$.

Input: Complete path $\left\langle \left(m_{\mathrm{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{\mathrm{f}}^{\copyright}\right)\right\rangle$.
Output: Complete movement sequence λ.
Initialize: $\lambda \leftarrow \langle\rangle$.

1: Map complete path $\left\langle \left(m_{\mathrm{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{\mathrm{f}}^{\copyright}\right)\right\rangle$ to node path $\langle (v_1, e_1, v_2), \ldots, (v_n, e_n, v_{n+1})\rangle$;
2: $i \leftarrow 1$;
3: **while** $(i \neq n)$ **do**
4: $\quad e \leftarrow \pi_2((v_i, e_i, v_{i+1}))$;
5: $\quad t \leftarrow w(e)$;
6: $\quad$ **if** $(\pi_1(t) =$ " >> ") **then**
7: $\quad\quad x \leftarrow$ " >> ";
8: $\quad$ **else**
9: $\quad\quad x \leftarrow \alpha(t)$;
10: $\quad$ **end if**
11: $\quad y \leftarrow \pi_2(t)$;
12: $\quad \lambda \leftarrow \lambda \oplus < (x, y) >$;
13: $\quad i \leftarrow i + 1$;
14: **end while**
15: **return** λ;

Both the time complexity and space complexity of Algorithm 2 are related to the length of the complete path. Supposed that the length of the complete path is n in the transition system, both the time complexity and space complexity of Algorithm 2 are $O(n)$.

Taking transition system $G_{\text{lm}*\text{pm}}$ as an example, path $\langle(([p'_1, p_1], 0), (t'_1, t_1), ([p'_2, p_2, p_3], 0)), (([p'_2, p_2, p_3], 0), (t'_2, >>), ([p'_3, p_2, p_3], 1)), (([p'_3, p_2, p_3], 1), (t'_3, t_4), ([p'_4, p_2, p_5], 1)), (([p'_4, p_2, p_5], 1), (>>, t_2), ([p'_4, p_4, p_5], 2)), (([p'_4, p_4, p_5], 2), (>>, t_5), ([p'_4, p_6], 3))\rangle$ from $m_{\text{i}}^{\copyright}$ to $m_{\text{f}}^{\copyright}$ is a complete path. The weight sequence on the path is $\langle(t'_1, t_1), (t'_2, >>), (t'_3, t_4), (>>, t_2), (>>, t_5)\rangle$, and its corresponding movement sequence $\langle(a, t_1), (e, >>), (d, t_4), (>>, t_2), (>>, t_5)\rangle$ is a complete movement sequence. Obviously, this complete movement sequence is an optimal alignment between trace σ and process model $N_{\text{lm}*\text{pm}}$.

Theorem 3. Let $N_1 = (P_1, T_1; F_1, \alpha_1, m_{\text{i},1}, m_{\text{f},1})$ be a log model of trace σ and $N_2 = (P_2, T_2; F_2, \alpha_2, m_{\text{i},2}, m_{\text{f},2})$ be a process model. $N_3 = (P_3, T_3; F_3, \alpha_3, m_{\text{i},3}, m_{\text{f},3})$ is their product model and its transition system is $G = (V, E)$. λ is a complete movement sequence based on G. $\Gamma^o_{\sigma,N,lc}$ is the set of all optimal alignments between trace σ and model N_2. Then, $\lambda \in \Gamma^o_{\sigma,N,lc}$ is true.

Proof. We suppose that $\left\langle \left(m_{\text{i}}^{\copyright}, (t'_1, t_1), m_2^{\copyright}\right), \ldots, \left(m_n^{\copyright}, (t'_n, t_n), m_{n+1}^{\copyright}\right)\right\rangle$ is a complete path and its complete movement sequence is λ, where $m_1^{\copyright} = m_{\text{i}}^{\copyright}$, $m_{n+1}^{\copyright} = m_{\text{f}}^{\copyright}$. The transition sequence is $\rho = \left\langle w\left(m_1^{\copyright}, m_2^{\copyright}\right), w\left(m_2^{\copyright}, m_3^{\copyright}\right), \ldots, w\left(m_n^{\copyright}, m_{n+1}^{\copyright}\right)\right\rangle$. According to Algorithm 2, the mapping relationship between complete movement sequence λ and transition sequence ρ can be determined.

In the product of two Petri nets, the name of the transition meets the following conditions: $\pi_1(t_3) = t_1$ or $\pi_1(t_3) =>>$, $\pi_2(t_3) = t_2$ or $\pi_2(t_3) =>>$, where $t_1 \in T_1$, $t_2 \in T_2$, $t_3 \in T_3$. According to Algorithm 1,

1. $m_{\text{i},1} \xrightarrow{\pi_1(\rho)_{\downarrow T_1}} m_{\text{f},1}$;
2. $m_{\text{i},2} \xrightarrow{\pi_2(\rho)_{\downarrow T_2}} m_{\text{f},2}$.

According to Algorithm 2,

1. $\pi_1(\lambda)_{\downarrow A} = \sigma$;
2. $m_{\text{i},2} \xrightarrow{\pi_2(\lambda)_{\downarrow T_2}} m_{\text{f},2}$.

In addition, the final state $m_{\text{f}}^{\copyright}$ guarantees that $\forall\gamma \in \Gamma_{\sigma,N} : \pi_2\left(m_{\text{f}}^{\copyright}\right) \leq \sum_{(a,t)\in\gamma} lc((a, t))$ is true. Based on the transition cost function, $\pi_2\left(m_{\text{f}}^{\copyright}\right)$ is the number of the deviations in λ. According to the definitions of the alignment and optimal alignment [15], $\lambda \in \Gamma^o_{\sigma,N,lc}$. □

Theorem 3 shows that a complete path corresponds to an optimal alignment between the trace and the process model in the transition system. If we want to get an optimal alignment between the trace and the model, we only need to access any path from the initial node to the final node in the transition system. An optimal alignment can be obtained by recording the weights of the visited edges and mapping them to the moves.

Algorithm 3 is presented to describe the specific implementation steps of calculating an optimal alignment based on the transition system.

Algorithm 3 The search algorithm of an optimal alignment between trace σ and model N.

Input: Min-cost transition system $G = (V, E)$.
Output: Optimal alignment γ.
Initialize: $\gamma \leftarrow \langle\rangle$.

1: $curnode \leftarrow m_{\mathrm{i}}^{\textcircled{c}}$;
2: **while** $\left(curnode \neq m_{\mathrm{f}}^{\textcircled{c}}\right)$ **do**
3: Choose any out edge of the current node as *curedge*;
4: $t \leftarrow w(curedge)$;
5: Translate t to the corresponding move *curmove*;
6: $\gamma \leftarrow \gamma \oplus \langle curmove \rangle$;
7: Consider the end node of edge *curedge* as the current node *curnode*;
8: **end while**
9: **return** γ;

We can discuss the complexity of Algorithm 3 from the prespective of graph and alignment, respectively. Algorithm 3 is to traverse any path from the initial node to the final node in the min-cost transition system. Supposing that the number of the nodes is v, the time complexity and space complexity of Algorithm 3 are $O(v)$. However, the time complexity and space complexity of the algorithm are relatively low, which are related to the longest path between the initial node and the final node in the transition system. The path in the graph corresponds to the optimal alignment between the trace and the model, so the maximum length of the path is equal to that of the optimal alignment. Supposed that the maximum length of the optimal alignment between the trace and the model is n, both the time complexity and space complexity of Algorithm 3 are $O(n)$.

In transition system $G_{\mathrm{lm}*\mathrm{pm}}$, v_0 is the initial node and v_{10} is the final node. According to Algorithm 3, optimal alignment γ_1 between trace σ and model N_{pm} is obtained. The specific search procedures are shown in Figure 6.

In Figure 6, there are three types of connection lines between nodes: the solid lines represent the logical relationship between nodes; the dashed lines represent the access order between nodes according to Algorithm 3; the dotted lines represent the output sequence of the moves when calculating the optimal alignment.

4.2 Search Algorithm of All Optimal Alignments

In the transition system, a complete path from the initial node to the final node can be mapped to an optimal alignment, and then all complete paths correspond to all optimal alignments.

Theorem 4. Let $N_1 = (P_1, T_1; F_1, \alpha_1, m_{\mathrm{i},1}, m_{\mathrm{f},1})$ be a log model of trace σ and $N_2 = (P_2, T_2; F_2, \alpha_2, m_{\mathrm{i},2}, m_{\mathrm{f},2})$ be a process model. $N_3 = (P_3, T_3; F_3, \alpha_3, m_{\mathrm{i},3}, m_{\mathrm{f},3})$

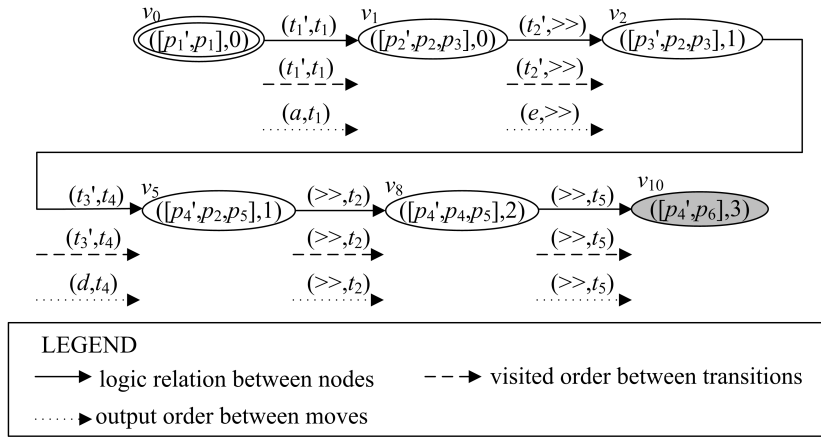


Figure 6. A search process of an optimal alignment in $G_{\text{lm}*\text{pm}}$

is their product model and its transition system is $G = (V, E)$. Λ is the set of all complete movement sequences based on G. $\Gamma^{o}_{\sigma,N,lc}$ is the set of all optimal alignments between trace σ and model N_2. Then, $\Lambda = \Gamma^{o}_{\sigma,N,lc}$ is true.

Proof. According to Theorem 3, $\forall\lambda \in \Lambda \Rightarrow \lambda \in \Gamma^{o}_{\sigma,N,lc}$. Then, $\Lambda \subseteq \Gamma^{o}_{\sigma,N,lc}$.

For $\gamma \in \Gamma^{o}_{\sigma,N,lc}$, there are three expressions that hold as follows:

1. $\pi_1(\gamma)_{\downarrow A} = \sigma$;
2. $m_{\text{i},2} \xrightarrow{\pi_2(\gamma)_{\downarrow T_2}} m_{\text{f},2}$;
3. given standard likelihood cost function $lc()$, $\forall\gamma' \in \Gamma_{\sigma,N,lc} : \sum_{(a,t)\in\gamma} lc((a,t)) \leq \sum_{(a',t')\in\gamma'} lc((a',t'))$.

According to the definition of the alignment and optimal alignment [15], optimal alignment γ is a movement sequence.

In the log model, $\alpha_1(t_{1,j}) = \sigma[j]$, where $t_{1,j} \in T_1$. So there is an inverse function $\alpha_1^{-1}(\sigma[j]) = t_{1,j}$, which maps optimal alignment γ to transition sequence ρ. In the product of two Petri nets, $\partial_{\text{set}}(\rho) \in T_3$. According to expression 1 mentioned above, $m_{\text{i},1} \xrightarrow{\pi_1(\rho)_{\downarrow T_1}} m_{\text{f},1}$; according to expression 2, $m_{\text{i},2} \xrightarrow{\pi_2(\rho)_{\downarrow T_2}} m_{\text{f},2}$. In the product of two Petri nets, $m_{\text{i},3} = m_{\text{i},1} \biguplus m_{\text{i},2}$, $m_{\text{f},3} = m_{\text{f},1} \biguplus m_{\text{f},2}$, $T_3 \subseteq (T_1^{>>} \times T_2^{>>})$, then $m_{\text{i},3} \xrightarrow{\rho} m_{\text{f},3}$. According to Algorithm 1, $\pi_1\left(m_{\text{i}}^{\copyright}\right) \xrightarrow{\rho} \pi_1\left(m_{\text{f}}^{\copyright}\right)$.

According to Algorithm 1, given the transition cost function, for $\forall\gamma' \in \Gamma_{\sigma,N,lc}$: $\pi_2\left(m_{\text{f}}^{\copyright}\right) \leq \sum_{(a',t')\in\gamma'} lc((a',t'))$. Combining with expression 3 mentioned above, $\sum_{(a,t)\in\gamma} lc((a,t)) = \pi_2\left(m_{\text{f}}^{\copyright}\right)$. So $\sum_{(t',t)\in\rho} lc((t',t)) = \pi_2\left(m_{\text{f}}^{\copyright}\right)$.

Hence, there is a complete path, just as $\left\langle m_1^{\copyright}, m_2^{\copyright}, \ldots, m_j^{\copyright}, \ldots, m_n^{\copyright} \right\rangle$, where $m_1^{\copyright} = m_{\text{i}}^{\copyright}$, $m_n^{\copyright} = m_{\text{f}}^{\copyright}$, $m_j^{\copyright} = \left(\pi_1\left(m_{j-1}^{\copyright}\right)[\rho[j-1]>, lc(\rho[j-1])\right)$ $(1 < j < n)$.

This complete path corresponds to a complete movement sequence, denoted by γ, which makes $\lambda = \gamma$ founded. So $\gamma \in \Lambda$, and then $\Gamma^o_{\sigma,N,lc} \subseteq \Lambda$.

In conclusion, $\Lambda = \Gamma^o_{\sigma,N,lc}$. □

Theorem 4 shows that the set of all complete movement sequences in the min-cost reachability graph is identical with the set of all optimal alignments between the trace and the model. All complete paths in the min-cost transition system correspond to all optimal alignments between the trace and the model. In other words, the complete movement sequence is equal to the optimal alignment, and they are legal movement sequences.

Next, Algorithm 4 is proposed to calculate all the optimal alignments between the trace and the model based on the standard likelihood cost function by the min-cost transition system. In Algorithm 4, two stacks are used. The description and operation of the stack are as follows:

- *nodestack*: node stack to store the visited nodes on the paths;
- *movestack*: move stack to store the weights of the directed edges between nodes in the node stack;
- *empty*(*stack*): a function to judge whether the stack is empty. If the stack is empty, it returns True; otherwise, it returns False;
- *gettop*(*stack*): get the top element of stack *stack*;
- *pop*(*stack*): pop up the top element of stack *stack*;
- *push*(*stack*, *node*): push element *node* into stack *stack*.

Algorithm 4 The search algorithm of all optimal alignments between trace σ and model N.

Input: Min-cost transition system $G = (V, E)$.
Output: Optimal alignment set $\Gamma^o_{\sigma,N,lc}$.
Initialize: $\Gamma^o_{\sigma,N,lc} \leftarrow \emptyset$, $nodestack \leftarrow \emptyset$, $movestack \leftarrow \emptyset$, $\gamma \leftarrow \langle\rangle$.

for (all ($edge \in E$)) **do**
  $flag(edge) \leftarrow 0$;
end for
$push\left(nodestack, m_{\mathrm{i}}^{\textcircled{c}}\right)$;
while (!$empty(nodestack)$) **do**
  $curnode \leftarrow gettop(nodestack)$;
  if ((each out edge *edge* of *curnode* has been visited) **or** $\left(curnode = m_{\mathrm{f}}^{\textcircled{c}}\right)$) **then**
    $pop(nodestack)$;
    if $\left(curnode \neq m_{\mathrm{i}}^{\textcircled{c}}\right)$ **then**
      $pop(movestack)$;
    end if
    $\gamma \leftarrow \gamma - \gamma[|\gamma|]$;
    if $\left(curnode = m_{\mathrm{f}}^{\textcircled{c}}\right)$ **then**
      $\Gamma^o_{\sigma,N,lc} \leftarrow \Gamma^o_{\sigma,N,lc} \cup \{\gamma\}$;

```
15:       else
16:         for (each out edge curedge of curnode) do
17:           flag(edge) ← 0;
18:         end for
19:       end if
20:     else
21:       flag(edge) ← 1;
22:       Consider the end node of edge as the current node curnode;
23:       push(nodestack, curnode);
24:       t ← w(edge);
25:       Translate t to the corresponding move move;
26:       push(movestack, move);
27:       γ ← γ ⊕ ⟨move⟩;
28:     end if
29: end while
30: return Γ^o_{σ,N,lc};
```

Similarly, we can discuss the complexity of Algorithm 4 from the prespective of graph and alignment, respectively. Algorithm 4 is to traverse all the paths from the initial node to the final node in the min-cost transition system. Because all paths between two nodes are required, every possibility must be examined, and backtracking is the only way. As for the complexity of this algorithm, it depends on the size of the graph and the number of the nodes. Supposing that the number of the nodes is v and the number of the edges is e, the time complexity and space complexity of Algorithm 4 are $O(ve)$.

However, the time complexity and space complexity of Algorithm 4 are related to the lengths of the complete paths and the number of complete paths between the initial node and the final node in the min-cost transition system. A complete path in the graph corresponds to an optimal alignment between the trace and the process model based on the given cost function, so the length of the complete path is equal to that of the optimal alignment. Meanwhile, the number of complete paths in the graph is equal to that of the optimal alignments. Supposed that the maximum length of optimal alignment is n and the number of optimal alignments is m, the time complexity and space complexity of Algorithm 4 are $O(mn)$.

According to Algorithm 4, the values of γ_1 to γ_7 are shown in Figure 7. There are seven different paths from the initial node to the final node in Figure 5, while there are seven different optimal alignments in Figure 7. In Figure 7, a vertical list represents a move in each optimal alignment. In order to explicitly compare the events in the trace with the activities mapped on the transitions in the model, the activities mapped on the transitions are also marked out in each optimal alignment.

According to Algorithm 4, the correspondence between complete paths, transition sequences and optimal alignments is shown in Table 5.

γ_1=

a	e	d	$>>$	$>>$
a	$>>$	d	b	e
t_1		t_4	t_2	t_5

γ_2=

a	e	$>>$	d	$>>$
a	$>>$	b	d	e
t_1		t_2	t_4	t_5

γ_3=

a	$>>$	e	d	$>>$
a	b	$>>$	d	e
t_1	t_2		t_4	t_5

γ_4=

a	$>>$	$>>$	e	d
a	b	c	e	$>>$
t_1	t_2	t_3	t_5	

γ_5=

a	$>>$	$>>$	e	d
a	b	d	e	$>>$
t_1	t_2	t_4	t_5	

γ_6=

a	$>>$	$>>$	e	d
a	c	b	e	$>>$
t_1	t_3	t_2	t_5	

γ_7=

a	$>>$	$>>$	e	d
a	d	b	e	$>>$
t_1	t_4	t_2	t_5	

Figure 7. All optimal alignments between trace σ and model $N_{\mathrm{lm}*\mathrm{pm}}$

Complete Path (Only Nodes)	Transition Sequence	Optimal Alignment
$(v_0, v_1, v_2, v_5, v_8, v_{10})$	$\langle (t'_1, t_1), (t'_2, >>), (t'_3, t_4), (>>, t_2), (>>, t_5) \rangle$	γ_1
$(v_0, v_1, v_2, v_6, v_8, v_{10})$	$\langle (t'_1, t_1), (t'_2, >>), (>>, t_2), (t'_3, t_4), (>>, t_5) \rangle$	γ_2
$(v_0, v_1, v_3, v_6, v_8, v_{10})$	$\langle (t'_1, t_1), (>>, t_2), (t'_2, >>), (t'_3, t_4), (>>, t_5) \rangle$	γ_3
$(v_0, v_1, v_3, v_7, v_9, v_{10})$	$\langle (t'_1, t_1), (>>, t_2), (>>, t_3), (t'_2, t_5), (t'_3, >>) \rangle$	γ_4
$(v_0, v_1, v_3, v_7, v_9, v_{10})$	$\langle (t'_1, t_1), (>>, t_2), (>>, t_4), (t'_2, t_5), (t'_3, >>) \rangle$	γ_5
$(v_0, v_1, v_4, v_7, v_9, v_{10})$	$\langle (t'_1, t_1), (>>, t_3), (>>, t_2), (t'_2, t_5), (t'_3, >>) \rangle$	γ_6
$(v_0, v_1, v_4, v_7, v_9, v_{10})$	$\langle (t'_1, t_1), (>>, t_4), (>>, t_2), (t'_2, t_5), (t'_3, >>) \rangle$	γ_7

Table 5. Mapping the optimal alignments to the paths in $G_{\mathrm{lm}*\mathrm{pm}}$

5 SIMULATION EXPERIMENTS

Given the process model, the trace and the standard likelihood cost function, Section 3 explains how to generate the min-cost transition system. In Section 4, we propose two algorithms to compute an optimal alignment and all optimal alignments according to the min-cost transition system, respectively. To facilitate the description of our approaches, we name the algorithm to compute an optimal alignment as *Min-cost algorithm-One*, and the algorithm to compute all optimal alignments as *Min-cost algorithm-All*.

This section presents several evaluations of simulation experiments about the proposed approaches to illustrate that our approaches can be finished in limited time by occupying limited space. The experiments are performed on a computer with Intel Core i7-6500U, 2.50 GHz CPU, 16.0 GB RAM, JDK 1.8, and Windows 7. We will compare the proposed approaches to illustrate their feasibility and effectiveness.

The approaches in this paper have been implemented as two plug-ins in the process mining framework ProM and are publicly available. The plug-ins are called "*Min-cost Algorithm-One*" and "*Min-cost Algorithm-All*". The first plug-in implements Algorithm 1, Algorithm 2 and Algorithm 3 presented in this paper and

its function is to compute an optimal alignment between the trace and the model based on the standard likelihood cost function. However, the second plug-in implements Algorithm 1, Algorithm 2 and Algorithm 4 presented in this paper and its function is to compute all optimal alignments between the trace and the model based on the standard likelihood cost function. Both plug-ins can be accessible at: `https://pan.baidu.com/s/11sAA_w5TBec08t7evHMyg`. The extraction code for downloading files is "57bg".

Through Algorithm 1, the min-cost transition system is obtained, which is the search space for the following search algorithms. So Algorithm 1 is the preparation. However, Algorithm 3 and Algorithm 4 do the search work in the search space. Algorithm 2 is invoked by Algorithm 3 and Algorithm 4 in the actual implementation. When we consider the performance of our approaches, Algorithms 3 and 4 are the main study objects.

Taking a business process from the inclined shaft in a coal mine as an example, the process model is constructed manually [32]. In order to enhance the safety of the transportation of the coal mine, a PLC-based distributed control system for the inclined shaft of coal mine is analyzed. And Petri nets are adopted to build the model of the system that simulates the process for building the route in the inclined shaft of the coal mine, shown in Figure 8 and denoted as N_{SE}. The meanings of the transitions in model N_{SE} are explained in Table 6.

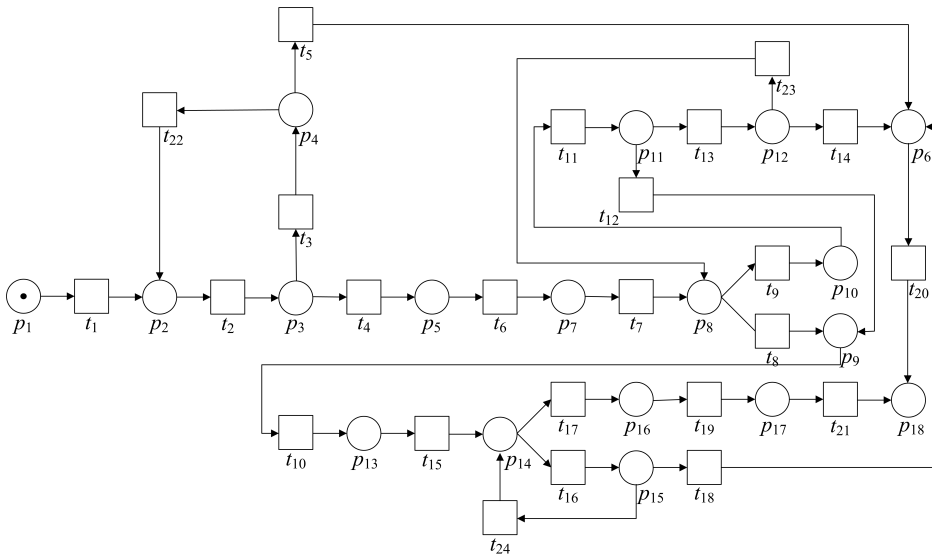


Figure 8. Petri net model N_{SE} for building the route in the inclined shaft of the coal mine

After the process model is introduced, the corresponding event logs can be obtained. We generate completely fit traces according to the process model with various lengths from 11 to 50 activities. Noise is introduced by randomly adding

Transition	Label	Meaning
t_1	a	build the route
t_2	b	obtain the target route state
t_3	c	fail to obtain the target route state
t_4	d	succeed in obtaining the target route state
t_5	e	report errors in obtaining the target route state
t_6	f	set the target route
t_7	g	check the consistency of turnouts
t_8	h	be consistent between turnout positions
t_9	i	be inconsistent between turnout positions
t_{10}	j	fix the turnout positions
t_{11}	k	switch the turnouts
t_{12}	l	succeed in switching the turnouts
t_{13}	m	fail to switch the turnouts
t_{14}	n	report errors in switching the turnouts
t_{15}	o	turn on the car stopper
t_{16}	p	fail to turn on the car stoppers
t_{17}	q	succeed in turning on the car stoppers
t_{18}	r	report errors in turning on the car stoppers
t_{19}	s	succeed in building the route
t_{20}	t	end abnormally and return
t_{21}	u	end normally and return
t_{22}	v	rebuild the route
t_{23}	w	recheck the consistency of turnouts
t_{24}	x	turn on the car stopper again

Table 6. The meanings of the transitions in model N_{SE}

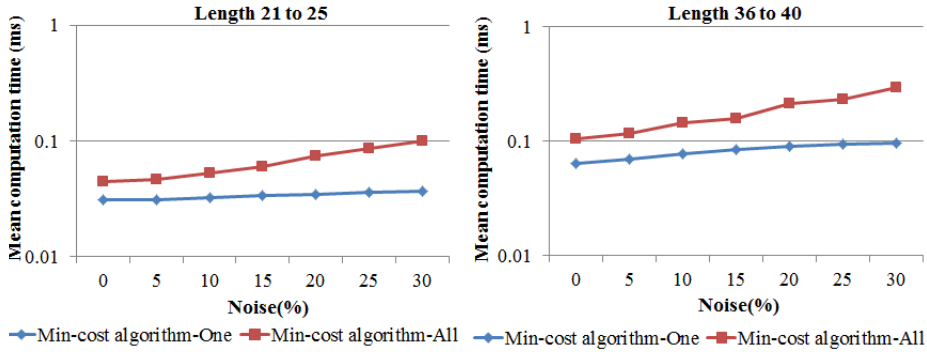
and/or deleting activities for every trace. The noise ratio is measured by the formula *noise = the number of the deviations/the length of the original fit trace*. Here, the number of the deviations is equal to that of the inserted activities and deleted activities.

For the traces with different noise ratios, the mean computation time and the mean queued states of constructing alignments are compared between both approaches. For the traces of lengths from 21 to 25 and 36 to 40, the computation time and the queued states for constructing alignments are also compared between the two approaches. For different trace lengths, we also compare the computation time and the queued states between the two approaches. In this experiment, two kinds of event logs are used. One includes the completely fit traces of various lengths and the other is the unfit traces of various lengths between the specified values. In this paper, each unfit trace has a noise ratio between 5 % and 30 %.

The experiment data in this paper are 28 event logs, and each event log includes 100 traces. Every result recorded in this paper is the average value of the same experiment repeated for 10 times.

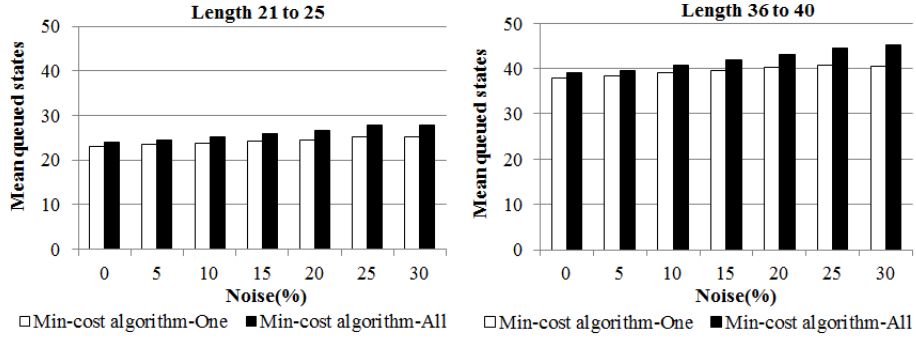
5.1 Noise Level

In this subsection, the mean computation time and the mean queued states are compared between Min-cost algorithm-One and Min-cost algorithm-All. The traces used in this experiment have various noise ratios. For the process model, two event log sets are generated, in which the trace lengths are from 21 to 25 and from 36 to 40, respectively. Each event log has the traces with the fixed noise ratio. The comparison results of the mean computation time are shown in Figure 9, and that of the mean queued states are shown in Figure 10. In Figure 9, y-axis is shown by a logarithmic scale.



a) b)

Figure 9. Comparison of computation time with different noise levels



a) b)

Figure 10. Comparison of queued states with different noise levels

Figure 9 shows that the mean computation time of constructing alignments increases as the length of traces and the noise ratios. When a trace length is from 21

to 25 in Figure 9 a), the mean computation time of the two approaches is less than 0.1 milliseconds, so it is very short. Min-cost algorithm-One needs the shorter time. In this experiment, the computation time of Min-cost algorithm-All is between 0.04 ms and 0.1 ms, and that of Min-cost algorithm-One is between 0.03 ms and 0.04 ms. The computation time of the two approaches increases exponentially as the noise ratios of the traces. However, the computation time of Min-cost algorithm-All increases faster than the other approach. In Figure 9 b), the growths of the computation time for the approaches are similar to the results shown in Figure 9 a). However, all data of the computation time shown in Figure 9 b) are higher than the results shown in Figure 9 a). In this experiment, the computation time of Min-cost algorithm-One lies in between 0.06 ms and 0.1 ms, and that of Min-cost algorithm-All lies in between 0.1 ms and 0.3 ms.

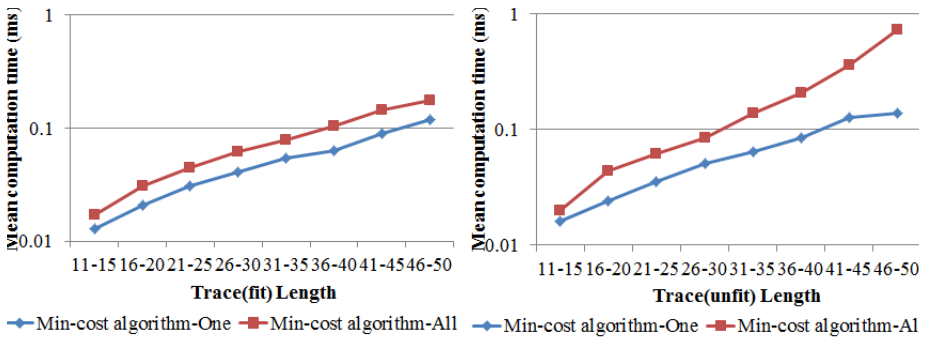
Whether in Figure 10 a) or Figure 10 b), the mean queued states of Min-cost algorithm-All are obviously higher than the other approach. However, the queued states of the approaches increase relatively slowly along with the growth of the noise ratios. Compared the data in Figure 10 a) with that in Figure 10 b), when the traces have the same noise ratios, the queued nodes of the traces with the lengths from 36 to 40 is almost 1.5 times as many as that of the traces with the lengths from 21 to 25 by the same approaches. On the study of the related data, the average length of the traces from 36 to 40 is also 1.5 times as that of the traces from 21 to 25, which is in accordance with the above-mentioned conclusion.

Hence, as the lengths of traces and the noise ratios increase, the time complexity and space complexity of Min-cost algorithm-One grow much slower than that of Min-cost algorithm-All.

5.2 Trace Length

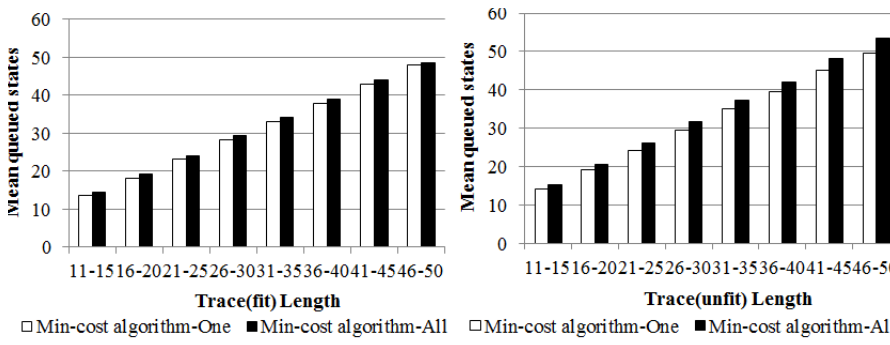
This experiment is conducted to illustrate the effect of the trace lengths. We aim to compare Min-cost algorithm-One and Min-cost algorithm-All for different trace lengths. For the process model, two kinds of event log sets are generated with different average lengths of the traces from 11 to 50. One includes the completely fit traces, and the other includes the unfit traces with different noise ratios. Here, the noise ratio of each trace is a random value from 5 % to 30 %. The comparison results of the mean computation time are shown in Figure 11, and that of the mean queued states are shown in Figure 12. In Figure 11, y-axis is shown by a logarithmic scale.

In Figure 11, no matter the traces are fit or unfit to process model N^{SE}, the mean computation time of Min-cost algorithm-One is lower than that of Min-cost algorithm-All. Figure 11 a) shows the computation time when the traces can be rightly replayed by the given model, and Figure 11 b) shows the computation time when the traces have noise ratios from 5 % to 30 %. In Figure 11 a), the growth rates of the computation time for the two approaches are almost equal. In Figure 11 b), the computation time of Min-cost algorithm-All grows the faster along with the growth of the trace lengths. When the lengths of the traces are larger than 45, the



a) b)

Figure 11. Comparison of computation time between different trace lengths



a) b)

Figure 12. Comparison of queued states between different trace lengths

computation time of the Min-cost algorithm-All has an obvious growth. However, the computation time of Min-cost algorithm-One increases relatively slow.

In Figure 12, the queued states of Min-cost algorithm-One are slightly fewer than that of Min-cost algorithm-All, but the two values are very close to each other. Compared the dots in Figure 12 a) with those in Figure 12 b), the queued states of each approach for the fit traces are less than those for the unfit traces. The queued states of the two approaches have a linear growth along with the increase of the trace lengths.

In conclusion, Min-cost algorithm-One outperforms Min-cost algorithm-All. Of course, Min-cost algorithm-One can only obtain the optimal alignment, but Min-cost algorithm-All can get all. In addition, we ignore the generation process of the min-cost transition system. The focus of these experiments is not to compare the two approaches, but to show that both approaches can be completed in limited time and space.

6 CONCLUSIONS

Conformance checking plays an increasingly important role in information management systems. Alignment is one of the most advanced and comprehensive conformance checking. By means of alignment approaches, the optimal alignments between traces and models based on the cost functions can be obtained. The results of optimal alignments can be applied to all aspects of process mining. However, the search space generated by some existing alignment approaches is so large that seriously affects the search efficiency of optimal alignments. In this paper, we propose new approaches that can align observed and modeled behaviors based on the min-cost transition systems. In the transition system, all paths from the initial node to the final node can be mapped to all the optimal alignments between the trace and the process model. All optimal alignments can be obtained and output by the traversing algorithm of graphs.

The alignment approaches proposed in this paper generate a min-cost transition system which includes all the optimal alignments between the trace and model based on the given cost function. In the min-cost transition system, the optimal alignments can be quickly found. Finally, all the algorithms in this paper are simulated on ProM. The simulation results show that the alignment approaches proposed in this paper are feasible and effective.

The alignment approaches presented in this paper are feasible and effective when dealing with the artificial logs and models. In the future work, we intend to mainly carry out the following research: Firstly, we can try to propose more efficient algorithms to generate the min-cost transition system for the Petri net. Secondly, the approaches presented in this paper will be simulated using more real-life cases to verify its robustness and stability. Then, it will be further to compare the min-cost transition systems in this paper with the classic reachability graphs of Petri nets, and find the differences between the optimal alignments under the different cost functions, as well as determine their own application areas. Finally, the idea of both products of two Petri nets and min-cost transition systems will be applied to process discovery as well as model repair and enhancement in order to improve the fitness between observed and modeled behaviors.

Acknowledgement

This work was supported in part by the Natural Science Foundation of China under the grant No. 61973180, in part by the Taishan Scholar Construction Project of Shandong Province, in part by the Key Research and Development Program of Shandong Province under the grant No. 2018GGX101011, and in part by the Natural Science Foundation of Shandong Province under the grants No. ZR2018MF001 and No. ZR2019MF033.

REFERENCES

[1] van der Aalst, W. M. P: Business Process Management: A Comprehensive Survey. ISRN Software Engineering, Vol. 2013, 2013, Art. No. 507984, 37 pp., doi: 10.1155/2013/507984.

[2] van der Aalst, W. M. P.—Adriansyah, A.—de Medeiros, A. K. A. et al.: Process Mining Manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (Eds.): Business Process Management Workshops (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 99, 2012, pp. 169–194, doi: 10.1007/978-3-642-28108-2_19.

[3] Li, C.—Reichert, M.—Wombacher, A.: Mining Business Process Variants: Challenges, Scenarios, Algorithms. Data and Knowledge Engineering, Vol. 70, 2011, No. 5, pp. 409–434, doi: 10.1016/j.datak.2011.01.005.

[4] van der Aalst, W. M. P.—Weijters, A. J. M. M.—Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, 2004, No. 9, pp. 1128–1142, doi: 10.1109/TKDE.2004.47.

[5] Weber, P.—Bordbar, B.—Tino, P.: A Framework for the Analysis of Process Mining Algorithms. IEEE Transactions on Systems Man and Cybernetics: Systems, Vol. 43, 2013, No. 2, pp. 303–317, doi: 10.1109/TSMCA.2012.2195169.

[6] van der Aalst, W. M. P.—Stahl, C.: Modeling Business Processes: A Petri Net Oriented Approach. The MIT Press, Cambridge, USA, 2011, doi: 10.7551/mitpress/8811.001.0001.

[7] Rozinat, A.—van der Aalst, W. M. P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems, Vol. 33, 2008, No. 1, pp. 64–95, doi: 10.1016/j.is.2007.07.001.

[8] Bose, R. P. J. C.—van der Aalst, W. M. P.: Process Diagnostics Using Trace Alignment: Opportunities, Issues, and Challenges. Information Systems, Vol. 37, 2012, No. 2, pp. 117–141, doi: 10.1016/j.is.2011.08.003.

[9] van der Aalst, W. M. P.—Adriansyah, A.—van Dongen, B. F.: Replaying History on Process Models for Conformance Checking and Performance Analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 2, 2012, No. 2, pp. 182–192, doi: 10.1002/widm.1045.

[10] Wang, Y. Y.—Du, Y. Y.: Conformance Checking Based on Extended Footprint Matrix. Journal of Shandong University of Science and Technology (Natural Science), Vol. 37, 2018, No. 2, pp. 9–15 (in Chinese).

[11] Adriansyah, A.—van Dongen, B. F.—van der Aalst, W. M. P.: Towards Robust Conformance Checking. In: zur Muehlen, M., Su, J. (Eds.): Business Process Management Workshops (BPM 2010). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 66, 2010, pp. 122–133, doi: 10.1007/978-3-642-20511-8_11.

[12] Rozinat, A.: Process Mining: Conformance and Extension. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2010.

[13] de Leoni, M.—Maggi, F. M.—van der Aalst W. M. P.: Aligning Event Logs and Declarative Process Models for Conformance Checking. In: Barros, A., Gal, A., Kindler, E. (Eds.): Business Process Management (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7481, 2012, pp. 82–97, doi: 10.1007/978-3-642-32885-5_6.

[14] Adriansyah, A.—van Dongen, B. F.—van der Aalst, W. M. P.: Conformance Checking Using Cost-Based Fitness Analysis. Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference (EDOC'11), Helsinki, Finland, 2011, pp. 55–64, doi: 10.1109/EDOC.2011.12.

[15] Adriansyah, A.—van Dongen, B. F.—van der Aalst, W. M. P.: Memory-Efficient Alignment of Observed and Modeled Behavior. Technical report. BPMcenter.org, BPM Reports, Vol. 1303, 2013.

[16] van Eck, M. L.: Alignment-Based Process Model Repair and Its Application to the Evolutionary Tree Miner. Master Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2013.

[17] Fahland, D.—van der Aalst, W. M. P.: Model Repair – Aligning Process Models to Reality. Information Systems, Vol. 47, 2015, No. 1, pp. 220–243, doi: 10.1016/j.is.2013.12.007.

[18] Adriansyah, A.—MunozGama, J.—Carmona, J.—van Dongen, B. F.—van der Aalst, W. M. P.: Alignment Based Precision Checking. In: La Rosa, M., Soffer, P. (Eds.): Business Process Management Workshops (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 132, 2013, pp. 137–149, doi: 10.1007/978-3-642-36285-9_15.

[19] de Leoni, M.—Maggi, F. M.—van der Aalst, W. M. P.: An Alignment-Based Framework to Check the Conformance of Declarative Process Models and to Preprocess Event-Log Data. Information Systems, Vol. 47, 2015, No. 3, pp. 258–277, doi: 10.1016/j.is.2013.12.005.

[20] van Eck, M. L.—Buijs, J. C. A. M.—van Dongen, B. F.: Genetic Process Mining: Alignment-Based Process Model Mutation. In: Fournier, F., Mendling, J. (Eds.): Business Process Management Workshops (BPM 2014). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 202, 2014, pp. 291–303, doi: 10.1007/978-3-319-15895-2_25.

[21] Cook, J. E.—Wolf, A. L.: Software Process Validation: Quantitatively Measuring the Correspondence of a Process to a Model. ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 8, 1999, No. 2, pp. 147–176, doi: 10.1145/304399.304401.

[22] Lu, X.—Fahland, D.—van der Aalst, W. M. P.: Conformance Checking Based on Partially Ordered Event Data. In: Fournier, F., Mendling, J. (Eds.): Business Process Management Workshops (BPM 2014). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 202, 2014, pp. 75–88, doi: 10.1007/978-3-319-15895-2_7.

[23] Wang, L.—Du, Y. Y.—Liu, W.: Aligning Observed and Modelled Behaviour Based on Workflow Decomposition. Enterprise Information Systems, Vol. 11, 2017, No. 8, pp. 1207–1227, doi: 10.1080/17517575.2016.1193633.

[24] Song, W.—Xia, X.—Jacobsen, H.-A.—Zhang, P.—Hu, H.: Efficient Alignment Between Event Logs and Process Models. IEEE Transactions on Services Computing, Vol. 10, 2017, No. 1, pp. 136–149, doi: 10.1109/TSC.2016.2601094.

[25] Tian, Y. H.—Du, Y. Y.—Li, M. Z.—Dong, H.—Hu, Q.: Reduced Alignment Based on Petri Nets. Concurrency and Computation: Practice and Experience, Vol. 30, 2018, No. 23, Art. No. e4411, doi: 10.1002/cpe.4411.

[26] van der Aalst, W. M. P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin, Heidelberg, 2011, doi: 10.1007/978-3-642-19345-3.

[27] Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceeding of the IEEE, Vol. 77, 1989, No. 4, pp. 541–580, doi: 10.1109/5.24143.

[28] Du, Y. Y.—Jiang, C. J.—Zhou, M. C.: A Petri Net-Based Model for Verification of Obligations and Accountability in Cooperative Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 39, 2009, No. 2, pp. 299–308, doi: 10.1109/TSMCA.2008.2010751.

[29] Ran, N.—Su, H. Y.—Wang, S. G.: An Improved Approach to Test Diagnosability of Bounded Petri Nets. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 2, pp. 297–303, doi: 10.1109/JAS.2017.7510406.

[30] Hu, Q.—Liu, M. H.—Zhao, Z.—Du, J. W.: A Path Detecting Method to Analyze the Interactive Compatibility of Service Processes Based on WS-BPEL. Concurrency and Computation: Practice and Experience, Vol. 30, 2018, No. 19, Art. No. e4699, doi: 10.1002/cpe.4699.

[31] Liu, G. J.: Complexity of the Deadlock Problem for Petri Nets Modeling Resource Allocation Systems. Information Sciences, Vol. 363, 2016, pp. 190–197, doi: 10.1016/j.ins.2015.11.025.

[32] Zhang, W. W.—Liu, W. D.: Research of Inclined Shaft Monitoring and Control System of Coal Mine Based on Petri Net. Computer Engineering and Application, Vol. 48, 2012, No. 20, pp. 240–243 (in Chinese).

Xiwen Feng received his Ph.D. degree from Shandong University of Science and Technology, Qingdao, China. He is currently Professor at the College of Mining and Safety, Shandong University of Science and Technology, Qingdao, China. He has long been engaged in mining system engineering, industrial engineering and logistics, logistics system simulation and optimization of teaching and research work. He presided over 30 vertical and horizontal research topics such as provincial and ministerial-level key planning projects, natural science fund projects, and corporate commissions, of which 8 results reached the international advanced level and won 10 provincial and ministerial science and technology awards. He has published 60 scientific and technological papers and 1 monograph.

Dong Han received his B.Sc. degree from Weifang College, Weifang, China, in 2004. He received his M.Sc. degree from Shandong University of Science and Technology, Qingdao, in 2007, where he is currently pursuing the Ph.D. degree with College of Mining and Safety. His current research interests include resource management, Petri nets, and workflow. He is currently Lecturer of computer science and technology with Shandong University of Science and Technology.

Yinhua Tian received her B.Sc. degree in computer science and technology, the M.Sc. degree and the Ph.D. degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2004, 2007 and 2018, respectively. She is currently Lecturer of computer science and technology with Shandong University of Science and Technology. She has authored over 10 technical papers in journals and conference proceedings. Her current research interests include Petri nets, process mining, and optimization algorithms.

Computing and Informatics, Vol. 39, 2020, 246–263, doi: 10.31577/cai_2020_1-2_246

ADAPTIVE FAULT DIAGNOSIS OF MOTORS USING COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZER WITH FUZZY PETRI NET

Xuezhen Cheng, Changan Wang, Jiming Li, Xingzhen Bai

College of Electrical Engineering and Automation
Shandong University of Science and Technology, Qingdao, China
e-mail: xzbai@sdust.edu.cn

Abstract. This study proposes and applies a comprehensive learning particle swarm optimization (CLPSO) fuzzy Petri net (FPN) algorithm, which is based on the CLPSO algorithm and FPN, to the fault diagnosis of a complex motor. First, the transition confidence is replaced by a Gaussian function to deal with the uncertainty of fault propagation. Then, according to the Petri net principle, a competition operator is introduced to improve the matrix reasoning. Finally, a CLPSO-FPN model for motor fault diagnosis is established based on the motor failure mechanism and fault characteristics. The CLPSO algorithm is used to generate the system parameters for fault diagnosis and to improve the adaptability and accuracy of fault diagnosis. This study considers the example of a three-phase asynchronous motor. The results show that the proposed algorithm can diagnose faults in this motor with satisfactory adaptability and accuracy compared with the traditional FPN algorithm. By establishing the system model, the fault propagation process of motors can be accurately and intuitively expressed, thus improving the fault treatment and equipment maintenance of motors.

Keywords: Fuzzy Petri net, CLPSO, fault diagnosis, motor, adaptive

Mathematics Subject Classification 2010: 93D21

1 INTRODUCTION

A motor is a complex mechanical system that usually comprises multiple functional modules. Owing to the complex correlations among these modules, the

fault characteristics are uncertain and nonlinear [1]. Either quantitative analysis or qualitative analysis is used for fault diagnosis of motors. At present, the data-driven quantitative analysis method is commonly used to process the fault characteristic signals of the rotor and bearing in a motor system for fault classification [2]. Deng et al. [3] proposed a motor bearing fault diagnosis method based on the combination of empirical wavelet transform and Hilbert transform; however, this method ignores the noise interference of low-frequency signals and is prone to misjudgment. Deng et al. [4] combined empirical mode decomposition, fuzzy information, and an improved support vector machine method and used the particle swarm optimization (PSO) algorithm to perform parameter optimization, feature extraction, and accurate classification of rotor fault signals. Quantitative analysis can effectively deal with fault signals. However, in complex motor systems, fault signals are easily affected by the environment and motor modules, and most functional modules cannot extract fault signals. Therefore, quantitative analysis cannot easily satisfy the fault diagnosis requirements of motor systems.

Qualitative analysis can be used to establish a system model by using internal knowledge of the system [5]. For example, the Petri net method can be used for graphical and mathematical modeling. In recent years, studies worldwide have used Petri nets to deal with discrete event sequences, concurrency, and conflict relationships [6, 7]. Therefore, Petri nets are increasingly being used for fault diagnosis. Sheng et al. [8] defined the probability transition method of Petri nets for dealing with the uncertainty in the fault propagation process, thereby overcoming the disadvantage that Petri nets only focus on the previous state of places in the fault diagnosis process and that their probability transition mode is not adaptive. Cheng et al. [9] proposed the concept of fuzzy fault Petri net and its modeling method to overcome the difficulty of dealing with the uncertainty of fault information in the fault diagnosis process. Zhang et al. [10] conducted a rigorous mathematical investigation of a fuzzy Petri net (FPN) and proposed a matrix reasoning process, thereby providing a theoretical foundation for applying FPNs. However, the acquisition of its initial weight value still relies on expert experience and has poor adaptability. The neural fuzzy Petri net (NFPN) concept [11] can effectively improve the algorithm's adaptability; however, it cannot satisfy the fault diagnosis requirements of complex motor systems.

To improve the poor accuracy and adaptability of traditional FPN fault diagnosis methods, this study uses the comprehensive learning particle swarm optimization (CLPSO) algorithm to optimize the FPN algorithm and improve the traditional FPN reasoning method. Further, a fault-diagnosis method based on CLPSO-FPN is proposed. The main contributions of this study are as follows:

- A new competition operator is proposed to solve the competition relationship between different modules in complex systems and the matrix reasoning process of the algorithm is optimized.

- A new representation of transition confidence is proposed, which uses a Gauss function instead of traditional transition confidence and reflects the impact of transition on its output places through a transition influence factor.
- A CLPSO algorithm is used to generate system parameters that reflect the relationship between different modules.

2 CLPSO-FPN

2.1 FPN

A Petri net can be used to study a network structure based on the known logical relationships between inputs and outputs in a system. In the structure diagram of a Petri net, circles, strips, and directed arcs respectively represent places, transitions, and the relationships between places and transitions. Places represent resources or conditions, and transitions represent events or actions. Figure 1 shows the structure diagram of a Petri net [12].

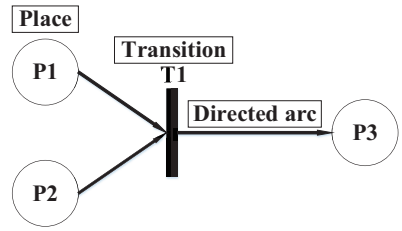


Figure 1. Structure diagram of Petri net

The traditional Petri net with different FPNs uses the place value between $[0, 1]$ instead of the token value, and defines the threshold as the condition of transition trigger. The FPN is defined as an 8-tuple, as follows [13].

$$FPN = (P, T, I, O, M, W, \alpha, \lambda) \tag{1}$$

where:

- $P = \{p_1, p_2, \ldots, p_n\}$, P represents a set of places.
- $T = \{t_1, t_2, \ldots, t_m\}$, T represents a set of transitions.
- $I = (\delta_{ij})_{(n\times m)}$ is an input matrix describing the mapping of transitions to places. For the input matrix element $\delta_{ij} = \{0, 1\}$. If P_i is the input of t_j, $\delta_{ij} = 1$. If P_i is not an input of t_j, $\delta_{ij} = 0$. $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
- $O = (\gamma_{ij})_{(n\times m)}$ is an output matrix describing the mapping of places to transitions. For the output matrix element $\gamma_{ij} = \{0, 1\}$. If t_j is the input of P_i, $\gamma_{ij} = 1$. If t_j is not an input of P_i, $\gamma_{ij} = 0$. $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

- $M = (m_1, m_2, \ldots, m_n)$ represents a distribution vector of marked places, which is the distribution of tokens in the Petri net.
- $W = (\omega_{ij})_{(n\times m)}$ is a weight matrix representing the impact of input places on transitions. $\sum_{i=1}^{n} \omega_{ij} = 1$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
- $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the place value vector, where $\alpha_i \in [0, 1]$ is the place value.
- $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ is the threshold vector, where $\lambda_i \in [0, 1]$ is the threshold.

In complex systems, the relationship between different modules is fuzzy and uncertain, therefore, extracted fault information has fuzzy characteristics [14]. Studies [15, 16] that combined fuzzy theory and Petri nets to obtain an FPN validated that it could be used for the fault diagnosis of motor systems.

2.2 CLPSO

The PSO algorithm can be used to find an optimal solution by simulating cooperation and information transfer among individuals in a group. It mainly includes two elements: speed and position [17, 18]. Each particle position represents a possible solution to the equation, and the speed represents the direction and step size of position movement.

When the PSO algorithm is updated, each particle's position and speed are randomly generated. Then, individuals in the group update their individual speeds by judging the group's global and local optimal positions to search for the global optimal position and thereby achieve the purpose of optimization [19].

However, in complex motor systems, the nonlinearity and complexity of the interrelationships between modules lead to large differences between dimensional parameters in particles. If particles are updated uniformly, the difference between dimensional parameters will be lost, resulting in local optimal conditions. The CLPSO algorithm enables independently optimizing different dimensional parameters in particles. It can effectively solve the problem of different dimensional parameters in particles and affords improved optimization ability. The algorithm is comparable to the BP neural network algorithm, which has a strong global optimization ability and prevents the problem of missing faults during fault diagnosis of the motor system. Compared with the PSO algorithm, it has an improved local optimization ability. It can prevent the problems of fault misjudgment during fault diagnosis of each module of the system. Therefore, it is more suitable for complex motors than the traditional optimization algorithm [20].

The CLPSO speed updated formula is as follows:

$$V_i^d = \omega * V_i^d + c * \mathrm{rand}_i^d * \left(X_g^d - X_i^d\right). \tag{2}$$

The CLPSO position updated formula is as follows:

$$X_i^d = X_i^d + V_i^d \tag{3}$$

where ω is the inertia constant and is a real number in the range of $[0, 1]$, c is a learning factor and is a real number in the range of $[0, 2]$, rand_i^d is a random number in the range of [0, 1], V_i^d and X_i^d are respectively the speed and position of the d^{th} dimension of the i^{th} particle, and X_g^d is the particle value of the global optimal position. The error formula is as follows:

$$E = \frac{1}{2} * \sum_{i=1}^{n} \left(\alpha\left(P_i\right) - \alpha^E\left(P_i\right)\right)^2 \tag{4}$$

where $\alpha\left(P_i\right)$ and $\alpha^E\left(P_i\right)$ are respectively the i^{th} place value obtained by reasoning and by Bayesian treatment.

2.3 CLPSO-FPN

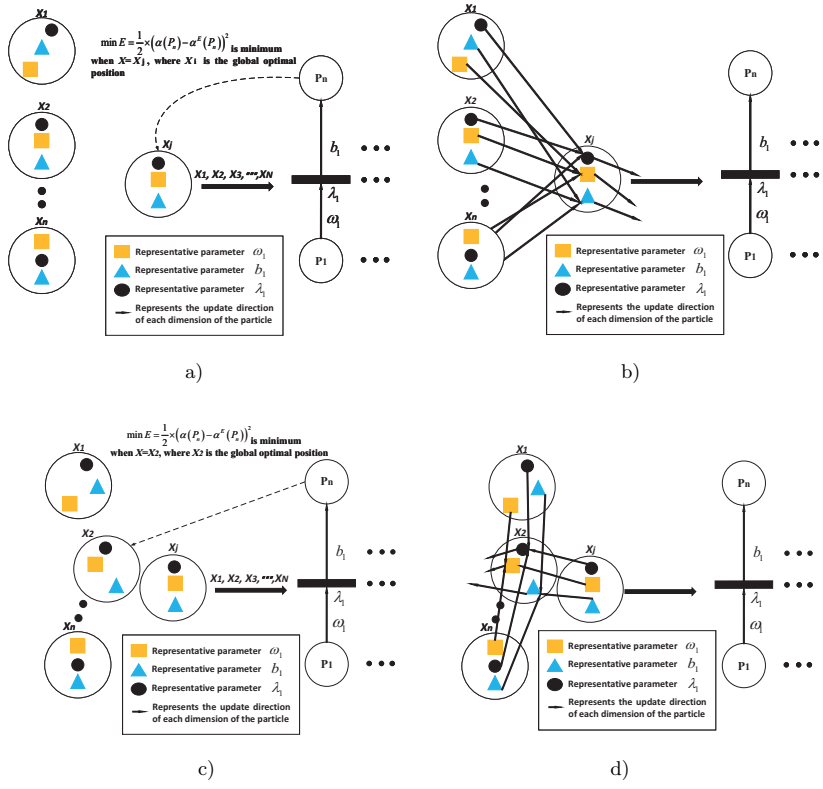


Figure 2. The CLPSO-FPN algorithm parameter generation process

Figure 2 shows the parameter generation process of the proposed CLPSO-FPN algorithm. The steps in this process are outlined below.

- The CLPSO algorithm randomly generates n particles, each of which includes ω_1, b_1, and λ_1. The parameters of the n particles are input into the FPN, and E for place P_n is obtained by supervised learning. Then, the minimum error $minE$ is obtained, and the corresponding particle X_j is the global optimal value in the next iteration process.
- The particles in the global optimal position are determined, and respective dimensional parameters in other particles are updated to the corresponding parameter directions in X_j. Further, X_j randomly updates the positions of the respective dimensional parameters.
- The n updated particles are input into the FPN to obtain particles with minimum error $minE$.
- The position of n particles is updated until the smallest error found in the end of the iteration is the optimal particle.

The CLPSO-FPN algorithm is mainly used for fault diagnosis of motor systems. First, based on the FPN principle, the system model is established according to the fault operation mechanism and fault characteristics of the system. Second, the CLPSO algorithm is used to randomly generate the weights, threshold, and transition influencing factor. Third, supervised learning is performed according to the actual place value to obtain the system parameter set of the FPN model with the smallest error. Finally, fault diagnosis of the motor system is performed according to the FPN fault diagnosis principle. This method can generate adaptive system parameters for different fault models, and effectively solve the problems of poor accuracy and adaptability of fault diagnosis in the traditional FPN algorithm that are caused by the assignment of system parameters based on experts' experience.

Next, complex motor systems involve multiple mapping relationships between the physical structure of the device and the faults [21, 22]. According to the Petri net principle, this study introduces competition operators, big operators, and a direct multiplication operator. The ability of the CLPSO-FPN algorithm to deal with the competition between different modules in complex motor systems is improved by the operator characteristics, and the matrix reasoning process of the algorithm is optimized.

- ∇:$C = \nabla A$, where A is an $(m \times n)$-dimensional matrix and C is an n-dimensional vector, such that $c_j = \max\limits_{1 \le i \le m} (a_{ij})$.
- $\oplus$:$C = A \oplus B$, where A, B, and C are all $(m \times n)$-dimensional matrices, such that $c_{ij} = \max\limits_{1 \le i \le m} (a_{ij}, b_{ij})$.
- $\otimes$:$C = A \otimes b$, where A and C are $(m \times n)$-dimensional matrices and b is an m-dimensional vector, such that $c_{ij} = a_{ij} \times b_i$.

Finally, to solve the nonlinear characteristics of the interrelationships among different modules in complex motor systems, a Gaussian function is used to replace the transition confidence, and the transition influencing factor reflects the influence of the transition on the output place. According to the above principle, CLPSO-FPN is defined as a 13-tuple as follows [23]:

$$S_{clpso-fpn} = (P, T, I, O, M, W, \alpha, \lambda, B, X, N, D, K) \tag{5}$$

where

- $B = (b_1, b_2, \ldots, b_m)$ is a transition influence factor vector representing the ability to influence the transition on output place,
- $X = \{W, B, \lambda\}$ is the particle value,
- N is the number of particles,
- D is the number of dimension,
- K is the number of iterations.

3 CLPSO-FPN FAULT DIAGNOSIS

According to the CLPSO-FPN principle, the following reasoning calculations are performed for fault diagnosis.

3.1 Transition Trigger Reasoning

$H_k = (h_1, h_2, \ldots, h_m)$ is an m-dimensional vector that is the sum of the marked place value and the corresponding weight product.

$$H_k = (\alpha_k * M_k) \otimes W. \tag{6}$$

To determine the transition trigger, the Sigmoid function is as follows:

$$s = 1/\left(1 + exp\left(-z\left(h - \lambda\right)\right)\right). \tag{7}$$

$S_k = (s_1, s_2, \ldots, s_m)$ is the pre-trigger matrix of the transition, where z is plus infinity and λ is the transition threshold. If $h \geq \lambda$, then $s(h) = 1$, otherwise $s(h) = 0$.

3.2 Fault Propagation Reasoning

The NFPN's forward dynamic fault reasoning process reflects the propagation direction of the system fault. Further, tokens reflect the occurrence of system faults. As a transition is triggered, a token is passed from the input place to the output place [23]. The fault propagation reasoning formula is as follows:

$$M_{k+1} = M_k \oplus \left[\frac{1}{1 + \exp\left[-z\left(\left(S_k \cdot O^T\right) - 1\right)\right]}\right]. \tag{8}$$

3.3 Place Value Reasoning

To judge the place value in the CLPSO-FPN model, based on the traditional FPN, this study uses the Gaussian function $1/\exp\left(-10b \times (x-1)^2\right)$ to replace the transition confidence, reflect the influence of the change in output place through the change transition influence factor, and reflect the influence of transition on the output place through the transition influence factor. For dealing with competitive characteristics in the fault diagnosis process of Petri net, the competition operator is introduced, and the matrix reasoning method is optimized. The place value reasoning is as follows:

$$\alpha_{k+1} = \alpha_k \oplus \left[\nabla \left(\left(\frac{X_k}{e^{10B_k * (X_k - 1)^2}} \right)^T \otimes O^T \right) \right], \tag{9}$$

according to the requirements of the NFPN algorithm for fault diagnosis, when the reasoning is complete.

3.4 CLPSO-FPN Fault Diagnosis Progress

In the traditional FPN-based fault diagnosis method, system parameters are usually assigned based on experts' experience, and therefore, the accuracy and adaptability of fault diagnosis are poor. To solve these problems, this study uses the CLPSO algorithm to randomly generate the weights, threshold, and transition influencing factor as dimensional parameters in particles. The update direction of each dimensional parameter of other particles is determined by finding the global optimal particle. Then, according to the comprehensive learning strategy, each dimensional parameter is differentiated, and its speed value and parameter value in the particle are updated. This method can perform differential training according to the characteristics of different parameters and can find the system parameter set under the optimal conditions. Finally, using the parameter set in the particle for fault diagnosis reasoning, the probability of each module failure occurs. The CLPSO-FPN fault diagnosis process is in Figure 3.

4 DATA PROCESSING AND MODEL CONSTRUCTION

In this study, a three-phase asynchronous motor is taken as an example to conduct model construction and data simulation. According to the data in [9], failure mode analysis and FPN are combined to analyze fault data and to establish a fault relation table that conforms to the complex motor system [24]. Table 1 shows the training sample set for the complex motor system. Further, Figure 4 shows the CLPSO model established for this motor based on its structure and fault analysis as well as the data in Table 1.

For this CLPSO-FPN model, this study uses a Bayesian method to process the fault data and combines the fault propagation mode of the motor system to determine the actual place value [25]. This method could effectively and accurately

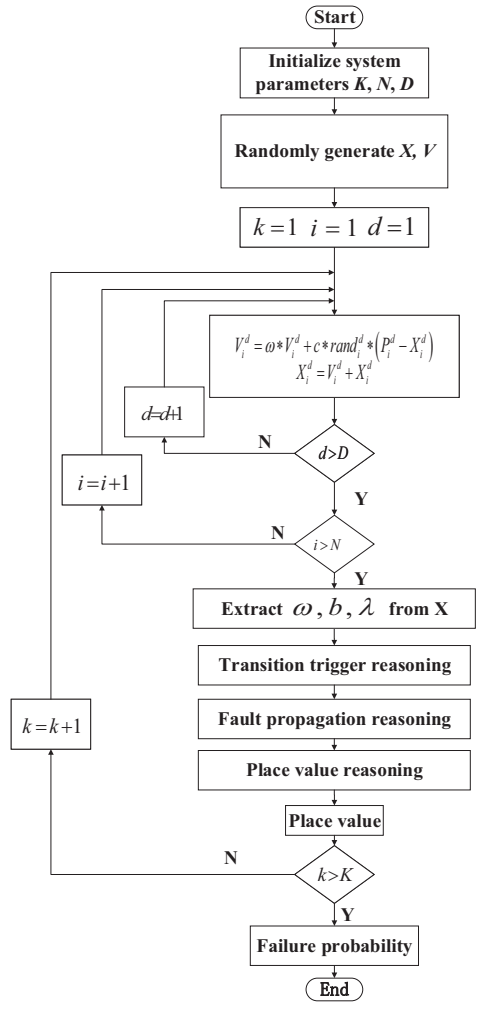


Figure 3. The CLPSO-FPN fault diagnosis process

convert knowledge and experience to rules, this is helpful in solving the problem of empirical assignment of parameters and in achieving accurate fault diagnosis of the motor.

5 METHOD IMPLEMENTATION AND VERIFICATION

5.1 Initial Value Determination

In this study, the "motor winding insulation burned" is taken as an example. According to the place value of the actual place, the CLPSO-FPN algorithm is used for supervised learning. The actual place values are shown in Table 2. When $K = 1\,000$ and $N = 40$, the error curve of the CLPSO algorithm is obtained under different numbers of iterations, as shown in Figure 5. The system parameters set is shown in Table 3.

5.2 Verification of the Method

To verify the optimization performance of the CLPSO algorithm for FPN fault diagnosis, the place values and accuracies obtained using the CLPSO, PSO, and back-propagation (BP) algorithms are compared in Figure 6 and Figure 7. These figures show that the accuracy of the place value obtained using the BP algorithm is poor, whereas that obtained using the PSO algorithm is satisfactory; however, the local optimization ability is poor. By contrast, the accuracy of the place value obtained using the CLPSO algorithm is high, and the overall optimization ability is satisfactory. The result proves that the algorithm is more suitable for fault diagnosis of complex motor systems than PSO and BP algorithms.

The effectiveness and accuracy of the CLPSO-FPN algorithm for the fault diagnosis of three-phase asynchronous motors is verified for three different fault conditions through comparisons with two previous algorithms [9, 12]; the results are shown in Table 4.

These results indicate that the FFPN and NFPN algorithms show misjudgments in the motor fault diagnosis process. By contrast, the proposed CLPSO-FPN fault diagnosis method can accurately determine the "motor winding insulation burned" problem and can effectively solve the problems of fault diagnosis and misjudgment. Further, it shows better accuracy than the FFPN and NFPN algorithms. Therefore, it can satisfy the fault diagnosis requirements of complex motor systems and shows higher accuracy and adaptability than the traditional FPN fault diagnosis method.

6 CONCLUSIONS

With the motor developing toward large-scale, complication and integrated direction, which leads to the traditional fault diagnosis method, it is difficult to meet the fault diagnosis requirements of the motor system. To solve these problems, this study proposes and applies a CLPSO-FPN algorithm to the fault diagnosis of a complex motor. In the proposed fault diagnosis method the CLPSO-FPN model of motor, using a reasoning process to diagnose fault, is established. Then a Gaussian function is used to replace the traditional transition confidence in the CLPSO-FPN

Code	Meaning	Code	Meaning
P_1	Phase winding resistance becomes smaller	P_{23}	Motor overload or irregular impact load
P_2	Rotor winding short circuit	P_{24}	Excessive bearing wear
P_3	Overload of motor	P_{25}	Motor holding shaft
P_4	Fuse melt failure	P_{26}	Bearing locking device failure
P_5	Shaft seal ring structure damage	P_{27}	Rotor core deformation
P_6	Oil seal material overheated	P_{28}	Magnetic slot wedge fracture or detachment
P_7	Seal surface axis roughness value is too large	P_{29}	Rotor winding open circuit
P_8	Temperature is too high	P_{30}	Junction box joint loosening
P_9	Exciting current is too large	P_{31}	Poor contact of the power control loop switch
P_{10}	A phase current is too large	P_{32}	Rotor winding mechanical failure
P_{11}	Rotational speed abnormality	P_{33}	The central line of motor is not consistent with the center line of shearer
P_{12}	Loss of phase voltage	P_{34}	Axial movement of rotor
P_{13}	Bearing is thermally expanded	P_{35}	Spring attachment device failure
P_{14}	Oil entering the motor	P_{36}	Scratching of motor
P_{15}	Bearing is thermally expanded	P_{37}	Stator current increase
P_{16}	Motor overheating	P_{38}	Excessive pressure drop
P_{17}	Motor in Open-phase State	P_{39}	Excessive operational shock of motor
P_{18}	Motor rotation is abnormal or card machine	P_{40}	Excessive noise of bearing
P_{19}	Motor insulation aging	P_{41}	The motor turns weak or does not rotate and buzz
P_{20}	Reduction of lubricating oil content	P_{42}	Motor running abnormal sound
P_{21}	Curved ring and axis hole produce friction	P_{43}	Motor failure
P_{22}	Motor winding insulation burned		

Table 1. Event table of places

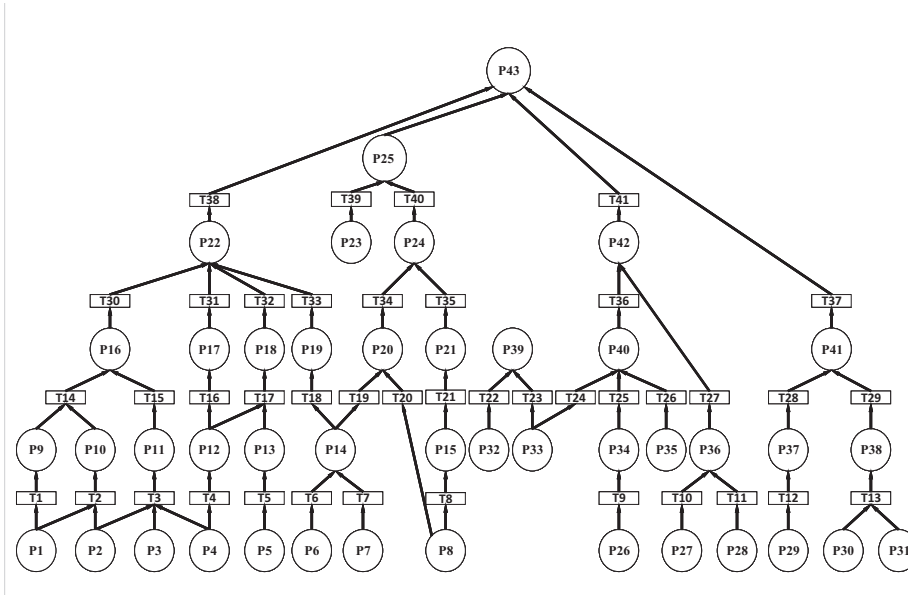


Figure 4. CLPSO-FPN model of three phase asynchronous motor

algorithm to reflect the influence of transition on the output place and a competition operator combined with a sigmoid function is proposed to determine the transition trigger reasoning. This method optimizes the matrix reasoning process compared with the traditional FPN algorithm. Finally, using the CLPSO algorithm to generate system parameters can effectively solve the problem of human's subjective

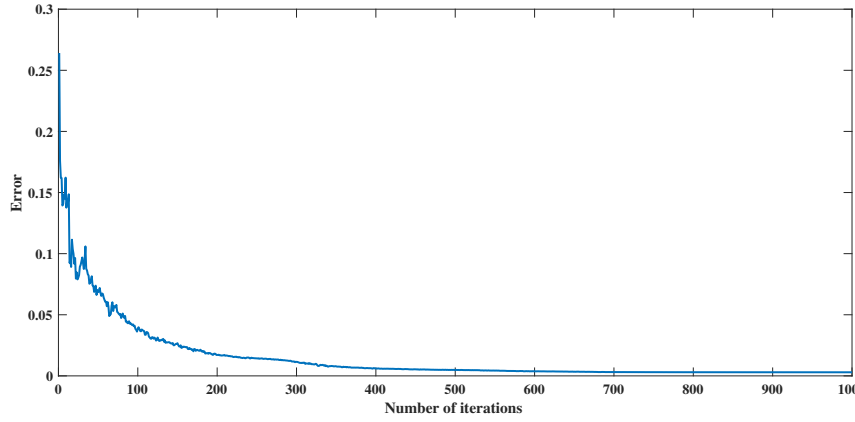


Figure 5. CLPSO iteration number error curve

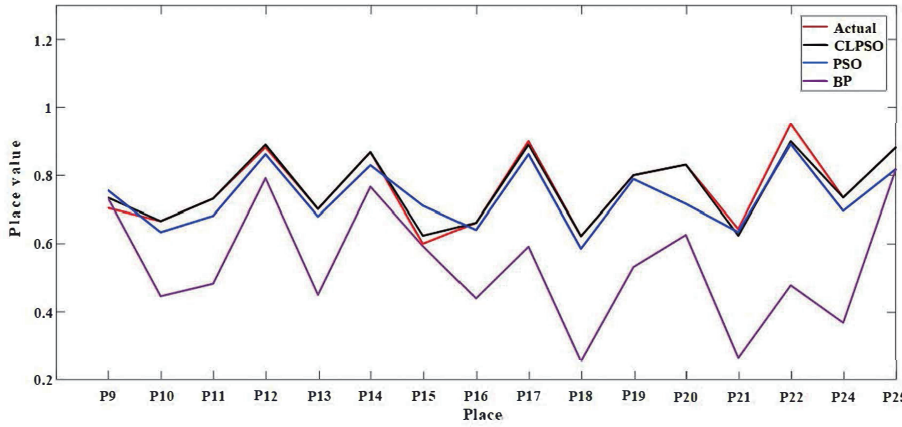


Figure 6. Place value curve

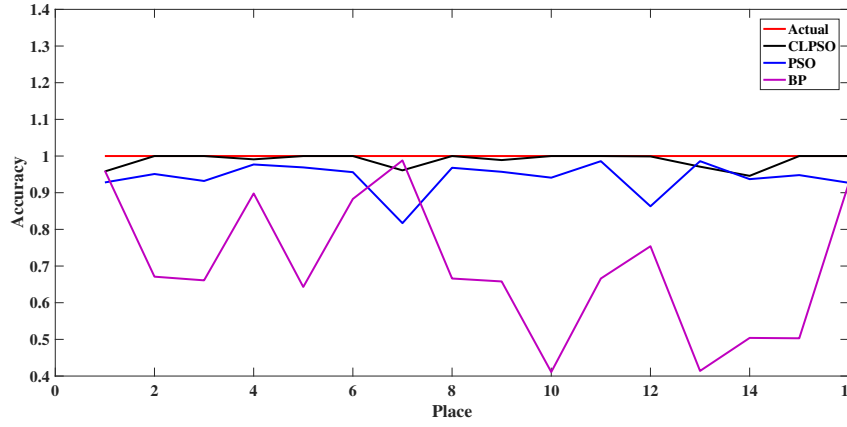


Figure 7. Accuracy curve

Code	Actual Place Value	Code	Actual Place Value
P_9	0.7051	P_{17}	0.9001
P_{10}	0.6670	P_{18}	0.6230
P_{11}	0.7320	P_{19}	0.8002
P_{12}	0.8821	P_{20}	0.8311
P_{13}	0.7020	P_{21}	0.6438
P_{14}	0.8682	P_{22}	0.9510
P_{15}	0.7320	P_{24}	0.7351
P_{16}	0.6620	P_{25}	0.8812

Table 2. Actual place values

Weights		Transition Influencing Factor		Threshold	
$\omega_{1,1}$	1	b_1	0.635	λ_1	0.488
$\omega_{1,2}$	0.519	b_2	0.557	λ_2	0.526
$\omega_{2,2}$	0.481	b_3	0.391	λ_3	0.498
$\omega_{2,3}$	0.370	b_4	0.517	λ_4	0.483
$\omega_{3,3}$	0.300	b_5	0.479	λ_5	0.484
$\omega_{4,3}$	0.330	b_6	0.334	λ_6	0.517
$\omega_{4,4}$	1	b_7	0.347	λ_7	0.524
$\omega_{5,5}$	1	b_{14}	0.485	λ_{14}	0.464
$\omega_{6,6}$	1	b_{15}	0.592	λ_{15}	0.510
$\omega_{7,7}$	1	b_{16}	0.463	λ_{16}	0.447
$\omega_{9,14}$	0.536	b_{17}	0.437	λ_{17}	0.495
$\omega_{10,14}$	0.464	b_{18}	0.298	λ_{18}	0.543
$\omega_{11,15}$	1	b_{30}	0.526	λ_{30}	0.455
$\omega_{12,16}$	1	b_{31}	0.541	λ_{31}	0.465
$\omega_{12,17}$	0.392	b_{32}	0.480	λ_{32}	0.486
$\omega_{13,17}$	0.608	b_{33}	0.404	λ_{33}	0.456
$\omega_{14,18}$	1				
$\omega_{16,30}$	1				
$\omega_{17,31}$	1				
$\omega_{18,32}$	1				
$\omega_{19,33}$	1				

Table 3. Set of system parameters

factor effectively due to the assignment of parameters based on experts' experience. Therefore, the results indicate that the method can effectively improve the accuracy and adaptability of fault diagnosis, which can improve the fault treatment and equipment maintenance of motors.

Several interesting research topics are to be considered in the future, for example,

1. the online fault diagnosis problem for motor and wireless localization systems [26],
2. the online fault prediction problem for wireless sensor networks [27].

REFERENCES

[1] Chen, X.—Bai, X.—Zhang, Q.: Micro Grid Fault Diagnosis Based on Redundant Embedding Petri Net. Systems Science and Control Engineering, Vol. 6, 2018, No. 3, pp. 289–296, doi: 10.1080/21642583.2018.1554801.

[2] Bai, X.—Wang, Z.—Zou, L.—Alsaadi, E. F.: Collaborative Fusion Estimation over Wireless Sensor Networks for Monitoring CO2 Concentration in a Greenhouse. Information Fusion, Vol. 42, 2018, pp. 119–126, doi: 10.1016/j.inffus.2017.11.001.

Method	Cause of Fault	Diagnosed Fault	Fault	Actual Failure Probability	Diagnostic Failure Probability	Accuracy
FFPN	P_4	P_{12}, P_{17}, P_{22}	P_{22}	0.8910	0.4270	47.9 %
	P_1, P_3	P_9	P_{22}	0.4121	0	0
	P_1, P_2, P_5	P_9, P_{10}, P_{13}, P_{16}	P_{22}	0.5120	0.6020	82.4 %
NFPN	P_4	P_{12}, P_{17}, P_{22}	P_{22}	0.8910	0.4794	53.8 %
	P_1, P_3	P_9, P_{16}	P_{22}	0.4121	0	0
	P_1, P_2, P_5	P_9, P_{10}, P_{13}, P_{16}	P_{22}	0.5120	0	0
CLPSO FPN	P_4	P_{11}, P_{12}, P_{16}, P_{17}, P_{18}, P_{22}	P_{22}	0.8910	0.8950	99.6 %
	P_1, P_3	P_9, P_{10}, P_{16}, P_{22}	P_{22}	0.4121	0.3665	88.9 %
	P_1, P_2, P_5	P_9, P_{10}, P_{13}, P_{16}, P_{18}, P_{22}	P_{22}	0.5120	0.3666	71.6 %

Table 4. Fault diagnosis table

[3] Deng, W.—Zhang, S.—Zhao, H.—Yang, X.: A Novel Fault Diagnosis Method Based on Integrating Empirical Wavelet Transform and Fuzzy Entropy for Motor Bearing. IEEE Access, Vol. 6, 2018, pp. 35042–35056, doi: 10.1109/ACCESS.2018.2834540.

[4] Deng, W.—Yao, R.—Zhao, H.—Yang, X.—Li, G.: A Novel Intelligent Diagnosis Method Using Optimal LS-SVM with Improved PSO Algorithm. Soft Computing, Vol. 23, 2019, No. 7, pp. 2445–2462, doi: 10.1007/s00500-017-2940-9.

[5] Lu, Y.—Wang, F.—Jia, M.: Qualitative Simulation and Fuzzy Knowledge Based Fault Diagnosis of Centrifugal Compressor Insufficient Discharge. Acta Automatica Sinica, Vol. 41, 2015, No. 11, pp. 1867–1876, doi: 10.16383/j.aas.2015.c150117 (in Chinese).

[6] Kabir, S.—Papadopoulos, Y.: Applications of Bayesian Networks and Petri Nets in Safety, Reliability, and Risk Assessments: A Review. Safety Science, Vol. 115, 2019, pp. 154–175, doi: 10.1016/j.ssci.2019.02.009.

[7] Bian, L.—Bian, C.: Review on Intelligence Fault Diagnosis in Power Networks. Power System Protection and Control, Vol. 42, 2014, No. 3, pp. 146–153 (in Chinese).

[8] Sheng, S.—Xiao, M.—Zhao, L.—Wen, Y.—Hu, B.: Research on Probability Transition Method for Fault Petri Net. Chinese Journal of Scientific Instrument, Vol. 35, 2014, No. 3, pp. 714–720, doi: 10.19650/j.cnki.cjsi.2014.03.033 (in Chinese).

[9] Cheng, X.—Wang, C.—Yu, Y.—Yi, L.—Chen, Q.: An Approach for Three-Phase Asynchronous Motor Failure Analysis Based on Fuzzy Fault Petri Net. Transactions of China Electrotechnical Society, Vol. 30, 2015, No. 17, pp. 132–139, doi: 10.19595/j.cnki.1000-6753.tces.2015.17.015 (in Chinese).

[10] Zhang, Y.—Zhang, Y.—Wen, F.—Chun, C. Y.—Tseng, C.—Zhang, X.—Zeng, F.—Yuan, Y.: A Fuzzy Petri Net Based Approach for Fault Diag-

nosis in Power Systems Considering Temporal Constraints. International Journal of Electrical Power and Energy Systems, Vol. 78, 2016, pp. 215–224, doi: 10.1016/j.ijepes.2015.11.095.

[11] Cheng, X.—Zhu, X.—Du, Y.—Wang, C.—Cao, M.: High Voltage Circuit Breaker Fault Diagnosis Based on Neural Fuzzy Petri Nets. Transactions of China Electrotechnical Society, Vol. 33, 2018, No. 11, pp. 2535–2544, doi: 10.19595/j.cnki.1000-6753.tces.170533 (in Chinese).

[12] Latsou, C.—Dunnett, S. J.—Jackson, L. M.: A New Methodology for Automated Petri Net Generation: Method Application. Reliability Engineering and System Safety, Vol. 185, 2019, pp. 113–123, doi: 10.1016/j.ress.2018.12.017.

[13] Wang, H.—Jiang, C.—Liao, S.: Concurrent Reasoning of Fuzzy Logical Petri Nets Based on Multi-Task Schedule. IEEE Transactions Fuzzy Systems, Vol. 9, 2001, No. 3, pp. 444–449, doi: 10.1109/91.928740.

[14] Dâmaso, A.—Rosa, N.—Maciel, P.: Using Coloured Petri Nets for Evaluating the Power Consumption of Wireless Sensor Networks. International Journal of Distributed Sensor Networks, Vol. 10, 2014, No. 6, Art. No. 423537, 13 pp., doi: 10.1155/2014/423537.

[15] Gong, M.—Song, H.—Tan, J.—Xie, Y.—Song, J.: Fault Diagnosis of Motor Based on Mutative Scale Back Propagation Net Evolving Fuzzy Petri Nets. 2017 Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 3826–3829, doi: 10.1109/CAC.2017.8243447.

[16] Kong, D.—Li, H.: Fuzzy Petri Nets and Its Application in Fault Diagnosis of Compressor. Computer Engineering and Design, Vol. 39, 2018, No. 1, pp. 271–275, doi: 10.16208/j.issn1000-7024.2018.01.047 (in Chinese).

[17] Lynn, N.—Suganthan, P. N.: Heterogeneous Comprehensive Learning Particle Swarm Optimization with Enhanced Exploration and Exploitation. Swarm and Evolutionary Computation, Vol. 24, 2015, pp. 11–24, doi: 10.1016/j.swevo.2015.05.002.

[18] Masdari, M.—Salehi, F.—Jalali, M.—Bidaki, M.: A Survey of PSO-Based Scheduling Algorithms in Cloud Computing. Journal of Network and Systems Management, Vol. 25, 2017, No. 1, pp. 122–158, doi: 10.1007/s10922-016-9385-9.

[19] Xu, J.—Yan, F.—Yun, K.—Ronald, S.—Li, F.—Guan, J.: Dynamically Dimensioned Search Embedded with Piecewise Opposition-Based Learning for Global Optimization. Scientific Programming, Vol. 2019, 2019, No. 1, Art. No. 2401818, 20 pp., doi: 10.1155/2019/2401818.

[20] Wang, J.—Gao, Y.—Liu, W.—Sangaiah, A. K.—Kim, H.: An Improved Routing Schema with Special Clustering Using PSO Algorithm for Heterogeneous Wireless Sensor Network. Sensors, Vol. 19, 2019, No. 3, Art. No. 671, 17 pp., doi: 10.3390/s19030671.

[21] Cheng, Y.—Wang, Z.—Zhang, W.—Huang, G.: Particle Swarm Optimization Algorithm to Solve the Deconvolution Problem for Rolling Element Bearing Fault Diagnosis. ISA Transactions, Vol. 90, 2019, pp. 244–267, doi: 10.1016/j.isatra.2019.01.012.

[22] Bai, X.—Wang, Z.—Sheng, L.—Wang, Z.: Reliable Data Fusion of Hierarchical Wireless Sensor Networks with Asynchronous Measurement for Greenhouse Moni-

toring. IEEE Transactions on Control Systems Technology, Vol. 27, 2019, No. 3, pp. 1036–1046, doi: 10.1109/TCST.2018.2797920.

[23] Li, J.—Zhu, X.—Cheng, X.: Sensor Fault Diagnosis Based on Fuzzy Neural Petri Net. Complexity, Vol. 2018, 2018, Art. No. 8261549, 11 pp., doi: 10.1155/2018/8261549.

[24] Liu, H.—Liu, L.—Lin, Q.—Liu, N.: Knowledge Acquisition and Representation Using Fuzzy Evidential Reasoning and Dynamic Adaptive Fuzzy Petri Nets. IEEE Transactions on Cybernetics, Vol. 43, 2013, No. 3, pp. 1059–1072, doi: 10.1109/TSMCB.2012.2223671.

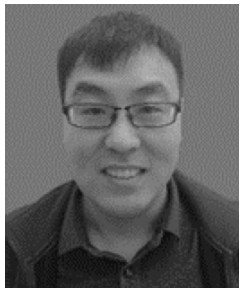
[25] Tolosana-Calasanz, R.—Bañares, J. Á.—Colom, J.-M.: Model-Driven Development of Data Intensive Applications over Cloud Resources. Future Generation Computer Systems, Vol. 87, 2018, pp. 888–909, doi: 10.1016/j.future.2017.12.046.

[26] Bai, X.—Wang, Z.—Zou, L.—Cheng, C.: Target Tracking for Wireless Localization Systems with Degraded Measurements and Quantization Effects. IEEE Transactions on Industrial Electronics, Vol. 65, 2018, No. 12, pp. 9687–9697, doi: 10.1109/TIE.2018.2813982.

[27] Bai, X.—Liu, L.—Cao, M.—Panneerselvam, J.—Sun, Q.—Wang, H.: Collaborative Actuation of Wireless Sensor and Actuator Networks for the Agriculture Industry. IEEE Access, Vol. 5, 2017, pp. 13286–13296, doi: 10.1109/ACCESS.2017.2725342.

Xuezhen Cheng received her Ph.D. degree in control theory and engineering in 2011 from the Shandong University of Science and Technology, China. She is currently serving as Professor in the School of College of Electrical Engineering and Automation, Shandong University of Science and Technology, China. Her research interests include detection technology and power system automation.

Changan Wang has his M.Sc. degree in electric power system and automation from the Shandong University of Science and Technology. His main research field is fault diagnosis.



Jiming Li is currently Ph.D. student at the Department of Electrical Engineering and Automation, Shandong University of Science and Technology. His research interests are signal processing, inspection technology and system integration.

Xingzhen Bai received his Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2010. He is currently serving as Associate Professor with the College of Electrical Engineering and Automation, Shandong University of Science and Technology. His current research interests include distributed estimation, fault diagnosis, wireless sensor network, and smart grid.

Computing and Informatics, Vol. 39, 2020, 264–297, doi: 10.31577/cai_2020_1-2_264

A LOGIC PETRI NET-BASED REPAIR METHOD OF PROCESS MODELS WITH INCOMPLETE CHOICE AND CONCURRENT STRUCTURES

Yuanxiu Teng, Liang Qi*, Yuyue Du

The College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
e-mail: 392828580@qq.com, {qiliangsdkd, yydu001}@163.com

Abstract. Current model repair methods cannot repair incomplete choice and concurrent structures precisely and simply. This paper presents a repair method of process models with incomplete choice and concurrent structures via logic Petri nets. The relation sets are constructed based on process trees, including branch sets, choice activity sets and concurrent activity sets. The deviations are determined by analyzing the relation between relation sets and activities in the optimal alignment. The model repair method is proposed for models with incomplete choice and concurrent structures via logic Petri nets according to different deviation positions. Finally, the correctness and effectiveness of the logic Petri net-based repair method are illustrated by simulation experiments.

Keywords: Process model, model repair, process tree, alignment, logic Petri net

1 INTRODUCTION

Process mining builds a bridge between data mining and process modeling and analysis. It extracts effective information from the data and resources of the real business process system, and builds the process model based on different algorithms according to the required information. Process mining is widely used in the design, analysis, implementation and adjustment of the system process [1]. The three types of applications for process mining is process discovery, conformance checking, and process improvement. The process discovery algorithm is a function that maps the

* Corresponding author

event log to a process model, which can be a BPMN [2], YAWL [3], Petri net [4, 5] and so on. α algorithm takes an event log as the input, finds out the possible causal dependence according to the sequence of activities, and outputs a Petri net with the initial identification [6]. Heuristic mining builds models with the use of representation preference and frequency of causal networks. The basic idea of heuristic mining algorithm is that infrequent paths should not be included in the model [7]. Conformance checking is used to compare the behaviors of process models with the behaviors recorded in the event logs, and it looks for their commonality and heterogeneity, so as to ensure that the information system and the actual business process keep a good compliance. The classic conformance checking contains token replay and alignments [8]. Besides, the method proposed by [9] projects both systems and system models or logs onto sub-sets of activities to determine their performance, and is applicable to both log–model and model–model conformance checking. The literature [10] can analyze and classify deviations with respect to the intended purpose of data, and provides an algorithm to identify wide range of deviations.

Conformance checking can also be used to improve business processes, repair models, and evaluate process discovery algorithms [11]. When the process model and event logs do not match on the process, the process model needs to be repaired, which is the process improvement. The aim of process improvement is to make the process model better reflect the real business system and improve the performance of the model. Fitness, simplicity, precision, and generalization are four main types of model performance, and those four performances are used to evaluate the quality of process models. A new genetic process mining algorithm is proposed to discover a process model from event logs, and it uses the tree representation to ensure the soundness of the model [12]. To improve the quality of process models, many approaches of process improvement are proposed. The Fahland's approach uses alignments to align the runs of the given process models to the traces in the logs [13]. It mines loops that can replay sub-logs of non-fitting sub-traces. The Goldratt's approach and Knapsack's approach are proposed to improve the correspondence between a model and event logs, and speed up the repair [14]. The work in [15] presents a judgment to mine the sub-process as the branch of choice structures, instead of inserting the sub-process directly into the original model.

Logic Petri net is the further abstraction and extension of the Petri net with inhibitor arcs [16, 17]. It is more concise and can describe a large number of logic relations among complex activities. From the perspective of analyzing business process operation and resources, logic Petri nets can better analyze batch processing function and the uncertainty of activity enablement of business process systems. The work in [16] proposes a vector matching method, and analyzes the reachability, liveness, conservativeness, and reversibility of logic Petri nets based on reachability trees and the state equations. The precursor and successor of activities in the traces are defined in [17], and an extended log-based ordering relationship is proposed. The work in [18] is based on alignments to repair unfitting transitions in concurrent blocks and generates a new branch containing new activities. In some real business processes, concurrency and choice exist at the same time, we call this structure is an

incomplete choice structure or incomplete concurrent structure. The current model repair methods based on Petri nets consider to improve the fitness of models, often ignore the precison and simplicity, and cannot correctly describe the logic relation among activities.

Therefore, we present a logic Petri net-based repair method. This work has the following contributions:

1. Relation sets are constructed based on process trees, including choice and concurrent activity sets, and branch sets. These relation sets can precisely locate deviations for concurrent and choice structures combining with optimal alignments.
2. The algorithms of determining deviations are presented based on relation sets and event logs. The model repair method is proposed for models with incomplete choice and concurrent structures via logic Petri nets.
3. Experimental results illustrate the correctness and effectiveness of the repair method presented in the paper.

The rest of the paper is organized as follows. The background in relevant fields is introduced in Section 2. Section 3 presents an approach to repair models with incomplete choice structures. The method of repairing models with incomplete concurrent structures is proposed in Section 4. The results and performance analysis of simulation experiments are given in Section 5. Section 6 concludes our work and draws the future work.

2 PRELIMINARIES

This section introduces some basic notions, mainly including Petri nets [4], logic Petri nets [16], alignments [11], process trees [14], and the precursor and successor [17]. In the following content, $\mathcal{N}$ represents a natural number set, i.e., $\mathcal{N} = \{0, 1, 2, \dots\}$.

Definition 1 (Trace, event log)**.** Let $A \subseteq \mathcal{A}$ be a set, and $\mathcal{A}$ is all sets of activities. $\sigma \in A^*$ is called a trace that denotes a sequence of activities. An event log is a multi-set of traces denoted as $L \in \mathcal{B}(A^*)$.

Definition 2 (Tuple)**.** Let A be a set and a tuple with n elements is denoted by $r = (b_1, b_2, \dots, b_n) \in A \times A \times \cdots \times A$. The i^{th} element of r is denoted as $\pi_i(r)$.

For example, there is a tuple $r = (x, y, z) \in A \times A \times A$ with 3 elements, $\pi_1(r) = x$, $\pi_2(r) = y$, and $\pi_3(r) = z$.

Definition 3 (Pre-set, post-set)**.** Let $N = (P, T; F)$ be a net. P denotes a finite set of places, T denotes a finite set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ denotes a finite set of directed arcs with each one from p to t or from t to p, where $p \in P$

and $t \in T$. x is a node in N and $\forall x \in P \cup T$, we have

$$^{\bullet}x = \{y | y \in (P \cup T) \wedge (y, x) \in F\},$$
$$x^{\bullet} = \{y | y \in (P \cup T) \wedge (x, y) \in F\}$$

where $^{\bullet}x$ and $x^{\bullet}$ represent the pre-set and post-set of x, respectively.

Definition 4 (Petri net)**.** A four-tuple $PN = (P, T; F, M)$ is a Petri net, where

1. $N = (P, T; F)$ is a net;
2. $M : P \to \mathcal{N}$ is a marking, $M(p)$ denotes the number of tokens in p, where $p \in P$; and
3. PN has the following transition firing rules:
   (a) for $t \in T$, if $\forall p \in^{\bullet} t : M(p) \geq 1$, then t is enabled under M, denoted as $M[t >$; and
   (b) if $M[t >$, then t can fire, and a new marking M' is generated, denoted as $M[t > M'$, and for $\forall p \in P$, we have

$$M(P)' = \begin{cases} M(P) - 1, & p \in^{\bullet} t - t^{\bullet}, \\ M(P) + 1, & p \in t^{\bullet} -^{\bullet} t, \\ M(P), & \text{otherwise.} \end{cases}$$

Definition 5 (Logic Petri net)**.** A six-tuple $LPN = (P, T; F, I, O, M)$ is called a logic Petri net, where

1. P denotes a finite set of places;
2. $T = T_D \cup T_I \cup T_O$ denotes a finite set of transitions, and $T \cap P = \phi$, for $\forall t \in T$, $^{\bullet}t \cap t^{\bullet} = \phi$, where
   (a) T_D is a set of classic transitions as in a Petri net;
   (b) T_I is a set of logic input transitions, for $\forall t \in T_I$, the input place of t is restricted by a logic input function $f_I(t)$; and
   (c) T_O is a set of logic output transitions, for $\forall t \in T_O$, the output place of t is restricted by a logic output function $f_O(t)$;
3. $F \subseteq (P \times T) \cup (T \times P)$ denotes a set of directed arcs with each one from p to t or from t to p, where $p \in P$ and $t \in T$;
4. I denotes a logic input function of t, where $t \in T_I$, and for $\forall t \in T_I$, $I(t) = f_I(t)$;
5. O denotes a logic output function of t, where $t \in T_O$, and for $\forall t \in T_O$, $O(t) = f_O(t)$;
6. $M : P \to \mathcal{N}$ is a marking, $M(p)$ denotes the number of tokens in p, where $p \in P$; and
7. LPN has the following transition firing rules:

(a) for $\forall t \in T_D$, the transition firing rule is consistent with that of Petri nets;
(b) for $\forall t \in T_I$, $I(t) = f_I(t)$. If $f_I(t)|_M =_\bullet T_\bullet$, then t can fire and is denoted as $M[t > M'$, and for $\forall p \in^\bullet t$, $M(p) = 1$, $M'(p) = 0$; for $\forall p \in t^\bullet$, $M(p) = 0$, $M'(p) = 1$; and for $\forall p \notin^\bullet t \cup t^\bullet$, $M'(p) = M(p)$; and
(c) for $\forall t \in T_O$, $O(t) = f_O(t)$. If $f_O(t)|_M =_\bullet T_\bullet$, then t can fire and is denoted as $M[t > M'$, and $\forall p \in^\bullet t$, $M'(p) = 0$; for $\forall p \in t^\bullet$, $M'(p) = 1$; and for $\forall p \notin^\bullet t \cup t^\bullet$, $M'(p) = M(p)$.

8. There are three symbols of the logic function: $\bigotimes$, $\wedge$ and $\vee$. $p_1 \bigotimes p_2 \cdots \bigotimes p_n$ denotes only one of $p_1 - p_n$ contains tokens; $p_1 \wedge p_2 \cdots \wedge p_n$ denotes each of $p_1 - p_n$ contains tokens; $p_1 \vee p_2 \cdots \vee p_n$ denotes at least one of $p_1 - p_n$ contains tokens; where $n \geq 2$.

For example, a logic Petri net denoted by LPN_1 is presented in Figure 1, where t_1 and t_3 are two logic transitions and t_2 is a classic transition. t_1 is a logic input transition, and $I(t_1) = p_2 \wedge (p_1 \bigotimes p_3)$. If t_1 fires, p_1 and p_3 cannot contain tokens at the same time, and $f_I(t_1) = p_2 \wedge (p_1 \bigotimes p_3) =_\bullet T_\bullet$, there will be two cases:

1. both p_1 and p_2 contain a token; or
2. both p_2 and p_3 contain a token.

Besides, t_3 is a logic output transition, and $O(t_3) = p_6 \vee p_7$. When t_3 fires, its logic output function needs to satisfy $f_O(t_3) = p_6 \vee p_7 =_\bullet T_\bullet$, there are three cases:

1. only p_6 contains a token;
2. only p_7 contains a token; or
3. both p_6 and p_7 contain a token.

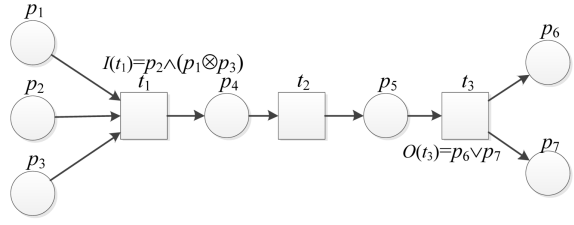


Figure 1. A logic Petri net model LPN_1

Definition 6 (Alignment)**.** Let $\sigma \in A^*$, and $PN = (P, T; F, M)$. A move is a pair $(a, t) \in (A \cup >>) \times (T \cup >>)$, where $>>$ denotes no move. A move sequence denoted by $\gamma = ((a_1, t_1)(a_2, t_2) \ldots (a_{|\gamma|}, t_{|\gamma|}))$ is called an alignment between σ and PN, where

1. $\pi_1(\gamma) = \sigma$ denotes a trace sequence generated by γ (ignoring $>>$); and

2. $m_i[\pi_2(\gamma) > m_f$ denotes a complete firing sequence generated by γ (ignoring $>>$); and
3. for each move (a, t), it is called a log activity if $a \in A$ and $t =>>$; it is called a model activity if $a =>>$ and $t \in T$; it is called a synchronous activity if $a \in A$ and $t \in T$; it is called an illegal activity otherwise.

$\Gamma_{\sigma,PN}$ denotes a set of all alignments between σ and PN.

Definition 7 (Optimal alignment)**.** Let $\sigma \in A^*$ and $PN = (P, T; F, M)$. $\gamma' \in \Gamma_{\sigma,PN}$ denotes an optimal alignment between σ and PN, if for $\forall\gamma' \in \Gamma_{\sigma,PN}$, we have $\sum_{(a,t)\in\gamma} lc(a,t) \leq \sum_{(a',t')\in\gamma} lc(a',t')$, where $lc(a,t)$ denotes a likelihood cost function. For each move $(a,\ t)$, if $a \in A$ and $t =>>$, $lc(a,t) = 1$; if $a =>>$ and $t \in T$, $lc(a,t) = 1$; if $a \in A$ and $t \in T$, $lc(a,t) = 0$. $\Gamma_{\sigma,PN,lc}$ denotes a set of all optimal alignments between σ and PN.

Definition 8 (Process tree)**.** Let $A \in \mathcal{A}$ be a set of activities. The notation of $\bigoplus$ denotes an operator set, and $\bigoplus = \{\times, \rightarrow, \bigodot, \wedge\}$, where

1. $a \in A \cup \{\tau\}$ denotes a process tree, and τ is an invisible transition; and
2. if $PT_1, PT_2, \ldots, PT_n (n > 0)$ are process trees, and then $\bigoplus(PT_1, \ldots, PT_n)$ is also a process tree; $\times$ represents the choice relation among $PT_1, \ldots, PT_n$; $\rightarrow$ represents the sequential execution of $PT_1, \ldots, PT_n$; $\bigodot$ represents the loop structure of $PT_1, \ldots, PT_n$; and $\wedge$ represents the parallel structure of $PT_1, \ldots, PT_n$.

Definition 9 (Precursor, successor)**.** Let $L \in B(A^*)$ be a log where $A \in \mathcal{A}$. For $\forall\sigma \in L$, if an activity $a \in \delta(\sigma)$ and the position index of a in σ is i, the precursor of a is denoted as ${}^{\triangle}a$ at the position with index $i-1$; and the successor of a is denoted as $a^{\triangle}$ at the position with index $i+1$.

For example, there is a trace $\langle t_1, t_2, t_5, t_3, t_4, t_5\rangle$, t_5 is the precursor of t_3, i.e., ${}^{\triangle}t_3 = t_5$; t_4 is the successor of t_3, i.e., $t_3^{\triangle} = t_4$.

3 INCOMPLETE CHOICE STRUCTURES DEVIATION REPAIR

This section proposes a repair method of process models with incomplete choice structures. For incomplete choice structures, we first present choice relation sets based on process trees of process models, including head-to-tail places, branch sets and choice activity sets. By comparing and analyzing the logic relation between activities in traces and transitions of choice relation sets, a model repair method via logic Petri nets is proposed.

In the following, we use $PN = (P, T; F, M)$ to denote a four-tuple Petri net, and use PT to denote a process tree of PN.

Definition 10 (Tree relation)**.** Let $|\ \rightarrow$ be a relation symbol of PT, and $(t_i \cup \bigoplus)^*| \rightarrow (t_j \cup \bigoplus)^*| \rightarrow \ldots | \rightarrow (t_n \cup \bigoplus)^*$ is a tree relation, where $t_i, t_j, \ldots, t_n \in T$ and $\bigoplus = \{\times, \rightarrow, \bigodot, \wedge\}$.

Definition 11 (Node relation)**.** Let $n \in (T \cup \bigoplus)$ be a node of PT, $\bigoplus = \{\times, \rightarrow, \odot, \wedge\}$. The top layer of PT is called the root node, there are six node relation:

1. if $m| \rightarrow n$ and $m \in (T \cup \bigoplus)$, then m is called the parent node of n, denoted as $n.parent = m$;
2. if $n| \rightarrow m$ and $m \in (T \cup \bigoplus)$, then m is called the child node of n, denoted as $n.child = m$;
3. if $m| \rightarrow n$, $m| \rightarrow l$, $m| \rightarrow r$ and $m, l, r \in (T \cup \bigoplus)$, then l and r are called the left and right sibling nodes of n, respectively, where l and r are on the left and right sides of n, denoted as $n.lsib = l$ and $n.rsib = r$;
4. if $\exists n.child \in (T \cup \bigoplus)$ and $|n.child| > 1$, then $n.child.p_i$ denotes the i^{th} child node of n, where $i \in [1, |n.child|]$;
5. if $n| \rightarrow \ldots | \rightarrow m$, $m.child =$ null, and $m.lib =$ null, then m is called the leftmost leaf node of n, denoted as $n.child.lp = m$; and
6. if $n| \rightarrow \ldots | \rightarrow m$, $m.child =$ null, and $m.rib =$ null, then m is called the rightmost leaf node of n, denoted as $n.child.rp = m$.

For example, a Petri net model denoted by PN_1 is presented in Figure 2, and PN_1 contains choice structures. Let $\sigma_1 = \langle a, t_1, t_5, t_2, t_4, b\rangle$, $\sigma_2 = \langle a, t_6, t_7, c, b\rangle$ be two traces. PN_1 has a choice structure with transitions t_1, t_2, t_3, t_4, t_5, t_6, t_7. In the trace σ_1, we have that t_1, t_2, t_4 and t_5 are concurrently enabled. It shows that the branch containing t_1, t_2, t_4 can occur concurrently with the branch containing t_5. In the original model PN_1, either of these two branches can be selected to occur. So there is an incomplete choice structure.

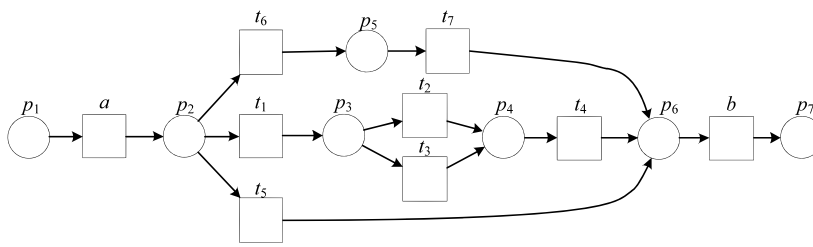


Figure 2. A Petri net model PN_1

Figure 3 shows the process tree of PN_1 represented by PT_1. $PT_1 = \rightarrow (a, \times((\rightarrow (t_1, \times(t_2, t_3), t_4), t_5, \rightarrow (t_6, t_7)), b)$. For PT_1, "$\rightarrow$"$| \rightarrow$ "$\times$", "$\times$"$| \rightarrow$ "$\rightarrow$", and "$\rightarrow$"$| \rightarrow t_1$ are three tree relations of PT_1. The root node is "$\rightarrow$". If $n =$ "$\times$", then $n.parent =$ "$\rightarrow$", $n.lsib = a$, and $n.rsib = b$. It can be seen that t_5 is the child node of n, $n.child.p_1 =$ "$\rightarrow$", $n.child.p_2 = t_5$, and $n.child.p_3 =$ "$\rightarrow$". Besides, $n.child.lp = t_1$, and $n.child.rp = t_7$.

Definition 12 (Head-to-tail place)**.** $[SF_P, SL_P]$ is called the head-to-tail place of a choice structure, where SF_P and SL_P are called the head place and the tail place, respectively. Let $n =$ "$\times$" be a node of PT, and it needs to satisfy:

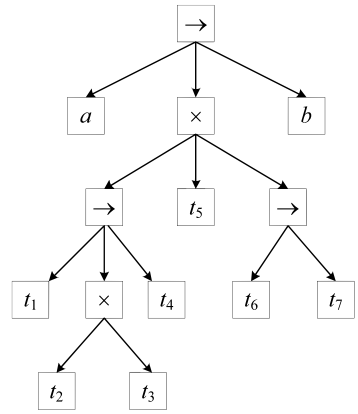


Figure 3. The process tree PT_1 of PN_1

1. if $n.child.p_i = m$, $m.child =$ null, and $0 < i \leq |n.child|$, then $SF_P =^{\bullet} m$ and $SL_P = m^{\bullet}$; and
2. if $n.child.p_i = m$, $\exists m.child \in (T \cup \bigoplus)$, $\bigoplus = \{\times, \rightarrow, \odot, \wedge\}$ and $0 < i \leq |n.child|$, then $SF_P =^{\bullet} (n.child.lp)$ and $SL_P = (n.child.rp)^{\bullet}$.

Definition 13 (Branch set)**.** Let n be a node of PT, $n =$ "$\times$" or $n =$ "$\wedge$", and a branch is defined as follows:

1. If $n.child = m$, $m.child =$ null, i.e., $n.child.child =$ null, then $\{m\}$ is a branch; and
2. if $n.child = m$, $\exists m.child.p_1, m.child.p_2, \ldots, m.child.p_k$, and $0 < k \leq |m.child|$, where
   (a) if $m =$ "$\rightarrow$", then $\{m.child.p_1,\ m.child.p_2, \ldots, m.child.p_k\}$ is a branch;
   (b) if $m =$ "$\times$", then $\{m.child.p_1\}$, $\{m.child.p_2\}$, $\ldots$, $\{m.child.p_k\}$ are each a branch; and
   (c) if $m =$ "$\wedge$", then $\{m.child.p_1\}$, $\{m.child.p_2\}$, $\ldots$, $\{m.child.p_k\}$ are each a branch.

B_S denotes a branch set that contains all branches of choice and concurrent structures.

Theorem 1. For $n =$ "$\times$", if $\exists n.child = m$ and $m.child =$ null, then $\{m\}$ is a branch.

Proof. If $\exists n.child = m$ and $m.child =$ null, i.e., $\exists n.child.child =$ null, it means that the child node of n is a leaf node. $\neg\exists m.child| \rightarrow \bigoplus$, where $\bigoplus = \{\times, \rightarrow, \odot, \wedge\}$, i.e., there are no structures behind m until SL_P. We have $SL_P = m^{\bullet}$. Since $m.parent = n$ and $n =$ "$\times$", n is the initial operator notation of the choice structure, then $SF_P =^{\bullet} m$. Thus, $\{m\}$ is a branch.

If $\{m\}$ is a branch and $n =$ "$\times$", it means $m^{\bullet} = SL_P$ and ${}^{\bullet}m = SF_P$. We have $\neg\exists m.child| \rightarrow \bigoplus$ and "$\times$"$| \rightarrow m$, i.e., $m = n.child$ and $m.child =$ null. □

Theorem 1 indicates that for the node of the process tree, if the node is "$\times$" and the child node of the node is a leaf node, then the child node alone is a branch of the choice structure.

Theorem 2. For $n =$ "$\times$" or $n =$ "$\wedge$", $m = n.child$, $0 < k \leq |m.child|$, $\{m.child.p_1\}$, $\{m.child.p_2\}$, ..., $\{m.child.p_k\}$ are each a branch.

Proof. For $n =$ "$\times$" or $n =$ "$\wedge$", there is $\exists q = m.lsib$ or $\exists q{=}m.rsib$, and $n| \rightarrow m.lsib$ or $n| \rightarrow m.rsib$. If $q.child.lp$ and m are contained in a branch, we have $M[q.child.lp >$, $\neg M[m >$, $M[q.child.lp > M'$, $M'[m >$ or $M[m >$, $\neg M[q.child.lp >$, $M[m > M'$, $M'[q.child.lp >$. For $n =$ "$\times$", $\exists M$, $M[q.child.lp >$, $\neg M[m >$, when $M[q.child.lp > M'$, $\neg M'[m >$ or $M[m >$, $\neg M[q.child.lp >$, $M[m > M'$, $\neg M'[q.child.lp >$, i.e., $q.child.lp$ and m are not contained in a branch. For $n =$ "$\wedge$", $\exists M, M[q.child.lp >$, $M[m >$, $M[q.child.lp > M'$, $M'[m >$ or $M[m > M'$, $M'[q.child.lp >$, i.e., $q.child.lp$ and m are not contained in a branch. Thus, $\{m.child.p_1\}$, $\{m.child.p_2\}$, ..., $\{m.child.p_k\}$ are each a branch. □

By Theorem 2, for the node denoted by "$\times$" or "$\wedge$" of the process tree, those child nodes of this node are not in the same branch.

Definition 14 (Choice activity set)**.** The choice activity set contains first and last choice activity sets. FC_S and LC_S denote the first and the last choice activity sets, respectively. If $n =$ "$\times$" is a node of PT, we have:

1. if $n.child = m$, $m.child =$ null, then $FC_S = \{m\}$, and $LC_S = \{m\}$;
2. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\wedge$" or $m =$ "$\times$", $k = |m.child|$, $q = n.child.lp$ and $m = q.parent$, then $FC_S = \{m.child.p_1, m.child.p_2, \ldots, m.child.p_k\}$;
3. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\wedge$" or $m =$ "$\times$", $k = |m.child|$, $q = n.child.rp$ and $m = q.parent$, then $LC_S = \{m.child.p_1, m.child.p_2, \ldots, m.child.p_k\}$; and
4. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\rightarrow$", and $m.child.child =$ null, then $FC_S = \{m.child.lp\}$, and $LC_S = \{m.child.rp\}$.

The algorithms of calculating branch sets, head-to-tail place and choice activity sets are proposed in the following.

Algorithm 1 calculates the branch set. For each node n of the process tree, if n is "$\wedge$" or "$\times$", and the child node of n is a leaf node, then this child node of n is contained in a branch. If the child node of n is "$\rightarrow$", and those child nodes of "$\rightarrow$" are leaf nodes, then all those leaf nodes are contained in a branch. If the child node of "$\rightarrow$" is "$\times$", then each node of the child node of "$\times$" is contained in a branch.

Algorithm 2 calculates the head-to-tail place and the choice activity set. For each node n of the process tree, if n is "$\times$", its head place is the pre-set of the leftmost leaf node of n, and its tail place is the post-set of the rightmost leaf node

Algorithm 1 $Calculate - B_S(PT)$

Input: A process tree denoted by PT
Output: The branch set denoted by B_S

```
1: B_S ← φ; C_TB ← φ;
2: for each n ∈ PT do
3:     if n = "∧"||n = "×" then
4:         for (i = 1; i ≤ |n.child| ; i++) do
5:             m_i ← n.child.p_i;
6:             if m_i.child = null then
7:                 B_S ← B_S ∪ {m_i};
8:             end if
9:             if m_i = "→" and m_i.child.child = null then
10:                for (j = 1; j ≤ |m_i.child| ; j++) do
11:                    C_TB ← C_TB ∪ m_i.child.p_j;
12:                end for
13:            end if
14:            if m_i = "→" and m_i.child = "×" then
15:                for each m_i.child.child do
16:                    C_TB ← C_TB ∪ m_i.child.child;
17:                end for
18:            end if
19:            B_S ← B_S ∪ C_TB;
20:        end for
21:    end if
22: end for
23: return B_S
```

of n. If the child node is a leaf node, then it is both contained in the first and last choice activity set. If the child node g of n is "$\times$" or "$\wedge$", and g is the parent node of the leftmost of rightmost leaf node of n, then each child node of g is contained in the first or last choice activity set. If g is "$\rightarrow$" and the child node of g is a leaf node, then the leftmost leaf node of g in contained in the first choice activity set, and the rightmost leaf node of g in contained in the last choice activity set.

For PT_1, we can obtain its head-to-tail place, branch set and choice activity set. Its head-to-tail place is denoted by $[SF_P, SL_P] = [p_2, p_6]$, and its branch set is denoted by $B_S = \{\{t_1, t_2, t_4\}, \{t_1, t_3, t_4\}, \{t_5\}, \{t_6, t_7\}\}$. The first choice activity set is denoted by $FC_S = \{t_1, t_5, t_6\}$ and the last choice activity set is denoted by $LC_S = \{t_4, t_5, t_7\}$.

Theorem 3. $|B_S|_{max}$ is the maximum length of elements in B_S. For $n =$ "$\times$" or $n =$ "$\wedge$", $n| \rightarrow w| \rightarrow \ldots| \rightarrow m$, if $w =$ "$\rightarrow$", $\exists m =$ "$\rightarrow$", $x = |w.child|$ and $y = |m.child|$, then $|B_S|_{max} = x + y$, where $w.child.child =$ null and $m.child.child =$ null; If $\neg\exists m =$ "$\rightarrow$", $|B_S|_{max} = x + 1$.

Algorithm 2 $Calculate - FLC(PT)$

Input: A process tree denoted by PT

Output: The head-to-tail place denoted by $[SF_P, SL_P]$, the first choice activity set denoted as FC_S, and the last choice activity set denoted as LC_S

1: $SF_P \leftarrow \phi$; $SL_P \leftarrow \phi$; $FC_S \leftarrow \phi$; $LC_S \leftarrow \phi$; $u \leftarrow \phi$; $q_1 \leftarrow \phi$; $q_2 \leftarrow \phi$;
2: **for** each $n \in PT$ **do**
3: **if** $n =$ "$\times$" **then**
4: $u \leftarrow |n.child|$;
5: $m \leftarrow n.child.p_1$;
6: $k \leftarrow n.child.p_u$;
7: $SF_P \leftarrow SF_P \cup^{\bullet} (m.child.lp)$;
8: $SL_P \leftarrow SL_P \cup (k.child.rp)^{\bullet}$;
9: **for** $(i = 1; i \leq |n.child|\,; i++)$ **do**
10: $g_i \leftarrow n.child.p_i$;
11: **if** $g_i.child = null$ **then**
12: $FC_S \leftarrow FC_S \cup \{g_i.child\}$;
13: $LC_S \leftarrow LC_S \cup \{g_i.child\}$;
14: **end if**
15: $q_1 \leftarrow n.child.lp$;
16: $q_2 \leftarrow n.child.rp$;
17: **if** $g_i =$ "$\times$"$||g_i =$ "$\wedge$" and $(g_i = q_1.parent)$ **then**
18: **for** each $g_i.child$ **do**
19: $FC_S \leftarrow FC_S \cup g_i.child$;
20: **end for**
21: **end if**
22: **if** $g_i =$ "$\times$"$||g_i =$ "$\wedge$" and $(g_i = q_2.parent)$ **then**
23: **for** each $g_i.child$ **do**
24: $LC_S \leftarrow LC_S \cup g_i.child$;
25: **end for**
26: **end if**
27: **if** $g_i =$ "$\rightarrow$" and $g_i.child.child$ **then**
28: $FC_S \leftarrow FC_S \cup g_i.child.lp$;
29: $LC_S \leftarrow LC_S \cup g_i.child.rp$;
30: **end if**
31: **end for**
32: **end if**
33: **end for**
34: **return** $[SF_P, SL_P]$, FC_S, LC_S

Proof. For $n =$ "$\times$" or $n =$ "$\wedge$", $n| \rightarrow w| \rightarrow \ldots| \rightarrow m$, $w =$ "$\rightarrow$", if $\exists d_1 = w.child.p_i$, $d_1.child =$ null, $d_2 = w.child.p_j$, $d_2.child =$ null, where $i, j \in [1, |w.child|]$, it means that $M[d_1 >$, $\neg M[d_2 > M'$, $M[d_1 > M'$, $M'[d_2 >$, so the maximum length of the element of B_S only containing d_1 and d_2 is 2. So if $\exists d = w.child$, $d.child =$ null, $|d| = x$, where $0 < x \leq |w.child|$, then the length of the element in B_S containing d is x. If $\exists m =$ "$\rightarrow$", if $\exists e_1 = m.child.p_i$, $e_1.child =$ null, $e_2 = m.child.p_j$, $e_2.child =$ null, where $i, j \in [1, |m.child|]$, it means that $M[e_1 >$, $\neg M[e_2 > M'$, $M[e_1 > M'$, $M'[e_2 >$, in the same way, the length of the element in B_S only containing e_1 and e_2 is 2. So if $\exists e = m.child$, $e.child =$ null, $|e| = y$, where $0 < y \leq |m.child|$, then the maximum length of the element in B_S containing e is y, thus, $|B_S|_{max} = x + y$; If $\exists m =$ "$\rightarrow$", it means that $M[m.child.p_i >$, $M[m.child.p_j >$, $M[m.child.p_i > M'$, $M'[m.child.p_j >$ or $M[m.child.p_i >$, $M[m.child.p_j >$, $M[m.child.p_i > M'$, $\neg M'[m.child.p_j >$, $i \neq j$ and $0 < i, j \leq |m.child|$, so the maximum length of the element in B_S containing $m.child$ is 1, thus, $|B_S|_{max} = x + 1$. □

By Theorem 3, in the process tree, the maximum length of the branch of the choice or concurrent structure is the sum of the number of leaf nodes of the child node of "$\rightarrow$" and the number of nodes "$\times$" and "$\wedge$".

Deviations are determined by judging whether the corresponding transitions of log activities are in a branch. For those log activities, we can obtain the branches that need to be concurrent with other branches. New activities that need to be inserted into the original models can also be obtained from the optimal alignment. The algorithm for determining the deviation position is given in the following.

Algorithm 3 calculates the deviation position for incomplete choice structures. We obtain the positions of SF_P and SL_P in the optimal alignment, and collect new activities. For activities between SF_P and SL_P in the optimal alignment, if the activity is a log activity and the corresponding transition is contained in the first choice activity set, we determine whether all transitions of the whole branch headed by this transition are log activities; if they are, we regard the first and last transitions of the branch as a deviation position.

γ_1=

a	t_1	t_5	t_2	t_4	b
a	t_1	>>	t_2	t_4	b

Figure 4. An optimal alignment γ_1 between σ_1 and PN_1

γ_2=

a	t_6	t_7	c	b
a	t_6	t_7	>>	b

Figure 5. An optimal alignment γ_2 between σ_2 and PN_1

Algorithm 3 Determining deviation positions for incomplete choice structures

Input: A process tree denoted by PT, a Petri net denoted by $PN = (P, T; F, M)$, and the optimal alignment denoted by $\Gamma_{\sigma,PN,lc}$

Output: The deviation position denoted by DP_{COS}, and the new activity set denoted by AL

1: $DP_{COS} \leftarrow \phi$; $AL \leftarrow \phi$;
2: $k \leftarrow 0$; $g \leftarrow 0$; $h \leftarrow 0$;
3: $B_S \leftarrow Calculate - B_S(PT)$;
4: $[SF_P, SL_P], FC_S, LC_S \leftarrow Calculate - FLP(PT)$;
5: **for** $(i = 1; i \leq |\gamma|; i++)$ **do**
6: **if** $\pi_1(\gamma[i]) = SF_P$ **then**
7: $k \leftarrow i$;
8: **end if**
9: **if** $\pi_1(\gamma[i]) = SL_P$ **then**
10: $g \leftarrow i$;
11: **end if**
12: **if** $\pi_2(\gamma[i]) =>>$ and $\pi_1(\gamma[i]) \notin \delta(PN)$ **then**
13: $AL \leftarrow AL \cup \{(\pi_1(\gamma[i]), >>)\}$;
14: **end if**
15: **end for**
16: **for** $(i = k + 1; i < g; i++)$ **do**
17: **if** $\pi_1(\gamma[i]) \in FC_S$ and $\pi_2(\gamma[i]) =>>$ **then**
18: **for** $(j = 1; j \leq |B_S|; j++)$ **do**
19: **if** $\pi_1(\gamma[i]) \in B_S$ **then**
20: $h \leftarrow j$;
21: **for** $(v = 2; \pi_h(\pi_v(B_S)) \in \delta(\sigma); v++)$ **do**
22: Continue;
23: **end for**
24: **end if**
25: **end for**
26: **end if**
27: **end for**
28: $DP_{COS} \leftarrow DP_{COS} \cup \{(\pi_h(\pi_1(B_S)), (\pi_h(\pi_v(B_S)))\}$;
29: **return** DP_{COS}, AL

Figure 4 is an optimal alignment γ_1 between σ_1 and PN_1, and Figure 5 is an optimal alignment γ_2 between σ_2 and PN_1. σ_1 has transitions of choice structures with t_1, t_5, t_2, t_4, and the branch set of PN_1 is denoted as $B_S = \{\{t_1, t_2, t_4\}, \{t_1, t_3, t_4\}, \{t_5\}, \{t_6, t_7\}\}$. Its head-to-tail place is denoted as $[SF_P, SL_P] = [p_2, p_6]$. Its first choice activity set is denoted as $FC_S = \{t_1, t_5, t_6\}$, and the last choice activity set is denoted as $LC_S = \{t_4, t_5, t_7\}$. We can find that the branch $\{t_5\}$ occurs concurrently with the branch $\{t_1, t_2, t_4\}$ by traversing the optimal alignment. For γ_1, $(t_5, >>)$ is a log activity, and t_5 is both contained in first and last choice activity sets, so

its deviation position denoted by $DP_{COS} = \{(t_5, t_5)\}$. For γ_2, its new activity set denoted by AL is $\{c\}$.

An algorithm is proposed to repair models with incomplete choice structures via logic Petri nets according to the deviation position in the following content.

Algorithm 4 Repair models for incomplete choice structures

Input: The deviation position denoted by DP_{COS}, a Petri net denoted by $PN = (P, T; F, M)$, the head-to-tail place denoted by $[SF_P, SL_P]$, and the new activity set denoted by AL

Output: A logic Petri net denoted by $LPN' = (P', T'; F', I', O', M')$

1: $LPN' \leftarrow PN$;
2: **for** each $(t_i, t_j) \in DP_{COS}$ **do**
3: $P' \leftarrow P' \cup P_{new1} \cup P_{new2}$;
4: $T' \leftarrow T'$;
5: $F' \leftarrow F' - \{SF_P \rightarrow t_i\} - \{t_j \rightarrow SL_P\} \cup \{\{^\bullet SF_P\} \rightarrow P_{new1}\} \cup \{P_{new1} \rightarrow t_i\} \cup \{t_j \rightarrow P_{new2}\} \cup \{P_{new2} \rightarrow \{SL_P^\bullet\}\}$;
6: $I' \leftarrow I' \cup \{I(SL_P^\bullet) = \{SL_P\} \bigotimes P_{new2} \bigotimes (\{SL_P\} \wedge P_{new2})\}$;
7: $O' \leftarrow O' \cup \{O(^\bullet SF_P) = \{SF_P\} \bigotimes P_{new1} \bigotimes (\{SF_P\} \wedge P_{new1})\}$;
8: **end for**
9: **for** each $t \in AL$ **do**
10: $P' \leftarrow P' \cup P_{new}$;
11: $T' \leftarrow T' \cup t$;
12: $F' \leftarrow F' \cup \{^\triangle t \rightarrow P_{new}\} \cup \{P_{new} \rightarrow t\} \cup \{t \rightarrow SL_P\}$;
13: $O' \leftarrow O' \cup \{O(^\triangle t) = P_{new} \bigotimes \{SL_P\}\}$;
14: **end for**
15: **return** LPN

Algorithm 4 repairs models with incomplete choice structures. For each deviation position and new activity, we add different logic input and output transitions.

For σ_1 and σ_2, its deviation position and new activity set are denoted as $DP_{COS} = \{(t_5, t_5)\}$ and $AL = \{c\}$. For $DP_{COS} = \{(t_5, t_5)\}$, we delete the arc from p to t_5 and the arc from t_5 to p', where $p \in^\bullet t_5$ and $p' \in t_5^\bullet$. Then we add two places, and add the arc from a to the new place and the arc from the new place to t_5, where $a \in^\bullet p$; we also add the arc from t_5 to the new place and the arc from the new place to b, where $b \in p'^\bullet$. We change a to a logic output transition and change b to a logic input transition. For $AL = \{c\}$, $^\triangle c = t_7$ and $c^\triangle = b$. We add a place and new transition c. Then we insert three arcs from t_7 to the new place and from the new place to c and from c to b into the model. Besides, we change t_7 to a logic output transition. The repaired model by our approach denoted as LPN_2 is shown in Figure 6.

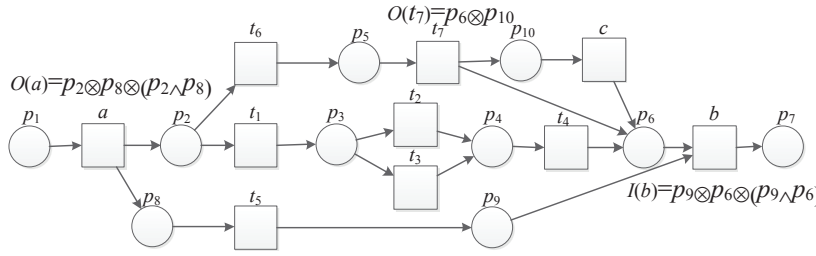


Figure 6. The repaired model LPN_2 by our approach

4 INCOMPLETE CONCURRENT STRUCTURES DEVIATION REPAIR

This section presents a repair method of the model with incomplete concurrent structures. For incomplete concurrent structures, we propose concurrent relation sets, including head-to-tail transitions and concurrent activity sets. By comparing and analyzing the logic relations between activities in traces and transitions in concurrent activity sets, we can determine the deviation position of incomplete concurrent structures, and repair models with incomplete concurrent structures via logic Petri nets.

In the following, we use $PN = (P, T; F, M)$ to denote a four-tuple Petri net, and use PT to denote a process tree of PN.

Definition 15 (Head-to-tail transition)**.** $[SF_T, SL_T]$ is called the head-to-tail transition of a concurrent structure, and SF_T and SL_T are called the head transition and the tail transition, respectively, where $SF_T = n.lsib$ and $SL_T = n.rsib$, $n =$ "$\wedge$" and $n \in PT$.

Theorem 4. For $n =$ "$\wedge$", if $\exists n.child = m$ and $m.child$=null, then $\{m\}$ is a branch.

Proof. If $\exists n.child = m$ and $m.child =$ null, it means that the child node of n is a leaf node, i.e., $\neg\exists m.child| \rightarrow \bigoplus$, where $\bigoplus = \{\times, \rightarrow, \odot, \wedge\}$. Since $n =$ "$\wedge$", $n.lsib = SF_T$ and $n.rsib = SL_T$. We have $SF_T.parent = SL_T.parent = n.parent =$ "$\rightarrow$". Since $m = n.child$ and $m.child =$ null, it means that SF_T, m and SL_T fire in order, i.e., $SF_T =^{\bullet} ({}^{\bullet}m)$ and $SL_T = (m^{\bullet})^{\bullet}$. Thus, $\{m\}$ is a branch.

If $\{m\}$ is a branch and $n =$ "$\wedge$", it means $n.lsib = SF_T$ and $n.rsib = SL_T$. We have $SF_T =^{\bullet} ({}^{\bullet}m)$ and $SL_T = (m^{\bullet})^{\bullet}$. Thus, $\neg\exists m.child| \rightarrow \bigoplus$ and "$\wedge$"$| \rightarrow m$, i.e., $m = n.child$ and $m.child =$ null. □

By Theorem 4, for the node denoted by "$\wedge$" of the process tree, if the child node of the node is a leaf node, then this child node is alone a branch in the concurrent structure.

Definition 16 (Concurrent activity set)**.** First and last concurrent activity sets are called the concurrent activity set together. FU_S and LU_S denote the first and the

last concurrent activity sets, respectively. If $n =$ "$\wedge$" and $n \in PT$, they need to satisfy:

1. if $n.child = m$, $m.child =$ null, then $FU_S = \{m\}$, and $LU_S = \{m\}$;
2. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\wedge$" or $m =$ "$\times$", $k = |m.child|$, $q = n.child.lp$ and $m = q.parent$, then $FU_S = \{m.child.p_1, m.child.p_2, \ldots, m.child.p_k\}$;
3. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\wedge$" or $m =$ "$\times$", $k = |m.child|$, $q = n.child.rp$ and $m = q.parent$, then $LU_S = \{m.child.p_1, m.child.p_2, \ldots, m.child.p_k\}$; and
4. if $n| \rightarrow \ldots| \rightarrow m$, $m =$ "$\rightarrow$", and $m.child.child =$ null, then $FU_S = \{n.child.lp\}$, and $LU_S = \{n.child.rp\}$.

The algorithm of calculating head-to-tail transitions and concurrent activity sets is given in the following.

Algorithm 5 calculates the head-to-tail transition and the concurrent activity set. For each node n of process tree, if n is "$\wedge$", its head transition is the pre-set of the pre-set of the leftmost leaf node of n, and its tail place is the post-set of the post-set of the rightmost leaf node of n. For each child node of n, if its child node is a leaf node, then the child node of n is contained in the first and last concurrent activity sets. If the child node g of n is "$\wedge$" or "$\times$", and g is the parent node of the leftmost or rightmost leaf node of n, then each child node of g is contained in the first or last concurrent activity set. If g is "$\rightarrow$" and the child node of g is a leaf node, then the leftmost leaf node of g is contained in the first concurrent activity set, and the rightmost leaf node of g is contained in the last concurrent activity set.

Figure 7 shows a Petri net model PN_2, and PN_2 contains concurrent structures. Let $\sigma_3 = \langle a, t_1, t_3, t_6, e, c\rangle$ and $\sigma_4 = \langle a, b, t_4, t_5, f, c\rangle$ be two traces. PN_2 has a concurrent structure with transitions t_1, t_2, t_3, t_4, t_5, t_6. In the trace σ_3, we have that t_1, t_3 and t_6 are concurrently enabled. For PN_2, according to Algorithm 1, we can calculate its branch set denoted by $B_S = \{\{t_1, t_3\}, \{t_2, t_3\}, \{t_4, t_5\}, \{t_6\}\}$. It shows that the branches $\{t_1, t_3\}$ and $\{t_6\}$ can occur selectively with the branch $\{t_4, t_5\}$. In the original model PN_2, all these branches need to occur. So this is an incomplete concurrent structure.

Figure 8 shows the process tree of PN_2 represented by PT_2. We have $PT_2 = \rightarrow (a, b, \wedge(\rightarrow (\times(t_1, t_2), t_3), \rightarrow (t_4, t_5), t_6), c)$. For PT_2, we can obtain the head-to-tail transition, the branch set and the concurrent activity set. Its head-to-tail transition is denoted as $[SF_T, SL_T] = [b, c]$, and its branch set is denoted as $B_S = \{\{t_1, t_3\}, \{t_2, t_3\}, \{t_4, t_5\}, \{t_6\}\}$. The first concurrent activity set is denoted as $FU_S = \{t_1, t_2, t_4, t_6\}$, and the last concurrent activity set is denoted as $LU_S = \{t_3, t_5, t_6\}$.

For incomplete concurrent structures, we can obtain log activities and model activities from the optimal alignment. For those model activities, we can obtain branches that need to occur selectively with other branches. For log activities, we can obtain new activities that need to be inserted into the original model. The algorithm of determining deviation positions is given in the following content.

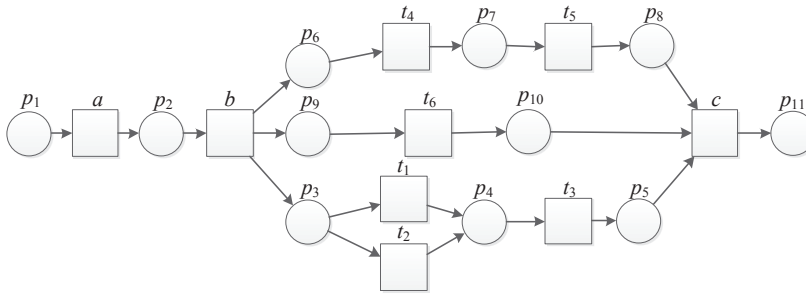


Figure 7. A Petri net model PN_2

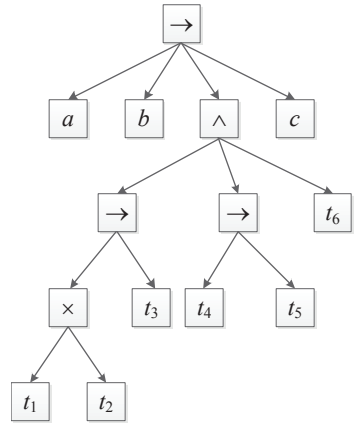


Figure 8. The process tree PT_2 of PN_2

Algorithm 6 calculates the deviation position for incomplete concurrent structures. We obtain the positions of SF_T and SL_T in the optimal alignment, and collect new activities. For activities between SF_T and SL_T in the optimal alignment, if the activity is a model activity and the corresponding transition is contained in the first concurrent activity set, we determine whether all transitions of the branch headed by this transition are model activities. If the corresponding activity of head transition is a model or synchronous activity, we regard the invisible or head transition and the first and last transitions in the branch as the deviation position.

An optimal alignment γ_3 between σ_3 and PN_2 is represented in Figure 9, and an optimal alignment γ_4 between σ_4 and PN_2 is shown in Figure 10. The branch set of PN_2 is $B_S = \{\{t_1, t_3\}, \{t_2,\ t_3\}, \{t_4, t_5\}, \{t_6\}\}$. For γ_3, $(>>, b)$, $(>>, t_4)$ and $(>>, t_5)$ are model activities, and we can find that the branch $\{t_4, t_5\}$ can occur selectively with other branches by traversing the optimal alignment. For γ_4, $(>>, t_1)$,

Algorithm 5 $Calculate - FLT(PT)$

Input: A process tree denoted by PT

Output: The head-to-tail transition denoted by $[SF_T, SL_T]$, the first concurrent activity set denoted as FU_S, and the last concurrent activity set denoted as LU_S

1: $SF_T \leftarrow \phi$; $SL_T \leftarrow \phi$; $FU_S \leftarrow \phi$; $LU_S \leftarrow \phi$; $u \leftarrow \phi$; $q_1 \leftarrow \phi$; $q_2 \leftarrow \phi$;
2: **for** each $n \in PT$ **do**
3: **if** $n =$ "$\wedge$" **then**
4: $u \leftarrow |n.child|$;
5: $m \leftarrow n.child.p_1$;
6: $k \leftarrow n.child.p_u$;
7: $SF_T \leftarrow SF_T \cup^{\bullet} ({}^{\bullet}(m.child.lp))$;
8: $SL_T \leftarrow SL_T \cup ((k.child.rp)^{\bullet})^{\bullet}$;
9: **for** $(i = 1; i \leq |n.child|; i++)$ **do**
10: $g_i \leftarrow n.child.p_i$;
11: **if** $g_i.child = null$ **then**
12: $FU_S \leftarrow FU_S \cup \{g_i.child\}$;
13: $LU_S \leftarrow LU_S \cup \{g_i.child\}$;
14: **end if**
15: $q_1 \leftarrow n.child.lp$;
16: $q_2 \leftarrow n.child.rp$;
17: **if** $g_i =$ "$\times$"$||g_i =$ "$\wedge$" and $(g_i = q_1.parent)$ **then**
18: **for** each $g_i.child$ **do**
19: $FU_S \leftarrow FU_S \cup g_i.child$;
20: **end for**
21: **end if**
22: **if** $g_i =$ "$\times$"$||g_i =$ "$\wedge$" and $(g_i = q_2.parent)$ **then**
23: **for** each $g_i.child$ **do**
24: $LU_S \leftarrow LU_S \cup g_i.child$;
25: **end for**
26: **end if**
27: **if** $g_i =$ "$\rightarrow$" and $g_i.child.child$ **then**
28: $FU_S \leftarrow FU_S \cup g_i.child.lp$;
29: $LU_S \leftarrow LU_S \cup g_i.child.rp$;
30: **end if**
31: **end for**
32: **end if**
33: **end for**
34: **return** $[SF_T, SL_T]$, FU_S, LU_S

Algorithm 6 Determining deviation positions for incomplete concurrent structures

Input: A process tree denoted by PT, a Petri net denoted by $PN = (P, T; F, M)$, and the optimal alignment denoted by $\Gamma_{\sigma,PN,lc}$

Output: The deviation position denoted by DP_{CUS}, and the new activity set denoted by AL

$DP_{CUS} \leftarrow \phi$; $AL \leftarrow \phi$;
$k \leftarrow 0$; $g \leftarrow 0$; $h \leftarrow 0$;
$B_S \leftarrow Calculate - B_S(PT)$;
$[SF_T, SL_T], FU_S, LU_S \leftarrow Calculate - FLT(PT)$;
for $(i = 1; i \leq |\gamma|; i++)$ **do**
  if $\pi_2(\gamma[i]) = SF_T$ **then**
    $k \leftarrow i$;
  end if
  if $\pi_2(\gamma[i]) = SL_T$ **then**
    $g \leftarrow i$;
  end if
  if $\pi_2(\gamma[i]) =>>$ and $\pi_1(\gamma[i]) \notin \delta(PN)$ **then**
    $AL \leftarrow AL \cup \{(\pi_1(\gamma[i]), >>)\}$;
  end if
end for
for $(i = k + 1; i < g; i++)$ **do**
  if $\pi_2(\gamma[i]) \in FU_S$ and $\pi_1(\gamma[i]) =>>$ **then**
    for $(j = 1; j \leq |B_S|; j++)$ **do**
      if $\pi_2(\gamma[i]) \in B_S$ **then**
        $h \leftarrow j$;
        for $(v = 2; \pi_h(\pi_v(B_S)) \in \delta(\sigma); v++)$ **do**
          Continue;
        end for
      end if
    end for
  end if
end for
if $\pi_1(\gamma[k]) =>>$ **then**
  $DP_{CUS} \leftarrow DP_{CUS} \cup \{\tau(\pi_h(\pi_1(B_S)), (\pi_h(\pi_v(B_S)))\}$;
else
  $DP_{CUS} \leftarrow DP_{CUS} \cup \{\pi_1(\gamma[k])(\pi_h(\pi_1(B_S)), (\pi_h(\pi_v(B_S)))\}$;
end if
return DP_{CUS}, AL

$(>>, t_3)$ and $(>>, t_6)$ are model activities, and we can find branches $\{t_1, t_3\}$ and $\{t_6\}$ occur selectively with other branches. Its head-to-tail transition is $[SF_T, SL_T] = [b, c]$, its first concurrent transition is $FU_S = \{t_1, t_2, t_4, t_6\}$, and the last concurrent transition is $LU_S = \{t_3, t_5, t_6\}$. Because $\{t_4, t_5\}$ is a branch of PN_2, and the head transition b also cannot fire, so $\{\tau(t_4, t_5)\}$ is a deviation. Because $\{t_1, t_3\}$ and $\{t_6\}$ are two branches of PN_2, and the head transition b is a synchronous activity, so $\{b(t_1, t_3), b(t_6, t_6)\}$ is a deviation. So we obtain its deviation position is $DP_{CUS} = \{\tau(t_4, t_5)), b(t_1, t_3), b(t_6, t_6)\}$. Besides, $(e, >>)$ and $(f, >>)$ are log activities, the new transitions e and f need to be inserted into the model, and its new activity set is $AL = \{e, f\}$.

γ_3=

a	>>	t_1	t_3	t_6	>>	>>	e	c
a	b	t_1	t_3	t_6	t_4	t_5	>>	c

Figure 9. An optimal alignment γ_3 between σ_3 and PN_2

γ_4=

a	b	t_4	t_5	>>	>>	>>	f	c
a	b	t_4	t_5	t_1	t_3	t_6	>>	c

Figure 10. An optimal alignment γ_4 between σ_4 and PN_2

Algorithm 7 repairs models for incomplete concurrent structures via logic Petri nets according to the deviation position. For different deviation positions, we add different logic input and output transitions.

For σ_3, σ_4 and PN_2, its deviation position is denoted by DP_{CUS}=$\{\tau(t_4, t_5))$, $b(t_1, t_3)$, $b(t_6, t_6)\}$, and its new activity set is denoted by $AL = \{e, f\}$. For $\tau(t_4, t_5))$, we add an invisible transition to skip transition b, and add two arcs from the invisible transition to p and p', where $p \in^\bullet t_1$ and $p' \in^\bullet t_6$. For $b(t_1, t_3)$ and $b(t_6, t_6)$, we change the head transition b to a logic output transition. For $AL = \{e, f\}$, we add two places and new transitions e and f, and change the tail transition c to a logic input transition. The model repaired by our approach denoted by LPN_3 is represented in Figure 11.

5 EXPERIMENTAL EVALUATION

This section will compare our repair approach with Fahland's approach, Knapsack's approach and Goldratt's approach. The data is from a routine examination and a CT index examination of a hospital in Qingdao, and event logs can be accessible at: `https://pan.baidu.com/s/1Nx2vf82NYKB9TGbf8uYIFQ`. The Fahland's approach is implemented in ProM6.6, which is a process mining tool with lots of plugins and can be available from: `http://www.promtools.org/prom6/`. The Goldratt's approach and Knapsack's approach are implemented in the DOS window and edited

Algorithm 7 Repair models for incomplete concurrent structures

Input: The deviation position denoted by DP_{CUS}, a Petri net denoted by $PN = (P, T; F, M)$, the head-to-tail transition denoted by $[SF_T, SL_T]$, and the new activity set denoted by AL

Output: A logic Petri net denoted by $LPN' = (P', T'; F', I', O', M')$

1: $LPN' \leftarrow PN$;
2: **for** each $SF_T(t_i, t_j) \in DP_{CUS}$ **do**
3: $P' \leftarrow P'$;
4: $T' \leftarrow T'$;
5: $F' \leftarrow F'$;
6: $O' \leftarrow O' \cup \{O(SF_T = [\wedge\{^\bullet t_i\}] \bigotimes \{\{SF_T^\bullet\} - \{^\bullet t_i\}\}\}$;
7: $I' \leftarrow I' \cup \{I(SL_T) = [\wedge\{t_j^\bullet\}]\}$;
8: **end for**
9: **for** each $\tau(t_i, t_j) \in DP_{CUS}$ **do**
10: $P' \leftarrow P'$;
11: $T' \leftarrow T' \cup \{\tau\}$;
12: $F' \leftarrow F' \cup \{\{^\bullet SF_T\} \rightarrow \tau\} \cup \{\tau \rightarrow \{\{SF_T^\bullet\} - \{^\bullet t_i\}\}\}$;
13: **end for**
14: **for** each $t \in AL$ **do**
15: $P' \leftarrow P' \cup P_{new}$;
16: $T' \leftarrow T' \cup t$;
17: $F' \leftarrow F' \cup \{\{(^\triangle t)^\bullet\} \rightarrow t\} \cup \{t \rightarrow P_{new}\} \cup \{P_{new} \rightarrow SL_T\}$;
18: $I' \leftarrow I' \cup \{I(SL_T) = I(SL_T) \cup [\bigotimes P_{new}]\}$;
19: **end for**
20: **return** LPN

in ProM6.6. Since there are no corresponding experimental tools for mining and repairing logic Petri nets, the model repair and analysis of our repair approach use manual simulation in this paper.

5.1 Experiment Data

We take two business processes from a routine examination and a CT index examination in a hospital as examples. A hospital routine examination business process is represented in Figure 12. First, a patient makes an appointment in the hospital and pays for an appointment. Then he (or she) can get a number and wait for his (or her) order. After that, a doctor will check what the patient needs by the routine examination. There are five types of examinations and the patient can do one of them, i.e., the electrocardiogram, the abdominal ultrasound, the lung function examination, the blood glucose and lipid, and the liver and kidney examination. Finally, a doctor will diagnose and cure disease according to examinations. Figure 13 shows a hospital CT index examination business process. First, a patient goes to the information desk to consult some related problems and makes an appointment.

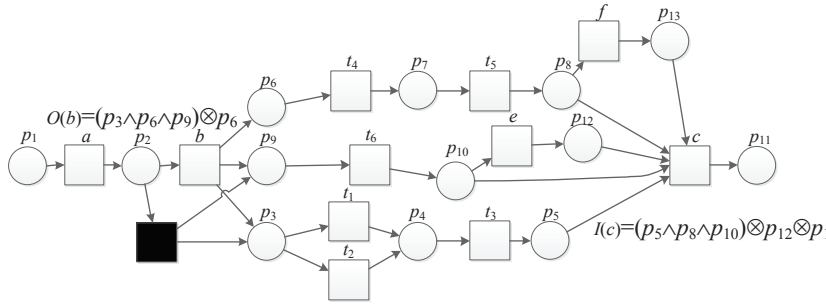


Figure 11. The repaired model LPN_3 by our approach

Then he (or she) will have an outpatient examination and take a brain CT and a head CT. Besides, a doctor checks for four symptoms, i.e., the sinusitis, the brain damage, the cerebral infarction, and the intracranial tumor. After that, the patient needs a surgery and is hospitalized for tests.

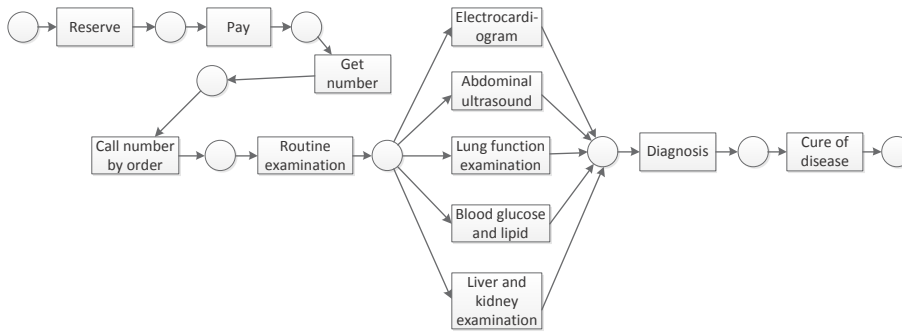


Figure 12. A hospital routine examination business process

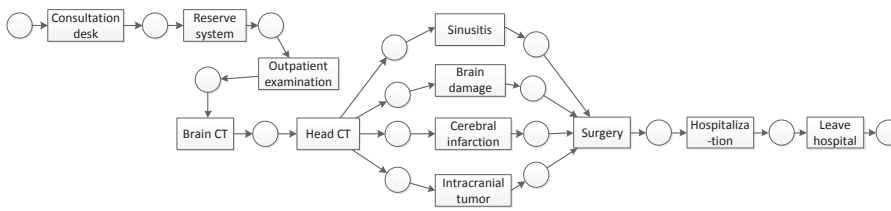


Figure 13. A hospital CT index examination business process

However, some event logs deviating from the original model are generated in the real process model systems. For example, in the hospital routine examination

business process, a patient can do four tests at once, i.e., the liver and kidney examination, the lung function examination, the blood glucose and lipid, and the electrocardiogram; the patient also can carry out the abdominal ultrasound and the gynecological examination together. In the hospital CT index examination business process, the patient may only have a brain examination, he (or she) is tested for only three conditions, i.e., the intracranial tumor, the brain damage, and the cerebral infarction; the patient also can take medicines after the sinusitis examination. In those situations, the process models need to be repaired, the choice structure needs to be repaired to an incomplete choice structure, and the concurrent structure needs to be repaired to an incomplete concurrent structure. These two repaired structures can describe both choice and concurrent structures. Petri net-based models cannot simply and accurately express those structures, and we can repair model based on logic Petri nets.

5.2 Model Repair

For event logs of a hospital routine examination business process, we first filter out event logs that are significantly deviated from the examination business process. And according to the preprocessed ten sets of event logs (as shown in Table 1), the process model (as shown in Figure 12) is repaired based on four model repair methods. Table 1 records the specific information of activities in event logs and the number of deviations.

Logs	Traces	Events	Transitions	Length	Deviations
L_1	100	1 020	13	8–11	220
L_2	200	2 103	13	8–12	503
L_3	300	3 143	13	8–12	743
L_4	400	4 306	13	8–12	1 106
L_5	500	5 447	13	8–12	1 447
L_6	600	6 129	13	8–11	1 378
L_7	700	7 169	13	8–11	1 570
L_8	800	8 179	13	8–11	1 780
L_9	900	9 169	13	8–11	1 970
L_{10}	10 00	10167	13	8–11	2 168

Table 1. Event logs L_1–L_{10} of a routine examination business process

Our proposed approach is compared with Fahland's approach, Knapsack's approach and Goldratt's approach to illustrate its correctness and effectiveness. For incomplete choice structures, the Fahland's approach repairs the choice structure by adding loop structures, and adds invisible transitions to skip transitions that are not enabled. The Goldratt's method and Knapsack's method add different self-loops of repeat transitions to improve the fitness of process models. For incomplete concurrent structures, the Fahland's approach collects new transitions as a sub-log and inserts it into the original model, and skips transitions that are not

enabled by adding invisible transitions. Besides, the Goldratt's method and Knapsack's method also add many invisible transitions to make repaired models better replay event logs generated in real systems. These invisible transitions and repeat transitions increase the uncertainty of the operation of the model, which leads to a poor performance of the model. These models cannot represent the logic relation among activities in incomplete choice structures and incomplete concurrent structures well, and most of them deviate from original structures of process models. It does not take advantage of the application and extension of process models.

For Figure 12 and Table 1, models repaired by four approaches are represented in Figures 14, 15, 16 and 17, respectively. The model repaired by Fahland's method adds a loop structure to make transitions in the choice structure fire simultaneously. The models repaired by Goldratt's approach and Knapsack's approach add different self-loops of single transitions. Our approach does not add any repeat transitions and does not add loop structures to change the basic structure of the original model. The model repaired by our method contains 3 logic transitions, the logic input function is $I(Diagnosis) = p_{10} \bigotimes p_{11} \bigotimes p_{13} \bigotimes p_{14} \bigotimes (p_{10} \wedge p_{11} \wedge p_{13} \wedge p_{14})$, and the logic output functions are $O(Routine\ examination) = p_6 \bigotimes p_7 \bigotimes p_8 \bigotimes p_9 \bigotimes (p_6 \wedge p_7 \wedge p_8 \wedge p_9)$ and $O(Abdominal\ ultrasound) = p_{12} \bigotimes p_{13}$. Our repaired model can describe the logic relation among transitions of incomplete choice structures. These input and output functions limit the enablement of transitions in the model, so the model generates fewer traces that are not included in event logs.

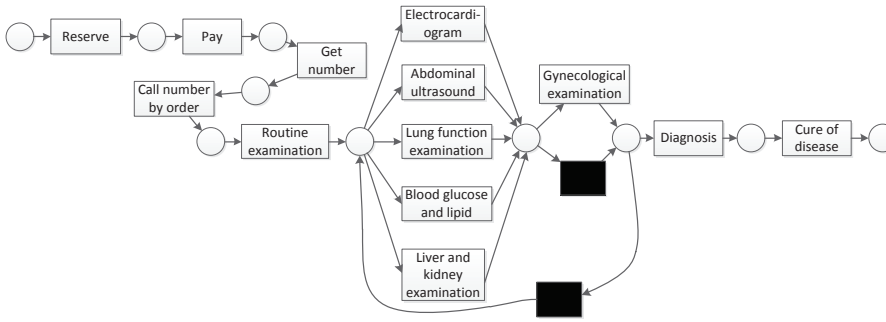


Figure 14. The repaired routine examination model by Fahland's method

Compared with the original model, the addition results of four repair models are represented in Table 2. As shown in Table 2, Goldratt's method adds the largest number of transitions (including invisible transitions), and our method only adds one transition without invisible transitions. The number of added repeat transitions by Goldratt's method and Knapsack's method are 6 and 4, respectively. Compared with other three repair methods, our method adds the least transitions, and does not add invisible transitions and repeat transitions.

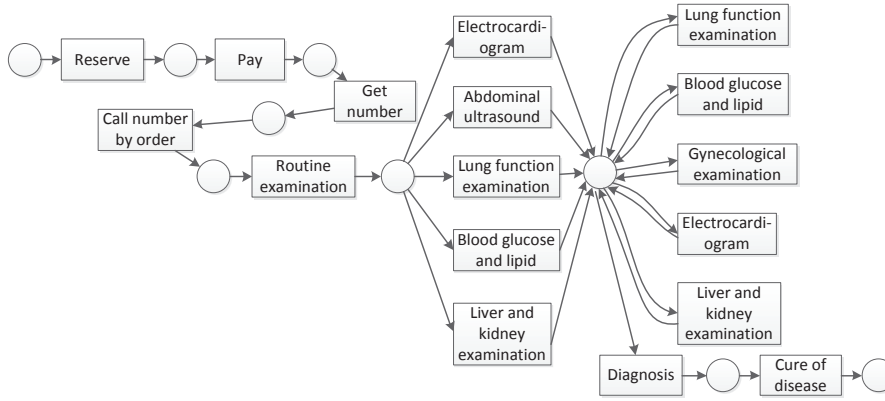


Figure 15. The repaired routine examination model by Knapsack's method

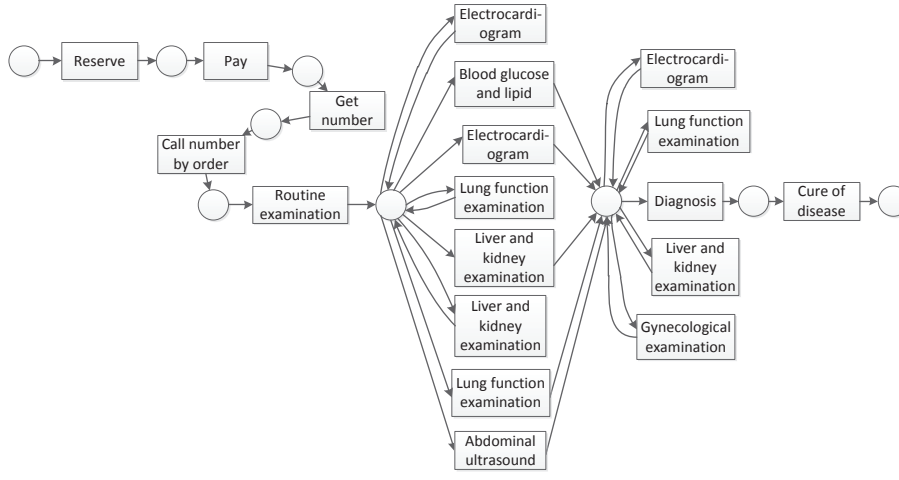


Figure 16. The repaired routine examination model by Goldratt's method

Four Repair Methods	Added $\|P\|$	Added $\|T+\tau\|$	Added $\|F\|$	Added Repeat $\|T\|$
Our method	7	1	9	0
Fahland's method	1	3	6	0
Goldratt's method	0	7	14	6
Knapsack's method	0	5	10	4

Table 2. The addition results of Figures 14, 15, 16 and 17

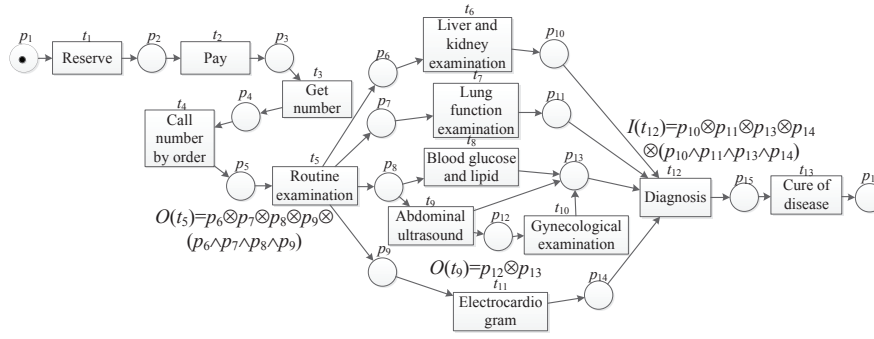


Figure 17. The repaired routine examination model by our approach

We process event logs of a CT index examination business process according to the same data filtering method. The filtered ten sets of event logs are described in Table 3. The process model (as shown in Figure 13) is repaired based on ten sets of event logs. The information of activities in event logs and the number of deviations are recorded in Table 3.

Logs	Traces	Events	Transitions	Length	Deviations
L_{11}	100	1 166	13	10–12	38
L_{12}	200	2 366	13	9–12	38
L_{13}	300	3 366	13	9–12	238
L_{14}	400	4 566	13	9–12	238
L_{15}	500	5 765	13	9–12	238
L_{16}	600	6 965	13	9–12	238
L_{17}	700	8 165	13	9–12	238
L_{18}	800	9 100	13	9–12	573
L_{19}	900	10 300	13	9–12	573
L_{20}	1 000	11 500	13	9–12	573

Table 3. Event logs $L_{11} - L_{20}$ of a CT index examination business process

Four repaired models repaired by three classic methods and our repair method are shown in Figure 18, 19, 20 and 21, respectively. Fahland's method inserts many invisible transitions into the model to skip concurrent transitions that cannot fire. Although it makes activities in event logs be well replayed in the model, it also reduces the precision and simplicity of the model. Goldratt's approach inserts activities deviating from the model into the original model by means of self-loops. Knapsack's approach repairs the model in the same way as Goldratt's approach with different constraints. Besides, these two methods add a lot of invisible transitions. The model repaired by our approach reduces the degree of uncertainty of transition firing and has a high simplicity of net structures. Besides, it contains one logic input function and one logic output function, and they are

$I(Surgery) = (p_{10} \wedge p_{11} \wedge p_{12} \wedge p_{13}) \bigotimes p_{14} \bigotimes (p_{10} \wedge p_{11} \wedge p_{12})$ and $O(Head\ CT) = (p_6 \wedge p_7 \wedge p_8 \wedge p_9) \bigotimes p_9$. The repaired model by our approach can describe the logic relation among transitions of incomplete concurrent structures, and it does not generate additional transitions to increase the uncertainty of the operation of the model.

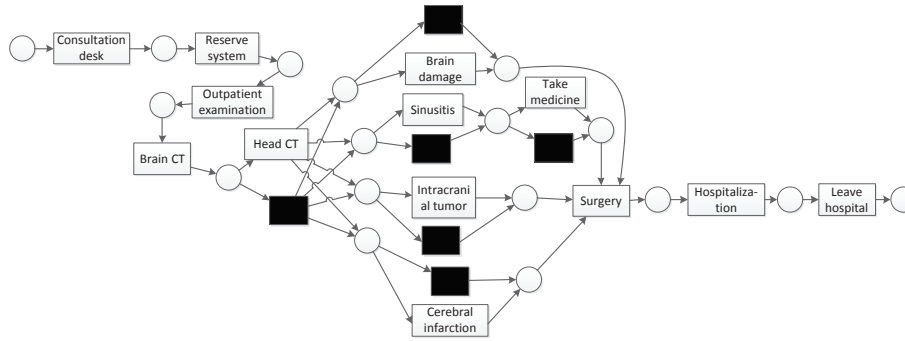


Figure 18. The repaired CT index examination model by Fahland's method

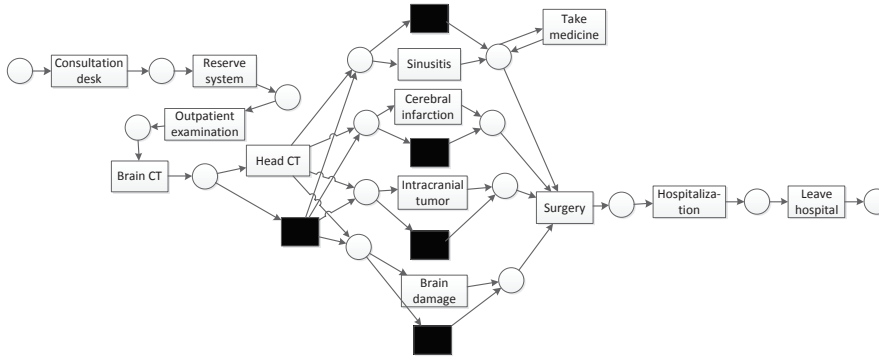


Figure 19. The repaired CT index examination model by Knapsack's method

Table 4 records addition results of four repair methods. Fahland's method adds the maximum number of transitions, including invisible transitions. The addition result of our repaired model is best, and the other three repaired models add a lot of invisible transitions and repeat transitions.

5.3 Performance Analysis

The sub-section describes the performance analysis of four repaired models combining with event logs L_1–L_{20}. Fitness is the proportion of traces that the model

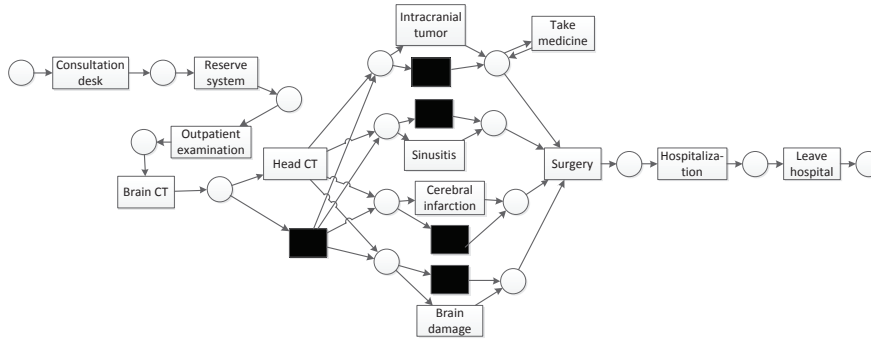


Figure 20. The repaired CT index examination model by Goldratt's method

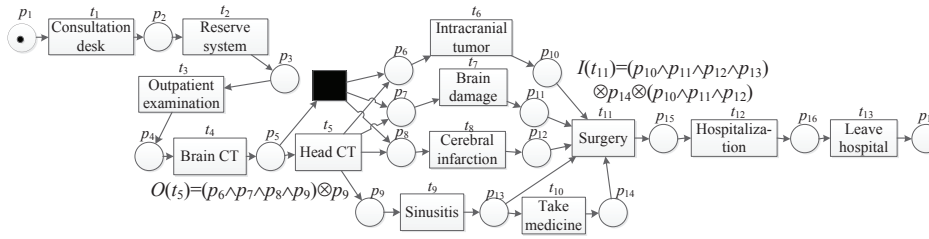


Figure 21. The repaired CT index examination model by our approach

can completely replay in the event log. A model with high fitness allows behaviors described in the event log to occur. If a model is precise, it does not allow too many activities that are not described in event logs to appear in the model. The formula for calculating fitness and precision of logic Petri nets are proposed in [19]. We consider the structure of the model and the repeatability of activities, and calculate the simplicity of net-based structures according to the method proposed in [18].

For four repaired models, we obtain the degree of fitness between models and event logs L_1–L_{20} represented in Table 5 and Table 6. As shown in Table 5, for our approach, Fahland's approach and Goldratt's approach, the fitness be-

Four Repair Methods	Added $\|P\|$	Added $\|T+\tau\|$	Added $\|F\|$	Added Repeat $\|T\|$
Our method	1	2	7	0
Fahland's method	1	7	17	0
Goldratt's method	0	6	15	1
Knapsack's method	0	6	15	1

Table 4. The addition results of Figures 18, 19, 20 and 21

tween these repaired models and event logs L_1–L_{10} are all 1. However, the fitness value of the model repaired by Goldratt's method is slightly lower than that of the other methods. As shown in Table 6, the degree of fitness between these four repaired models and event logs L_{11}–L_{20} are all 1. In general, the fitness of these four methods is very high, and those four repaired models all have a high fitness.

Four Repair Methods	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
Fahland's method	1	1	1	1	1	1	1	1	1	1
Our method	1	1	1	1	1	1	1	1	1	1
Goldratt's method	1	1	1	1	1	1	1	1	1	1
Knapsack's method	1	0.9884	0.9884	0.9723	0.9710	1	1	1	1	1

Table 5. The fitness between different models and event logs L_1–L_{10}

Four Repair Methods	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
Fahland's method	1	1	1	1	1	1	1	1	1	1
Our method	1	1	1	1	1	1	1	1	1	1
Goldratt's method	1	1	1	1	1	1	1	1	1	1
Knapsack's method	1	1	1	1	1	1	1	1	1	1

Table 6. The fitness between different models and event logs L_{11}–L_{20}

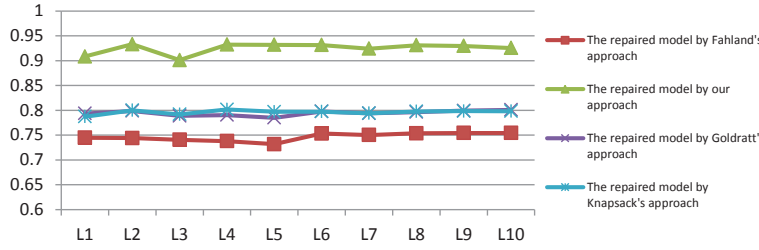


Figure 22. The precision between different models and event logs L_1–L_{10}

The results of precision between models proposed by four repair methods and event logs L_1–L_{20} are represented in Figure 22 and Figure 23. As shown in Figure 22, the model repaired by our approach has the highest precision, significantly higher than Knapsack's approach, Goldratt's approach and Fahland's approach. As shown in Figure 23, the model repaired by our approach also has the highest precision, and it has a clear performance advantage than the other three repair methods. In general, the precision of our repaired model for incomplete choice structures or incomplete concurrent structures is significantly higher than that of the three other classic repair methods. Our proposed method greatly improves the

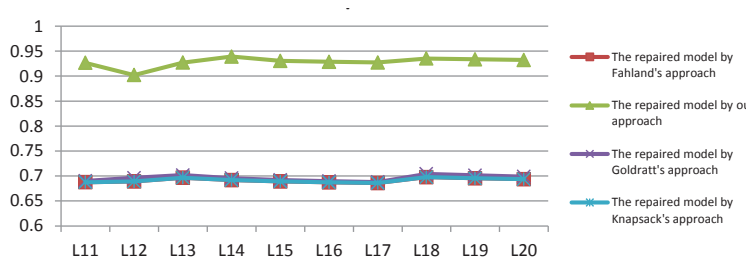


Figure 23. The precision between different models and event logs L_{11}–L_{20}

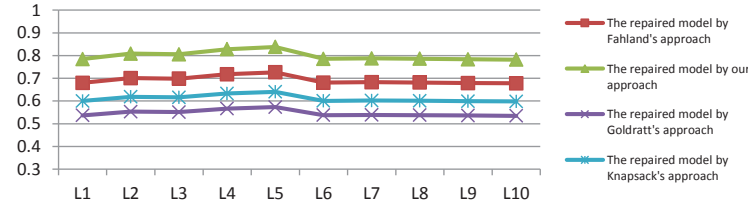


Figure 24. The simplicity between different models and event logs L_1–L_{10}

precision of the process models and also improves the performance of the models.

The simplicity of net structures is obtained by calculating the sum of the proportion of events of per trace in the total number of transitions of models. The results of simplicity of four repair approach combining with event logs L_1–L_{20} are represented in Figure 24 and Figure 25. As shown in Figure 24, the simplicity of the repaired model by our approach is highest, followed by Fahland's approach. These two methods are higher than Knapsack's approach and Goldratt's approach. As shown in Figure 25, our repair approach has the highest simplicity of net structures, higher than Goldratt's approach and Knapsack's approach. By comparison, the repaired model by Fahland's approach has the lowest simplicity of net structures. In terms of net structures, our repair approach has the highest simplicity of net structures, higher than Goldratt's approach, Knapsack's approach

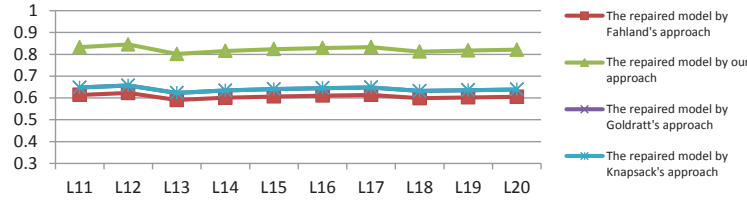


Figure 25. The simplicity between different models and event logs L_{11}–L_{20}

and Fahland's approach. Therefore, for business processes with incomplete concurrent and choice structures, our method not only enables the model to have a high precision and fitness, but also improves the simplicity of net structures of the model.

6 CONCLUSIONS

In the paper, the repair method for process models with incomplete choice and concurrent structures is proposed via logic Petri nets. Current model repair methods tend to change the choice structure to a loop structure, and generate a large number of invisible and repeat transitions, or add self-loops of transitions to improve fitness. The concepts of choice relation sets and concurrent relation sets are presented based on process trees. For incomplete choice structures, we judge whether the corresponding transitions of log activities are contained in a whole branch, and determine deviation positions based on choice activity sets. For incomplete concurrent structures, we find deviations by judging whether the corresponding transitions of model activities are contained in a whole branch and whether the head-to-tail transition is a synchronous activity. Then we determine deviation positions based on concurrent activity sets, and repair models via logic Petri nets according to different deviation positions. Through the simulation experiment, we prove that our repair approach greatly improves the precision and simplicity of the model, and the fitness is still very high. It also can describe the logic relation among transitions in incomplete choice and concurrent structures correctly. However, process trees can only represent Petri nets with block-distributed structures. For those process models that cannot be represented by process trees, we will further study how to determine the deviation position through other algorithms.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61903229 and Grant 61973180, in part by the Key Research and Development Program of Shandong Province under Grant 2018GGX101011, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2018MF001 and Grant ZR2019BF004.

REFERENCES

[1] Qi, H. D.—Du, Y. Y.—Liu, W.: Process Model Repairing Method Based on Reachable Markings. Journal of Shandong University of Science and Technology (Natural Science), Vol. 36, 2017, No. 1, pp. 118–124.

[2] Conforti, R.—Dumas, M.—García-Bañuelos, L.—La Rosa, M.: BPMN Miner: Automated discovery of BPMN Process Models with Hierarchical Structure. Information Systems, Vol. 56, 2016, pp. 284–303, doi: 10.1016/j.is.2015.07.004.

[3] van der Aalst, W. M. P.—ter Hofstede, A. H. M.: YAWL: Yet Another Workflow Language. Information Systems, Vol. 30, 2004, No. 4, pp. 245–275, doi: 10.1016/j.is.2004.02.002.

[4] Liu, G. J.: Complexity of the Deadlock Problem for Petri Nets Modeling Resource Allocation Systems. Information Sciences: An International Journal, Vol. 363, 2016, Iss. C, pp. 190–197, doi: 10.1016/j.ins.2015.11.025.

[5] Wang, Y. Y.—Du, Y. Y.: Conformance Checking Based on Extended Footprint Matrix. Journal of Shandong University of Science and Technology (Natural Science), Vol. 37, 2018, No. 2, pp. 9–15.

[6] van der Aalst, W. M. P.—Weijters, A. J. M. M.—Maruster, L.: Workflow Mining: Discovering Process Model from Event Logs. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, 2004, No. 9, pp. 1128–1142, doi: 10.1109/TKDE.2004.47.

[7] Weijters, A. J. M. M.—van der Aalst, W. M. P.: Rediscovering Workflow Models from Event-Based Data Using Little Thumb. Integrated Computer-Aided Engineering, Vol. 10, 2003, No. 2, pp. 151–162.

[8] Adriansyah, A.—Munoz-Gama, J.—Carmona, J.—van Dongen, B. F.—van der Aalst, W. M. P.: Aligning Based Precision Checking. In: La Rosa, M., Soffer, P. (Eds.): Business Process Management Workshops (BPM 2012). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 132, pp. 137–149, doi: 10.1007/978-3-642-36285-9_15.

[9] Leemans, S. J. J.—Fahland, D.—van der Aalst, W. M. P.: Scalable Process Discovery and Conformance Checking. Software and Systems Modeling, Vol. 17, 2018, pp. 599–631, doi: 10.1007/s10270-016-0545-x.

[10] Alizadeh, M.—Lu, X.—Fahland, D.—Yanonne, N.—van der Aalst, W. M. P.: Linking Data and Process Perspectives for Conformance Analysis. Computers and Security, Vol. 73, 2018, pp. 172–193, doi: 10.1016/j.cose.2017.10.010.

[11] Rozinat, A.—van der Aalst, W. M. P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems, Vol. 33, 2008, No. 1, pp. 64–95, doi: 10.1016/j.is.2007.07.001.

[12] Buijs, J. C. A. M.—van Dongen, B. F.—van der Aalst, W. M. P.: A Genetic Algorithm for Discovering Process Trees. IEEE Congress on Evolutionary Computation, 2012, 8 pp., doi: 10.1109/CEC.2012.6256458.

[13] Fahland, D.—van der Aalst, W. M. P.: Model Repair – Aligning Process Models to Reality. Information Systems, Vol. 47, 2015, pp. 220–243, doi: 10.1016/j.is.2013.12.007.

[14] Polyvyanyy, A.—van der Aalst, W. M. P.—ter Hofstede, A. H. M.—Wynn, M. T.: Impact-Driven Process Model Repair. ACM Transactions on Software Engineering and Methodology, Vol. 25, 2016, No. 4, pp. 25–33, doi: 10.1145/2980764.

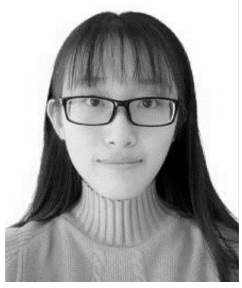
[15] Qi, H. D.—Du, Y. Y.—Qi, L.—Wang, L.: An Approach to Repair Petri Net-Based Process Models with Choice Structures. Enterprise Information Systems, Vol. 12, 2018, No. 8-9, pp. 1149–1179, doi: 10.1080/17517575.2018.1432768.

[16] Du, Y. Y.—Qi, L.—Zhou, M. C.: Analysis and Application of Logical Petri Nets to E-Commerce Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 44, 2014, No. 4, pp. 468–481, doi: 10.1109/TSMC.2013.2277696.

[17] Zhang, X. Z.—Du, Y. Y.—Qi, L.—Sun, H. C.: An Approach for Repairing Process Models Based on Logic Petri Nets. IEEE Access, Vol. 6, 2018, pp. 29926–29939, doi: 10.1109/ACCESS.2018.2843137.

[18] Teng, Y. X.—Du, Y. Y.—Qi, L.—Luan, W. J.: A Logic Petri Net-Based Method for Repairing Process Models with Concurrent Blocks. IEEE Access, Vol. 7, 2019, pp. 8266–8282, doi: 10.1109/ACCESS.2018.2890070.

[19] Adriansyah, A.: Aligning Observed and Modeled Behavior. Ph.D. Thesis, Technische Universiteit Eindhoven, 2014, pp. 139–149, doi: 10.6100/IR770080.



Yuanxiu Teng received her B.Sc. degree from Shandong University of Science and Technology, Qingdao, China, in 2017. She is now pursuing the M.Sc. degree in the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China. Her current research interests are process mining, Petri nets and workflow.

Liang Qi received his B.Sc. degree in information and computer science and M.Sc. degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2017. He is currently Lecturer of computer science and technology at Shandong University of Science and Technology, Qingdao, China. His current research interests include Petri nets, discrete event systems, process mining and optimization algorithms.

Yuyue Du received his B.Sc. degree from Shandong University, Jinan, China, in 1982, the M.Sc. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991, and the Ph.D. degree in computer application from Tongji University, Shanghai, China, in 2003. He is currently Professor at the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China. His research interests are in formal engineering, Petri nets, real-time systems, process mining and workflows.

Computing and Informatics, Vol. 39, 2020, 298–317, doi: 10.31577/cai_2020_1-2_298

TRAVEL MODE RECOGNITION FROM GPS DATA BASED ON LSTM

Shaowu Zhu, Haichun Sun*, Yongcheng Duan, Xiang Dai

College of Police Information Technology and Cyber Security
People's Public Security University of China
Beijing, China
e-mail: shaowuzhu@163.com, sunhaichun@ppsuc.edu.cn,
{443130851, 991836225}@qq.com

Sangeet Saha

School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
e-mail: sangeet.saha@essex.ac.uk

Abstract. A large amount of GPS data contains valuable hidden information. With GPS trajectory data, a Long Short-Term Memory model (LSTM) is used to identify passengers' travel modes, i.e., walking, riding buses, or driving cars. Moreover, the Quantum Genetic Algorithm (QGA) is used to optimize the LSTM model parameters, and the optimized model is used to identify the travel mode. Compared with the state-of-the-art studies, the contributions are: 1. We designed a method of data processing. We process the GPS data by pixelating, get grayscale images, and import them into the LSTM model. Finally, we use the QGA to optimize four parameters of the model, including the number of neurons and the number of hidden layers, the learning rate, and the number of iterations. LSTM is used as the classification method where QGA is adopted to optimize the parameters of the model. 2. Experimental results show that the proposed approach has higher accuracy than BP Neural Network, Random Forest and Convolutional Neural Networks (CNN), and the QGA parameter optimization method can further improve the recognition accuracy.

* Corresponding author

Keywords: GPS, LSTM, QGA, deep learning, travel mode

Mathematics Subject Classification 2010: 68-T10

1 INTRODUCTION

With the increasing usage of mobile communication devices, more and more personal location data are generated by GPS. These data could turn out to be useful information if utilized properly. Various types of applications are based on location data analysis. With the aid of GPS data, we can identify the individual's travel mode. It has great advantages, for example, we can display resident traveling information, guide people to choose a better travel mode, and help urban traffic planning and management. It thus can bring many benefits such as decreasing traffic congestion and environmental pollution. Along with this, through accurate recognition of the travel mode, we can analyze individual travel characteristics to improve urban transport efficiency. It may also be useful in recommending relevant services. Such as launching customized advertisements and launching sneaker advertisements for pedestrians.

Travel mode recognition has been studied for many years. From the beginning of paper-and-pencil interviews to computer telephone interviews and computer-assisted self-interviews, the way to collect information on residents' travel mode cannot achieve highly accurate results [1]. These methods are complicated.

In comparison, the information obtained utilizing GPS trajectory data is more comprehensive, detailed and accurate. There exist some researches on this topic. Dabiri et al. [2], Liang et al. [3] used GPS data to recognize the travel mode. However, such methods to recognize the travel mode using GPS trajectory data need to be improved to increase accuracy. People can simply use some rules to judge the way of travel, such as speed, directions, etc., however, the obtained results are not very effective. The same speed may correspond to different modes of transportation in different scenarios. Therefore, the extraction of more features is crucial when judging the travel mode. In the feature extraction process, it is also necessary to consider the redundancy of features and the complex relationship among the features [4]. Moreover, different data samples, the method for cutting the GPS trajectory, and classification models will affect the accuracy of the mode detection.

The use of GPS trajectory data and other sensors to infer traffic modes is constantly evolving, and many classification algorithms have been used in previous studies. On the one hand, individual GPS historical data is small, and the conventional machine and deep learning algorithm rely heavily on the amount of data. LSTM has a good effect on the timing prediction class. Thus, we have employed the LSTM as the GPS travel pattern recognition method. In this paper, the accuracy of the LSTM model is compared with the existing algorithms. The results show that the accuracy of the LSTM model is higher than that of the random forest and BP

neural network. Moreover, the QGA is used to optimize the LSTM model parameters, and the optimized model is used to identify the travel mode. The results reveal that the model recognition accuracy is higher after using QGA optimization.

The rest of the paper is organized as follows. Section 2 presents the literature review. Section 3 depicts the design of the algorithm in detail. Section 4 designs the preprocessing method and QGA. Section 5 evaluates the effectiveness of the proposed approach by comparing their results with classical machine learning algorithms and QGA. Finally, Section 6 concludes this work and discusses the future direction.

2 LITERATURE REVIEW

In recent years, research on the use of GPS trajectory data and other sensor data to infer travel modes has become more and more popular. In order to study different travel mode recognition, different classification algorithms have been employed and these techniques have a great influence on the accuracy of the final recognition result. Typical travel pattern recognition algorithms include neural networks, random forests, support vector machines. Specifically, Zheng et al. [5] evaluated four classification algorithms, i.e., decision tree, Bayesian network, support vector machine (SVM) and conditional random field (CRF), and employed a segmentation method to cut GPS trajectory data. The classification algorithm shows that the results based on decision trees are more accurate. Zhu et al. [6] designed the trajectory segmentation algorithm using a logic hypothesis, through random forest classifier on the classification method they achieved the final recognition degree of 82.85 %. Stenneth et al. [7] proposed a method combining the GPS sensor with the basic traffic network knowledge for the confusion of the past inferred motor vehicle model. The random forest algorithm was used as the main model, which improves detection accuracy by 17 %. Guvensan et al. [8] propose a novel post-processing algorithm, healing algorithm, to correct the classification error generated by the machine model algorithm. The segment-based treatment algorithm improves the average accuracy of travel mode detection by 11.7 %. Reddy et al. [9] use a classification system combining decision trees with first-order discrete hidden Markov models and use correlation feature-based feature selection (CFS) feature sets to eliminate irrelevant and redundant attributes. The final accuracy goes as high up to 93.6 %. Bolbol et al. [10] use a framework-based reasoning model based on support vector machine (SVM) classification and tests with coarse-grained GPS data, resulting in an accuracy of 88 %. Brunauer et al. [11] propose a fully data-driven classification method using a feedforward multilayer perceptron (MLP) to select the most beneficial feature subsets through evolutionary features and compare them with a logical model tree and a C4.5 tree. The achieved overall accuracy is 92.24 %, which is higher than the other two methods, which are 92.09 % and 84.48 %, respectively. Xia et al. [12] conduct an in-depth analysis of the stationary state during the travel mode, and divide the speed into zero and the pause and wait for modes. The significance of the pause during transportation is evaluated. The support vec-

tor machine (SVM) and the ant colony optimization are used to reduce the feature dimension, thus achieving 96.31 % detection accuracy. However, Zong et al. [13] use the support vector (SVC) classification model, with the genetic algorithm (GA) for optimization purposes in the SVC model and the accuracy reaches 92.2 %. Liang et al. [14] use CNN for identification and add some filtering algorithms to smooth the data (can reduce the fluctuation of data), and its highest recognition accuracy is about 94 %. Xiao et al. [4] use the Bayesian network and the K2 algorithm, and the resulting walking recognition rate is over 97 %. Besides, the researchers have proposed many new recognition algorithms. Martin et al. [15] develop a new classification algorithm and compare it with k-Nearest Neighbor (KNN) and random forest. Experiments show that the recognition accuracy combined with the random forest algorithm is 94 %. Zhu et al. [16] propose a travel mode selection model and a directed graph-guided fusion Lasso method, which reduces the time complexity of the algorithm.

Compared with other algorithms, the LSTM recognition algorithm combines the input and output states of the previous data, so it is more sensitive to time series and it also has a memory function for long-term data. Hence, it can produce good effects on travel mode recognition. Dai et al. [18] introduce shortcut connections between the inputs and the outputs of two consecutive LSTM layers to handle gradient vanishment. The result shows that the new model has a higher trajectory predicting accuracy. Yuan et al. [17] proposed a supervised LSTM (SLSTM) network to learn quality-relevant hidden dynamics for soft sensor applications, which are more relevant and useful for quality prediction. Using the LSTM model, the GPS data is analyzed and processed, the relevant parameters of the LSTM model are optimized by QGA, and the travel mode of the GPS data is finally identified.

3 OVERVIEW OF LSTM

The LSTM is obtained by improving the recurrent neural network (RNN). The RNN is difficult to deal with long-distance dependence, and gradient disappearance or gradient explosion are prone to occur. The LSTM introduces the cell state and adds the input and output of the previous time to the current time processing through the gating unit and linear connection. Hence, it can deal with the long-distance dependency problem.

3.1 Forward Computing

In forward computing, the LSTM saves the long-term memory of the model in the cell state. The current information, the cell state, and the output of the previous moment are the input. It controls the input information of different gated units and gets output. The flowchart is shown in Figure 1.

By adding a cell state for saving the long-term state of input data, the problem that the RNN hidden layer is sensitive to short-term input is solved. The key of the

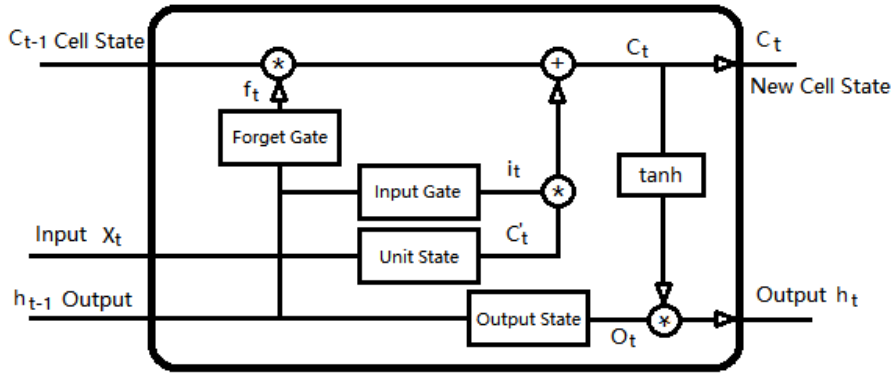


Figure 1. Forward computing structure diagram

LSTM is to control the long-term state. Here, the cell state is controlled by four control switches: input gate, output gate, forgetting gate, and unit state.

$$A * B = (A * B)_{ij} = a_{ij} b_{ij}. \tag{1}$$

Formula (1) represents the Hadamard product of A and B, which multiplies the corresponding elements in the matrix to form a new matrix. Formulas (2), (3), (4), (5), (6) and (7) describe each process of the LSTM forward computing.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{2}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{3}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{4}$$

$$C'_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * C'_t, \tag{6}$$

$$h_t = o_t * \tanh(C_t). \tag{7}$$

The x_t represents the processed data of the input, and after combining with the output h_{t-1}, the gate state f_t is formed by the function to control the output at the previous moment.

σ denotes a sigmoid function in (8), and tanh denotes hyperbolic tangent function in (9).

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{8}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{9}$$

3.2 Back-Propagation Algorithm

The LSTM uses the back-propagation algorithm to calculate the weight gradient of each cell through the error term and uses the gradient descent method to get the training model.

The error is propagated back in time to control the weight matrix of the gating unit at each time and adjust the model.

$$\delta_k^T = \prod_{j=k}^{t-1} \left(\delta_{oj}^T W_{oh} + \delta_{fj}^T W_{fh} + \delta_{ij}^T W_{ih} + \delta_{c'j}^T W_{ch}\right). \tag{10}$$

Formula (10) shows how the error term is transmitted forward to any time k. Among them, δ is the error term of different elements at different times, and W is the weight matrix of different elements at different times.

4 OPTIMIZATION OF LSTM PARAMETERS USING QGA

In order to evaluate the travel mode form the GPS data, firstly, a segment of GPS data is processed through filtering, eliminating dirty data. Secondly, the model is constructed by calculating the relevant features based on pure data. Then, the parameters of the LSTM model are optimized. Finally, the data are trained and identified to obtain the accuracy of model recognition. The overall framework of the whole method is shown in Figure 2.

At each point of the data, we adopt three characteristics, including the stopping point, velocity, and acceleration. Each data point contains these three characteristics. Each sample contains 300 data, and each sample label is marked as a travel mode.

In the output part, we select four modes, including walk, bike, bus, and car. So the output dimension is set as $(4, 1)$, i.e., the number of neurons in the output layer is 4.

In the LSTM model, input data is dimensionally reduced to form $(n, 300)$ data groups, where n is the number of travel records, and each travel record corresponds to a motion mode label. During each training, a set of data of batch_size parameter number in the LSTM model is read, and a set of current time output is calculated by using the weight coefficient of the current time. The error between the current

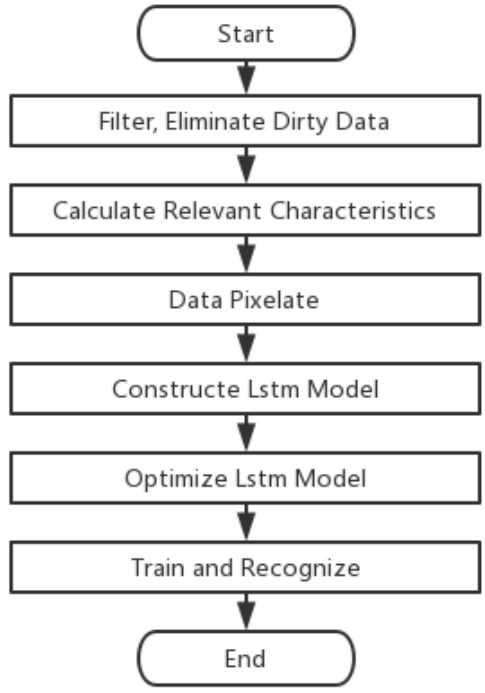


Figure 2. Framework for travel mode research and judgment

time output and the actual label is compared, and the error items are transmitted back to the end in time order. Finally, according to the error items of each time, the weight coefficients of the current time are updated to form the corresponding recognition model.

4.1 Data Process

4.1.1 Gaussian Filtering

Huang et al. [19], Sun [20], Liang [14] have used filtering methods to smooth the data. Because of the mechanical error, all the GPS positioning data cannot locate the collected position accurately. Thus the collected data will fluctuate in a certain range and produce unnecessary noise, and thus needs to be smoothed by employing Gaussian filtering algorithm, data are processed and the white noise of the data is filtered out.

$$F_i' = (F_{i-1} + F_i + F_{i+1})/3 \tag{11}$$

where F_i' is the filtered longitude and latitude of point i, F_{i-1} is the original GPS data longitude and latitude of point $i-1$, F_i is the original GPS data longitude and

latitude of i, F_{i+1} is the original GPS data longitude and latitude of $i+1$. Replacing the original value with the average value of a point and its adjacent points can reduce the sudden change of data, makes the trajectory smoother, more continuous and more realistic.

4.1.2 Characteristic

The original data provides the longitude, latitude, time, height difference and other information of the acquisition point. We use Geo Life project data [21, 22, 23] and introduce it in Section 5. In this paper, the stopping point, velocity, and acceleration parameters are selected as the experimental characteristics. Experiments show that the longitude and latitude data hamper model recognition as the absolute position of longitude and latitude is not closely related to the motion pattern and thus the recognition accuracy will be reduced in recognition. The elevation data are omitted because of the small variation and large error of the elevation measured in the experiment.

When the speed is below the threshold for a while, the stopping point is determined. The stopping point is according to the movement of motor vehicles under the condition of urban roads, so it is more useful to distinguish between motor vehicles and non-motor vehicles.

$$S_t = \left[\operatorname{sgn} \left(10 - \sum_{i=1}^{10} v_{t-i} \right) + 1 \right] /2. \tag{12}$$

Formula (12) shows the calculation of the stopping point, which means that when the velocity is less than 1 m/s in the first ten seconds of time t, the stopping point is denoted as 1, and 0 otherwise.

However, it is meaningful to select acceleration as the input variable of the LSTM. Different vehicles have different characteristics of acceleration and deacceleration, and travel pattern recognition can be realized by using acceleration. Traditional GPS devices do not provide velocity V_t and acceleration A_t, so they can acquire the acquired GPS data by post-processing.

$$V_t = \frac{\text{Vincenty}(P_{t-1}, P_t)}{\Delta t}. \tag{13}$$

In (13), we use Vincenty Formula [24] to process GPS data and obtain the distance between two points on the sphere. P_{t-1} represents the longitude and latitude of $t-1$ time, and P_t represents the longitude and latitude of t time.

The main travel modes of bus, car, bike, and walk are selected as the output of the LSTM.

4.1.3 Normalization and Data Pixelate

Through normalization, the input characteristics are changed to decimal numbers between 0 and 1, which makes data processing more convenient and faster. And

in sigmoid function, the output tends to be zero. In the calculation of gradient descent, it can improve the convergence speed of the model and make the training identification process faster.

For each data, it needs to be pixelated and the float data normalized by one characteristic becomes eight-bit data from 0 to 255. The set of data represents eight pixels, that is, the data dimension is expanded to $(100, 24)$. Specifically, all the data are normalized, and then each data are encoded with eight-bit binary, and finally form eight pixels data. In general, the input is three characteristics, the stopping point, velocity, and acceleration. In the grayscale image, there are 24 columns. One to eight columns are the stopping point, nine to sixteen columns are the velocity, seventeen to twenty-four columns are the acceleration. And every row is a one-time point, every one hundred rows is a group.

Each integer type data represents one pixel, and the value of 0–255 represents the gray value of each pixel. Each sample contains 2 400 pixels to form a grayscale image, which is input into the model. The grayscale image is shown in Figure 3. Pixelization of the data is effective, makes the model easier to recognize, and enhances the interpretability of the model.

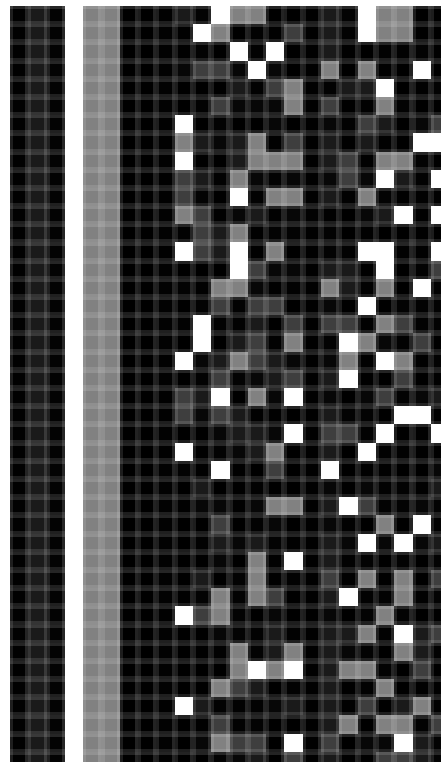


Figure 3. Grayscale images sample

4.2 Model Optimization Using the QGA

The QGA uses the qubit encoding method and uses the probability amplitude to represent the superposition states of 0 and 1. QGA imitates the situation of cross-variation of chromosomes in genetics, combines individuals with diversity, selects the best individuals of the population in the generational inheritance, and also generates the optimal solution of the parameters sought.

QGA [25] is an intelligent optimization algorithm combining the quantum algorithm and genetic algorithm. It solves the problem when traditional genetic algorithms fall into a locally optimal solution to a certain extent and have higher global search ability and convergence speed [26]. QGA replaces the binary coding method of chromosomes in traditional genetic algorithms by qubit and quantum superposition hence it increases the range of chromosome values and replaces the original chromosome crossover method by quantum full interference crossover method. It also improves chromosomes by the quantum revolving algorithm. The mutation method ensures the convergence of the algorithm.

Here, we optimize the number of neurons, the number of hidden layers, the learning rate, and the number of iterations. So the chromosome number is 4. In the experiment, the QGA parameters are initialized by combining the common parameter setting experience. The chromosome length is 20.

And in the QGA experiment, the iterations number is 100. As Figure 4 shows, with the addition of the iterations number, the accuracy increases. When the number is close to 100, there is hard to find more accurate parameters.

The flow chart of the QGA mechanism in order to complete the optimization is shown in Figure 5.

Step 1: Parameter initialization: This step determines population size, number of iterations, number of chromosomes, and chromosome length.

Step 2: Population initialization: This step randomly generates a population particle $Q_i(a, b, c, d)$, where a represents the number of neurons, b represents the number of hidden layers, and c represents the learning rate of LSTM, d represents the number of iterations of the LSTM.

Step 3: Measurement: The observed state P_i is obtained by measuring each individual in the population Q_i.

Step 4: Evaluation of fitness: The fitness function set in this paper is the accuracy of the LSTM model. This function is used to evaluate each individual in the observed state $P(t)$.

Step 5: Optimization: Using the most retained individual strategy, record the individuals with the greatest fitness in the observed state $P(t)$.

Step 6: Terminate condition: If the expected optimal result is obtained for the whole result, the algorithm ends; otherwise, return to Step 7.

Step 7: Update: Update the population with a quantum revolving door, return to Step 3.

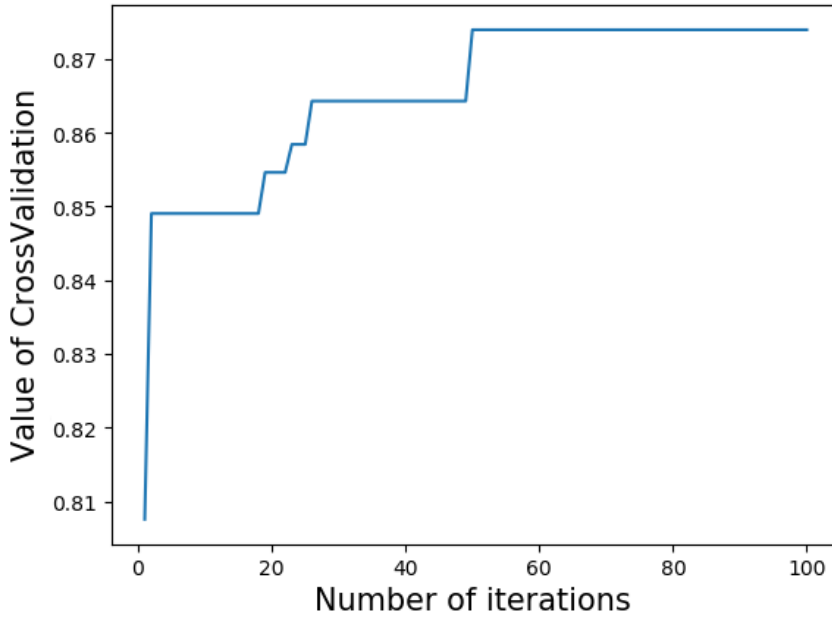


Figure 4. The effect of the number of iterations

5 EXPERIMENTAL RESULTS

5.1 Test Conditions and Data Sets

The operating environment used in the experiment is as follows: the motherboard: ASUS X455LJ, CPU: Intel® Core™ i5-5200U CPU @ 2.20 GHz (2 195 MHz), memory: 12.00 GB (1 600 MHz), main hard disk: Samsung SSD 860 EVO 250 GB; software environment: Microsoft Windows 10 (64 bit), and Pycharm integrated development environment.

As the collection methods of data sets used in each article are different, the amount of inconsistent data in each data set will vary and it may lead to different recognition accuracy. Hence, it is necessary to compare the data sets with large gaps separately. The collected data sets are usually divided into self-collection data sets and engineering data sets. Data collected independently is usually realized by the built-in function of mobile application which collects the movement of individuals for a while. Engineering data sets are collected and measured by a professional project, including Geo Life Project [21, 22, 23] and some other projects used in most studies. For example, Zhu et al. [16], Zhu et al. [6], and Zhu et al. [27] all use Geo Life's data. Geo Life Project is also used in this paper. Compared to the data collected by the application, the data collected by the project is more

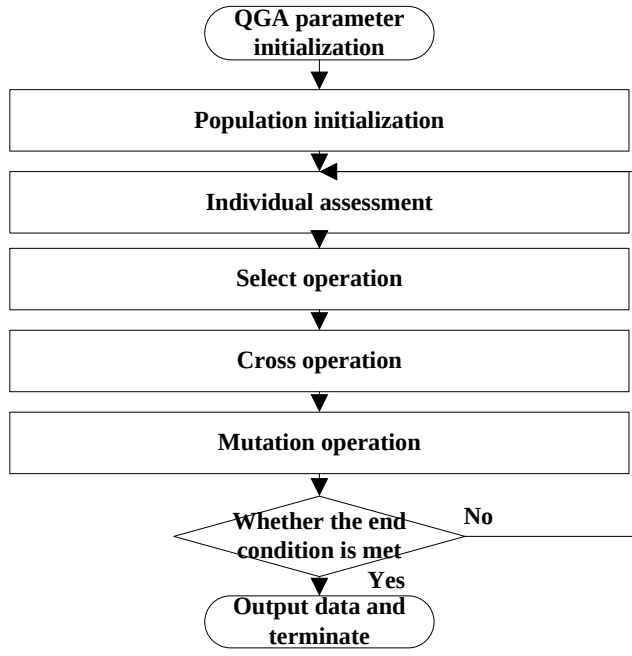


Figure 5. QGA flow chart

standard, accurate and stable. In reality, data will inevitably have inconsistency, and thus using data closer to the real environment will help to improve the value of application in real-life.

5.2 Comparing Results of Travel Mode Recognition Test

In this work, the LSTM model is used to identify the GPS data, and the three characteristics of stop point, speed and acceleration are added to recognize the travel pattern. This algorithm is compared with the existing algorithms in terms of recognition accuracy.

In the experiments, to ensure that the proportion of training set and test set samples is similar, 20 % of the data is used as a test set through a random selection.

5.2.1 Comparison with the Different Models

During testing, it is found that the accuracy of the algorithm varies slightly with each training recognition. To reduce the fluctuation of test results, the test is repeated 10 times. The average accuracy of 10 times is taken as the final accuracy of the improved algorithm.

Besides, the standard deviation of 10 recognition accuracy is calculated, and the stability of this algorithm and the existing algorithm is analyzed, as shown in Tables 1 and 2. By performing ten repeated tests, the fluctuation of the accuracy in one test is avoided, and the contingency of the experimental result is reduced to some extent.

LSTM	BP Neural Network	Random Forest	CNN
81.0 %	80.4 %	79.4 %	81.0 %
81.7 %	78.0 %	77.9 %	69.0 %
80.2 %	77.4 %	80.2 %	84.9 %
82.5 %	78.6 %	79.8 %	83.3 %
81.7 %	78.6 %	79.9 %	63.5 %
83.3 %	76.2 %	78.9 %	74.6 %
81.7 %	76.8 %	79.5 %	84.1 %
84.1 %	83.3 %	78.7 %	74.6 %
83.3 %	78.6 %	79.5 %	81.7 %
82.5 %	80.4 %	81.7 %	81.7 %

Table 1. Comparison of 10 recognition accuracy of different models

	LSTM	BP Neural Network	Random Forest	CNN
Accuracy	82.22 %	79.13 %	79.55 %	77.86 %
Average Standard Deviation	0.01132	0.00966	0.01970	0.071814

Table 2. Comparison of recognition of different models

Long-term memory is stored in the LSTM model. Long-term memory and short-term memory are used to fit different patterns, which is more suitable for identifying problems with continuous characteristics. In different modes, their travel modes are different. For example, the acceleration time of the walk mode is short, the speed is stable and low; the acceleration time of bus and car mode is longer and the speed is larger. Therefore, LSTM can distinguish between different modes more accurately. Compared to the existing algorithms, the average accuracy of the proposed algorithm is improved, and the fluctuation of recognition accuracy is stable.

5.2.2 Accuracy Analysis of Mode Recognition

In Figure 6, the horizontal axis is the data amount, which is represented by training set/test set, and the vertical axis is the recognition accuracy. It is evident from Figure 5 that as the number of data increases, the recognition accuracy of the algorithm also increases, and gradually stabilizes. The selection of more data could increase the recognition accuracy of the model, but it may also increase the time complexity of recognition. In the proposed experiment, the existing data has shown that the LSTM is better than other algorithms in travel mode recognition and can

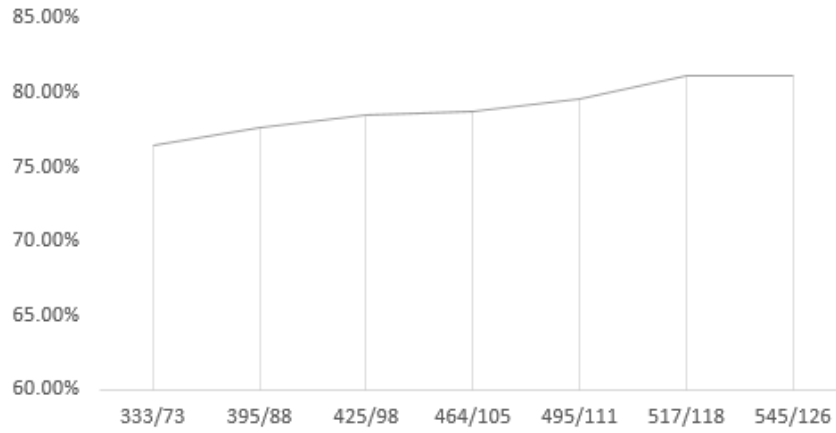


Figure 6. Algorithm identification accuracy analysis chart

have higher recognition accuracy. Therefore, although the experimental results are limited by the number of data sets, they can still make a good classification of the travel mode.

In the deep learning algorithm, the amount of data is important for the improvement of recognition accuracy. Data set covers the common modes of travel in people's daily life, and the proportion of different travel modes in the data set is also close to reality. Among the 671 travel data, we selected 80 % of the data as the training set and the remaining 20 % as the test set. All the data was split by absolutely random and ensured one data does not appear in both training and test sets. Among the 10 identifications, this paper chooses the best model to be used as the test travel mode recognition obfuscation matrix, as shown in Table 3. In the case of optimal recognition, the natural error can be sufficiently reduced, the influence of actual data and algorithms on the recognition accuracy can be better reflected, which facilitates the analysis of characteristics and algorithms.

Actual \ Prediction	Walk	Bike	Bus	Car	Total	Accuracy
Walk	36	4	0	0	40	90.0 %
Bike	3	18	2	1	24	75.0 %
Bus	0	3	7	6	16	43.8 %
Car	0	0	4	59	63	93.7 %
Total	39	25	13	66	143	83.9 %

Table 3. Travel mode recognition confusion matrix

In all our conducted experiments, the accuracy of bus mode recognition is significantly lower than that of other modes. According to the analysis provided in Table 3, due to the limitation of the number of data, there are fewer bus modes in the data set, so the characteristics of bus mode cannot be fitted well in the training process thus it reduces the recognition rate.

Velocity, acceleration and stop point are selected as features to be input into the model. In characteristics analysis, in the case of motor vehicles and non-motor vehicles, the polarization of features is more obvious – usually, the speed and acceleration of motor vehicles are greater, and there are more stop points. There will not be any major errors in model recognition. For the accurate recognition of motor vehicle and non-motor vehicle modes, there will be errors. For example, in Table 3, the left-lower and right-upper parts of the confusion matrix are 0, while the left-upper and right-lower parts are confusion.

5.2.3 Parameter Optimization Using the QGA

	Before Optimization	After Optimization
Accuracy	82.22 %	83.75 %
Error	17.78 %	16.25 %

Table 4. Comparison of the effects of different parameter optimization algorithms on classification results

To verify that the proposed QGA method has an impact on the identification results of the LSTM model, Table 4 shows the test results after optimization and without optimization. The results show that the LSTM model with optimization parameters has higher recognition accuracy and better classification effect than that of without optimization parameters.

6 CONCLUSION AND FUTURE WORK

In this paper, the travel mode of GPS data is extracted from a section of GPS data. Compared to previous studies, this paper converts data to the grayscale image and uses the LSTM model to identify the travel mode, and uses the QGA to optimize the parameters of the model. The identification accuracy of the model can be improved by using the QGA parameter optimization algorithm. Comprehensive analysis shows that the LSTM model and parameter optimization have a better effect on improving the accuracy of travel modes recognition algorithm, can effectively improve the accuracy of travel modes recognition for GPS data, and has higher application value for the problem of travel modes recognition for GPS data.

The adopted technique has greatly improved the overall recognition accuracy, but due to the limitation of data sets, the data used for training in this paper is

less, and the recognition problem is not enough to make accurate judgments, which should be considered in our future work.

7 DATA AVAILABILITY

The Geo-life project GPS data supporting this meta-analysis are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request. Data can be downloaded through the following address: `https://www.microsoft.com/en-us/download/details.aspx?id=52367`.

Acknowledgment

This work was supported by the National Key R & D Program of China (Grant No. 2017YFC0803700), the Beijing Natural Science Foundation Program (Grant No. 4184099), National Natural Science Foundation of China (Grant No. 41971367) and Construction and Development of Key Laboratory of the Ministry of Public Security.

REFERENCES

[1] Wang, B.—Gao, L.—Juan, Z.: Travel Mode Detection Using GPS Data and Socioeconomic Attributes Based on a Random Forest Classifier. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, 2018, No. 5, pp. 1547–1558, doi: 10.1109/TITS.2017.2723523.

[2] Dabiri, S.—Heaslip, K.: Inferring Transportation Modes from GPS Trajectories Using a Convolutional Neural Network. Transportation Research Part C: Emerging Technologies, Vol. 86, 2018, No. 1, pp. 360–371, doi: 10.1016/j.trc.2017.11.021.

[3] Liang, J.—Zhu, Q.—Zhu, M. et al.: An Enhanced Transportation Mode Detection Method Based on GPS Data. In: Zou, B., Li, M., Wang, H., Song, X., Xie, W., Lu, Z. (Eds.): Data Science (ICPCSEE 2017). Springer, Singapore, Communications in Computer and Information Science, Vol. 727, 2017, pp. 605–620, doi: 10.1007/978-981-10-6385-5_51.

[4] Xiao, G.—Juan, Z.—Zhang, C.: Travel Mode Detection Based on GPS Track Data and Bayesian Networks. Computers, Environment and Urban Systems, Vol. 54, 2015, pp. 14–22, doi: 10.1016/j.compenvurbsys.2015.05.005.

[5] Zheng, Y.—Liu, L.—Wang, L.—Xie, X.: Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. Proceedings of the 17th International Conference on World Wide Web (WWW 2008), 2008, pp. 247–256, doi: 10.1145/1367497.1367532.

[6] Zhu, Q.—Zhu, M.—Li, M.—Fu, M.—Huang, Z.—Gan, Q.—Zhou, Z.: Transportation Modes Behaviour Analysis Based on Raw GPS Dataset. International

Journal of Embedded Systems (IJES), Vol. 10, 2018, No. 2, pp. 126–136, doi: 10.1504/IJES.2018.090569.

[7] Stenneth, L.—Wolfson, O.—Yu, P. S.—Xu, B.: Transportation Mode Detection Using Mobile Phones and GIS Information. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11), 2011, pp. 54–63, doi: 10.1145/2093973.2093982.

[8] Guvensan, M. A.—Dusun, B.—Can, B.—Turkmen, H. I.: A Novel Segment-Based Approach for Improving Classification Performance of Transport Mode Detection. Sensors (Basel), Vol. 18, 2018, No. 1, Art. No. 87, 19 pp., doi: 10.3390/s18010087.

[9] Reddy, S.—Mun, M.—Burke, J.—Estrin, D.—Hansen, M.—Srivastava, M.: Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks, Vol. 6, 2010, No. 2, Art. No. 13, 27 pp., doi: 10.1145/1689239.1689243.

[10] Bolbol, A.—Cheng, T.—Tsapakis, I.—Haworth, J.: Inferring Hybrid Transportation Modes from Sparse GPS Data Using a Moving Window SVM Classification. Computers, Environment and Urban Systems, Vol. 36, 2012, No. 6, pp. 526–537, doi: 10.1016/j.compenvurbsys.2012.06.001.

[11] Brunauer, R.—Hufnagl, M.—Rehrl, K.—Wagner, A.: Motion Pattern Analysis Enabling Accurate Travel Mode Detection from GPS Data Only. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 404–411, doi: 10.1109/ITSC.2013.6728265.

[12] Xia, H.—Qiao, Y.—Jian, J.—Chang, Y.: Using Smart Phone Sensors to Detect Transportation Modes. Sensors (Basel), Vol. 14, 2014, No. 11, pp. 20843–20865, doi: 10.3390/s141120843.

[13] Zong, F.—Bai, Y.—Wang, X.—Yuan, Y.—He, Y.: Identifying Travel Mode with GPS Data Using Support Vector Machines and Genetic Algorithm. Information, Vol. 6, 2015, No. 2, pp. 212–227, doi: 10.3390/info6020212.

[14] Liang, X.—Wang, G.: A Convolutional Neural Network for Transportation Mode Detection Based on Smartphone Platform. 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017, pp. 338–342, doi: 10.1109/MASS.2017.81.

[15] Martin, B. D.—Addona, V.—Wolfson, J.—Adomavicius, G.—Fan, Y.: Methods for Real-Time Prediction of the Mode of Travel Using Smartphone-Based GPS and Accelerometer Data. Sensors (Basel), Vol. 17, 2017, No. 9, Art. No. 2058, 20 pp., doi: 10.3390/s17092058.

[16] Zhu, X.—Li, J.—Liu, Z.—Yang, F.: Learning Transportation Mode Choice for Context-Aware Services with Directed-Graph-Guided Fused Lasso from GPS Trajectory Data. 2017 IEEE International Conference on Web Services, 2017, pp. 692–699, doi: 10.1109/ICWS.2017.83.

[17] Yuan, X.—Li, L.—Wang, Y.: Nonlinear Dynamic Soft Sensor Modeling with Supervised Long Short-Term Memory Network. IEEE Transactions on Industrial Informatics, Vol. 16, 2020, No. 5, pp. 3168–3176, doi: 10.1109/TII.2019.2902129.

[18] Dai, S.—Li, L.—Li, Z.: Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction. IEEE Access, Vol. 7, 2019, pp. 38287–38296, doi: 10.1109/ACCESS.2019.2907000.

[19] Huang, R.—Tian, F.—Tian, W.: Motion Pattern Recognition Using Acceleration Transducer. Computer Engineering and Applications, Vol. 51, 2015, No. 6, pp. 235–239.

[20] Sun, B.—Lü, W.—Li, W.: Activity Recognition Based on Smartphone Sensors and SC-HMM Algorithm. Journal of Jilin University (Science Edition), Vol. 51, 2013, No. 6, pp. 1128–1132.

[21] Zheng, Y.—Zhang, L.—Xie, X.—Ma, W. Y.: Mining Interesting Locations and Travel Sequences from GPS Trajectories. Proceedings of International Conference on World Wide Web (WWW 2009), Madrid, Spain. ACM Press, 2009, pp. 791–800, doi: 10.1145/1526709.1526816.

[22] Zheng, Y.—Li, Q.—Chen, Y.—Xie, X.—Ma, W. Y.: Understanding Mobility Based on GPS Data. Proceedings of the 10th ACM Conference on Ubiquitous Computing (UbiComp 2008), Seoul, Korea. ACM Press, 2008, pp. 312–321, doi: 10.1145/1409635.1409677.

[23] Zheng, Y.—Xie, X.—Ma, W. Y.: GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. Invited Paper. IEEE Data(base) Engineering Bulletin, Vol. 33, 2010, No. 2, pp. 32–39.

[24] Vincenty, T.: Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations. Survey Review, Vol. 23, 1975, No. 176, pp. 88–93, doi: 10.1179/sre.1975.23.176.88.

[25] Zhang, H.—Guan, B.—Ren, T.: Inverse Synchronization with Heterogeneous Structure of Chaotic System Based on Quantum Genetic Algorithm. Chinese Journal of Quantum Electronics, Vol. 36, 2019, No. 1, pp. 75–81.

[26] Chen, T.—Guestrin, C.: XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[27] Zhu, Q.—Zhu, M.—Li, M.—Fu, M.—Huang, Z.—Gan, Q.—Zhou, Z.: Identifying Transportation Modes from Raw GPS Data. In: Che, W. et al. (Eds.): Social Computing (ICYCSEE 2016). Springer, Singapore, Communications in Computer and Information Science, Vol. 623, 2016, pp. 395–409, doi: 10.1007/978-981-10-2053-7_35.

Shaowu Zhu is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. His current research interests include machine learning and information service.

Haichun Sun received her Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2015. She is currently Assistant Professor at the College of Information Technology and Network Security, People's Public Security University of China, Beijing, China. She is a member of Professional Committee of Internet Information Service of Chinese Association of Automation. Her current research interests include information service, Petri nets, and service-oriented computing. She has published over 10 papers in journals and conferences such as the IEEE Transactions on Systems, Man, and Cybernetics, WISE 2014.

Yongcheng Duan is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. He is working on situational awareness.

Xiang Dai is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. His current research interests include natural language processing.

Sangeet Saha received his B.Tech. degree in information technology in 2011 and M.Tech. degree in computer science and engineering in 2013 from University of Calcutta, Kolkata in West Bengal, India. He completed his Ph.D. from the same institute as a Tata Consultancy Services (TCS) research fellow in 2018. After completing his Ph.D. in May 2018, he was appointed Senior Research Officer in the Embedded and Intelligent Systems (EIS) Research Group at the University of Essex in Colchester. His current research interests include real-time scheduling, scheduling for reconfigurable computers, real-time and fault-tolerant embedded systems, cloud computing.

Computing and Informatics, Vol. 39, 2020, 318–339, doi: 10.31577/cai_2020_1-2_318

DEEP CONVOLUTION AND CORRELATED MANIFOLD EMBEDDED DISTRIBUTION ALIGNMENT FOR FOREST FIRE SMOKE PREDICTION

Yaoli Wang, Xiaohui Liu

College of Information and Computer
Taiyuan University of Technology
Jinzhong, 030600, China
e-mail: wangyaoli@tyut.edu.cn, 1662010618@qq.com

Maozhen Li*

The Key Laboratory of Service Computing and Embedded Systems
Tongji University
Shanghai, China
e-mail: Maozhen.Li@gmail.com

Wenxia Di

Foreign Languages Department
Taiyuan Normal University
Jinzhong, 030600, China
e-mail: wendy_cn@263.net

Lipo Wang

School of Electrical and Electronic Engineering
Nanyang Technological University
639798, Singapore
e-mail: elpwang@ntu.edu.sg

* Corresponding author

Abstract. This paper proposes the deep convolution and correlated manifold embedded distribution alignment (DC-CMEDA) model, which is able to realize the transfer learning classification between and among various small datasets, and greatly shorten the training time. First, pre-trained Resnet50 network is used for feature transfer to extract smoke features because of the difficulty in training small dataset of forest fire smoke; second, a correlated manifold embedded distribution alignment (CMEDA) is proposed to register the smoke features in order to align the input feature distributions of the source and target domains; and finally, a trainable network model is constructed. This model is evaluated in the paper based on satellite remote sensing image and video image datasets. Compared with the deep convolutional integrated long short-term memory (DC-ILSTM) network, DC-CMEDA has increased the accuracy of video images by 1.50 %, and the accuracy of satellite remote sensing images by 4.00 %. Compared the CMEDA algorithm with the ILSTM algorithm, the number of iterations of the former has decreased to 10 times or less, and the algorithm complexity of CMEDA is lower than that of ILSTM. DC-CMEDA has a great advantage in terms of convergence speed. The experimental results show that DC-CMEDA can solve the problem of small sample smoke dataset detection and recognition.

Keywords: Transfer learning, domain adaptation, deep convolution, small dataset, forest fire smoke

1 INTRODUCTION

The research on smoke image detection technology mainly focuses on the classification and recognition of smoke images through deep learning training. This method has high classification accuracy, but a strong dependence on the dataset. It requires that the training and test data meet independent and identical distributions, and it needs enough training samples available. However, smoke images of different scenes and different resolutions cannot meet the same and independent distribution. It is also very expensive, and difficult to obtain many labeled sample images in different scenes. Therefore, this type of method requires training of various models, which is time-consuming and labor-intensive. In addition, there is an overfitting phenomenon in small sample scenario training. With the development of artificial intelligence, smoke detection technology based on transfer learning has a good application prospect. This technology is used to learn and train with large sample smoke image data, and to popularize what is collected in small sample smoke images, which not only reduces the time for model training, but also prevents overfitting.

The existing methods mainly proceed from the perspective of deep transfer learning. Literature [1] proposed a feature migration method based on isomorphic data by taking ImageNet dataset as source data and using VGG16 model, which provided a feasible method for smoke detection and identification; literature [2] used the pre-trained VGG16 network on the ImageNet dataset for effective smoke feature

extraction, and proposed an integrated long-term and short-term memory network, which uses this network to fuse smoke features in segments. Finally, a trainable deep neural network model is constructed, which can be used for forest fire smoke detection. This method uses pre-trained VGG16 model feature extraction, which requires high requirements for sample dataset, and the model has a shallow network depth and insufficient extraction of certain features, which will result in a low accuracy of recognition and classification.

Literature [3] proposed a recursive convolutional neural network based on RNN and successfully applied it in the field of video smoke detection; literature [4] introduced the Inception-v3 network trained on the ImageNet dataset, first remove and reset the last full connection layer in the Inception-v3 network, then freeze all parameters of the convolutional layer and pooling layer in the previous hidden layer, and then use the collected small smoke dataset for training to fine-tune the reset full connection layer. Finally, the smoke detection model of deep transfer learning convolutional neural network is obtained. This method reduces the number of datasets in the process of fine-tuning the fully connected layer, but it is still unable to accurately identify the classification for small sample datasets; and the deep network is complex, with many parameters, and the fine-tuning still takes a long time.

Another popular research area of transfer learning [5] at present is domain adaptation [6], which has a good effect on solving the problem of source domain data calibration target domain data, achieves data classification, and has advantages in the number of parameters and time consumption. The probability distribution adaptation method is mainly carried out from three aspects, namely: edge distribution adaptation, conditional distribution adaptation, and joint distribution adaptation. The earliest application of conditional distribution adaptation [7] to transfer learning was achieved by domain adaptive of conditional probability models through feature subsets, and then the Conditional Transition Component (CTC) [8] was modeled to make the method get developed. Transfer Component Analysis (TCA) [9] applies edge distribution adaptation to transfer learning, and some scholars have extended the Transfer Component Analysis (TCA), such as ACA [10], DTMKL [12], DME [13], CMD [14, 11], etc. Joint Distribution Alignment (JDA) [15] considers the edge distribution and the conditional distribution at the same time, and the effect is better, but the importance of the edge distribution and the conditional distribution is not considered, and the default weights are equal. The proposal of Balanced Distribution Adaptation (BDA) [16] improves the Joint Distribution Alignment (JDA). It considers the distribution adaptability between domains and enables the weight of each class to be changed adaptively. This type of method registers the probability distribution, but lacks consideration of subspace alignment. Therefore, its transfer effect is limited for smoke detection and recognition.

In addition, the subspace learning method is a method of transforming the source domain and the target domain to the same subspace, and then building a unified model. It solves the problem of domain adaptation mainly from two aspects, which are statistical property transformation and manifold learning [17]. In terms of statistical characteristics transformation, the Subspace Alignment (SA) [18] directly

reduces the distance between the two domains by optimizing the mapping function that converts the source domain subspace to the target subspace to bring the source domain subspace and the target domain subspace closer together. The Subspace Distribution Alignment (SDA) [19] adaptively expands SA by increasing the subspace variance without considering the local attributes of the subspace, and ignores the conditional distribution alignment; CORrelation Alignment (CORAL) [20] aligns subspaces in second-order statistics without considering distribution alignment. Scatter Component Analysis (SCA) [21] minimizes the divergence between them by converting the samples into a set of subspaces. In terms of manifold learning, the Sample Geodesic Flow (SGF) [21] treats the problem of domain adaptation as an incremental "walking" problem, and samples a limited number of points in the manifold space to construct a geodesic flow; Geodesic Stream Kernel (GFK) [18] extends the idea of sampling points in the manifold, and proposes a learning method for inter-domain geodesic stream kernels; Domain Invariant Projection (DIP) [23, 14] passed Grasmanian manifolds are used for domain adaptation, but conditional distribution alignment is ignored; Heilinger distance is used to approximate the geodesic distance in Riemann space, and a Statistical Manifold (SM) is proposed [24]. These methods solve the problem of domain adaptation from the perspective of subspace learning, but they lack the registration of probability distributions and still have limitations for smoke detection and recognition.

In 2018, Wang et al. proposed Manifold Embedded Distribution Alignment (MEDA) from the perspective of probability distribution adaptation method and subspace learning method [25], not only using the principle of structural risk minimization to learn the domain invariant classifier on manifold domain, but also aligning the edge distribution and conditional distribution dynamically, which provides a feasible method for quantitative calculation of adaptive factors. This method achieves good results in transfer learning classification, but still has the limitation of lacking feature extraction [26, 27] and original feature space alignment in solving smoke detection and recognition.

In view of the above problems, this paper proposes a smoke detection model by combining deep CNN and improved MEDA. This model can realize the transfer learning classification between various small datasets, and greatly reduce the time complexity. Firstly, pre-trained Resnet50 network is used for feature transfer to extract smoke features because of the difficulty in training small dataset of forest fire smoke; secondly, a correlated manifold embedded distribution alignment (CMEDA) is proposed to register the smoke features in order to align the input feature distributions of the source and target domains; finally, a trainable network model is constructed.

2 DATA FEATURES EXTRACTION IN ALL DOMAINS

This paper explores and compares several CNN models with different parameter settings for forest fire smoke detection, namely, AlexNet, Resnet, VGG, and GoogleNet.

In these models, AlexNet and GoogleNet use large convolution kernels with sizes of 11×11 and 7×7, and with step sizes of 3 and 5, which may ignore important features of the smoke area. Although VGG uses a small convolution kernel with a size of 3×3 and a step size of 1, VGG shallow network depth is not conducive to processing and extracting the features of each pixel of the smoke image, and VGG takes up a large space with a size of 528 MB. In addition, compared with Resnet34, Resnet50 replaces two convolution kernels of 3×3 with convolution kernels of sizes 1×1, 3×3, and 1×1. With similar time complexity, it is beneficial to process and extract the features of each pixel of smoke images, with higher accuracy and less computation. The parameter comparison between Resnet50 and other CNN models is shown in Table 1. As seen from Table 1, the top-1 accuracy, top-5 accuracy, and top-5 test error rate of Resnet50 on the ImageNet dataset are better than those of other state-of-the-art architectures of models. Therefore, this paper improves the architecture of the Resnet50 model based on the problem of forest fire smoke detection. The paper first uses the ImageNet dataset and small sample satellite remote sensing image dataset to fine-tune the parameters of the Resnet50 network model, then transfers the trained parameters corresponding to the convolutional layer and removes the fully connected layer to obtain the transfer learning network model, and finally, uses the above model for feature extraction of smoke and non-smoke images in the source and target domains.

Model	Parameter Amount/Million	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Top-5 Error Rate (%)
GoogleNet	60	69.8	89.3	6.7
AlexNet	7	57.5	80.3	16.4
VGG16	138	70.5	91	7.3
Resnet50	256	75.9	92.9	5.25

Table 1. Comparison of Resnet50 and other CNN model parameters

The transfer learning model based on Resnet50 network in this paper is shown in Figure 1. This model transfers the trained parameters of the model on the ImageNet dataset and the small sample satellite Remote Sensing (RS) image dataset, removes the fully connected layers, and finally performs feature extraction on the smoke and images in the two domains. As shown in Figure 1 on the left, the smoke feature extraction model used in this paper is mainly composed of convolutional layers and downsampling layers. The model contains a total of 49 convolutional layers and 4 downsampling layers, of which the first segment is a convolutional layer composed of a $7 \times 7 \times 64$ convolution kernel; the second segment consists of three bottleneck structures, each of which contains three convolutional layers consisting of $1 \times 1 \times 64$, $3 \times 3 \times 64$, and $1 \times 1 \times 256$ convolution kernels; the third segment (unshown in Figure 1) is composed of 4 bottleneck structures, and each bottleneck structure contains three convolution layers consisting of $1 \times 1 \times 128$, $3 \times 3 \times 128$, and $1 \times 1 \times 512$ convolution kernels; the fourth segment (unshown in Figure 1) is composed of 6 bottleneck structures, and each bottleneck structure contains three

convolution layers consisting of $1 \times 1 \times 256$, $3 \times 3 \times 256$, and $1 \times 1 \times 1\,024$ convolution kernels; the fifth segment is composed of three bottleneck structures, and each bottleneck structure contains three convolution layers consisting of $1 \times 1 \times 512$, $3 \times 3 \times 512$, and $1 \times 1 \times 2\,048$ convolution kernels.

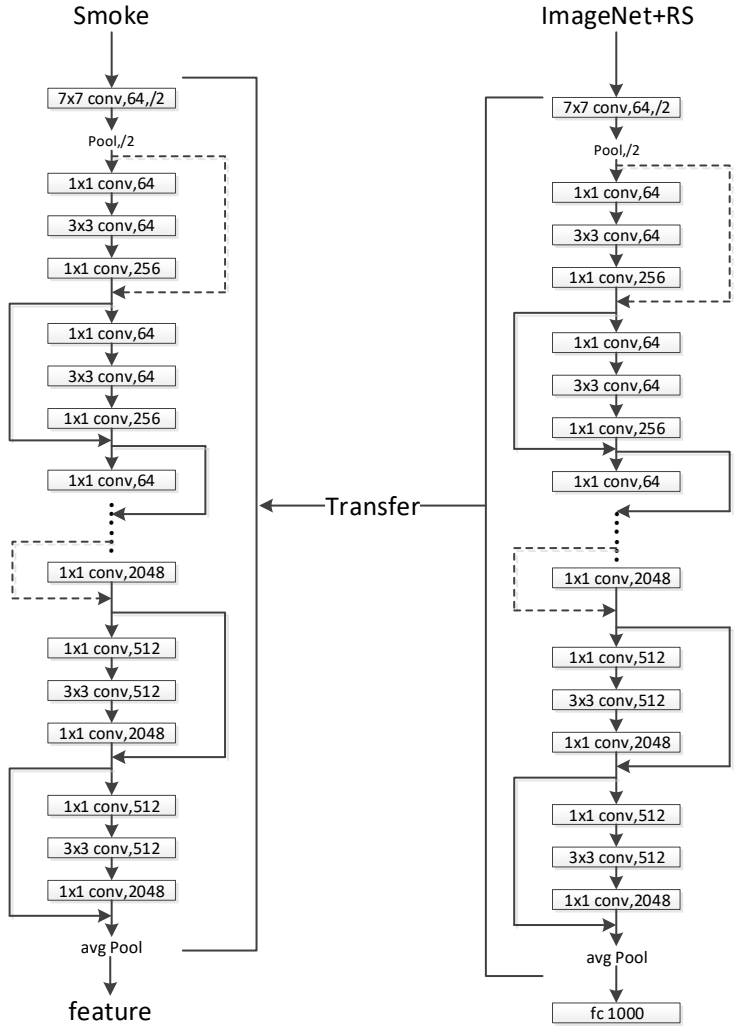


Figure 1. Transfer learning model based on Resnet50 network

This model is mainly obtained from the Resnet50 network, and the corresponding Resnet50 network training parameters are loaded at the same time. First, the convolutional layer is constructed based on the Resnet50 network; second, the smoke dataset is used as input to obtain the parameters of the convolutional layer in the

trained Resnet50 network on ImageNet; finally, image feature extraction is performed.

3 CORRELATED MANIFOLD EMBEDDED DISTRIBUTION ALIGNMENT ALGORITHM

From the perspective of the probability distribution adaptation method and the subspace learning method, MEDA uses the principle of structural risk minimization to learn the domain-invariant classifier on the manifold domain, and dynamically aligns the edge distribution and conditional distribution. Therefore, the drift between domains is greatly reduced. The specific flow of the algorithm is shown in Figure 2.

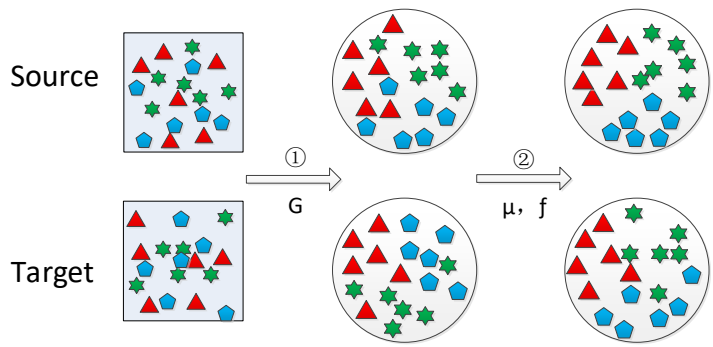


Figure 2. The idea of the manifold distribution registration method. ① Transform the features in the original space into the manifold space by learning the manifold kernel **G**. ② Align dynamically the distributions through learning the adaptive factor μ, and learn the final domain-invariant classifier f through structural risk minimization in the manifold space.

Compared with the features of the original space, the features of the manifold space have some good geometric structures, which can avoid the distortion of the features. Therefore, to eliminate the degenerate feature transformation, the manifold feature learning is an important processing step. When learning manifold feature transformations, MEDA uses d-dimensional subspaces to model the data domain, and then embeds these subspaces into the manifold **G**.

After obtaining the manifold characteristics, in order to dynamically measure the relative importance of the edge distribution and the conditional distribution, MEDA introduced an adaptive factor to adaptively balance the two distributions. In formal language, the adaptive distributed adaptation $\overline{D_f}$ can be expressed as:

$$\overline{D_f}\left(\mathrm{D}_s, \mathrm{D}_t\right) = (1-\mu)\, D_f\left(P_s, P_t\right) + \mu \sum_{c=1}^{C} D_f^{(c)}\left(Q_s, Q_t\right) \tag{1}$$

where $\mu \in [0,1]$ represents the adaptive factor, $c \in \{1, \cdots, C\}$ is a category indication. $D_f(P_s, P_t)$ represents edge distribution adaptation, $D_f^{(c)}(Q_s, Q_t)$ indicates a conditional distribution adaptation to category c.

MEDA uses Maximum Mean Discrepancy (MMD) to calculate the difference between two probability distributions. The MMD distance between two probability distributions p and q is defined as $d^2(p,q) = \left(\mathrm{E}_p\left[\phi\left(\mathbf{Z}_s\right)\right] - \mathrm{E}_q\left[\phi\left(\mathbf{Z}_t\right)\right]\right)_{\mathcal{H}_K}^2$, where $\mathcal{H}_K$ is the Reproducing Kernel Hilbert space (RKHS) expanded by the feature map $\phi(\bullet)$, and $\mathrm{E}(\bullet)$ is the mean of the embedded samples. Finally, MEDA summarizes the manifold learning and dynamic distribution alignment, and learns the final domain-invariant classifier through structural risk minimization:

$$f = \underset{f \in \sum_{i=1}^{n} \mathcal{H}_K}{\arg\min}\ l\left(f\left(g\left(\mathbf{x}_i\right)\right), y_i\right) + \eta \|f\|_K^2 + \lambda \overline{D_f}\left(\mathrm{D}_s, \mathrm{D}_t\right) + \rho \overline{R_f}\left(\mathrm{D}_s, \mathrm{D}_t\right). \tag{2}$$

In the formula, $g\left(\mathbf{x}_i\right)$ represents learning manifold features, $\overline{D_f}\left(\mathrm{D}_s, \mathrm{D}_t\right)$ represents dynamically aligned edge distribution and conditional distribution, and $\overline{R_f}\left(\mathrm{D}_s, \mathrm{D}_t\right)$ is a regularization term. This part can better learn the geometric properties of the closest point in the manifold space. MEDA is the first attempt to deal with degraded feature conversion and unassessed distribution alignment challenges. It has achieved a good result in classification accuracy, but it still has limitations in the detection and recognition of smoke images. If the input features of the two fields are first distributed and aligned in the original space before manifold feature learning, the two fields will be better registered and the classification accuracy will be higher. Therefore, this paper proposes the CMEDA module. The CMEDA module adds an input feature distribution alignment section to the MEDA module. The input feature distribution alignment first removes the feature correlation of the source domain, then re-associates the target domain, and finally adds the association of the target domain to the source characteristics. The process is shown in Figure 3.

In the original space, the input feature distributions of the source and target domains are aligned by comparing the second-order statistics of the two domains. This method can minimize the domain offset. In order to minimize the distance between the second-order statistics (covariance) of the two domains, this paper performs a linear transformation on the original source features and uses the Frobenius norm as the matrix distance metric, as follows:

$$\min_{\mathbf{A}} \left\|\mathbf{C}_{\hat{S}} - \mathbf{C}_T\right\|_F^2 = \min_{\mathbf{A}} \left\|\mathbf{A}^T \mathbf{C}_S \mathbf{A} - \mathbf{C}_T\right\|_F^2. \tag{3}$$

Calculated $\mathbf{A}$:

$$\mathbf{A} = \mathbf{U}_S \mathbf{E} = \left(\mathbf{U}_S \mathbf{\Sigma}_S{}^{\frac{1}{2}} \mathbf{U}_S{}^T\right)\left(\mathbf{U}_{T[1:r]} \mathbf{\Sigma}_{T[1:r]}{}^{\frac{1}{2}} \mathbf{U}_{T[1:r]}{}^T\right). \tag{4}$$

$\mathbf{U}_S \mathbf{\Sigma}_S{}^{\frac{1}{2}} \mathbf{U}_S{}^T$ in $\mathbf{A}$ can be regarded as removing the feature correlation of the source domain, and $\mathbf{U}_{T[1:r]} \mathbf{\Sigma}_{T[1:r]}{}^{\frac{1}{2}} \mathbf{U}_{T[1:r]}{}^T$ can be regarded as re-associating the target domain and adding the association of the target domain to the source characteristic.

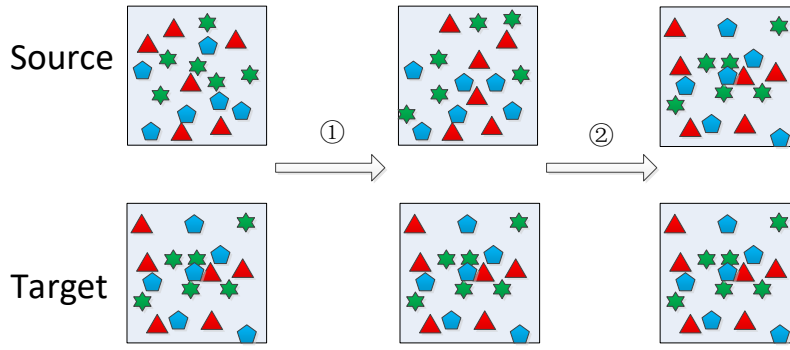


Figure 3. Associated alignment input feature distribution. ① Remove the feature correlation of the source domain and keep the target domain unchanged. ② Re-associate the target domain, add the correlation of the target domain to the source characteristics, and obtain the feature distribution which is aligned between the source domain and the target domain.

Therefore, the CMEDA module consists of three parts: the first part is the association alignment input feature distribution, the second part is the manifold feature learning, and the third part is the dynamic alignment of the edge distribution and the conditional distribution. The overall flowchart of the module is shown in Figure 4.

In manifold feature learning, $\mathbf{S}_s$ and $\mathbf{S}_t$ are used to represent the subspace of the source and target domains after Principal Component Analysis (PCA), respectively, then $\mathbf{G}$ can be regarded as a set of all d-dimensional subspaces. Each d-dimensional primitive subspace can be viewed as a point on $\mathbf{G}$ and the geodetic line $\{\mathbf{\Phi}(t) : 0 \leq t \leq 1\}$ between two points can form a path between two subspaces.

Let $\mathbf{S}_s = \mathbf{\Phi}(0)$, $\mathbf{S}_t = \mathbf{\Phi}(1)$, finding a geodesic from $\mathbf{\Phi}(0)$ to $\mathbf{\Phi}(1)$ is equivalent to transforming the original features into a space of infinite dimensions, and finally reducing the drift between domains. In particular, features in a manifold space can be represented as $\mathbf{Z} = \mathbf{\Phi}(t)^T\mathbf{X}$. The inner product of the transformed features $\mathbf{z}_i$ and $\mathbf{z}_j$ defines a semi-positive definite GFK.

$$\langle \mathbf{z}_i, \mathbf{z}_j \rangle = \int_0^1 \left(\mathbf{\Phi}(t)^T\mathbf{x}_i\right)^T \left(\mathbf{\Phi}(t)^T\mathbf{x}_j\right) dt = \mathbf{x}_i{}^T\mathbf{G}\mathbf{x}_j. \tag{5}$$

Therefore, through $\mathbf{Z} = \sqrt{\mathbf{G}}\mathbf{X}$, the features in the original space can be transformed into Grassmann manifold space, and the kernel $\mathbf{G}$ can be efficiently calculated by matrix singular value decomposition.

In addition, because the target domain data D_t has no labels, it is not feasible to directly evaluate the conditional probability distribution $Q_t = Q_t(y_t|\mathbf{Z}_t)$ of

the target domain, and when the number of samples is large enough, $Q_t(\mathbf{Z}_t|y_t)$ and Q_t have a good similarity. The class conditional probability $Q_t(\mathbf{Z}_t|y_t)$ is used to approximate Q_t. To approximate $Q_t(\mathbf{Z}_t|y_t)$, a weak classifier is trained on the source domain D_s, and then this weak classifier is used to predict on D_t to obtain pseudo-labels in the target domain. Since the confidence of these pseudo-labels may not be high, this weak classifier can iteratively modify the prediction results.

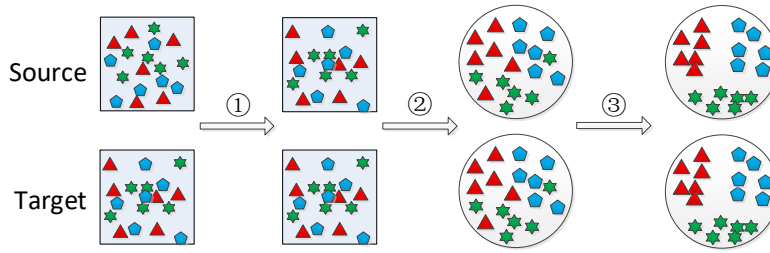


Figure 4. CMEDA flowchart. ① Remove the feature correlation of the source domain and add the correlation of the target domain to the source features to obtain the feature distribution of the source domain aligned with the target domain in the original space. ② Transform the distribution-aligned features in the original space into the manifold space by learning the manifold kernel. ③ Align adaptively the distribution in the manifold space by learning adaptive factors, and learn the final domain-invariant classifier through structural risk minimization.

4 FOREST FIRE SMOKE DETECTION METHOD BASED ON DC-CMEDA

The main purpose of the forest fire smoke detection method is to construct a classification algorithm which is able to realize the migration of trace datasets to achieve forest fire smoke detection. The structure of the DC-CMEDA model is shown in Figure 5. The model first uses Resnet50 to extract N-dimensional features from the source and target domain data. Then the source and target domain features are processed by the CMEDA module, that is, the input distribution features are aligned, and the structural risk minimization method is used to learn the domain-invariant classifier in the Grassmann manifold. Meanwhile, dynamic distribution alignment is performed by considering the different importance of edge distribution and conditional distribution, and finally, transfer classification of smoke images from source domain to target domain is realized.

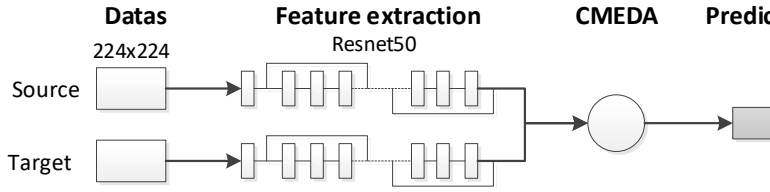


Figure 5. DC-CMEDA model structure

In transfer learning, we use the Resnet50 model based on the ImageNet dataset as a CNN model to extract smoke features for each image. Experiments show that the network trained on the ImageNet dataset has better generalization ability.

During the transfer from the source domain to the target domain, the CMEDA module aligns the input distribution features of the two domains to improvement of the accuracy of image detection. CMEDA not only learned the domain-invariant classifier on the manifold domain by using the principle of structural risk minimization, but also dynamically aligned the edge distribution and conditional distribution. Experiments show that the CMEDA network significantly increases the accuracy of smoke detection compared with the MEDA network.

This method performs feature extraction and transfer classification on images in two domains (satellite remote sensing and video images) and it outputs two types of results (smoked and smokeless). The main steps of the experimental algorithm are as follows.

Algorithm DC-CMEDA

Input: Source domain dataset $\{\mathbf{Simage}_i : 1 \leq i \leq M\}$,
Target domain dataset $\{\mathbf{Timage}_j : 1 \leq j \leq N\}$

Output: Classifier f

1: Preprocess Source and Target domain datasets: Adjust $\mathbf{Simage}_i$ and $\mathbf{Timage}_j$ to resolutions of $3 \times 224 \times 224$ and transform randomly and normalize them
2: Construct transfer learning model based on Resnet50 network
3: Perform feature extraction on $\mathbf{Simage}_i$ and $\mathbf{Timage}_j$, get feature matrices $\mathbf{X}_s$ and $\mathbf{X}_t$, and get source domain label $\mathbf{y}_s$
4: Obtain the feature distribution of the source domain aligned with the target domain in the original space by $\mathbf{X}_s{}' = \mathbf{X}_s * \mathbf{A}$, and get data matrix $\mathbf{X} = (\mathbf{X}_s{}', \mathbf{X}_t)$
5: Train a weak classifier using $\mathbf{D}_s$, then apply the classifier to predict pseudo-label $\hat{\mathbf{y}}_t$ in target domain $\mathbf{D}_s$
6: **repeat**
7: Calculate the adaptive factor μ using Formula (1) and obtain f via Formula (2)
8: Update the label of $\mathrm{D}_t : \hat{\mathbf{y}}_t = f(\mathbf{X}_t)$
9: **until** Convergence
10: **return** Classifier f

5 EXPERIMENTAL RESULTS AND ANALYSES

5.1 Experimental Dataset

This paper studies the transfer learning classification technology between smoke images of different resolution videos in a forest fire video monitoring system. Satellite remote sensing images and video images are selected as experimental data. For satellite remote sensing (RS) images, a large amount of data are available, the cost is low, and it is easy to apply in the real world. When a relatively obvious smoke image is detected, it means that the fire is already very large, and timely feedback and disaster relief cannot be achieved. The video images can just overcome the shortcomings of satellite remote sensing (RS) images which are unable to offer real-time feed back. Video images can quickly capture the fire situation and give real-time feedback, but they still have the disadvantage that they are unable to be automatically interpreted. The transfer learning of satellite remote sensing (RS) images and video images can give full play to their respective advantages and achieve the effect of real-time and automatic interpretation. The dataset in this paper is derived from the Academy of Forestry of Shanxi Province, China.

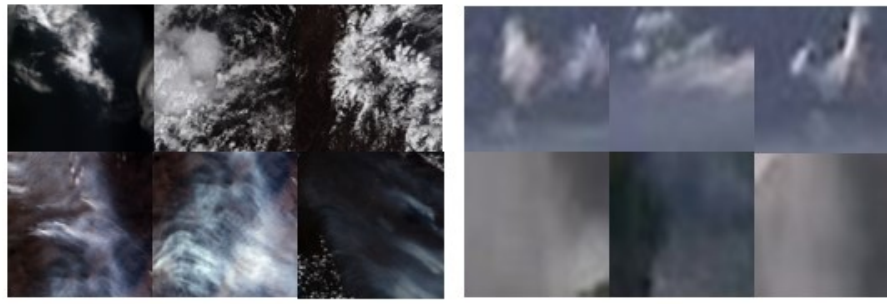


Figure 6. Image sample map

5.2 Experimental Evaluation Criteria

This paper uses Precision, Recall, and the Harmony Mean F1 of both Precision and Recall to measure the network performance, so that images with smoke are positive and images with no smoke are negative. The calculation formula is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

$$F1 = \frac{2TP}{2TP + FP + FN}. \tag{9}$$

TP means to predict positive class as positive class; TN means to predict negative classes as negative classes; FP means to predict negative class as positive class; FN means to predict positive class as negative class.

5.3 Results and Analyses

The followings are four comparative experiments in this paper. Experiment 1 shows the comparison of the parameters on the video image sample dataset and the satellite remote sensing sample dataset by applying the Resnet50 method and other methods. Experiment 2 gives a comparison of detection results of several domain adaptive methods based on Resnet50 network. Experiment 3 compares DC-CMEDA-based forest fire smoke detection methods with other state-of-the-art methods. Experiment 4 gives a comparison of the effects of Resnet modules on the DC-CMEDA algorithm.

The experimental dataset includes a small sample set of 200 satellite remote sensing (RS) images and 200 video images. Different image distributions represent two different domains. Each domain includes 100 images with smoke and 100 images with no smoke.

5.3.1 Experiment 1

In transfer learning, the Resnet50 network is compared with other CNN networks. Tables 2 and 3 show the comparison of the parameters on the video image sample dataset and the satellite remote sensing sample dataset by applying the Resnet50 method and other methods. In this paper, the cross-validation method is used to divide the sample set in each domain into training set, validation set, and test set in proportion. Among them, the training set accounts for 50 % of the total sample (50 images with smoke and 50 images with no smoke), and the validation set and test set each account for 25 % (each contains 25 images with smoke and 25 images with no smoke).

As can be seen from Table 2, the accuracy of the video image sample dataset using AlexNet and GoogleNet is the lowest, and the false positive and false negative score the worst. The detection results using VGG16 are better than the detection results using AlexNet and GoogleNet. But compared with the Resnet50 model, the accuracy of VGG16 is still lower and the false positive is higher. Among AlexNet, GoogleNet, VGG16, and Resnet50, Resnet50 has achieved the best results, which are 16.67 % in false positive, 11.54 % in false negative, and 86.00 % in accuracy.

Table 3 shows the satellite remote sensing verification set. The accuracy of the AlexNet, GoogleNet, and VGG16 models is also lower than that of the Resnet50

Model	False Positive (%)	False Negative (%)	Accuracy (%)
AlexNet	30.43	29.62	70.00
GoogleNet	26.09	25.93	74.00
VGG16	24.00	16.00	80.00
Resnet50	16.67	11.54	86.00

Table 2. Accuracy of video image sample set during training

model, and the false positive and false negative scores are also relatively poor compared to the Resnet50. That is, the experimental verification of the satellite remote sensing set using Resnet50 has relatively good results, of which false positive is 16.67 %, false negative is 19.23 %, and the accuracy is 82.00 %. Therefore, the performance of the Resnet50 model is better than other models.

Model	False Positive (%)	False Negative (%)	Accuracy (%)
AlexNet	29.17	26.92	72.00
GoogleNet	26.09	25.93	74.00
VGG16	17.39	25.93	78.00
Resnet50	16.67	19.23	82.00

Table 3. Accuracy of satellite remote sensing sample set during training

Based on the analyses and summaries of the results in Tables 2 and 3, the following conclusions can be drawn.

1. Learning model parameters through transfer learning can reduce the number of samples required for a dataset. But for too small sample datasets, even the most advanced CNN networks have low accuracy in recognition and classification detection due to the inability to fully adjust the parameters.
2. The model parameters in the experiment are obtained from the pre-trained model migration based on the ImageNet dataset, so there are fewer satellite remote sensing smoke images in the ImageNet dataset. Therefore, comparing Table 2 with Table 3, it is found that the accuracy under the satellite remote sensing verification set is significantly lower than the accuracy under the video image verification set.
3. The accuracy of both the satellite remote sensing verification set and the video image verification set is better than other CNN networks due to the advantages of the size of the convolution kernel and the depth of the network in the Resnet50 network model.

5.3.2 Experiment 2

This part uses the Resnet50 network along with JDA, BDA, GFK, MEDA and CMEDA methods for testing and comparison. Table 4 shows the compared results of false positives, false negatives, and accuracy of various detection and recognition

methods of the satellite remote sensing sample set as the source domain and the video image sample set as the target domain. Table 5 shows the comparison between the source domain, the video image sample set and the target domain, the satellite remote sensing sample set.

As seen from Table 4, when the satellite remote sensing sample set is used as the source domain and the video image sample set as the target domain, the transfer effect of MEDA is obviously better than that of GFK, TDA and BDA either from the perspective of false positives and false negatives or from the perspective of accuracy. However, when MEDA is compared with the CMEDA module, the combined CMEDA has a better effect in experimental verification. In CMEDA, false positive is 4.81 %, false negative is 3.13 %, and the accuracy is 96.00 %.

Model	False Positive (%)	False Negative (%)	Accuracy (%)
GFK	18.18	11.88	85.00
JDA	11.32	9.57	89.50
BDA	8.82	8.16	91.50
MEDA	7.29	3.85	94.50
CMEDA	4.81	3.13	96.00

Table 4. Transfer accuracy of satellite remote sensing images to video images

As seen from Table 5, when the video image sample set is used as the source domain and the satellite remote sensing sample set as the target domain, the transfer effect of CMEDA is much better than that of GFK, TDA, BDA, and even better than MEDA. In CMEDA, false positive is 11.76 %, false negative is 9.18 %, the accuracy is 89.50 %, and the accuracy is significantly improved by 2.00 % comparing with that of MEDA.

Model	False Positive (%)	False Negative (%)	Accuracy (%)
GFK	21.43	18.63	80.00
JDA	16.19	15.79	84.00
BDA	15.89	13.98	85.00
MEDA	13.08	11.83	87.50
CMEDA	11.76	9.18	89.50

Table 5. Transfer accuracy of video images to satellite remote sensing images

Based on the analyses and summaries of the results in Tables 4 and 5, the conclusions are drawn as follows.

1. GFK, JDA, and BDA solve the domain adaptation problem by taking either subspace learning method only or probability distribution adaptation method only into consideration while MEDA applies the two methods simultaneously, and thus has a better transfer effect.

2. The input feature distributions of the source and target domains in CMEDA are aligned in the original space before the manifold feature learning is conducted, so the improved CMEDA module performs better than MEDA.

3. The parameters of the CNN network model are obtained from transfering a pre-trained model based on the ImageNet dataset, so the accuracy of transfering from video image to satellite remote sensing image is lower than that of transfering from satellite remote sensing image to video image.

5.3.3 Experiment 3

This part compares the deep convolutional long-term short-term memory network (DC-ILSTM) proposed by Wei et al. in [2] and the smoke detection method using convolution and recursive network proposed by Filonenko et al. in [3] to verify the detection effect of the DC-CMEDA method shown in Tables 6 and 7.

Table 6 shows the accuracy test results transfered from satellite remote sensing images to video images. After having tested the two methods in [2] and [3], this paper uses satellite remote sensing images to train the model and to fine-tune the parameters, and uses video images as the test set to verify the detection effect. Table 6 shows that the DC-CMEDA method proposed in this paper has an efficient detection effect, and the accuracy is as high as 96.0 %.

Performance	Filonenko's Method [3]	DC-ILSTM [2]	DC-CMEDA
Accuracy (%)	93.5	94.5	96.0
Precision (%)	93.3	93.4	94.9
Recall (%)	94.2	96.2	96.8
F1 (%)	93.7	94.8	95.8

Table 6. Accuracy test results transfered from satellite remote sensing images to video images

Table 7 shows the test results of the accuracy transfered from video images to satellite remote sensing images. Models are trained with video images, the parameters are fine-tuned, and satellite remote sensing images are used as the test set to verify the detection effect. As shown in Table 7, after our having tested the two methods in [2] and [3], and our own method DC-CMEDA, the DC-CMEDA method proposed in this paper has an efficient detection effect, and the accuracy is as high as 89.5 %.

The analyses and summaries of the results in Tables 6 and 7 show that in terms of small sample datasets, the DC-CMEDA method proposed in this paper has a more efficient detection effect than other most advanced methods. Take the precision, recall, and the harmonic mean F1 of the former two into consideration, the proposed DC-CMEDA method performs the best.

Performance	Filonenko's Method [3]	DC-ILSTM [2]	DC-CMEDA
Accuracy (%)	83.0	85.5	89.5
Precision (%)	82.5	86.2	88.1
Recall (%)	82.5	83.5	90.8
F1 (%)	82.5	84.8	89.4

Table 7. Accuracy test results transfered from video images to satellite remote sensing images

5.3.4 Experiment 4

Under the condition of applying video images and satellite remote sensing smoke images small datasets to DC-CMEDA, the convergence accuracy and convergence speed of the DC-CMEDA-based algorithms are contrasted after combining the CMEDA module with module Resnet34 and module Resnet50, respectively. The experimental results are shown in Figure 7. The red line chart in Figure 7 shows the accuracy of each iteration in the process of learning the video image sample dataset and predicting the satellite remote sensing sample dataset. The blue line chart shows the accuracy of each iteration in the process of learning the satellite remote sensing sample dataset and predicting the video image sample dataset.

The experimental results show that (1) the convergence accuracy and convergence speed of the DC-CMEDA algorithm combined with Resnet50 module are better than those of the DC-CMEDA algorithm combined with Resnet34 module. Taking convergence speed and small sample datasets into consideration, this paper does not discuss cases combined with other Resnet size except Resnet34 and Resnet50. (2) The domain adaptive algorithm in the DC-CMEDA has converged once the number of iterations is less than 10 times. It is clear that under the same accuracy condition, the number of CMEDA iterations is much less than 150 times of the ILSTM algorithm, an example as in deep convolution integration [2], and the algorithm complexity of CMEDA is lower than that of ILSTM. The convergence speed of DC-CMEDA is much higher than that of other deep convolution integrated smoke detection methods [2, 3].

6 CONCLUSION

Aiming at solving the problem of small sample datasets in certain scenes in the forest fire smoke detection, this paper proposes a DC-CMEDA model. This model not only performs feature extraction on a small sample dataset of forest fires on a deep transfer learning architecture, it also proposes a smoke feature registration in combination with the CMEDA module. In the experiments, the model was evaluated based on satellite remote sensing image and video image datasets, and compared with various state-of-the-art forest fire smoke detection methods. The results show that, in terms of detection performance, the model discussed in this paper detects smoke faster and meanwhile, the detection accuracy is higher than other methods.

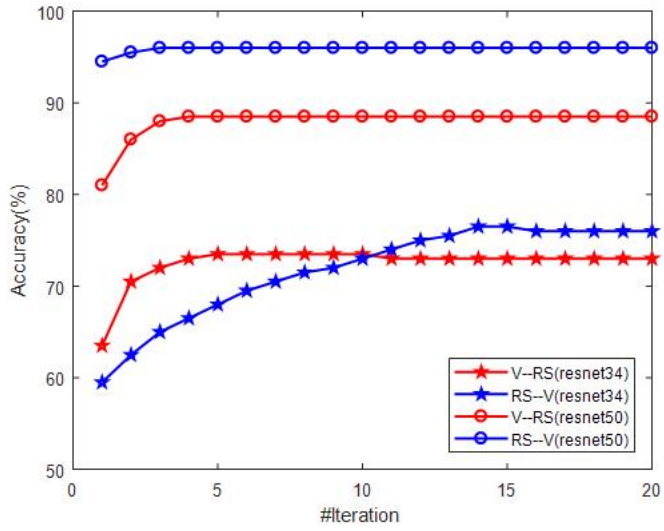


Figure 7. Convergence rate comparison

Among the various methods of smoke detection, the use of domain adaptive methods has not been reported. This paper first attempts to combine domain adaptation with deep transfer learning to solve the smoke detection problem. In the next stage of our work, we need to further optimize the model to improve the accuracy of forest fire smoke detection.

Acknowledgements

This study is funded by Joint Research Fund for Overseas Chinese Scholars and Scholars in Hong Kong and Macao (Grant No. 61828601), Natural Science Foundation of Shanxi Province (Grant No. 201801D121141), and Provincial Program on Key Research Projects of Shanxi (Social Development Area, Grant No. 201903D321003).

REFERENCES

[1] Wang, W.—Mao, W.—He, J.—Dou, Z.: Smoke Recognition Based on Deep Transfer Learning. Journal of Computer Applications, Vol. 37, 2017, No. 11, pp. 3176–3181.

[2] Wei, X.—Wu, S.—Wang, Y.: Forest Fire Smoke Detection Model Based on Deep Convolution Long Short-Term Memory Network. Journal of Computer Applications, Vol. 39, 2019, No. 10, pp. 2883–2887.

[3] Filonenko, A.—Kurnianggoro, L.—Jo, K. H.: Smoke Detection on Video Sequences Using Convolutional and Recurrent Neural Networks. In: Nguyen, N., Papadopoulos, G., Jędrzejowicz, P., Trawiński, B., Vossen, G. (Eds.): Computational Collective Intelligence (ICCCI 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10449, 2017, pp. 558–566, doi: 10.1007/978-3-319-67077-5_54.

[4] Han, C.—Ma, J.—Wu, W.—Chen, J.: Smoke Image Detection Based on Deep Transfer Learning. Journal of Wuhan Textile University, Vol. 32, 2019, No. 2, pp. 65–71.

[5] Pan, S. J.—Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, 2010, No. 10, pp. 1345–1359, doi: 10.1109/TKDE.2009.191.

[6] Wang, J. et al.: Everything about Transfer Learning and Domain Adapation. Available at: `http://transferlearning.xyz`, 2018.

[7] Satpal, S.—Sarawagi, S.: Domain Adaptation of Conditional Probability Models via Feature Subsetting. In: Kok, J. N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (Eds.): Knowledge Discovery in Databases: PKDD 2007. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4702, 2017, pp. 224–235, doi: 10.1007/978-3-540-74976-9_23.

[8] Gong, M.—Zhang, K.—Liu, T.—Tao, D.—Glymour, C.—Schölkopf, B.: Domain Adaptation with Conditional Transferable Components. Proceedings of the 33rd International Conference on Machine Learning (ICML'16), 2016, Vol. 48, pp. 2839–2848.

[9] Pan, S. J.—Tsang, I. W.—Kwok, J. T.—Yang, Q.: Domain Adaptation via Transfer Component Analysis. IEEE Transactions on Neural Networks, Vol. 22, 2011, No. 2, pp. 199–210, doi: 10.1109/TNN.2010.2091281.

[10] Dorri, F.—Ghodsi, A.: Adapting Component Analysis. IEEE 12th International Conference on Data Mining (ICDM 2012), 2012, pp. 846–851, doi: 10.1109/ICDM.2012.85.

[11] Lan, Z.—Sourina, O.—Wang, L. P.—Scherer, R.—Müller-Putz, G. R.: Domain Adaptation Techniques for EEG-Based Emotion Recognition: A Comparative Study on Two Public Datasets. IEEE Transactions on Cognitive and Developmental Systems, Vol. 11, 2019, No. 1, pp. 85–94, doi: 10.1109/TCDS.2018.2826840.

[12] Duan, L.—Tsang, I. W.—Xu, D.: Domain Transfer Multiple Kernel Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, 2012, No. 3, pp. 465–479, doi: 10.1109/TPAMI.2011.114.

[13] Baktashmotlagh, M.—Harandi, M.—Salzmann, M.: Distribution-Matching Embedding for Visual Domain Adaptation. The Journal of Machine Learning Research, Vol. 17, 2016, No. 1, pp. 3760–3789.

[14] Zellinger, W.—Lughofer, E.—Saminger-Platz, S.—Grubinger, T.—Natschläger, T.: Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. International Conference on Learning Representations (ICLR 2017), 2017, arXiv:1702.08811.

[15] Long, M.—Wang, J.—Ding, G.—Sun, J.—Yu, P. S.: Transfer Feature Learning with Joint Distribution Adaptation. 2013 IEEE International Conference on Computer Vision (ICCV 2013), 2013, pp. 2200–2207, doi: 10.1109/ICCV.2013.274.

[16] Wang, J.—Chen, Y.—Hao, S.—Feng, W.—Shen, Z.: Balanced Distribution Adaptation for Transfer Learning. 2017 IEEE International Conference on Data Mining (ICDM 2017), 2017, pp. 1129–1134, doi: 10.1109/ICDM.2017.150.

[17] Gopalan, R.—Li, R.—Chellappa, R.: Domain Adaptation for Object Recognition: An Unsupervised Approach. IEEE International Conference on Computer Vision (ICCV 2011), 2011, pp. 999–1006, doi: 10.1109/ICCV.2011.6126344.

[18] Fernando, B.—Habrard, A.—Sebban, M.—Tuytelaars, T.: Unsupervised Visual Domain Adaptation Using Subspace Alignment. Proceedings of the IEEE International Conference on Computer Vision (ICCV 2013), 2013, pp. 2960–2967, doi: 10.1109/ICCV.2013.368.

[19] Sun, B.—Saenko, K.: Subspace Distribution Alignment for Unsupervised Domain Adaptation. 26th British Machine Vision Conference (BMVC 2015), 2015, Art. No. 24, 10 pp., doi: 10.5244/C.29.24.

[20] Sun, B.—Feng, J.—Saenko, K.: Return of Frustratingly Easy Domain Adaptation. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016), 2016, pp. 2059–2065.

[21] Ghifary, M.—Balduzzi, D.—Kleijn, W. B.—Zhang, M.: Scatter Component Analysis: A Unifed Framework for Domain Adaptation and Domain Generalization. IEEE Transactions on Patern Analysis and Machine Intelligence, Vol. 39, 2017, No. 7, pp. 1414–1430, doi: 10.1109/TPAMI.2016.2599532.

[22] Gong, B.—Shi, Y.—Sha, F.—Grauman, K.: Geodesic Flow Kernel for Unsupervised Domain Adaptation. IEEE Conference on Computer Vision and Patern Recognition (CVPR 2012), 2012, pp. 2066–2073, doi: 10.1109/CVPR.2012.6247911.

[23] Baktashmotlagh, M.—Harandi, M. T.—Lovell, B. C.—Salzmann, M.: Unsupervised Domain Adaptation by Domain Invariant Projection. Proceedings of the IEEE International Conference on Computer Vision (ICCV 2013), 2013, pp. 769–776, doi: 10.1109/ICCV.2013.100.

[24] Baktashmotlagh, M.—Harandi, M. T.—Lovell, B. C.—Salzmann, M.: Domain Adaptation on the Statistical Manifold. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (ICCV 2014), 2014, pp. 2481–2488, doi: 10.1109/CVPR.2014.318.

[25] Wang, J.—Feng, W.—Chen, Y.—Yu, H.—Huang, M.—Yu, P. S.: Visual Domain Adaptation with Manifold Embedded Distribution Alignment. Proceedings of the 26th ACM International Conference on Multimedia (MM 2018), Seoul, Republic of Korea, 2018, pp. 402–410, doi: 10.1145/3240508.3240512.

[26] Huang, Z.—Li, M.—Chousidis, C.—Mousavi, A.—Jiang, C.: Schema Theory-Based Data Engineering in Gene Expression Programming for Big Data Analytics. IEEE Transactions on Evolutionary Computation, Vol. 22, 2018, No. 5, pp. 792–804, doi: 10.1109/TEVC.2017.2771445.

[27] Wang, L. P.—Wang, Y.—Chang, Q.: Feature Selection Methods for Big Data Bioinformatics: A Survey from the Search Perspective. Methods, Vol. 111, 2016, pp. 21–31, doi: 10.1016/j.ymeth.2016.08.014.

Yaoli Wang received his B.S. and M.Sc. degrees from the Northwestern Polytechnical University, China and his Ph.D. degree from the Taiyuan University of Technology, China. His research interests include intelligent techniques with applications to communications, image/video processing, robots. He is (co-)author of over 40 papers. He holds two Chinese patents. He is Member of the Expert Committee on Embedded Systems of the China Electronic Society, Member of the Seventh Forest and Grassland Fire Prevention Committee of the China Forestry Society, and Member of the Seventh Forest Fire Control Branch of the China Fire Control Association.

Xiaohui Liu received her B.Sc. degree from the Taiyuan University of Science and Technology, China. Her research interest is intelligent techniques with applications to robots. She is now studying for her M.Sc. degree at the Taiyuan University of Technology.

Maozhen Li is Professor in the Department of Electronic and Computer Engineering, Brunel University London, UK. He is also a Visiting Professor of Tongji University, Shanghai, China. He received the Ph.D. from the Institute of Software, Chinese Academy of Sciences, in 1997. His main research interests include high performance computing, big data analytics and intelligent systems with applications to smart grid, smart manufacturing and smart cities. He has over 180 research publications in these areas including 4 books. He has served over 30 IEEE conferences and is on the editorial board of a number of journals. He is a Fellow of the British Computer Society (BCS) and the Institute of Engineering and Technology (IET).

Wenxia Di received her B.Sc. degree from the Inner Mongolian Normal University, China. Her research interests include intelligent techniques with applications to speech signal processing. She is (co-)author of over 10 papers. She has co-authored 2 books. She is Member of English Voice Association.

Lipo Wang received his B.Sc. degree from the National University of Defense Technology, China, and Ph.D. from the Louisiana State University, USA. His research interests include intelligent techniques with applications to communications, image/video processing, biomedical engineering, and data mining. He is (co-)author of over 270 papers, of which more than 90 are in journals. He holds a U.S. patent in neural networks and a Chinese patent in VLSI. He has co-authored 2 monographs and (co-)edited 15 books. He was/will be a keynote/panel speaker for 21 international conferences. He is/was Associate Editor/Editorial Board Member of 30 international journals, including 3 IEEE Transactions, and the guest editor for 10 journal special issues. He is an AdCom Member (2010–2015) of the IEEE Computational Intelligence Society (CIS) and served as CIS Vice President for Technical Activities and Chair of Emergent Technologies Technical Committee. He is Member of the Board of Governors of the International Neural Network Society (2011–2016) and was an AdCom member of the IEEE Biometrics Council. He served as Chair of Education Committee, IEEE Engineering in Medicine and Biology Society (EMBS). He was President of the Asia-Pacific Neural Network Assembly (APNNA) and received the APNNA Excellent Service Award. He was founding Chair of both the EMBS Singapore Chapter and CIS Singapore Chapter. He serves/served as chair/committee member of over 200 international conferences.

Computing and Informatics, Vol. 39, 2020, 340–360, doi: 10.31577/cai_2020_1-2_340

AN IMPROVED PDR LOCALIZATION ALGORITHM BASED ON PARTICLE FILTER

Wei Wang*, Cunhua Wang, Zhaoba Wang, Xiaoqian Zhao

School of Information Science and Engineering
North University of China, Taiyuan, 030051, China
e-mail: 41695559@qq.com

Abstract. Pedestrian Dead Reckoning (PDR) helps to realize step frequency detection, step estimation and direction estimation through data collected by inertial sensors such as accelerometer, gyroscope, magnetometer, etc. The initial positioning information is used to calculate the position of pedestrians at any time, which can be applied to indoor positioning technology researching. In order to improve the position accuracy of pedestrian track estimation, this paper improves the step frequency detection, step size estimation and direction detection in PDR, and proposes a particle swarm optimization particle filter (PSO-IPF) PDR location algorithm. Using the built-in accelerometer information of the smartphone to carry out the step frequency detection, the step frequency parameter construction model is introduced to carry out the step estimation, the direction estimation is performed by the Kalman filter fusion gyroscope and the magnetometer information, and the positioning data is merged by using the particle filter. The fitness function in the particle swarm optimization process is changed in the localization algorithm to improve particle diversity and position estimation. The experimental results show that the error rate of the improved step frequency detection method is reduced by about 2.1 % compared with the traditional method. The angle accuracy of the direction estimation is about 4.12° higher than the traditional method. The overall positioning accuracy is improved.

Keywords: Indoor positioning, PDR, particle filtering, particle swarm optimization, data fusion

Mathematics Subject Classification 2010: 68W40

* Corresponding author

1 INTRODUCTION

In today's world, information technology is developing rapidly. With the in-depth development of the Internet, technologies such as the Internet and 5G are gradually maturing, and location-based services will play an increasingly important role. As an application based on location services, indoor positioning gradually penetrates into all aspects of social life. Currently the Global Positioning Systems (GPS) and the BeiDou Navigation Satellite System (BDS) are widely used in the United States. These systems have been able to provide users with higher-precision outdoor positioning, such as on-board map navigation, which makes people's needs for outdoor activities met. However, due to a small spatial pattern of the interior, the indoor positioning has stricter requirements on accuracy. As for indoors signals, outside signals are easily blocked, attenuated or reflected. Therefore, some problems exist in the outdoor positioning system, such as insufficient accuracy and signal instability in indoor positioning, resulting in relatively poor reliability, continuity and stability of indoor positioning. It is important to develop indoor positioning technology to provide high-precision, convenience and mature indoor location services [1].

Currently, as smartphones are widely embedded with magnetometers, accelerometers, gyroscopes and other sensors, it is possible for smartphone-based pedestrian dead-reckoning (PDR) to adapt smartphones. PDR technology is mainly divided into three parts: step frequency detection, step estimation and direction estimation [2]. Poulose et al. [3] proposed a step frequency detection by combining the acceleration information and the gyro information. Wang [4] proposed a dynamic constrained gait detection method, which removed the interference caused by jitter, through using dynamic constraint amplitude and accelerating peak time. In step estimation, it often combined people's walking frequency and experience value to estimate. Manos et al. [5] estimated the direction angle by establishing a model of gravity-like direction. Kang et al. [6] established indoor positioning for pedestrians by establishing an inseparable walking mode and real-time deep learning network module. Lu et al. [7] proposed a new regression model using accelerating data to perform stride frequency detection. At the same time, it combined with map information and barometer for spatial three-dimensional positioning. Hasan and Mishuk [8] used Kalman filtering to fuse acceleration, gyroscope and magnetometer data for indoor positioning. Since most of the current smartphones have built-in sensors such as accelerometers, gyroscopes and magnetometers, and the accuracy is also higher and higher, it is possible to use the smartphone for PDR indoor positioning.

In this paper, the accelerometer and gyroscope data are filtered and pre-processed by FIR low-pass filter. Firstly, in step frequency detection, setting the time threshold, the acceleration threshold and the change of the state value respectively. Secondly, in step estimation, the steps are counted, and the step frequency parameters are introduced to estimate the step size to improve the accuracy of the estimation. Then the magnetometer and gyroscope data are fused by Kalman filter to estimate the direction. Finally, the particle filter is improved by changing the

fitness function in the particle swarm optimization algorithm to improve particle diversity and the accuracy of indoor positioning.

The rest of the paper is organized as follows. Section 2 brings forward a few related works. Section 3 focuses on improved PDR algorithm, introduces the algorithm for step frequency detection, step size estimation and direction angle prediction as well as a new particle swarm optimization particle filter algorithm. Section 4 mainly verifies the advantages of the proposed algorithm in indoor positioning, which can improve the positioning accuracy, and Section 5 concludes the paper.

2 PEDESTRIAN MOBILE SENSING TECHNOLOGY BASED ON INERTIAL SENSOR

The most important technique for pedestrian motion perception in indoor positioning is track estimation. In indoor positioning, track estimation can be evolved into pedestrian trajectory estimation. Pedestrian dead reckoning mainly consists of three important parts: step detection, step estimation and direction estimation. The basis of the stride detection is that the pedestrian motion has periodic characteristics. The cycle of each step of the movement is from the beginning of a step to the end of a step. The output value of the acceleration sensor can visually see the waveform of the motion cycle. The fluctuation of the acceleration value is generally related to the height of the person and the individual, the exercise habits and road conditions. The step frequency of pedestrian movement can be obtained by analyzing the acceleration values.

The purpose of the stride detection is to detect whether a pedestrian has walked. When a person is walking with a smartphone, the horizontal acceleration and the vertical acceleration will exhibit periodic changes. Therefore, the pedestrian motion can be detected by periodically changing the acceleration in the walking motion. Figure 1 is an acceleration signal collected when the pedestrian is holding the smartphone while walking.

It can be seen from the figure that the acceleration in the Z direction is relatively obvious, the periodicity is better, the acceleration in the X and Y directions is weaker, and the change is not as obvious as the Z direction. Since the three-axis output component of the smartphone accelerometer is related to the attitude of the smartphone itself, in the actual situation, the hand-held smartphone has a certain randomness, and the result obtained directly by using the Z-axis output in some severe scenes will appear larger error. Therefore, in order to eliminate the acceleration signal fluctuation caused by the gesture of the smartphone itself, the three-axis output component of the acceleration information is modulo obtained to obtain the combined amount Zsyn.

$$Zsyn = \sqrt{a_x^2 + a_y^2 + a_z^2}. \tag{1}$$

a_x, a_y, a_z are the three-axis output components of the smartphone. Figure 2 is a comparison diagram of changes in acceleration information components and composite amounts while a person is walking with a mobile phone in his/her pocket.

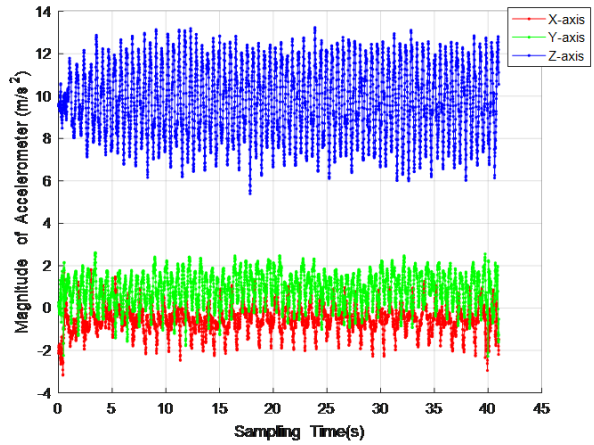


Figure 1. Change in acceleration value when walking

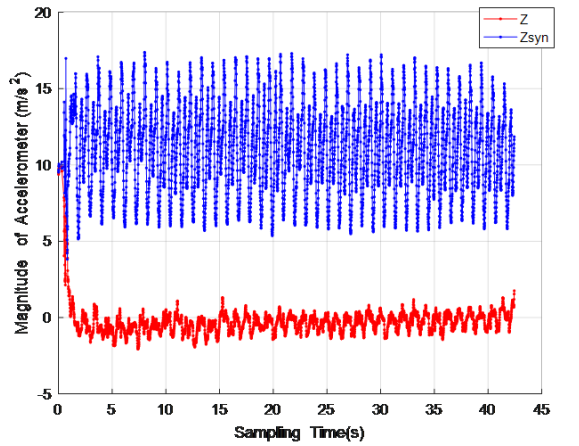


Figure 2. Comparison of dynamic Z component and composite quantity

It can be seen from Figure 2 that the Z-axis acceleration information fluctuation is not obvious when the smartphone is placed in the pocket. If the Z-axis acceleration information is also used to detect the pace, a large error will be caused, but the synthesis amount has a good periodicity. Therefore, the raw data of the acceleration information is used as the step to detect the original data. Moreover, the faster the walking speed from Figure 3, the larger the peak value and the valley value of the waveform.

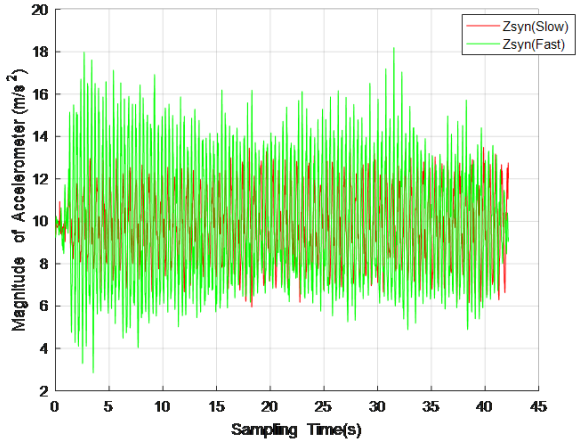


Figure 3. Acceleration waveform at different walking speeds

Considering that the human body is relatively fluctuating when walking, the collected acceleration information is mixed with random noise, so the acquired acceleration signal cannot be directly used, and it is necessary to perform related filtering processing to filter out large accidental noise and false peaks to obtain more obvious wave forms.

3 PDR ALGORITHM

3.1 Step Frequency Detection

The acceleration signal collected by the smartphone has high-frequency information interference, and it is particularly important to use the filter for filtering. Because the FIR filter has the advantages of good stability, high precision, small accumulation error, linear phase characteristics, etc., the acceleration signal is not easy to be distorted, so this paper selects the FIR filter to pre-filter the acceleration data.

The method of the digital filter design includes a window function method, a frequency sampling design method, and a multi-filter parallel processing method. In this paper, the window function method is selected to complete the filter design. The design idea is to choose an ideal frequency selection filter, which has an infinite impulse response and uses a suitable window function to cut off its impulse response to obtain a linear phase. The RF filter of the excited response is designed as follows:

$H_d(e^{jw})$ represents an ideal low-pass filter and the ideal bandwidth filter is given by Formula (2):

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega}, & |\omega| \le \omega_c, \\ 0, & \omega_c < |\omega| \le \pi. \end{cases} \tag{2}$$

In this formula, ω_c is the cutoff frequency, and the impulse response is:

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{jw})\,\mathrm{d}\omega = \frac{\sin[\omega_c(n-\alpha)]}{(n-\alpha)\pi}. \tag{3}$$

In the formula: $\alpha = \frac{M-1}{2}$, in order to obtain an FIR filter from the ideal filter, it must be truncated by a windowing function to obtain a linear phase filter whose length is M:

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le M-1, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The windowing function shows that $h(n)$ can be seen as the result of $h_d(n)$ multiplying by a window function:

$$\begin{aligned} h(n) &= h_d(n)\omega(n), \\ \omega(n) &= \begin{cases} 0 \le n \le M-1, & \text{on the function of } \alpha\text{-symmetry,} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

$h(n)$ is the filter required for the filter implementation. The filter order is M. At the same time, compared with the filters designed by different window functions, the Kaiser window not only has the lowest order, but also has a flat and minimum attenuation, so the Kaiser window is selected for design. After filtering and pre-processing the acceleration signal, the steps are counted by setting the time threshold, the acceleration threshold and the change of the state value. The specific algorithm steps are as follows:

- Set the status value to 0.
- Read the acceleration data to determine whether the time threshold is met, and if so, continue the subsequent judgment, if not, continue to read the next acceleration data.
- Process the acceleration data to obtain a target value, determine whether the target value is greater than the acceleration threshold, and if greater than the threshold, set the state value to 1, continue the third step of judgment, otherwise return to the first step.
- Determine whether the target value is greater than the current peak value. If it is greater, set the value to the new peak value, and then repeat the process until the maximum value is found, otherwise proceed to the next step.
- Determine whether the target value is less than zero. If it is less than 0, set the status value to 2, otherwise return to the first step.
- Determine whether the target value is less than the current minimum value, if it is less, set the value to the new minimum value, and then repeat the process until the minimum value is found, otherwise proceed to the next determination.

- Determine whether the current target value is greater than 0. If it is greater than 0, it means that the step counting process is completed, otherwise return to the first step.

3.2 Step Estimate

Since different people's step sizes are different, individual differences can be reflected in the walking frequency of pedestrians. The model calculation formula is as follows:

$$stepLen_k = (1-\eta)\left(0.56\sqrt[2]{ACC_{pv}} + 0.11b(ACC_{pv})\right) + \eta(af^2 + bf + c). \quad (6)$$

In Formula (6), ACC_{pv} is the difference between the maximum and minimum values of acceleration in each step cycle; a, b and c are the step control parameters of the acceleration model, respectively, taken as 0.3, 0.2 and 0.1, η is step frequency weighting factors, taking 0.4, f is steps frequency. The selection of the experimental data is based on the step size of a large number of different people for statistical analysis, and then calculate the average.

3.3 Direction Angle Estimation

Since the carrier coordinate system of the mobile phone is different from the navigation coordinate system used in the actual positioning process, the sensor data is transformed between the two coordinate systems. The acceleration of the mobile phone and the sensor data acquired by the gyroscope are all for the carrier coordinate system. During the movement, the position change of the user is for the navigation coordinate system. The navigation coordinate system used in this paper takes the carrier centroid as the origin and the X axis along the West to the East, the Y axis is northward along the meridian, and they are all in the local horizontal plane. The Z coordinate axis is vertically upward along the local geographic vertical line, that is, the east-north-up geographic coordinate system. The coordinate system transformation can be realized by rotating the matrix. In this paper, the quaternion method is used for conversion. The gyro-based incremental rotation matrix based on the quaternion is shown in Formula (7):

$$\Delta R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (7)$$

q_i $(i = 1, 2, 3, 4)$ corresponds to the four sub-elements in the quaternion and can be solved by Formula (8):

$$\begin{aligned} Q &= [q_0, q_1, q_2, q_3] \\ &= \left[\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\left(\frac{\omega_x}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}}, \frac{\omega_y}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}}, \frac{\omega_z}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}}\right)\right]. \end{aligned} \tag{8}$$

In Formula (8), $\omega_x, \omega_y, \omega_z$ are the output components of the smartphone in the carrier coordinates, θ is the angular velocity mode, $|\omega|$ is the angular velocity increment obtained by integrating the time. Since the gyroscope cannot obtain the initial direction, it is possible to obtain an initial rotation matrix Rc by the smartphone at rest and then multiply the incremental rotation matrix of the gyroscope with the previous matrix to obtain the rotation matrix of the gyroscope. The initial rotation matrix Rc can be obtained by the roll angle φ, pitch angle ϕ and heading angle θ at res.

Assume that the accelerometer measures the three-axis component of the acceleration: $a^b = [a_x, a_y, a_z]$ in the carrier coordinates. Under the navigation coordinate system, it is: $a^n = [a_E, a_N, a_U]$, we can get:

$$a^b = R_c a^n. \tag{9}$$

In the static state, $a^n = [0, 0, g]^T$, in which the local gravity acceleration is taken as 9.8. According to the attitude angle conversion formula, the formula can be obtained:

$$\theta = \arctan\left(\frac{M_x\cos\phi + M_z\sin\phi}{(M_x\sin\phi + M_z\cos\phi)\sin\varphi + M_y\cos\varphi}\right), \tag{10}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} -g\sin\varphi\cos\phi \\ g\sin\phi \\ g\cos\varphi\cos\phi \end{bmatrix}. \tag{11}$$

According to Formula (10), the heading angle can be obtained, and the roll angle and the pitch angle can be obtained according to Formula (11). Then solve the four variables of the quaternion by the relationship between the quaternion and the attitude angle:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\gamma}{2}\cos\frac{\theta}{2}\cos\frac{\varphi}{2} + \sin\frac{\gamma}{2}\sin\frac{\theta}{2}\sin\frac{\varphi}{2} \\ \sin\frac{\gamma}{2}\cos\frac{\theta}{2}\cos\frac{\varphi}{2} - \cos\frac{\gamma}{2}\sin\frac{\theta}{2}\sin\frac{\varphi}{2} \\ \cos\frac{\gamma}{2}\sin\frac{\theta}{2}\cos\frac{\varphi}{2} + \sin\frac{\gamma}{2}\cos\frac{\theta}{2}\sin\frac{\varphi}{2} \\ \cos\frac{\gamma}{2}\cos\frac{\theta}{2}\sin\frac{\varphi}{2} - \sin\frac{\gamma}{2}\sin\frac{\theta}{2}\sin\frac{\varphi}{2} \end{bmatrix}. \tag{12}$$

After the initial rotation matrix is calculated, the incremental rotation matrix

updated by the gyroscope is continuously multiplied by the previous rotation matrix to obtain the updated gyroscope rotation matrix.

$$R_g^k = R_c \prod_{i=0}^{k} \Delta R_g^k, \tag{13}$$

$$\begin{cases} \theta = \arctan\left(\frac{T(2,2)}{T(1,2)}\right), \\ \varphi = \arcsin\left(-\frac{T(3,1)}{1}\right), \\ \phi = \arctan\left(-\frac{T(3,3)}{T(3,1)}\right). \end{cases} \tag{14}$$

ΔR_g^k is the gyroscope incremental rotation matrix from the $k-1$ to the k sampling; R_g^k is the gyroscope rotation matrix incremented for the k sampling. After obtaining the rotation matrix of the gyroscope R_g^k, the posture information in the navigation coordinate system can be solved. $T(i,j)$ represents the i row and j column elements in the rotation matrix.

Considering the complementary advantages of gyroscope and magnetometer information, Kalman filter can be used to fuse the two kinds of information. The specific fusion process is: assuming that θ_k is the estimated direction of the pedestrian at time k; $\Delta\theta_k$ is the amount of change of the gyroscope at the time $k-1$ to k, then the system state equation is:

$$S_k = AS_{k-1} + B\Delta S_k + C_k. \tag{15}$$

In the formula, $S_k = [\theta_k]$ is the system state, $A = B = [1]$ is the system parameter; C_k is the system process noise, and the system observation variable D_k is the magnetometer output, so the observation equation is:

$$D_k = HS_k + \xi_k. \tag{16}$$

$H = [1]$ is the system parameter and ξ_k is the noise the system observes, so the Kalman filter prediction and update process is as follows:

Observation process:

$$\begin{cases} S_{k|k-1} = AS_{k-1|k-1} + B\Delta S_k, \\ P_{k|k-1} = AP_{k-1|k-1}A^T + B\Delta\theta_k + C^k. \end{cases} \tag{17}$$

Update process:

$$\begin{cases} K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + \xi^k)^{-1}, \\ S_{k|k} = S_{k|k-1} + K_kD_k - K_kHS_{k|k-1}, \\ P_{k|k} = P_{k|k-1} - K_kHP_{k|k-1}. \end{cases} \tag{18}$$

P_k is the system co-variance matrix; K_k is the Kalman filter gain.

3.4 (PSO-PF) Particle Swarm Optimization Particle Filter

Traditional particle swarm optimization particle filter algorithm (PSO-PF) in the search field to fitness function as evaluation standard updates the size of the particle state change quantity, and then adjusts the spatial distribution of particles, the particle state concentrated in the fitness function extremum near. Filtering algorithm introduces the latest measurement values to the sampling distribution, makes particles move backward probabilistic higher area, and in the process of optimization iterations, because each particle state change amount of random variation, it ensures the diversity of particles, thus improves the particle degradation problems of particle filter. However, the fitness function used by this algorithm which selects the optimal particle is based on the difference between the current measured value and the predicted value of measured value as the evaluation standard. When the noise variance becomes high, it directly affects the selection of the optimal particle, resulting in a significant decrease in the filtering performance with the increase of noise variance.

The PSO-IPF algorithm proposes a new fitness function, which takes the difference between each particle state and state estimation as the evaluation standard, weakens the influence of random measurement noise and improves the filtering accuracy of the algorithm.

General dynamic time-varying systems can be described as (19) and (20):

$$x_k = f(x_{k-1}) + v_k \sim p(x_k|x_{k-1}), \tag{19}$$

$$z_k = h(x_k) + w_k \sim p(z_k|x_k). \tag{20}$$

$\mathrm{X_k} \in \mathrm{R^n}$ is the n-dimensional state measurement vector of the system at the time of k; $\mathrm{Z_k} \in \mathrm{R^1}$ is the l-dimensional measurement of the system at the time of k, $\mathrm{V_k} \in \mathrm{R^a}$, $\mathrm{W_k} \in \mathrm{R^1}$ are process noise and measurement noise, respectively. The algorithm takes the particle prior probability as the importance function, calculates the corresponding particle weight by extracting the particle sample, and then obtains the weight to normalize, and finally obtains the weighted particle set. After resampling, the particle set is $\{x_k^i, 1/N\}_{i=1}^N$.

In the optimization process, the particle state set is taken as the initial state of the particle swarm optimization, $x_p^i(m)$ and $x_g(m)$ are the individual optimal value and the local optimal value of the m^{th} iterative particle state, respectively, and finally according to the particle fitness function (21) to choose. In the formula, σ^2 is the measuring noise variance, and $z_{pre} = h(x_k^i(m-1))$ is the predicted value.

$$s_k^i(m-1) = \exp\left[-\frac{1}{2\sigma^2}(z_k - z_{pre})^2\right]. \tag{21}$$

In the m times iteration, if the value of the n^{th} particle fitness function is larger,

then the current state of the particle is updated to its individual optimal value, otherwise the result of the last iteration is retained. Then, the model is updated according to the state variable, and the particle state change is calculated as follows:

$$v_x^i(m) = \omega v_x^i(m-1) + \varphi_1(x_p^i(m-1) - x_k^i(m-1)) + \varphi_2(x_g(m-1) - x_k^i(m-1)),$$
$$m = 1, 2, \ldots, D. \tag{22}$$

In Formula (22), ω is the inertia weight factor, $V_x^i(m)$ is the uniform distribution of [0,1], $\varphi_1 : \varphi_2$ is the random number of [0,1], and D is the total number of iterations. The updated particle state value is:

$$x_k^i(m) = x_k^i(m-1) + v_x^i(m). \tag{23}$$

Although this algorithm improves the particle degradation problem and maintains the particle diversity and improves the filtering accuracy, the fitness function is based on the difference between the current measured value and the measured predicted value as the evaluation standard, which is largely affected by the measurement noise, causing lower filtering accuracy. Therefore, the filter value (state estimation value) which is less affected by the noise variance is selected as the reference value of the selected particle, and the improved fitness function expression is in Formula (24).

$$s_k^i(m) = \exp\left(-\frac{1}{C}\left(x_k^i(m) - \overset{\wedge}{\underset{k}{x}}\right)^2\right). \tag{24}$$

In Formula (24), $x_k^i(m)$ is the state value of the m^{th} iteration of the i^{th} particle; $x_k^\wedge$ is the state estimation value of the pre-optimization particle filter, and C is the constant selected according to the convergence of the iteration.

The algorithm steps are as follows:

Step 1: Importance sampling process. According to the system model, obtain particles $\{x_k^i \mid x_k^i \sim p(x_k^i|x_{k-1}^i)\}$, and then obtain the normalized weight of the particle filter part of the particle, and finally get the weighted particle set $\{x_k^i, w_k^i\}_{i=1}^N$.

Step 2: The initial state estimation process. The state estimation of the particle filtering part is performed according to Formula (25), the initial filtering result $x_k^\wedge$ is obtained, and the estimation result is taken as an important parameter of the fitness function in the optimization step.

$$\overset{\wedge}{\underset{k}{x}} = \sum_{i=1}^{N} w_k^i x_k^i. \tag{25}$$

Step 3: PSO processing is divided into three steps:

1. Initialize the population, and use the particles obtained by the current particle filtering as the initial population $\{x_k^i(0), w_k^i\}_{i=1}^N = \{x_k^i, w_k^i\}_{i=1}^N$.

2. Calculate the fitness function and update the individual optimal value and the global optimal value by calculating the fitness function of each particle.
   - If $s_k^i(m) \geq s_k^i(m-1)$, then $x_p^i(m) = x_k^i(m)$.
   - If $s_k^i < s_k^i(m-1)$, then $x_p^i(m) = x_p^i(m-1)$. Find the maximum particle number $I(m)$ in the $s_k^i(m)$.
   - If $s_k^{I(m)}(m) \geq s_k^{I(m-1)}(m-1)$, then $x_g(m) = x_k^{I(m)}(m)$. If $s_k^{I(m)}(m) < s_k^{I(m-1)}(m-1)$, then $x_g(m) = x_g(m-1)$.
3. Adjust the state of the particles, and update the position and velocity of the particles to obtain the state change $v_x^i(m+1)$ of the $(m+1)^{\text{th}}$ iteration of each particle and the state $x_k^i(m+1), i = 1, 2, \ldots, N$.
4. Return to step 2 until the optimization is complete.

Step 4: Output state estimation: Select the final global optimal value of the particle $\widehat{x}_k = x_g(D)$ as the exact filtered output.

In order to verify the filtering effect of the PSO-IPF algorithm proposed in this paper, the nonlinear non-Gaussian system model is filtered by the algorithm, standard PF algorithm and PSO-PF algorithm, respectively.

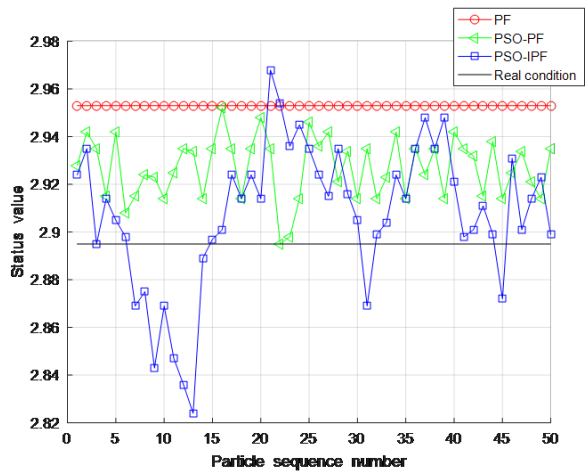


Figure 4. The particle distribution of the three algorithms ($k = 20$)

It can be seen from Figure 4 that at the time of $k = 20$, the particle state of PF is unique, which is due to the serious loss of particle diversity caused by re-sampling, while PSO-PF and PSO-IPF can ensure the diversity of particles. In contrast, PSO-IPF algorithm has the best results.

4 EXPERIMENTAL VERIFICATION

The experimental site was held on the third floor corridor of the Science Building of North University of China. The total length of the route in the corridor is 40 m, and the width of the corridor is 1.6 m. The plan view is shown in Figure 5. The size of the floor tile (0.8 m × 0.8 m) divides the experimental area into 100 grid cells of the same size, and the vertices of each grid cell serve as sampling reference points for a total of 303 reference points.

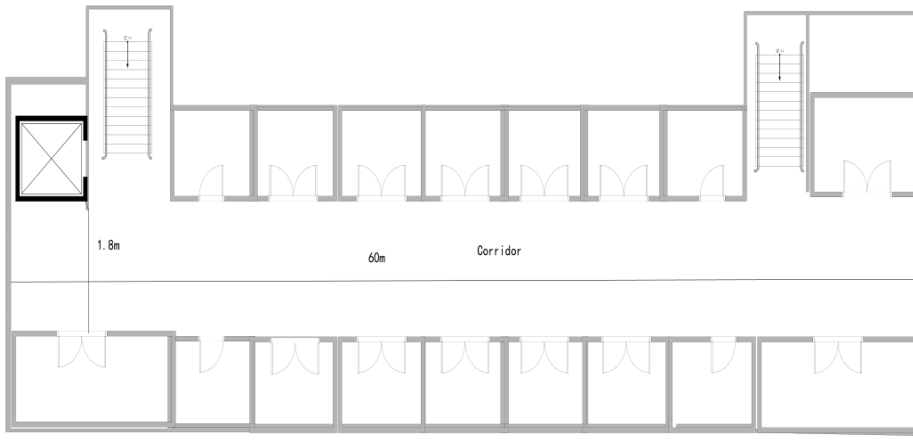


Figure 5. A plan view

The mobile phone used in this article is equipped with a three-axis magnetometer model AK09915 manufactured by AKM. The magnetometer has a measurement accuracy of 0.12856238 and is equipped with sensors such as the LSM6DS3 accelerometer and the LSM6DS3 gyroscope. Set the mark on the ground to record the pedestrian walking process, and use the smartphone to collect pedestrian walking data, collect acceleration, gyroscope and magnetometer data through the mobile phone, and set the sampling frequency to 50Hz; the experimental scene and mobile phone collection information are shown in Figures 6 and 7.

4.1 Step Frequency Test

In order to verify the accuracy of the step detection, the experiment chose to allow three experimenters with different heights and weights to carry the same smartphone to walk in the same experimental environment. Each person made three experiments to record the actual number of steps and walking of the experimenter. Acceleration sensor data in the process, for the reliability of the result, each person's route is different each time. Then after the collected original acceleration data is subjected to FIR low-pass filtering, the step frequency detection method and the traditional peak

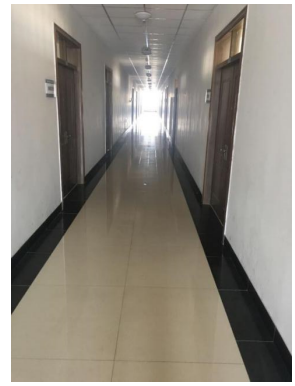


Figure 6. Experimental scene diagram

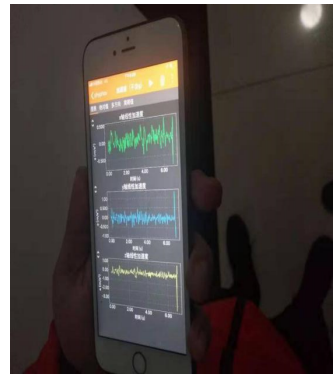


Figure 7. Smartphone data collection site map

detection method of this article are used. The zero detection method is compared, and the experimental results are shown in Table 1.

It can be seen from Table 1 that the accuracy of our algorithm in 9 groups experiments is:

$$1 - \frac{0+2+2+1+3+3+1+2+3}{63+69+80+71+85+79+72+79+83} \times 100\,\% = 97.6\,\%.$$

The accuracy of the peak detection in the experiments is:

$$1 - \frac{4+4+3+4+3+10+4+7+10}{63+69+80+71+85+79+72+79+83} \times 100\,\% = 92.9\,\%.$$

The accuracy of the zero-crossing detection in the experiments is:

$$1 - \frac{3+2+3+3+1+5+3+4+6}{63+69+80+71+85+79+72+79+83} \times 100\,\% = 95.5\,\%.$$

Experiment	Group	Real Step	Calculation Step			Error [Step]		
			Peak Detection	Zero-Cross Detection	Improved Method	Peak Detection	Zero-Cross Detection	Improved Method
Tester 1	1	63	67	66	63	4	3	0
	2	69	73	71	69	4	2	2
	3	80	83	83	81	3	3	2
Tester 2	4	71	75	74	72	4	3	1
	5	85	88	86	85	3	1	3
	6	79	89	84	77	10	5	3
Tester 3	7	72	76	75	72	4	3	1
	8	79	86	83	78	7	4	2
	9	83	93	89	86	10	6	3

Table 1. Analysis of the results of the step detection test

Figure 8 is an acceleration detection waveform diagram using the method herein.

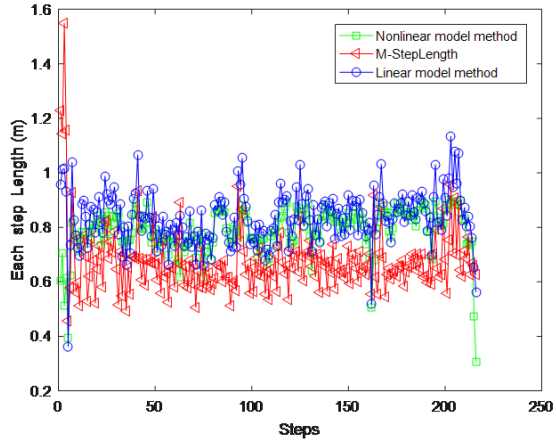


Figure 8. Step detection waveform and step count statistics

4.2 Pedestrian Step Estimation Experiment

In order to verify the accuracy of the step estimation method, experiments were carried out on the corridor, and the data collected by the experiment was compared with the conventional nonlinear model by using the step size and linear model calculated by the method. Figure 9 compares the three different methods of step size calculation, Figure 10 shows the average error of each step (the absolute value of the step difference between each step and the measured average).

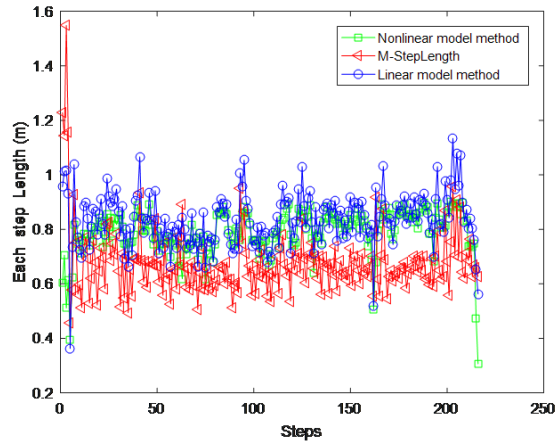


Figure 9. Comparison of three methods of step size calculation

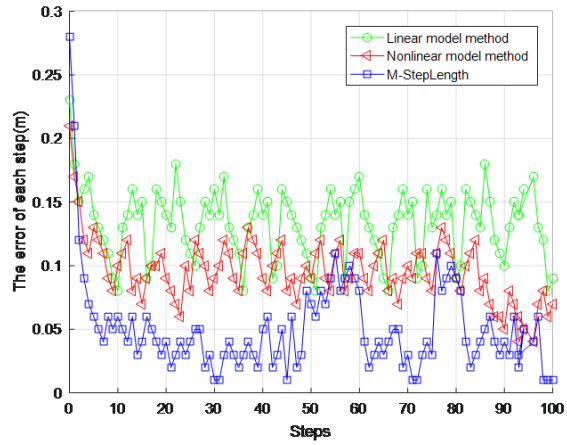


Figure 10. Average error per step

It can be seen from Figure 9 that the step sizes calculated by the method are distributed between 0.5 m and 0.8 m, which is in accordance with the actual value of the walking step of the person. As can be seen from Figure 10, the error of each step of the method is relatively small, and is distributed between 0 m and 0.1 m. Considering this, the method has certain advantages.

4.3 Direction Angle Estimation Experiment

In order to verify the effectiveness of the direction estimation algorithm, a path that changes direction is selected in the laboratory environment to verify the validity of the method estimation. The gyro and magnetometer data collected by the smartphone are used for Kalman filter fusion and localization, and compared with the method of using the gyroscope and the magnetometer alone. Figure 11 presents a direction angle estimation value obtained by different methods, and Figure 12 shows a direction estimation cumulative error probability distribution curve.

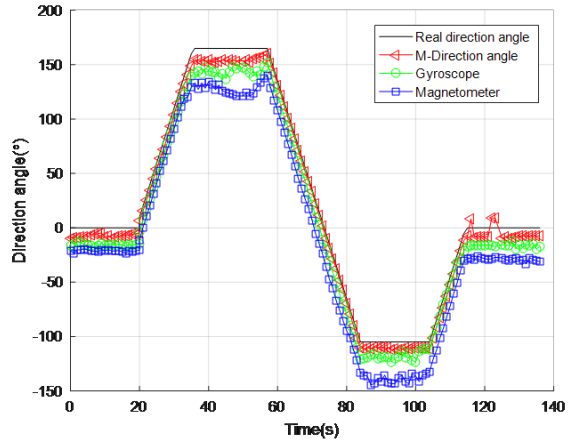


Figure 11. Comparison of different method direction angles

It can be seen from Figure 11 that the direction angle estimation method proposed in this paper is applicable to the process of changing direction. At the same time, it can be seen from Figure 12 that the cumulative probability distribution of the estimation method error within 5 degrees is 0.78, which is significantly higher than the other two. 0.52 using of the gyroscope alone and 0.29 using the magnetometer alone, the cumulative probability distribution with an error of less than 10 degrees is 0.91, which is significantly higher than 0.81 using the gyroscope alone and 0.65 using the magnetometer alone.

4.4 PDR Indoor Positioning Experiment

In order to verify the positioning of the improved PDR positioning algorithm in indoor positioning, two different experiments were performed in the laboratory environment. Figure 13 shows the positioning of the experimenter's rectangular path, and Figure 14 presents the positioning of the experimenter's straight path.

It can be seen from Figures 13 and 14 that whether the pedestrian walks a rectangular path or a straight path, the improved PDR localization algorithm is improved

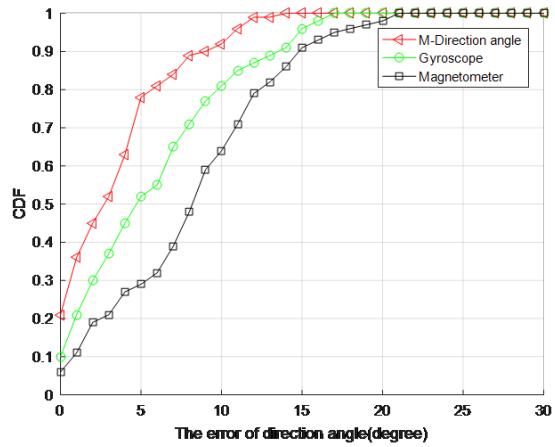


Figure 12. Directional estimation cumulative error probability distribution curve

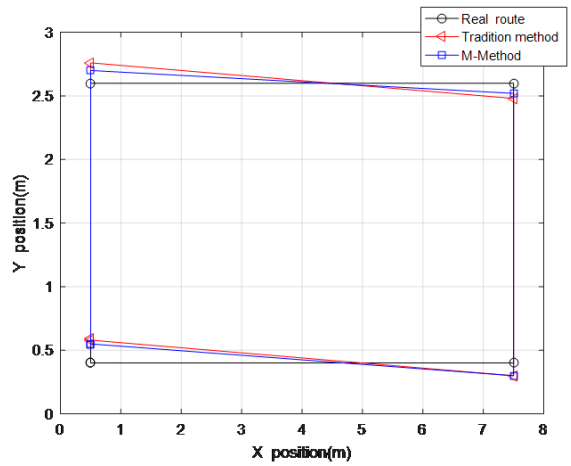


Figure 13. Position trajectory of rectangular path

compared with the traditional PDR localization algorithm, and the positioning accuracy is also improved.

5 CONCLUSION

In order to increase the indoor positioning accuracy of PDR, this paper firstly improves the step frequency detection, step size estimation and direction angle estimation in PDR algorithm, and then passes the problem of particle diversity weakening

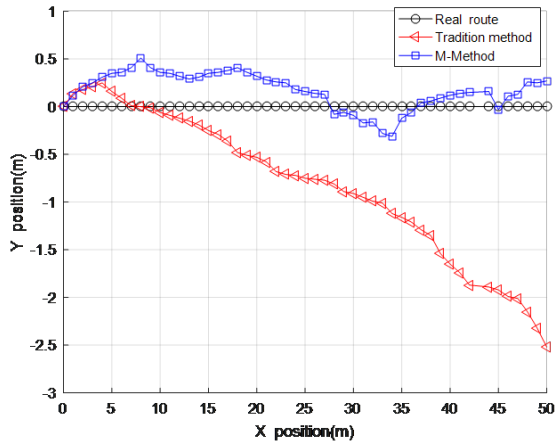


Figure 14. Position trajectory of straight path

over time in the particle filtering process. The fitness function in the particle swarm optimization algorithm is modified to optimize the particle filter to increase particle diversity and improve positioning accuracy.

REFERENCES

[1] Maghdid, H. S.—Lami, I. A.—Ghafoor, K. Z.—Lloret, J.: Seamless Outdoors-Indoors Localization Solutions on Smartphones: Implementation and Challenges. ACM Computing Surveys, Vol. 48, 2016, No. 4, Art. No. 53, 34 pp., doi: 10.1145/2871166.

[2] Li, N.—Chen, J.—Yuan, Y.—Song, C.: A Fast Indoor Tracking Algorithm Based on Particle Filter and Improved Fingerprinting. 2016 35th Chinese Control Conference (CCC), Chengdu, China, IEEE, 2016, pp. 5468–5472, doi: 10.1109/ChiCC.2016.7554206.

[3] Poulose, A.—Eyobu, O. S.—Han, D. S.: An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data. IEEE Access, Vol. 7, 2019, pp. 11165–11175, doi: 10.1109/ACCESS.2019.2891942.

[4] Wang, Z.: Research on Indoor Location Algorithm Based on Inertial Sensor and WiFi. University of Electronic Science and Technology, 2018.

[5] Manos, A.—Klein, I.—Hayan, T.: Gravity-Based Methods for Heading Computation in Pedestrian Dead Reckoning. Sensors (Basel), Vol. 19, 2019, No. 5, Art. No. 1170, 19 pp., doi: 10.3390/s19051170.

[6] Kang, J.—Lee, J.—Eom, D.-S.: Smartphone-Based Traveled Distance Estimation Using Individual Walking Patterns for Indoor Localization. Sensors (Basel), Vol. 18, 2018, No. 9, Art. No. 3149, 18 pp., doi: 10.3390/s18093149.

[7] Lu, C.—Uchiyama, H.—Thomas, D.—Shimada, A.—Taniguchi, R. I.: Indoor Positioning System Based on Chest-Mounted IMU. Sensors (Basel), Vol. 19, 2019, No. 2, Art. No. 420, 20 pp., doi: 10.3390/s19020420.

[8] Hasan, M. A.—Mishuk, M. N.: MEMS IMU Based Pedestrian Indoor Navigation for Smart Glass. Wireless Personal Communications, Vol. 101, 2018, pp. 287–303, doi: 10.1007/s11277-018-5688-3.

[9] Jimenez, A. R.—Seco, F.—Prieto, C.—Guevara, J.: A Comparison of Pedestrian Dead-Reckoning Algorithms Using a Low-Cost MEMS IMU. 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 2009, pp. 37–42, doi: 10.1109/WISP.2009.5286542.

[10] Ibarra Bonilla, M. N.—Escamilla-Ambrosio, P. J.—Ramírez Cortés, J. M.: Pedestrian Dead Reckoning Towards Indoor Location Based Applications. 2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control, Merida City, Mexico, 2011, 6 pp., doi: 10.1109/ICEEE.2011.6106608.

[11] Basiri, A.—Lohan, E. S.—Moore, T. et al.: Indoor Location Based Services Challenges, Requirements and Usability of Current Solutions. Computer Science Review, Vol. 24, 2017, pp. 1–12, doi: 10.1016/j.cosrev.2017.03.002.

[12] Phan, A. V.—Nguyen, M. L.—Bui, L. T.: Feature Weighting and SVM Parameters Optimization Based on Genetic Algorithms for Classification Problems. Applied Intelligence, Vol. 46, 2017, No. 2, pp. 455–469, doi: 10.1007/s10489-016-0843-6.

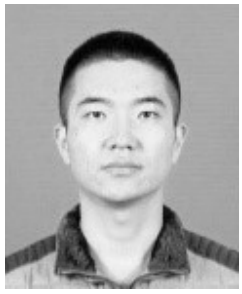
[13] Wang, W.—Liu, X. M.—Li, M. Z.—Wang, Y. B.—Wang, C. H.: Optimizing Node Localization in Wireless Sensor Networks Based on Received Signal Strength Indicator. IEEE Access, Vol. 7, 2019, pp. 73880–73889, doi: 10.1109/ACCESS.2019.2920279.

[14] Wang, H.: Implementation of RSSI-Based Localization Algorithm in Wireless Sensor Network. Beijing University of Posts and Telecommunications, Vol. 8, 2010, pp. 26–27.

[15] Lin, S. W.—Ying, K. C.—Chen, S. C.—Lee, Z. J.: Particle Swarm Optimization for Parameter Determination and Feature Selection of Support Vector Machines. Expert Systems with Applications, Vol. 35, 2008, No. 4, pp. 1817–1824, doi: 10.1016/j.eswa.2007.08.088.

[16] Song, J.—Xu, Y.—Liu, Y.—Zhang, Y.: Investigation on Estimator of Chirp Rate and Initial Frequency of LFM Signals Based on Modified Discrete Chirp Fourier Transform. Circuits, Systems, and Signal Processing, Vol. 38, 2019, No. 12, pp. 58–61, doi: 10.1007/s00034-019-01171-5.

Wei Wang received his B.Sc. degree from China University of Mining and Technology, Xuzhou, Jiangsu, China, in 2002, the M.Sc. degree from North University of China, Taiyuan, Shanxi, China, in 2007, and the Ph.D. degree from Taiyuan University of Technology, Taiyuan, Shanxi, China, in 2011. His research interest includes wireless sensor networks and signal processing, fundamental study of WSN node location, and intelligent algorithms. Since 2016, he has served as Assistant Professor in the Information and Communication Engineering Department, North University of China. He is the author of one book and more than 20 articles.



Cunhua Wang received his B.Eng. degree from North University of China. He is currently pursuing the M.Sc. degree with the North University of China. His research interests include signal processing and magnetic field indoor positioning.

Zhaoba Wang is Professor in the Department of Information and Communication Engineering, North University of China. He received his Ph.D. degree from the Detection Technology, Nanjing University of Science and Technology in 2002. His main research interests include signal and information processing. He is the author of more than 80 articles.

Xiaoqian Zhao received her B.Eng. degree from Xinzhou Teachers University. She is currently pursuing her M.Sc. degree with the North University of China. Her research interests include signal processing and magnetic field indoor positioning.