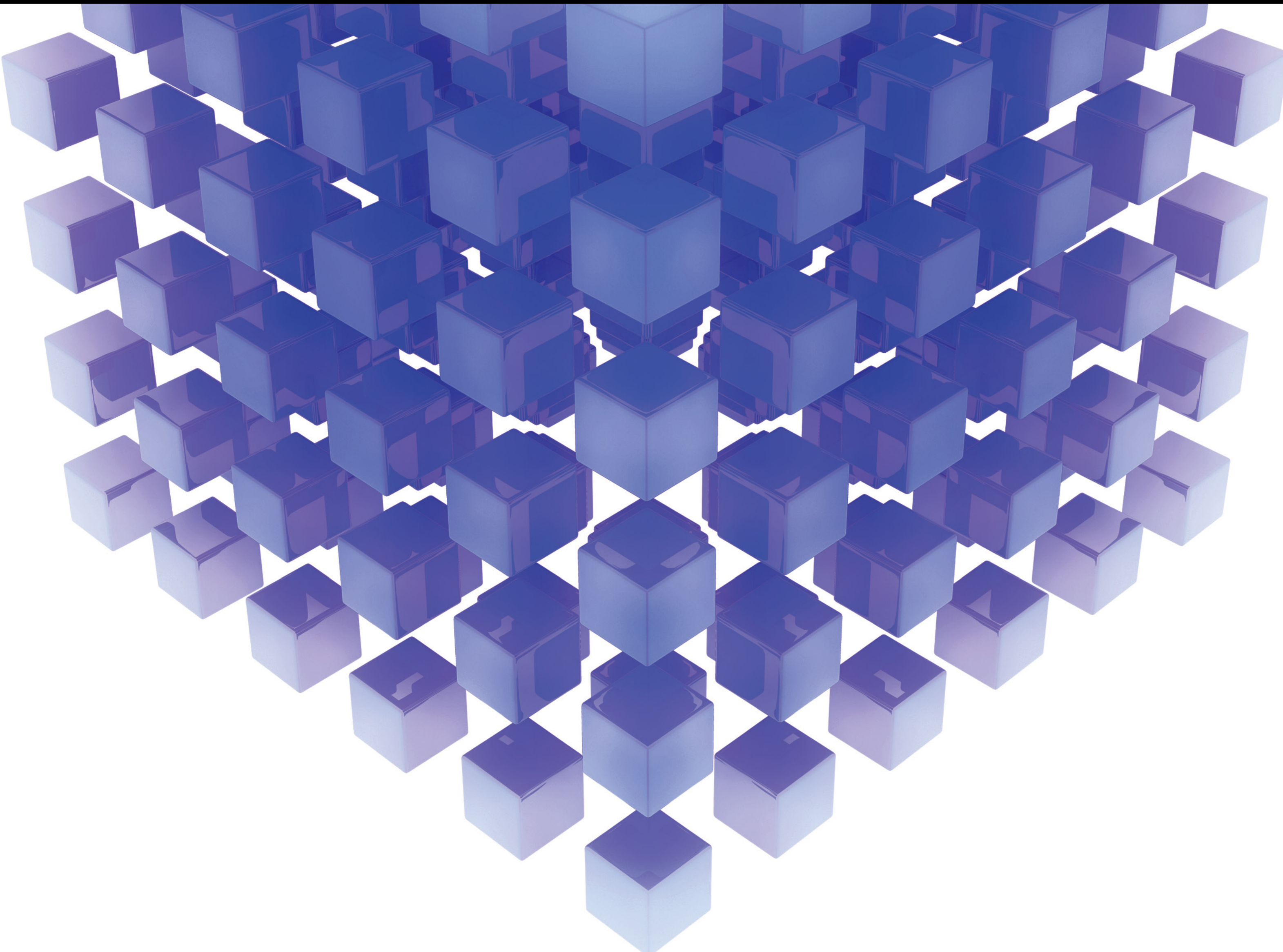
Mathematical Problems in Engineering

Nature-Inspired Optimization Algorithms for Routing, Resource Assignment, and Schedule-Related Problems

Lead Guest Editor: Ali Asghar Rahmani Hosseinabadi
Guest Editors: Ali Shokouhi Rostami and Adam Slowik



Nature-Inspired Optimization Algorithms for Routing, Resource Assignment, and Schedule-Related Problems

Mathematical Problems in Engineering

Nature-Inspired Optimization Algorithms for Routing, Resource Assignment, and Schedule-Related Problems

Lead Guest Editor: Ali Asghar Rahmani Hosseinabadi
Guest Editors: Ali Shokouhi Rostami and Adam Slowik

José António Fonseca De Oliveira Correia, Portugal
Jose Renato De Sousa, Brazil
Michael Defoort, France
Alessandro Della Corte, Italy
Laurent Dewasme, Belgium
Sanku Dey, India
Gianpaolo Di Bona, Italy
Roberta Di Pace, Italy
Francesca Di Puccio, Italy
Ramón I. Diego, Spain
Yannis Dimakopoulos, Greece
Hasan Dinçer, Turkey
José M. Domínguez, Spain
Georgios Dounias, Greece
Bo Du, China
Emil Dumic, Croatia
Madalina Dumitriu, United Kingdom
Premraj Durairaj, India
Saeed Eftekhar Azam, USA
Said El Kafhali, Morocco
Antonio Elipe, Spain
R. Emre Erkmen, Canada
John Escobar, Colombia
Leandro F. F. Miguel, Brazil
FRANCESCO FOTI, Italy
Andrea L. Facci, Italy
Shahla Faisal, Pakistan
Giovanni Falsone, Italy
Hua Fan, China
Jianguang Fang, Australia
Nicholas Fantuzzi, Italy
Muhammad Shahid Farid, Pakistan
Hamed Faroqi, Iran
Yann Favennec, France
Fiorenzo A. Fazzolari, United Kingdom
Giuseppe Fedele, Italy
Roberto Fedele, Italy
Baowei Feng, China
Mohammad Ferdows, Bangladesh
Arturo J. Fernández, Spain
Jesus M. Fernandez Oro, Spain
Francesco Ferrise, Italy
Eric Feulvarch, France
Thierry Floquet, France
Eric Florentin, France
Gerardo Flores, Mexico
Antonio Forcina, Italy
Alessandro Formisano, Italy
Francesco Franco, Italy
Elisa Francomano, Italy
Juan Frausto-Solis, Mexico
Shujun Fu, China
Juan C. G. Prada, Spain
HECTOR GOMEZ, Chile
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, USA
Jaime Gallardo-Alvarado, Mexico
Mosè Gallo, Italy
Akemi Gálvez, Spain
Maria L. Gandarias, Spain
Hao Gao, Hong Kong
Xingbao Gao, China
Yan Gao, China
Zhiwei Gao, United Kingdom
Giovanni Garcea, Italy
José García, Chile
Harish Garg, India
Alessandro Gasparetto, Italy
Stylianos Georgantzinos, Greece
Fotios Georgiades, India
Parviz Ghadimi, Iran
Ştefan Cristian Gherghina, Romania
Georgios I. Giannopoulos, Greece
Agathoklis Giaralis, United Kingdom
Anna M. Gil-Lafuente, Spain
Ivan Giorgio, Italy
Gaetano Giunta, Luxembourg
Jefferson L.M.A. Gomes, United Kingdom
Emilio Gómez-Déniz, Spain
Antonio M. Gonçalves de Lima, Brazil
Qunxi Gong, China
Chris Goodrich, USA
Rama S. R. Gorla, USA
Veena Goswami, India
Xunjie Gou, Spain
Jakub Grabski, Poland

Francisco J. Martos, Spain
Elio Masciari, Italy
Paolo Massioni, France
Alessandro Mauro, Italy
Jonathan Mayo-Maldonado, Mexico
Pier Luigi Mazzeo, Italy
Laura Mazzola, Italy
Driss Mehdi, France
Zahid Mehmood, Pakistan
Roderick Melnik, Canada
Xiangyu Meng, USA
Jose Merodio, Spain
Alessio Merola, Italy
Mahmoud Mesbah, Iran
Luciano Mescia, Italy
Laurent Mevel, France
Constantine Michailides, Cyprus
Mariusz Michta, Poland
Prankul Middha, Norway
Aki Mikkola, Finland
Giovanni Minafò, Italy
Edmondo Minisci, United Kingdom
Hiroyuki Mino, Japan
Dimitrios Mitsotakis, New Zealand
Ardashir Mohammadzadeh, Iran
Francisco J. Montáns, Spain
Francesco Montefusco, Italy
Gisele Mophou, France
Rafael Morales, Spain
Marco Morandini, Italy
Javier Moreno-Valenzuela, Mexico
Simone Morganti, Italy
Caroline Mota, Brazil
Aziz Moukrim, France
Shen Mouquan, China
Dimitris Mourtzis, Greece
Emiliano Mucchi, Italy
Taseer Muhammad, Saudi Arabia
Ghulam Muhiuddin, Saudi Arabia
Amitava Mukherjee, India
Josefa Mula, Spain
Jose J. Muñoz, Spain
Giuseppe Muscolino, Italy
Marco Mussetta, Italy
Hariharan Muthusamy, India
Alessandro Naddeo, Italy
Raj Nandkeolyar, India
Keivan Navaie, United Kingdom
Soumya Nayak, India
Adrian Neagu, USA
Erivelton Geraldo Nepomuceno, Brazil
AMA Neves, Portugal
Ha Quang Thinh Ngo, Vietnam
Nhon Nguyen-Thanh, Singapore
Papakostas Nikolaos, Ireland
Jelena Nikolic, Serbia
Tatsushi Nishi, Japan
Shanzhou Niu, China
Ben T. Nohara, Japan
Mohammed Nouari, France
Mustapha Nourelfath, Canada
Kazem Nouri, Iran
Ciro Núñez-Gutiérrez, Mexico
Wlodzimierz Ogryczak, Poland
Roger Ohayon, France
Krzysztof Okarma, Poland
Mitsuhiro Okayasu, Japan
Murat Olgun, Turkey
Diego Oliva, Mexico
Alberto Olivares, Spain
Enrique Onieva, Spain
Calogero Orlando, Italy
Susana Ortega-Cisneros, Mexico
Sergio Ortobelli, Italy
Naohisa Otsuka, Japan
Sid Ahmed Ould Ahmed Mahmoud, Saudi Arabia
Taoreed Owolabi, Nigeria
EUGENIA PETROPOULOU, Greece
Arturo Pagano, Italy
Madhumangal Pal, India
Pasquale Palumbo, Italy
Dragan Pamučar, Serbia
Weifeng Pan, China
Chandan Pandey, India
Rui Pang, United Kingdom
Jürgen Pannek, Germany
Elena Panteley, France
Achille Paolone, Italy

George A. Papakostas, Greece
Xosé M. Pardo, Spain
You-Jin Park, Taiwan
Manuel Pastor, Spain
Pubudu N. Pathirana, Australia
Surajit Kumar Paul, India
Luis Payá, Spain
Igor Pažanin, Croatia
Libor Pekař, Czech Republic
Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Jian Peng, China
Mingshu Peng, China
Xiang Peng, China
Xindong Peng, China
Yuexing Peng, China
Marzio Pennisi, Italy
Maria Patrizia Pera, Italy
Matjaz Perc, Slovenia
A. M. Bastos Pereira, Portugal
Wesley Peres, Brazil
F. Javier Pérez-Pinal, Mexico
Michele Perrella, Italy
Francesco Pesavento, Italy
Francesco Petrini, Italy
Hoang Vu Phan, Republic of Korea
Lukasz Pieczonka, Poland
Dario Piga, Switzerland
Marco Pizzarelli, Italy
Javier Plaza, Spain
Goutam Pohit, India
Dragan Poljak, Croatia
Jorge Pomares, Spain
Hiram Ponce, Mexico
Sébastien Poncet, Canada
Volodymyr Ponomaryov, Mexico
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Sivakumar Poruran, India
Francesc Pozo, Spain
Aditya Rio Prabowo, Indonesia
Anchasa Pramuanjaroenkij, Thailand
Leonardo Primavera, Italy
B Rajanarayan Prusty, India
Krzysztof Puszynski, Poland
Chuan Qin, China
Dongdong Qin, China
Jianlong Qiu, China
Giuseppe Quaranta, Italy
DR. RITU RAJ, India
Vitomir Racic, Italy
Carlo Rainieri, Italy
Kumbakonam Ramamani Rajagopal, USA
Ali Ramazani, USA
Angel Manuel Ramos, Spain
Higinio Ramos, Spain
Muhammad Afzal Rana, Pakistan
Muhammad Rashid, Saudi Arabia
Manoj Rastogi, India
Alessandro Rasulo, Italy
S.S. Ravindran, USA
Abdolrahman Razani, Iran
Alessandro Reali, Italy
Jose A. Reinoso, Spain
Oscar Reinoso, Spain
Haijun Ren, China
Carlo Renno, Italy
Fabrizio Renno, Italy
Shahram Rezapour, Iran
Ricardo Riaza, Spain
Francesco Riganti-Fulginei, Italy
Gerasimos Rigatos, Greece
Francesco Ripamonti, Italy
Jorge Rivera, Mexico
Eugenio Roanes-Lozano, Spain
Ana Maria A. C. Rocha, Portugal
Luigi Rodino, Italy
Francisco Rodríguez, Spain
Rosana Rodríguez López, Spain
Francisco Rossomando, Argentina
Jose de Jesus Rubio, Mexico
Weiguo Rui, China
Rubén Ruiz, Spain
Ivan D. Rukhlenko, Australia
Dr. Eswaramoorthi S., India
Weichao SHI, United Kingdom
Chaman Lal Sabharwal, USA
Andrés Sáez, Spain

Bekir Sahin, Turkey
Laxminarayan Sahoo, India
John S. Sakellariou, Greece
Michael Sakellariou, Greece
Salvatore Salamone, USA
Jose Vicente Salcedo, Spain
Alejandro Salcido, Mexico
Alejandro Salcido, Mexico
Nunzio Salerno, Italy
Rohit Salgotra, India
Miguel A. Salido, Spain
Sinan Salih, Iraq
Alessandro Salvini, Italy
Abdus Samad, India
Sovan Samanta, India
Nikolaos Samaras, Greece
Ramon Sancibrian, Spain
Giuseppe Sanfilippo, Italy
Omar-Jacobo Santos, Mexico
J Santos-Reyes, Mexico
José A. Sanz-Herrera, Spain
Musavarah Sarwar, Pakistan
Shahzad Sarwar, Saudi Arabia
Marcelo A. Savi, Brazil
Andrey V. Savkin, Australia
Tadeusz Sawik, Poland
Roberta Sburlati, Italy
Gustavo Scaglia, Argentina
Thomas Schuster, Germany
Hamid M. Sedighi, Iran
Mijanur Rahaman Seikh, India
Tapan Senapati, China
Lotfi Senhadji, France
Junwon Seo, USA
Michele Serpilli, Italy
Silvestar Šesnić, Croatia
Gerardo Severino, Italy
Ruben Sevilla, United Kingdom
Stefano Sfarra, Italy
Dr. Ismail Shah, Pakistan
Leonid Shaikhet, Israel
Vimal Shanmuganathan, India
Prayas Sharma, India
Bo Shen, Germany
Hang Shen, China
Xin Pu Shen, China
Dimitri O. Shepelsky, Ukraine
Jian Shi, China
Amin Shokrollahi, Australia
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, Canada
Angelo Sifaleras, Greece
Nuno Simões, Portugal
Mehakpreet Singh, Ireland
Piyush Pratap Singh, India
Rajiv Singh, India
Seralathan Sivamani, India
S. Sivasankaran, Malaysia
Christos H. Skiadas, Greece
Konstantina Skouri, Greece
Neale R. Smith, Mexico
Bogdan Smolka, Poland
Delfim Soares Jr., Brazil
Alba Sofi, Italy
Francesco Soldovieri, Italy
Raffaele Solimene, Italy
Yang Song, Norway
Jussi Sopanen, Finland
Marco Spadini, Italy
Paolo Spagnolo, Italy
Ruben Specogna, Italy
Vasilios Spitas, Greece
Ivanka Stamova, USA
Rafał Stanisławski, Poland
Miladin Stefanović, Serbia
Salvatore Strano, Italy
Yakov Strelniker, Israel
Kangkang Sun, China
Qiuqin Sun, China
Shuaishuai Sun, Australia
Yanchao Sun, China
Zong-Yao Sun, China
Kumarasamy Suresh, India
Sergey A. Suslov, Australia
D.L. Suthar, Ethiopia
D.L. Suthar, Ethiopia
Andrzej Swierniak, Poland
Andras Szekrenyes, Hungary
Kumar K. Tamma, USA

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 4670997, 15 pages
https://doi.org/10.1155/2021/4670997

Research Article

Path Optimization of Joint Delivery Mode of Trucks and UAVs

Qingkui Cao,[1,2] Xuefei Zhang,[1] and Xiangyang Ren [1]

[1] *School of Management Engineering and Business, Hebei University of Engineering, Handan 056038, China*
[2] *School of Economics and Management, Langfang Normal University, Langfang 065000, China*

Correspondence should be addressed to Xiangyang Ren; xiangyangren@hebeu.edu.cn

Received 9 June 2021; Revised 16 July 2021; Accepted 21 October 2021; Published 17 November 2021

Academic Editor: Adnan Maqsood

With the development of e-commerce and information technology, new modes of distribution are emerging. A new type of distribution tool, UAV (unmanned aerial vehicle), has entered into the public's field of vision. In the background of growing e-commerce, this paper proposes a new delivery mode of joint delivery of trucks and UAVs which particularly has been popular in recent years, with the advantages of prompt delivery, low cost, and independence from terrain restrictions, while traditional transportation tools such as trucks have more advantages in terms of flight distance and load capacity. Therefore, the joint delivery mode of trucks and UAVs proposed in this paper can well realize the complementary advantages of trucks and UAVs in the distribution process and consequently optimize the distribution process. Moreover, the growing e-commerce promotes customers' higher needs for delivery efficiency and the integrity of the delivered goods which urges companies to pay more attention to customers' satisfaction. This paper analyzes the joint delivery mode of trucks and UAVs, aims to minimize total delivery cost and maximize customer satisfaction, and builds a multiobjective optimization model for joint delivery. Furthermore, an improved ant colony algorithm is proposed in order to solve the mode in this paper. In order to effectively avoid prematurity of the ant colony algorithm, the limited pheromone concentration and the classification idea of the artificial bee colony algorithm are introduced to improve the ant colony algorithm. Finally, some experiments are simulated by MATLAB software, and the comparison shows that the joint delivery of trucks and UAVs has more advantages, and the improved ant colony algorithm is more efficient than the traditional ant colony.

1. Introduction

Recently, the application of UAVs has remained high heat and has been widely used in people's daily life and public services. On one hand, UAVs are not restricted by geographical conditions or road conditions and can greatly expand the efficiency of space usage. In urban areas with traffic congestion, remote areas with complex terrain, or disaster areas with dynamic environmental changes, they can respond quickly. On the other hand, UAVs are highly flexible and do not require a dedicated pilot, making them easy to operate and significantly reducing manpower costs. In addition, UAVs use batteries as a power source, making them more energy efficient and environmentally friendly. Being battery-powered and the small size make the UAVs limited in range and vulnerable to outside interference [1]. However, some tasks require ground coordination, such as logistics delivery, rescue, and target tracking, which require vehicles and UAVs to work together [2]. In the logistics distribution process, the joint delivery model of trucks and UAVs not only reduces the delivery tasks of trucks but also improves the efficiency. It has significant economic advantages. Trucks and UAVs can complement each other to better and accomplish tasks such as logistics delivery, disaster area rescue, and target tracking.

From an application perspective, UAVs and the joint delivery model of trucks and UAVs are already widely used in various fields. Firstly, in the agricultural sector, UAVs are mainly used for seed sowing and pollination, agricultural irrigation, and plant protection [3]. Torres et al. [4] conducted the first study of early weed management using UAVs. They pointed out that the number and resolution reflected in the images are a contradiction and the relationship between the two needs to be further optimized. Zhang et al. [5] studied the use of UAVs to apply pesticides to rice, which included reducing droplet floating, improving pesticide utilization, and increasing droplet deposition. Ayaz et al. [6] allowed the collaboration of UAVs with seeders to achieve more efficient seeding. Han et al. [7] gave a technical system for the application of UAVs in precision irrigation technology and suggested that, to meet the needs for efficient detection and precise dynamic management at different scales, UAVs should be combined with microsensing and ground monitoring systems. Secondly, in the field of rescue, Li et al. [8] proposed a new way of air-ground search and rescue with high altitude reconnaissance aircraft, groups of UAVs, and groups of unmanned vehicles for earthquake search and rescue scenarios. Zhang et al. [9] proposed a multi-UAV mission planning method for firefighting and rescue of multifire scenes. A multi-UAV mission planning model with changing target values over time is established to achieve a rational rescue according to the commander's decision intent. Furthermore, the first issue that needs to be addressed when UAVs and vehicles work together on policing patrols is energy resupply. Kim [10] used multiple shared static charging stations at different geographical locations to solve the UAV energy constraint problem and established a path planning model based on MILP (mixed integer linear program) to find the optimal flight path for UAVs through genetic algorithms. Maini et al. [11] transformed collaborative air-ground coverage monitoring into a two-stage optimization problem and built a complex mixed integer linear programming model to optimize the path of UAV and sequence of refueling points. Hu et al. [12] established an air-ground collaborative patrol path planning model to optimize the UAVs' patrol path and the unmanned vehicle energy resupply path. Finally, in the field of logistics distribution, Murray and Chu [13] proposed the joint delivery model of trucks and UAVs, where trucks carry UAVs from a distribution center to sequentially pass by some customer points where UAVs take off from the trucks to deliver packages for the customers in their vicinity. Ham [14] extended PDSTSP, considering the multitruck, multiwarehouse, and multi-UAV joint delivery that can be delivered and picked up by UAVs and using constraint planning solutions. Agatz et al. [15] modeled the travelling salesman problem with UAV (TSP-D), considered the same take-off and return positions of the UAV, and proposed a dynamic programming solution. Mario et al. [16] studied the TSP-D where the takeoff and return positions of the UAVs can be different and designed a greedy algorithm to solve the problem. Poikonen et al. [17] analyzed that the time for joint parallel delivery of trucks and UAVs is less than the delivery time of trucks alone in the worst case. Chu et al. [18] proposed a rural e-commerce logistics delivery model in which a soft time window and simultaneous pickup and delivery of trucks and UAVs were considered. With the goal of minimizing the total cost and using intelligent optimization algorithms, the total cost of terminal logistics is then significantly reduced compared to the delivery method using trucks alone. Han et al. [19] studied the delivery model of "vehicle + UAV" joint transportation of military materials in alpine mountain environment and wrote CTDEA algorithm to solve the problem. Among various application fields, logistics distribution is the most popular, which is also the main research field of this paper.

The vehicle routing problem is an NP-hard problem [20]. The more complex problem of joint delivery model of trucks and UAVs is also an NP-hard problem. Modern heuristic algorithms are the main method for solving such problems. These include genetic algorithm (GA), simulated annealing (SA), artificial bee colony algorithm (ABC), neural network (NN), and ant colony algorithm (ACA). Different algorithms have different solving performances, and they are the most important methods for solving vehicle routing problems at present. William et al. [21] introduced genetic algorithms for the study of vehicle routing problem and to solve the VRPTW problem. Shim et al. [22] designed a hybrid genetic algorithm to solve the vehicle routing problem for multiple vehicle models in multiple warehouses. The simulated annealing algorithm was first applied to combinatorial optimization problems by Tarantilis et al. [23]. Osman et al. [24] used the simulated annealing algorithm to solve the vehicle routing problem effectively. Li et al. [25] designed a multistage solution of the hybrid variable neighbourhood artificial bee colony algorithm to solve the dynamic demand vehicle routing problem with continuously updated customer points. Pei and Yu [26] proposed an improved artificial bee colony algorithm based on probability matrix model and elite retention strategy for solving perishable product distribution routing optimization problems in logistics industry. Li et al. [27] combined the ant colony system (ACS) with the maximum and minimum ant system (MMAS) to design an improved ant colony algorithm to solve the vehicle routing problem with time windows. Aiming at the "last mile" problem of fresh produce delivery, Fu and Liu [28] constructed an open time-varying vehicle routing optimization model with the objective of minimizing the total delivery cost and designed an improved ant colony algorithm to solve the problem according to the characteristics of the model.

Recently, in China, the e-commerce industry has been growing rapidly with the scale of online shopping users increasing gradually year by year. Data showed that, by December 2020, the scale of China's online shopping users has reached 782 million, increasing 72.15 million from March 2020, accounting for 79.1% of the total Internet users. With the booming economy, the disposable income per capita increasing, the scale of online shopping expanding, and online retail increasing year by year, online retail sales grew from RMB 5,155.6 billion in 2016 to RMB 1,176.01 billion in 2020, with a compound annual growth rate of 22.89% and online sales are expected to reach RMB 1,375.93 billion in 2021 indicating that the e-commerce is growing at an incredible speed. E-commerce makes shopping more convenient for customers which is an essential factor that enables e-commerce to develop at a rapid pace. However, the key aspect of the development of e-commerce is the delivery of logistics. After placing orders, the top concerns for the customers are the delivery time and the integrity of the delivered goods, which means the distribution process fundamentally determines the service level of e-commerce. Therefore, it is crucial to optimize the delivery efficiency and deliver goods with integrity in time for customers, while optimizing routes and reducing distribution costs for logistics companies are the top priorities. Therefore, it is of great significance to study e-commerce logistics delivery patterns with the aim of minimizing total delivery cost and maximizing customer satisfaction.

To sum up, the joint mode of trucks and UAVs is feasible. It has been widely used in agriculture, rescue, public security patrol, and logistics distribution, and the application of various modern heuristic algorithms in solving vehicle routing problems is discussed. This paper mainly studies the path optimization problem of the joint delivery mode of trucks and UAVs under the background of e-commerce. From the above analysis, it can be seen that the current researches mostly consider trucks carrying UAVs and UAVs can only serve one customer point and do not consider that UAVs will be limited by load capacity and mileage, resulting in some customer points not being able to get delivery services; also customer satisfaction was not taken into account. Therefore, in the joint delivery process of trucks and UAVs in this paper, the UAVs take off from the joint distribution transfer station to provide delivery services for nearby customer points, and, after completing the delivery task, they can return to other transfer stations that are closer. In addition, while trucks supply goods to be delivered at the joint distribution transfer station, they must also provide delivery services for those customer points that cannot be reached by UAVs due to UAV load capacity and mileage restrictions. Based on this, a mathematical model is established with the goal of minimizing total distribution cost and maximizing customer satisfaction, which is solved by the ABC-ACO algorithm. The results show that the joint delivery mode of trucks and UAVs has lower costs with higher customer satisfaction compared to trucks alone, and the improved ant colony algorithm is more efficient than the traditional ant colony algorithm.

The major contributions of this study are as follows:

(1) This paper proposes and analyzes a joint delivery mode of trucks and UAVs which is different from the existing research.

(2) A joint delivery multiobjective optimization model is developed with the objectives of minimizing total delivery cost and maximizing customer satisfaction with delivery time and integrity of goods.

(3) To avoid prematurity, the classification idea of the artificial bee colony algorithm is introduced into the ant colony algorithm and the ant colony algorithm is improved. Limits are placed on the upper and lower limits of pheromone concentration to prevent the algorithm from stopping iterations after prematurity has occurred. Finally, the ABC-ACA is used to solve the model in this paper.

Section 2 describes the problem of joint delivery of trucks and UAVs and builds the mathematical model. Section 3 is dedicated to algorithm design. Section 4 is dedicated to calculation examples analysis and Section 5 gives a summary of this paper and prospects for future research.

2. Problem Description and Model Establishment

The traditional truck-alone delivery mode is shown in Figure 1. In the traditional model, the truck departs from the distribution center carrying goods needed by several customer points, in order to serve each customer point in turn, and finally returns to the distribution center from which it departs. With the continuous development of urban e-commerce, the range of customers to be served is getting wider and wider, and how to achieve customer satisfaction in terms of time window and integrity of goods has gradually become the goal pursued by logistics enterprises. Presently, there are a series of problems in e-commerce distribution, such as urban traffic congestion, scattered customer points, and numerous distribution points with small delivery volume. These problems make the traditional delivery model of trucks not able to complete the delivery tasks efficiently. In order to make delivery more efficient, this paper proposes a joint delivery mode of trucks and UAVs based on the characteristics of UAVs with less affection by terrain, low delivery cost, and high distribution efficiency, which combines the advantages of trucks and UAVs to make delivery process more efficient and competitive for logistics companies.

2.1. Problem Description. The joint delivery mode of trucks and UAVs is shown in Figure 2. Joint distribution transfer stations are established near congestion-prone roads, and a certain number of UAVs are equipped in these transfer stations, which will deliver to customers near the transfer stations. Analogous to the two-level vehicle routing problem, the first-level distribution is the truck routing problem; that is, the truck departs from the distribution center, when the truck is loaded with goods for the customers near the joint distribution transfer station and goods for customer points beyond the UAV load or range, then the truck visits these transfer stations and customer points in turn and finally returns to the distribution center. Secondary distribution is a delivery mission for UAVs. Each UAV departs from a joint distribution transfer station and then provides delivery services to customer points near the transfer station which accord with the UAVs' loading capacity and flight range. What needs to be emphasized is that, after the UAVs' completion of the delivery tasks, they can return to the transfer stations from which they departed or other available transfer stations nearby.

As the issue requires logistics companies to consider both minimizing delivery costs and maximizing customer satisfaction, this paper builds a multiobjective optimization model for joint delivery.

Without losing generality, this paper makes the following assumptions:

(1) There is only one distribution center.

(2) Each customer point can only be served by one UAV or one truck, because each customer point can only be served once in real life.

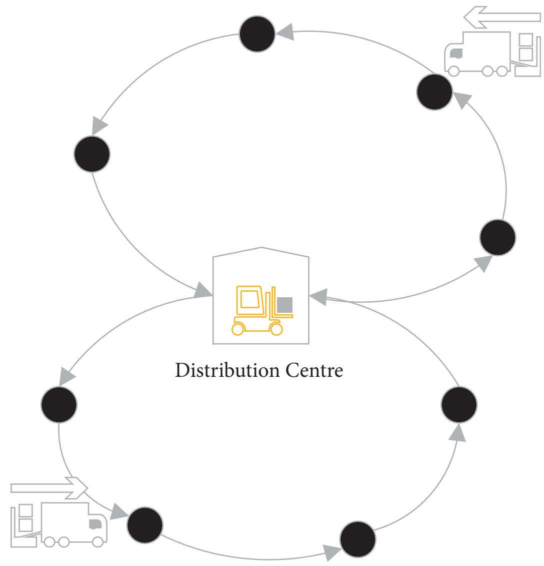


Figure 1: Traditional truck-alone delivery mode.

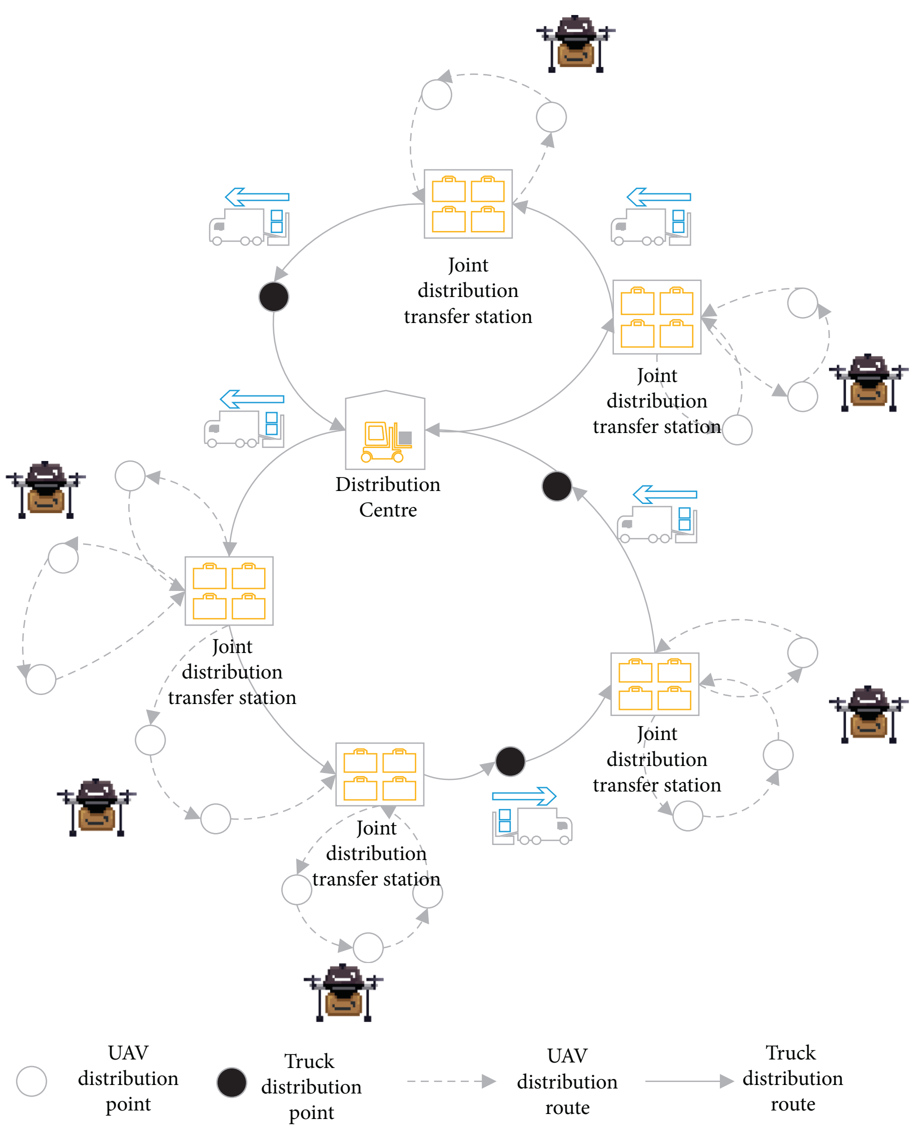


Figure 2: Joint distribution mode of trucks and UAVs.

(3) Distribution centers have multiple trucks of the same type and multiple UAVs of the same type in each joint distribution transfer station.

(4) The maximum payload and the longest flight distance of the UAV are known.

(5) Each UAV can load multiple packages at once. In other words, the UAV can serve multiple customers at a time. This assumption is reasonable based on real-life UAV delivery experience and the load of the UAVs.

(6) The UAV can continue delivery after battery replacement or charging at the joint delivery transfer station.

(7) UAV battery changes time and loading time at joint distribution transfer station are disregarded.

(8) The location coordinates of the distribution center, customers, and joint distribution transfer stations are known. These are known in the actual distribution process.

(9) The time windows of all customers are known.

(10) UAVs and trucks move at uniform speed.

2.2. Customer Satisfaction Function

2.2.1. Time Satisfaction Function. Unlike considering soft time windows or hard time windows, in the actual delivery process, customers may prefer to get the delivery service at a specific time period within the time window, and earlier or later than this time period can lead to different degrees of dissatisfaction [29]. As a result, this paper blurs the time windows of each customer point. This fuzzy time window includes the time range in which the customer most wants to get the delivery service and the service time range that can be tolerated. Specifically, the level of satisfaction is 100% if the delivery service is received within the time range $[e_i, l_i]$ by consumer i. The time window within this time range is called the most satisfactory time window. Satisfaction would gradually decrease as the gap with the most satisfactory time window increases if the delivery service is received within the time ranges $[E_i, e_i]$ and $[E_i, e_i]$ by consumer i. The time window $[E_i, L_i]$ is called tolerable time window. Completely unsatisfied or satisfaction is 0 if the delivery service is received beyond the time range $[E_i, L_i]$ by consumer i. The time satisfaction function is shown in Figure 3.

The time satisfaction function can be expressed as follows:

$$f_i(t_i) = \begin{cases} \left[\dfrac{t_i - E_i}{e_i - E_i}\right]^{\alpha}, & E_i < t_i < e_i, \\ 1, & e_i \le t_i \le l_i, \\ \left[\dfrac{L_i - t_i}{L_i - l_i}\right]^{\beta}, & l_i < t_i < L_i, \\ 0, & \text{else}, \end{cases} \tag{1}$$

where t_i is the time when the truck or UAV starts service for customer point i and α, β is the customer's sensitivity factor for time.

2.2.2. Goods Damage Satisfaction Function. Throughout the delivery process, the degree of damage to the goods after they are received by the customer affects customer satisfaction to some extent. E-commerce goods may be damaged to a certain extent in the process of distribution due to goods extrusion, improper operation of handling staff, or goods collision. Therefore, goods damage comes as another important issue for logistics enterprises to consider. There is a negative correlation between the goods damage rate and customer satisfaction: the higher the damage rate is, the lower the customer satisfaction is. This paper only considers the goods damage caused by the accumulation of transportation time during the transportation process. The goods damage rate can be calculated as follows:

$$y_i = q(t_i - t_0), \tag{2}$$

where y_i is the damage rate of customer point i, t_0 is the departure time from the distribution center or joint distribution transfer station, and q is the goods damage coefficient per unit time.

The most satisfied or the satisfaction is 100% if the goods damage rate is within the range $[0, h]$ by consumer i. Customer satisfaction decreases as the rate of cargo damage increases if the goods damage rate is within the range $[h, n]$ by consumer i. When the damage rate exceeds n, the customer satisfaction is 0. Therefore, the damage satisfaction function is shown in Figure 4.

The goods damage satisfaction function can be expressed as follows:

$$u_i(y_i) = \begin{cases} 0, & y_i > n, \\ \dfrac{n - y_i}{n - h}, & h \le y_i \le n, \\ 1, & 0 \le y_i < h. \end{cases} \tag{3}$$

2.3. Definition of Parameters and Variables. The joint delivery problem of trucks and UAVs can be defined on an undirected graph $G = (P, E, F)$, where $P = \{P_0 \cup P_S \cup P_C\}$ is the set of points, P_0 is distribution center, P_S is the set of joint distribution transfer stations, $P_C = \{P_{ck} \cup P_{cv}\}$ is the set of customer points, P_{ck} is the set of customer points that cannot be visited by UAV due to load and range restrictions and can only be served by trucks, P_{cv} is the set of customer points that are serviced by UAVs, and $M_S = P_S \cup P_{cv}$ is the set of customer points accessible by UAVs and joint distribution transfer stations. $E = \{(i, j) | i, j \in P, i \neq j\}$ is the set of trucks driving edges, and $F = \{(i, j) | i, j \in M_S, i \neq j\}$ is the set of UAVs flighting edges.

K: the set of trucks;

V: the set of UAVs;

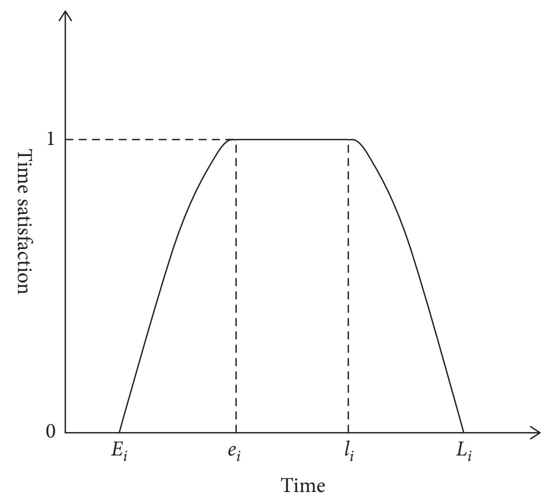


Figure 3: Time satisfaction function.

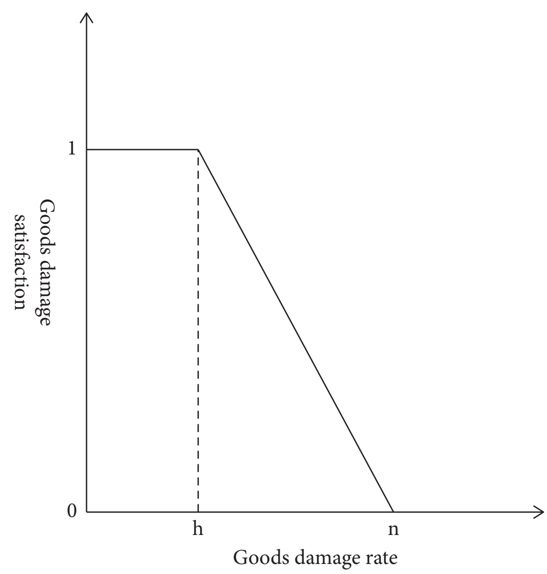


Figure 4: Goods damage satisfaction function.

Q_K: the maximum load of each truck;

D_V: the maximum flight distance of each UAV;

q_i: the demand of consumer i;

S_v: the transfer station for the i-th UAV;

$W^v_{iS_v}$ $(0 \leq W^v_{iS_v} \leq Q_V)$: the load capacity of UAV v taking off from station S_v when leaving customer point i;

v: the truck speed;

$v\prime$: the UAV speed;

d_{ij}: the linear distance between nodes i and j;

η: the multiple of the distance of the truck greater than the straight line distance;

t^k_i: the time when the truck $k$arrives at node i;

t^v_i: the time when the UAV $v$arrives at node i;

t_{ki}: the time for the truck to serve the customers;

t_{vi}: the time for the UAV to serve the customers;

$[e_i, l_i]$: the most satisfying time window of customer i;

$[E_i, L_i]$: the tolerable service time of customer i;

$f_i(t_i)$: the time satisfaction function;

$[0, h]$: the range of damage rate acceptable to customers;

$[h, n]$: the range of damage rate that customers can tolerate;

$u_i(y_i)$: the function of satisfaction of goods damage;

C^a_k: the fixed cost of single truck;

C^a_v: the fixed cost of single UAV;

C^b_k: the transport cost per unit distance of the truck;

C^b_v: the flight cost per unit distance of the UAV;

P_{kn}: the set of the n-th distribution routes of truck k, where $P_{kn} \subseteq P, 0 \leq n \leq \text{num}\,1, \text{num}\,1 = |P|$; that is, num 1 is the number of elements in P;

P_{vm}: the set of the m-th distribution routes of UAV v, where $P_{vm} \subseteq P_{cv}, 0 \le m \le \text{num }2, \text{num }2 = |P_{cv}|$; that is, num 2 is the number of elements in P_{cv};

$x_{ij}^{k} = \begin{cases} 1, & \text{Truck } k \text{ runs from node } i \text{ to node } j, \\ 0, & \text{else.} \end{cases}$;

$y_{ij}^{v} = \begin{cases} 1, & \text{UAV } v \text{ flights from node } i \text{ to node } j, \\ 0, & \text{else.} \end{cases}$;

$P_{ij}^{k} = \{1,$ In the path of truck K, node i precedes node j (but not necessarily continuous) , 0, else.

2.4. Mathematical Model. According to the above analysis, the total cost Z_1 is composed with the fixed cost and delivery cost of trucks and UAVs as follows:

$$Z_1 = C_a + C_b, \tag{4}$$

where C_a is the total start-up cost of trucks and UAVs and C_b is the total delivery cost of trucks and UAVs. Since the start-up cost is related to the number of trucks and UAVs dispatched and the delivery cost is related to the moving distance of trucks and UAVs, C_a and C_b can be expressed as follows:

$$C_a = C_k^a \sum_{j \in P_S \cup P_{ck}} \sum_{k \in K} x_{0j}^{k} + C_v^a \sum_{i \in P_S} \sum_{j \in P_{cv}} \sum_{v \in V} y_{ij}^{v}, \tag{5}$$

$$C_b = C_k^b \sum_{i,j \in P_0 \cup P_S \cup P_{ck}} \sum_{k \in K} (1+\eta) d_{ij} x_{ij}^{k} + C_v^b \sum_{i,j \in P_S \cup P_{cv}} \sum_{v \in V} d_{ij} y_{ij}^{v}. \tag{6}$$

Based on the above description and hypotheses, the multiobjective optimization model for the joint delivery of trucks and UAVs can be established as follows:

$$\min Z_1 = C_a + C_b, \tag{7}$$

$$\min Z_2 = \omega_1 \left[1 - \frac{1}{N} \sum_{i \in P_C} f_i(t_i) \right] + \omega_2 \left[1 - \frac{1}{N} \sum_{i \in P_C} u_i(y_i) \right], \tag{8}$$

$$\text{s.t.} \sum_{i \in P} \sum_{k \in K} x_{ij}^{k} + \sum_{i \in P} \sum_{v \in V} y_{ij}^{v} = 1, \quad \forall j \in P_{cv}, \tag{9}$$

$$\sum_{i \in P} \sum_{j \in P} x_{ij}^{k} \le K, \quad \forall k \in K, \tag{10}$$

$$\sum_{i \in P_S \cup P_{ck}} x_{i0}^{k} = \sum_{i \in P_S \cup P_{ck}} x_{0i}^{k}, \quad \forall k \in K, \tag{11}$$

$$\sum_{i \in P} x_{ij}^{k} = \sum_{i \in P} x_{ji}^{k} \le 1, \quad \forall j \in P, \forall k \in K, i \ne j, \tag{12}$$

$$\sum_{i \in P} \sum_{j \in P} x_{ij}^{k} q_i \le Q_K, \quad \forall k \in K, \tag{13}$$

$$\sum_{i \in P_{cv}} y_{ij}^{v} = \sum_{i \in P_{cv}} y_{ji}^{v} \le 1, \quad \forall j \in P_{cv} \cup S_v, \forall v \in V, \tag{14}$$

$$\sum_{i \in P} \sum_{k \in K} x_{ij}^{k} \le \sum_{i \in P} \sum_{v \in V} y_{ij}^{v}, \quad \forall j \in P_S, \tag{15}$$

$$\sum_{i \in P} \sum_{k \in K} x_{ij}^{k} = 1, \quad \forall j \in P_{ck}, \tag{16}$$

$$\sum_{i\in P_S\cup P_{cv}} y_{ij}^v = \sum_{i\in P_S\cup P_{cv}} y_{ji}^v = 0,\quad \forall j\in P_{ck}, \forall v\in V, \tag{17}$$

$$d_{gi} + \sum_{i\in P_{cv}}\sum_{j\in P_{cv}} d_{ij} + d_{jg} \le D_V,\quad \forall g\in P_S, \tag{18}$$

$$\sum_{i\in P_{cv}} y_{gi}^v = 0,\quad \forall g\in P_0, \forall v\in V, \tag{19}$$

$$\sum_{i\in P_{cv}} y_{ig}^v = 0,\quad \forall g\in P_0, \forall v\in V, \tag{20}$$

$$\sum_{i\in P_S}\sum_{j\in P_{cv}} y_{ij}^v = \sum_{i\in P_{cv}}\sum_{j\in P_S} y_{ij}^v,\quad \forall v\in V, \tag{21}$$

$$W_{S_vS_v}^v - q_j y_{S_vj}^v - Q_v\left(1-y_{S_vj}^v\right) \le W_{jS_v}^v \le W_{S_vS_v}^v - q_j y_{S_vj}^v + Q_v\left(1-y_{S_vj}^v\right),\quad \forall j\in P_{cv}, \forall v\in V, \tag{22}$$

$$W_{iS_v}^v - q_j y_{ij}^v - Q_V\left(1-y_{ij}^v\right) \le W_{jS_v}^v \le W_{iS_v}^v - q_j y_{ij}^v + Q_V\left(1-y_{ij}^v\right),\quad \forall i\in P_{cv}, \forall j\in P_{cv}, \forall v\in V, \tag{23}$$

$$\sum_{i\in P_{kn}}\sum_{j\in P_{kn}} x_{ij}^k \le \left|P_{kn}\right| - 1,\quad \forall k\in K, \forall P_{kn}\subseteq P, 0\le n\le \text{num } 1, \tag{24}$$

$$\sum_{i\in P_{vm}}\sum_{j\in P_{vm}} y_{ij}^v \le \left|P_{vm}\right| - 1,\quad \forall v\in V, \forall P_{vm}\subseteq P_{cv}, 0\le m\le \text{num } 2, \tag{25}$$

$$t_j^k = \sum_{i,j\in P_0\cup P_S\cup P_{ck}} \left(t_i^k + \frac{d_{ij}(1+\eta)}{v} + t_{ki}\right),\quad \forall k\in K, \tag{26}$$

$$t_j^v = \sum_{i,j\in P_S\cup P_{cv}} \left(t_i^v + \frac{d_{ij}}{v\prime} + t_{vi}\right),\quad \forall v\in V, \tag{27}$$

where equations (7) and (8) are objective functions. Equation (7) implies the minimum total cost of delivery. Equation (8) implies that the average customer dissatisfaction is minimal. Equation (9) implies that any customer point can only be served once by an UAV or a truck. Equation (10) indicates the limit of the number of trucks used. Equation (11) indicates that the truck departs from the distribution center and returns to the distribution center after completing the delivery task. Equation (12) implies that the truck drives into a node and then drives out from this node. Equation (13) indicates that the good loaded by the truck cannot exceed its maximum load constraint. Equation (14) indicates that after the UAV flies into a certain customer point, it will fly out from the customer point. Equation (15) shows that the UAVs at the joint delivery transfer station can perform delivery services only after the trucks have visited the joint delivery transfer station. Equation (16) indicates customer points that cannot be visited by UAVs due to load and range restrictions and can only be served by trucks for delivery. Equation (17) indicates customer points that exceed the load and mileage constraints of the UAV and therefore cannot be accessed by the UAV. Equation (18) implies the flight distance constraint of the UAV. Equation (19) indicates that the UAV cannot be launched directly from the distribution center to the customer points. Equation (20) indicates that the UAV cannot fly directly from the customer point to the distribution center. Equation (21) indicates that the UAV departs from a joint distribution transfer station and can return to any transfer station after completing its delivery task. Equation (22) represents the load capacity of the UAV from the joint delivery transfer station to the customer point j after the delivery is completed. Equation (23) represents the load capacity of the UAV from customer point i to customer point j after completing this delivery task. Equation (24) is the detrucking branch constraint, which is the removal of incomplete routes from the truck routes. Equation (25) is the de-UAV branch constraint, which is the removal of incomplete routes from the UAV routes. Equation (26) indicates the time when truck k arrives at node j. Equation (27) represents the time when the UAV v arrives at node j.

In this paper, a multiobjective optimization model is established. In order to make the solution more convenient, the two objective functions are transformed into a single-objective optimization model by assigning weights. The transformed objective function can be expressed as follows:

$$\min Z = \omega_1 (C_a + C_b) + \omega_2 \left[1 - \frac{1}{N} \sum_{i \in P_C} f_i(t_i)\right] + \omega_3 \left[1 - \frac{1}{N} \sum_{i \in P_C} u_i(y_i)\right]. \quad (28)$$

3. Algorithm Design

This paper studies the multiobjective optimization problem, which belongs to the NP-hard problem. When solving such problems, heuristic algorithms have been widely used [30]. This paper presents the joint delivery model of trucks and UAVs as a two-level path planning problem that requires a heuristic algorithm with parallel computing mechanisms to solve, and, due to the complexity of the problem, sufficient stability needs to be maintained in the solution process. The ant colony algorithm uses a distributed parallel computer system, which has the advantages of positive feedback and good robustness and can get a more satisfactory feasible solution within an acceptable time range. However, the ant colony algorithm tends to be premature and inefficient in the solving process [31]. Therefore, in this paper, the ant colony algorithm is improved by combining the idea of artificial bee colony algorithm, and the upper and lower limits of the global pheromone concentration are given to prevent the algorithm from affecting the optimization effect due to prematurity.

3.1. Principle of Ant Colony Algorithm. Ant colony algorithm (ACA) was first proposed by Italian scholar Dorigo and colleagues [32] in the 1990s. They found that single ant behaved simply when foraging, while ants exhibited an intelligent behavior when they foraged in groups. For example, the ant colony will search for the shortest path for food, because the ants in the ant colony will transmit information to each other through an information mechanism. After further research, it was found that ants would release “pheromone” on the path they passed during the foraging process. Ants within a colony have the ability to perceive “pheromones” and will follow paths with high concentrations of “pheromones.” Besides, each ant will leave “pheromones” on the path it passes. This is known as a positive feedback mechanism. It is because of such a process that the entire colony will follow the shortest path to the food source.

3.2. Principle of Artificial Bee Colony Algorithm. The artificial bee colony (ABC) algorithm was proposed by the group of Karaboga and Basturk [33] in 2005. It is an algorithm inspired by the behavior of bee colonies. The artificial bee colony algorithm imitates the process of honey collection by bees. This algorithm divides all bees into leading bees, detecting bees, and following bees, and these bees can complete their missions in the process of finding the optimal honey source [34]. In the process of solving the artificial bee colony algorithm, firstly leading bees look for nectar sources based on existing information and tell onlooker bees about the sources they have found. Following bees then select a nectar source based on the information obtained with a certain probability and find a candidate nectar source to compare with the previous one. Finally, when a nectar source is not renewed after a finite number of cycles, the bees associated with that source are transformed into leading bees to continue the search for potential new nectar sources.

3.3. Algorithm Improvement. Based on the ACA, the classification idea of ABC algorithm is introduced to dynamically divide the ant colony into two types, leading ants and detecting ants, according to the fitness value. Among them, leading ants search for a better path, and detecting ants search for other paths and more feasible solutions. Secondly, the weighting coefficients and fitness values are used to dynamically update the local pheromone concentrations. Finally, the pheromone concentration on each path was limited to $[\tau_{\min}, \tau_{\max}]$ in order to prevent all ants from rapidly aggregating and stalling the search with too high a pheromone concentration on a particular path.

3.3.1. Ant Colony Classification. Combining the classification idea of artificial bee colony algorithm, the dynamic classification operation is introduced on the basis of ant colony algorithm. The purpose is to divide the entire ant colony into two ant forms, leading ants and detecting ants, and search in parallel. The specific operation is to divide the ant colony into leading ants and detecting ants according to the different pheromone concentrations of the paths taken by different ants. Leading ants are responsible for searching for a better path, highlighting the pheromone concentration on this path, which speeds up the convergence of the algorithm. Detecting ants are responsible for searching for feasible solutions with better quality in addition to these better paths. This operation ensures the diversity of the algorithm, so that it will not fall into a local optimum and at the same time improve the quality of the solution.

The ant colony classification formula can be expressed as follows:

$$f_i = \frac{1}{Z_i}, \quad (29)$$

$$R_i = \begin{cases} \text{detecting ant}, & 0 < f_i \leq M, \\ \text{leading ant}, & M < f_i \leq 1. \end{cases} \quad (30)$$

In equation (29), Z_i is the objective function of the model, and f_i is the fitness of each ant, which is the reciprocal of the objective function. According to the different fitness value f_i of each ant, the ant colony was divided into two types: detecting ant and leading ant. It can be expressed as equation (30), where M is the classification boundary point that divides ants into two classes $(0 < M < 1)$: when $0 < f_i \leq M$, it is the detection ant; when $M < f_i \leq 1$, it is the leading ant.

3.3.2. Dynamic Updating of Local Pheromone. In order to better maintain the diversity of the population and improve the convergence speed of the algorithm, the weighted coefficient and fitness value are introduced to update the local pheromone [35], so that different types of ants can implement different local pheromone update strategies. They can be expressed as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_{ij}, \tag{31}$$

$$\tau_{ij} = \lambda_1 f_i, \tag{32}$$

$$\tau_{ij} = \lambda_2 f_i, \tag{33}$$

where λ_1 and λ_2 are weighting coefficients, f_i is the fitness of the i-th ant, and ρ is the rate of pheromone volatilization. Equation (32) is the local pheromone update formula of leading ants. Equation (33) is the local pheromone update formula of detecting ants. Two kinds of ants perform different updating strategies of dynamic pheromone, and $\lambda_1 > \lambda_2$, to shorten the optimization time by highlighting the pheromone concentration on the optimal solution path.

3.3.3. Pheromone Concentration Limit. In order to avoid prematurity and stagnant phenomena, the pheromone concentration on each path is controlled within $[\tau_{\min}, \tau_{\max}]$. The pheromone less than $\tau_{\min}$ is assigned as $\tau_{\min}$, and the pheromone greater than $\tau_{\max}$ is assigned as $\tau_{\max}$. The purpose of this method is to avoid the fact that the pheromone concentration is too high to iterate, which will attract most ants to gather quickly and lead to premature phenomenon. The value formulas of pheromone upper and lower limits can be expressed as follows:

$$\tau_{\max}(t) = \frac{1}{2(1-\rho)C(t)} + \frac{\sigma}{C(t)}, \tag{34}$$

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)}{20}, \tag{35}$$

where ρ is the pheromone volatilization rate, $C(t)$ is the optimal objective function in the t-th iteration, and σ is the number of optimal solutions of the t-th iteration.

3.4. Algorithm Steps

Step 1. Initialize parameters.

Step 2. Create a tabu list and put m ants in the distribution center.

Step 3. Provided that the constraints are satisfied, each ant calculates the transfer probability according to equation (36) and selects the next visited node according to the transfer probability. In this equation, α is the relative importance factor of pheromone concentration, and β is the importance factor of heuristic function, which reflects the importance of ant to heuristic information. $J_k(i)$ is the set of cities that ant k is allowed to visit in the next step.

$$P_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\left(1/Z_i\right)^{\beta}(t)}{\sum_{j\in J_k(i)} \tau_{ij}^{\alpha}(t)\left(1/Z_i\right)^{\beta}(t)}, & j \in J_k(i), \\ 0, & else. \end{cases} \tag{36}$$

Step 4. According to the transition probability, roulette is used to select the next node to visit.

Step 5. According to the ant colony classification, the fitness value is calculated and the ants are classified.

Step 6. Update the pheromone of leading ant and detecting ant.

Step 7. Exchange high-quality solution. If $Z_{\text{detecting ants}} < Z_{\text{leading ants}}$, let two kinds of ants exchange high-quality solutions. This step is done by swapping the identity functions of the two types of ants, with the better detecting ant transforming into the new leading ant and the replaced leading ant converting into a detecting ant and performing the functions of a detecting ant. If the solution of the detecting ant is not better than the solution of the leading ant during the iteration, which means the algorithm does not perform a high-quality solution exchange, then proceed directly to Step 8.

Step 8. Update global pheromone. It is limited to $[\tau_{\min}, \tau_{\max}]$.

Step 9. Judge whether the end condition is satisfied at this time. If not, go to Step 3 and start a new cycle. Otherwise, go to Step 10.

Step 10. If the maximum number of iterations is reached, the iteration is terminated.

The flow-process diagram of ABC-ACA is shown in Figure 5.

4. Analysis of Calculation Examples

4.1. Background Analysis and Parameter Setting. This paper takes the order of a distribution center of an e-commerce logistics enterprise in city A as the customer sample. The ABC-ACO algorithm is used to plan the joint delivery of trucks and UAVs, so as to minimize the delivery cost and maximize customer satisfaction under the above constraints. The location coordinates of 15 customer points, 1 distribution center, and 5 joint distribution transfer stations, the time window requirements of customer points, and the requirement and service time of each customer point are shown in Table 1.

The distance between nodes is measured by vector distance, which can be expressed by the following equation:

$$d_{ij} = \left[\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right]^{1/2} \tag{37}$$

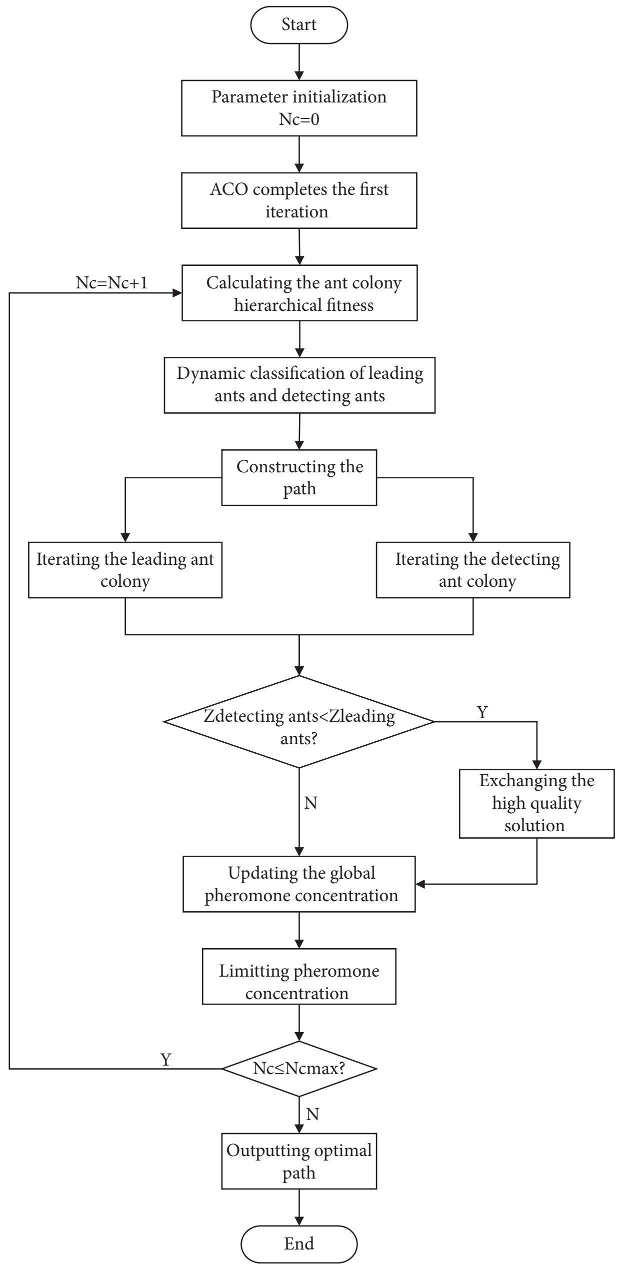


Figure 5: ABC-ACA flow chart.

The relevant parameters of trucks and UAVs are shown in Table 2.

In addition, the multiple of the distance of the truck greater than the straight distance $\eta = 0.3$, time sensitivity coefficient $\alpha = 0.5, \beta = 0.8$, goods damage coefficient $q = 0.1\%$, the range of goods damage rate acceptable to customers is $[0, 0.2\%]$, the range of goods damage rate tolerable to customers is $[0.2\%, 1]$, and the weight of objective function is $\omega_1 = 0.5, \omega_2 = 0.3, \omega_3 = 0.2$.

4.2. Result Analysis. According to the ABC-ACA described above, using MATLAB software, the joint delivery model of trucks and UAVs is solved. Through multiple simulations of the example, the parameters are set as follows: $\alpha = 1$, $\beta = 3$, $\rho = 0.4$, Ant_num = 20, $M = 0.7$, $\lambda_1 = 4$, and $\lambda_2 = 2$.

Figure 6 shows the route map of the optimal delivery scheme for the joint delivery model of trucks and UAVs by using ABC-ACA.

Figure 7 shows the optimal delivery scheme when only trucks are used for delivery by using ABC-ACA.

Figure 8 shows the optimal delivery scheme when only trucks are used for delivery by using ACA.

Table 3 shows the optimization results of using ABC-ACA to solve the joint delivery mode and the trucks-alone delivery.

As can be seen from the above table, when using joint delivery of trucks and UAVs, only 2 trucks need to be

Table 1: Information table of each node.

No.	Node	Position coordinates (X,Y)	Demands q_i/kg	Time windows			
				e_i/h	l_i/h	E_i/h	L_i/h
0	Distribution center	(94,100)	0	7	20	7	20
1	Customer point 1	(60,135)	4	10	11	7	12
2	Customer point 2	(40,36)	17	8	11.5	7.5	12.5
3	Customer point 3	(48,96)	3	8	12	7	12.5
4	Customer point 4	(92,154)	25	8.5	13	8	13.5
5	Customer point 5	(97,50)	10	10	16	11	17
6	Customer point 6	(120,90)	11	9	13.5	8	16
7	Customer point 7	(44,160)	22	9.5	14	9	14.5
8	Customer point 8	(20,54)	20	9.5	15	9	15.5
9	Customer point 9	(108,32)	5	10	16	9	17
10	Customer point 10	(130,88)	5	11	16.5	10	17
11	Customer point 11	(56,100)	5	11.5	17	11	17.5
12	Customer point 12	(50,68)	6	10	17.5	9.5	18
13	Customer point 13	(43,110)	18	13	15	10	16
14	Customer point 14	(135,24)	31	17	19	15	9.5
15	Customer point 15	(23,108)	8	10	19	9.5	20
A	Joint distribution transfer station A	(124,103)	0	7	20	7	20
B	Joint distribution transfer station B	(52,120)	0	7	20	7	20
C	Joint distribution transfer station C	(30,114)	0	7	20	7	20
D	Joint distribution transfer station D	(50,50)	0	7	20	7	20
E	Joint distribution transfer station E	(105,43)	0	7	20	7	20

Table 2: Description of relevant parameters.

Related parameters	Trucks in distribution centers	UAVs in joint distribution transfer stations
Quantity (vehicle, frame)	5	10
Maximum load (kg)	100	20
Maximum transport distance (km)	—	30
Speed (km/h)	30	80
Transportation cost per unit distance (yuan/km)	5	1
Start-up cost (yuan/time)	100	20
Service hours (h)	0.5	0.5

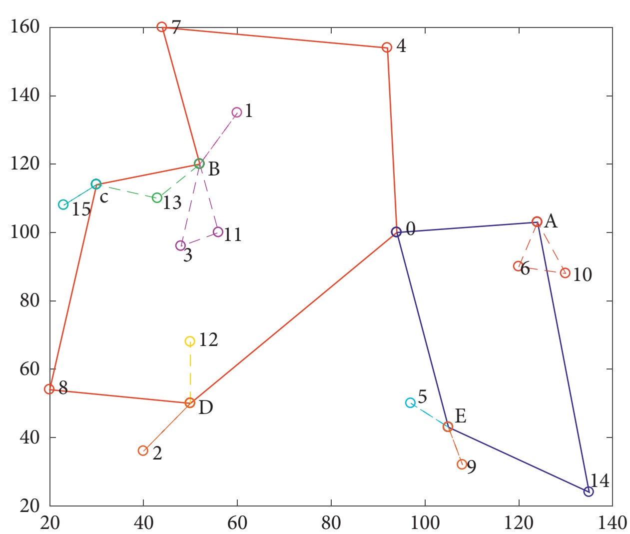


Figure 6: The optimal delivery scheme for joint delivery of trucks and UAVs by using ABC-ACA.

activated. There are three joint distribution transfer stations on the path of the first truck, and a total of six UAVs are used for delivery. There are two joint distribution transfer stations on the path of the second truck, and three UAVs are used for delivery. The delivery and satisfaction cost incurred during the joint delivery of trucks and UAVs process was RMB

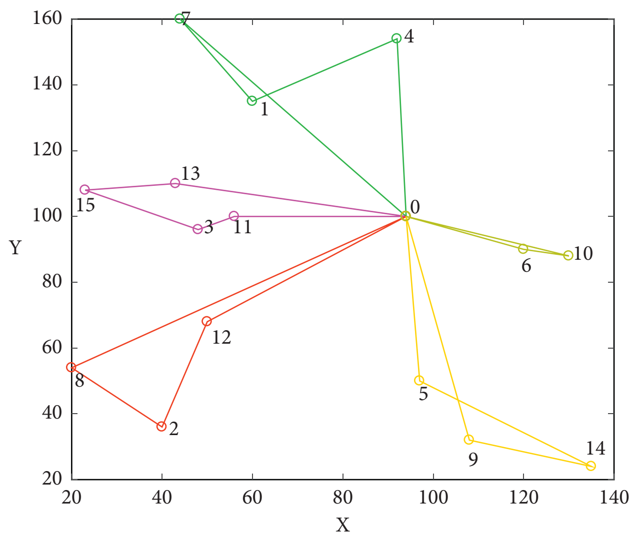


Figure 7: The optimal distribution scheme of trucks-alone distribution by using ABC-ACA.

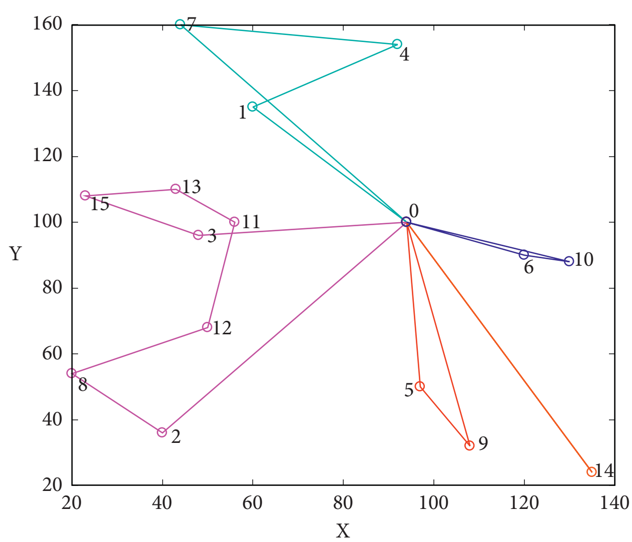


Figure 8: The optimal distribution scheme of trucks-alone distribution by using ACA.

Table 3: Comparison of results.

Delivery mode	Algorithm	Truck	Truck routes	UAV	UAV routes	Distribution and satisfaction cost (yuan)
Joint distribution of trucks and UAVs	ABC-ACA	1	0-4-7-B-C-8-D-0	1	B-1-B	1991.9
				2	B-3-11-B	
				3	C-13-B	
				4	C-15-C	
				5	D-12-D	
				6	D-2-D	
		2	0-A-14-E-0	7	A-6-10-A	
				8	E-9-E	
				9	E-5-E	

Table 3: Continued.

Delivery mode	Algorithm	Truck	Truck routes	UAV	UAV routes	Distribution and satisfaction cost (yuan)
Trucks-only distribution	ABC-ACA	1	0-4-7-1-0	—	—	2431.5
		2	0-13-15-3-11-0			
		3	0-8-2-12-0			
		4	0-5-14-9-0			
		5	0-6-10-0			
Trucks-only distribution	ACA	1	0-1-4-7-0	—	—	2824.7
		2	0-3-15-13-11-12-8-2-0			
		3	0-5-9-0			
		4	0-14-0			
		5	0-6-10-0			

1991.9 yuan, the average time dissatisfaction was 8.9%, and the average goods damage dissatisfaction was 6.3%. When the trucks are used for single delivery by using ABC-ACA, it necessarily starts five trucks, all of which start from the distribution center and return to it after completing the delivery tasks. The cost of delivery and satisfaction in the process of trucks-alone delivery is RMB 2431.5 yuan, the average time dissatisfaction is 45.2%, and the average goods damage dissatisfaction is 36.4%. When the trucks are used for single delivery by using ACA, it starts five trucks, all of which start from the distribution center and return to it after completing the delivery tasks. The cost of delivery and satisfaction in the process of trucks-alone delivery is RMB 2824.7 yuan, the average time dissatisfaction is 47.8%, and the average goods damage dissatisfaction is 38.6%. From the above analysis, it is obvious that the cost of the joint delivery mode of trucks and UAVs is lower than that of the trucks-alone delivery model, and the customer satisfaction of the joint delivery model is much higher than that of the trucks-alone delivery mode. Based on the comparison results, it can also be seen that the trucks-alone delivery mode solved by the ABC-ACA has better solution results and lower cost than the nonimproved ACA. Therefore, the joint delivery mode of trucks and UAVs has advantages not only in cost but also in meeting customer needs; that is, the ABC-ACA proposed in this paper is more efficient.

5. Conclusion

This paper proposes a joint delivery mode of trucks and UAVs for e-commerce delivery, which realizes the complementary advantages of UAVs and traditional delivery tools. A multiobjective optimization model for joint delivery is developed and the proposed model is solved by an improved ant colony algorithm. It proves that the joint delivery mode of trucks and UAVs is able to achieve lower costs, higher customer satisfaction, and better delivery results compared to the traditional truck-alone delivery mode. With e-commerce growing rapidly today, there is an urgent need for an efficient and economical delivery mode to meet market demand. As a new type of distribution tool, UAVs form a perfect complementary advantage with trucks, so that both distribution tools could give full play to their respective advantages and work together to achieve efficient and economic distribution. The joint delivery mode of trucks and UAVs provides an important tool for logistics companies to save resources, achieve economic benefits, and gain market competitiveness. The results of the research have laid a solid foundation for future research into the joint delivery of trucks and UAVs.

Although the content of this paper provides a certain reference value for the future joint delivery of trucks and UAVs, the model constructed in this article is based on certain assumptions and constraints and will face more uncertainties in reality, for example, constraints on carbon emissions and the existence of regional restrictions. Considering carbon emissions and regional restrictions in a joint delivery mode of trucks and UAVs will be the focus of our future research.

Data Availability

The data used are shown in Tables 1 and 2.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This paper was supported by Social Science Grand Research of Hebei Education Department (ZD202105).

References

[1] X. Ren, H. Huang, S.-W. Yu, S. Feng, and G.-Q. Liang, "Review on vehicle-UAV combined delivery problem," *Control and Decision*, vol. 10, pp. 2313–2327, 2021.

[2] Y. Wu, S.-B. Wu, and X.-T. Hu, "Cooperative Path Planning of UAVs & UGVs for a persistent surveillance task in urban environments," *IEEE Internet of Things Journal*, vol. 99, pp. 1–14, 2020.

[3] H.-J. Li, X.-K. He, J.-L. Song, and Y. Yang, "Comparison of global R&D of agricultural unmanned aerial vehicle, based on bibliometrics," *Journal of China Agricultural University*, vol. 9, pp. 154–167, 2021.

[4] J. Torres, F. Granados, A. Castro, and J. M. Pea, "Configuration and specifications of an unmanned aerial vehicle (UAV) for early site specific weed management," *PLoS One*, vol. 8, pp. 1–15, 2013.

[5] J. Zhang, X.-K. He, J.-L. Song, A.-J. Zeng, Y.-J. Liu, and X.-F. Li, "Influence of spraying parameters of unmanned aircraft on droplets deposition," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 43, pp. 94–96, 2012.
[6] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, "Internet-of-Things (IoT)-Based smart agriculture: toward making the fields talk," *IEEE access*, vol. 7, pp. 129551–129583, 2019.
[7] W.-T. Han, L.-Y. Zhang, Y.-X. Niu, and X. Shi, "Review on UAV romote sensing application in precision irrigation," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 2, pp. 1–14, 2020.
[8] M. Li, W.-J. Yang, X.-D. Yi, Y.-Z. Wang, and J. Wang, "Swarm robot task planning based on air and ground coordination for disaster search and rescue," *Journal of Mechanical Engineering*, vol. 55, no. 11, pp. 1–9, 2019.
[9] X.-M. Zhang, Y.-J. Hu, W.-G. Li, Q.-W. Pang, and G.-G. Yuan, "Multi-UAV fire fighting mission planning based on improved artificial bee colony algorithm," *Journal of Chinese Inertial Technology*, vol. 28, pp. 528–536, 2020.
[10] J. Kim, B. D. Song, and J. R. Morrison, "On the scheduling of systems of UAVs and fuel service stations for long-term mission fulfillment," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 347–359, 2013.
[11] P. Maini, K. Sundar, M. Singh, S. Rathinam, and P. B. Sujit, "Cooperative aerial–ground vehicle route Planning with fuel constraints for coverage applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3016–3028, 2019.
[12] Z.-F. Hu, Y. Chen, X.-J. Zheng, and H.-Y. Wu, "Cooperative patrol path planning method for air-ground heterogeneous robot system," *Control Theory & Applications*, vol. 10, pp. 1–13, 2021.
[13] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
[14] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.
[15] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman Problem with drone," *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.
[16] M. Mario, C. Leonardo, O. Michele, and D. Mauro, "En route truck–drone parcel delivery for optimal vehicle routing strategies," *Transport Systems*, vol. 4, pp. 253–261, 2018.
[17] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, 2017.
[18] Y.-C. Chu, X.-T. Wang, and N. Zhang, "Rural e-commerce logistics transportation path planning of truck combined with UAV," *Logistics technology*, vol. 9, pp. 82–88, 2020.
[19] M. Han, Y.-B. Wang, and L.-Y. Ding, "Vehicle +UAV distribution path optimization based on CTDEA algorithm," *Journal of weapon equipment engineering*, vol. 11, pp. 149–154, 2019.
[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co, New York, NY, USA, 1979.
[21] H. A. William, T. S. George, B. Ho, J. B. Ping, and C. W. Henry, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 548–557, 2008.
[22] J. Y. Shim, "A vehicle routing problem solved by using a hybrid genetic algorithm," *Computers & Industrial Engineering*, vol. 53, pp. 2959–2986, 2007.
[23] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, "A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *European Journal of Operational Research*, vol. 152, no. 1, pp. 148–158, 2004.
[24] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research*, vol. 41, no. 4, pp. 421–451, 1993.
[25] Y. Li, H.-M. Fan, and X.-N. Zhang, "Periodic optimization model and solution of vehicle routing problem under dynamic demand," *Chinese Journal of Management Science*, vol. 10, pp. 1–23, 2021.
[26] X.-B. Pei and X.-Y. Yu, "Research on Perishable Product distribution Problem based on improved artificial bee colony algorithm," *Journal of Lanzhou University (Natural Sciences)*, vol. 2, pp. 261–266, 2019.
[27] L. Li, S.-X. Liu, and J.-F. Tang, "Improved ant colony algorithm for solving vehicle routing problem with time windows," *Control and Decision*, vol. 25, pp. 1379–1383, 2010.
[28] Z.-H. Fu and C.-S. Liu, "Research on the opentime-dependent vehicle routing problem of fresh food e-commerce distribution," *Computer Engineering and Applications*, vol. 20, pp. 1–12, 2021.
[29] Z.-A. Hu, Y.-Z. Jia, and B.-W. Li, "Study on vehicle routing optimization considering customer satisfaction," *Industrial Engineering*, vol. 1, pp. 100–107, 2019.
[30] Y.-C. Hou, Y.-P. Bai, and H.-P. Hu, "Multi objective fresh food distribution path optimization based on elite ant colony algorithm," *Practice and understanding of mathematics*, vol. 20, pp. 50–57, 2018.
[31] T. Ren, Y. Chen, and Y. –C. Xiang, "Low carbon cold chain vehicle routing optimization considering customer satisfaction," *Computer integrated manufacturing system*, vol. 4, pp. 1108–1117, 2020.
[32] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
[33] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
[34] S. D. Li, X. M. You, and S. Liu, "Double ant colony algorithm based on ABC algorithm," *Computer engineering and application*, vol. 12, pp. 37–46, 2020.
[35] B. Yu, Z.-Z. Yang, and J.-X. Xie, "A parallel improved ant colony optimization for multi-depot vehicle routing problem," *Journal of the Operational Research Society*, vol. 62, no. 1, pp. 183–188, 2011.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6629449, 14 pages
https://doi.org/10.1155/2021/6629449

Research Article

Fuzzy Rule-Based Trust Management Model for the Security of Cloud Computing

Mona Soleymani,[1] **Navid Abapour**,[2] **Elham Taghizadeh**,[3] **Safieh Siadat**,[4] **and Rasoul Karkehabadi**[5]

[1]*Department of Computer Engineering, Islamic Azad University, Parand Branch, Tehran, Iran*
[2]*Department of Computer Science, Faculty of Science, University of Mohaghegh Ardabili, Ardabil, Iran*
[3]*Wayne State University, 4815 Fourth Street, Detroit 48202, MI, USA*
[4]*Department of Computer Engineering and Information Technology, Payame Noor University (PNU), P.O. Box 19395-4697, Tehran, Iran*
[5]*Department of Computer Science and Information Technology, Mahdishahr Branch, Islamic Azad University, Mahdishahr, Iran*

Correspondence should be addressed to Safieh Siadat; safieh.siadat@gmail.com

Received 27 December 2020; Revised 25 May 2021; Accepted 30 May 2021; Published 15 June 2021

Academic Editor: Ali Asghar Rahmani Hosseinabadi

In the last few years, due to the benefit of solving large-scale computational problems, researchers have been developed multicloud infrastructures. The trust-related issue in multiclouds includes more complicated content and new problems. A new trust management framework for multicloud environments is proposed in this article. The proposed framework used a combination of objective and subjective trust values to calculate the cloud service provider's trust values. This new framework can identify and rectify fake feedbacks from other feedbacks. Another advantage of this framework is applying fuzzy rules to calculate trust values. Two main components of the proposed framework are simulated in this paper. The simulation results confirm the important role of applied components. Also, this paper proposed a framework compared with other frameworks (feedback-based model, SLA-based model, and multicloud model). Simulation results show the proposed framework increased trust values rather than other models. Also, compared with other models, our framework gives better mean trust values.

1. Introduction

In recent years, cloud computing has attracted the attention of many researchers around the world, and various programs, infrastructures, and frameworks have been created for it by several companies in the world [1–4]. In fact, cloud computing, as a new technology, provides a fully scalable, accessible, and flexible computing platform for a variety of applications [5]. Due to the various applications that cloud computing has found in various aspects of life, the issue of providing security in cloud computing communications and data stored in it, has been considered by users and providers of cloud computing services. According to some research conducted at Berkeley University, trust management and security optimization have been identified as the most important issues in using various cloud computing services [3, 6–8]. Cloud computing due to its distributive nature, very dynamic space, and lack of transparency in performing cloud computing faces many challenges in providing security and gaining trust. In order to improve security in performing cloud computing, trust management can play a very effective role [9, 10].

This article is organized into five sections. In the second part, we examine some of the related work done by various researchers. In the third section, we present a new framework for trust management in multicloud environments. In the fourth section, we bring the simulation results of the framework presented in this research, and finally, in the fifth section, we will conclude.

In this paper, we have tried to present a trust management model in multicloud environments that uses both objective and subjective parameters to calculate trust values. And it looks at the category of trust instead of a one-dimensional category in a multidimensional way and selects and assigns the cloud service provider that is closest to the cloud service user request. In this model, an attempt has been made to design a component to detect fake feedback and distinguish fake feedback from the real one and use only real feedback to calculate trust values, which leads to increasing the accuracy of the proposed framework, so, in this paper, for the first time, the calculation of multidimensional trust values in multicloud environments is presented along with the detection of fake feedback.

In previous models and methods, trust has been considered as a one-dimensional category, whereas trust is a multidimensional and relative value, and for each individual, some parameters of service quality are important. Someone needs to have accessibility, someone else reliability, the other security, and so on. And after receiving the service, the feedback will also vary according to the importance of each of these parameters. Therefore, trust cannot be viewed as a one-dimensional parameter, and it should be calculated multidimensional, and in this model, based on the type of user's request and prioritizing the parameters of the cloud service provider that is closest to the request, it is selected. And also, after receiving the service despite the appropriate cloud service, users may generate fake feedbacks in order to reduce the trust values of cloud service providers, in the proposed framework which has been tried to provide a component for detecting fake feedbacks in this component with higher accuracy and calculate the trust values. So, in general, it can be stated that in other previous studies, the issue of trust was viewed as a one-dimensional way, and besides there was no appropriate mechanism to detect fake feedback in this study for the first time.

2. Related Works

The level of trust and confidence in cloud service providers is one of the important parameters to provide a reliable service for the cloud service user. Liu and colleagues [11] proposed a method in which reliable cloud service providers for SaaS applications were selected based on their credibility and trust [12]. Many of the proposed models for measuring trust are based on records of trust in various cloud service providers [13, 14]. Accordingly, these models can be divided into two general categories, which are subjective and objective trust models [15].

To measure the level of objective trust, parameters related to the quality of service (QoS) delivery are used. Fan and colleagues [16] proposed a concept called "objective trust" for software agents. The researchers explained the trust between agents based on real-life experiences. Lin et al. [17] proposed a new framework for MANET networks in which one node evaluates the reliability of another node using direct observations. To calculate the level of trust in the subjective method, we can use the amount of feedback received from users using various cloud services [16]. Uikey et al. [18] proposed a new trust management model in which all information about different cloud service providers and the level of trust is recorded and stored. In this study, SLA models were used to calculate the reliability of cloud service providers. Alhamad et al. [3] proposed a new SLA-based trust management model to predict the level of trust in cloud service providers. In the proposed model, SLA-based conceptual framework is integrated with a trust value management. Chakraborty et al. [19] applied the parameters extracted using the SLA to measure the reliability of cloud services. Some of the most commonly used SLA-based models are probability-based trust model, intuitive reasoning-based trust model [20], Bayesian-based trust model, Dempster–Shafer model, Fuzzy logic-based trust [21], cloud computing trust model [22], and so on. Siadat et al. [23] proposed a new model for managing trust in cloud computing that uses game theory to detect fake feedback [24]. Chen et al. [25] proposed a new trust management model for the Internet of Things (IoT) in which trust management at different levels of the IoT was examined. In the study by Guo et al. [26], a new model for managing trust in the IoT suggests that methods for assessing trust are examined based on five common design dimensions (including trust composition, dissemination, aggregation, updating, and shaping). Din et al. [27] examined trust management methods without performing any classification. Various studies have been conducted to combine methods of objective and subjective trust. Yuan et al. [28] proposed a framework for assessing trust that uses a combination of objective and subjective trust methods that calculate and rate trust based on a combination of users' trust and credibility. Ngo et al. [29] examined the relationship between the level of objective trust and subjective trust and expressed the characteristics of each of these two types of trust.

Sangaiah et al. [30] with using machine-learning techniques proposed a new method to maintain the confidentiality of the geographical location of PBS portable users. The proposed method had three phases. During these phases, using the integration of decision tree techniques and the nearest neighbor, the user's geographical location was determined, and using the sequence of routes transferred and using hidden Markov models, the user's destination was identified. Along with maintaining the confidentiality of users' position, these researchers showed that the accuracy of this method in establishing position in PBS was equal to 90%. The results of the implementation of the proposed method by these researchers showed that the accuracy of this method in establishing position confidentiality in PBS was equal to 90%.

Sangaiah et al. [31] defined a weight called relay ability for each node according to the sensor network topology. These weights are calculated by the head and reported to all sensor nodes. When a target enters the area covered by sensor nodes, a signal is sent to CH via a path that has a predefined maximum weight in the network. The simulation of the proposed method in this research showed that this method has better results than other tracking methods based on the criteria of network power consumption, power consumption and power for GRTT, dynamic energy efficient routing (DEER) protocol, and virtual power-based energy consumption (VFEM).

In the study by Sangaiah et al. [32], an energy-aware green adversary model has been proposed for use in intelligent industrial environments by achieving confidentiality. In this study, researchers explored various aspects of preserving geographic location information and information confidentiality. Finally, we proposed a new model which has the capacity to make prediction based on a schedule in real-time situations, it can make connections, respond to user demands, and minimize energy consumption. The experimental results of these researchers showed that their proposed model can be five times more energy saving compared with other methods.

Mousa et al. [33] used a trust model based on the fuzzy logic system to evaluate trust values. They proposed a new method for this purpose, which gives cloud users the ability to assess the reliability of cloud service providers. Simulations of their proposed method showed that the accuracy of the evaluations performed by these researchers was higher compared with other works.

Sule et al. [34] using a combination of fuzzy logic and several different security mechanisms (such as identification and trust) proposed a multilayered security model based on an integrated cloud platform. The simulation of the proposed model showed that this model can provide the possibility of determining and verifying the trust status of the cloud computing service. Therefore, this model can be used to improve end-user confidence when selecting or consuming cloud computing resources.

Fan et al. [35] assessed the objective and subjective trust of CSPs. Using different clouds, they proposed a trust dissemination network among TSPs. This network can be used by the TSP to obtain trust information about a service from other TSPs. The researchers also proposed a framework for trust management in a multicloud environment based on a trust assessment and a proposed trust dissemination network. The results of their experiments showed that the proposed framework is more reliable than other CSPs in a multicloud environment.

In the study by Kumar et al. [36], a fuzzy-based trust management system was proposed to facilitate cloud service delivery and identify trusted providers. The proposed system simulation showed that the degree of reliability and trust of the proposed system was higher compared with that of other CSPs.

Over the past few years, numerous studies have been conducted on multicloud infrastructure and cloud computing environments to solve large-scale computing problems. The problems associated with building and managing trust across multiple clouds are vast. Nielsen et al. [37] proposed a new framework for trust management in multicloud environments for cloud service providers. Among the models offered for trust management, only a few of them have considered trust in a multidimensional way, and for this purpose, they have combined objective and subjective trust. Also, a limited number of these models are able to detect fake feedback models. The main purpose of this paper is to provide a framework for trust management in multicloud environments using fuzzy logic to enhance security in these networks. The research questions that this research follows are how can trust management help to enhance security in multicloud networks? Is it possible to calculate trust values from the combination of subjective parameters and objective parameters together using fuzzy logic? How can we distinguish fake feedback from nonfake feedbacks in the proposed framework to increase the accuracy of trust values?

3. Our Proposed New Trust Management Framework in Multicloud Environments

We propose a new framework for trust management using cloud service providers (TSPs). We try to cover several problems that exist in the field of trust management in multicloud environments with this proposed framework. The trust management framework proposed in this study is seen in Figure 1.

CSPs (cloud service providers) are responsible for providing services to cloud service users. In cloud computing, a variety of services are provided to users by CSPs. The most common types of services are SaaS, PaaS, and IaaS. CSPs also provide services to CSUs (cloud service users). CSUs send their requests to TSP (which is one of the CSPs). The selection of this TSP among CSPs is done by different selection algorithms. The main task of TSP is to select the appropriate CSP to receive and respond to requests sent from CSU. Another function of TSP is to verify CSP reliability.

3.1. SLA Monitor Agent. SLA monitor agent is located on CSUs side. It monitors services behavior and services performance that if CSUs meet SLA or not. SLA monitor agent collects data in the interaction between CSPs and CSUs and also its responses to control requests. A control request is sent by TSP. SLA monitor agent continuously collects control information from server side. Control information contains SLA performance parameters.

3.2. Monitoring Information Collection Agent. The responsibility of monitoring information collection agent is collecting monitor agents information on SLA which has an agreement with TSP. Collected information by this agent is applied to evaluate objective trust values. The following information is maintained by the monitoring information collection agent:

(i) CSPs list is monitored by TSP
(ii) SLA monitoring information received from SLA monitor agent that is in agreement with TSP

Before receiving the CSUs service from the CSPs, an SLA contract is agreed between them with various parameters. This SLA contract is the output of the SLA negotiation component, which determines the level of service that the CSUs and CSPs agree on. Some of the parameters in the SLA are availability, response time, and so on, which are agreed upon, based on which the server agreements that are closest to the CSUs request are selected in the next steps.

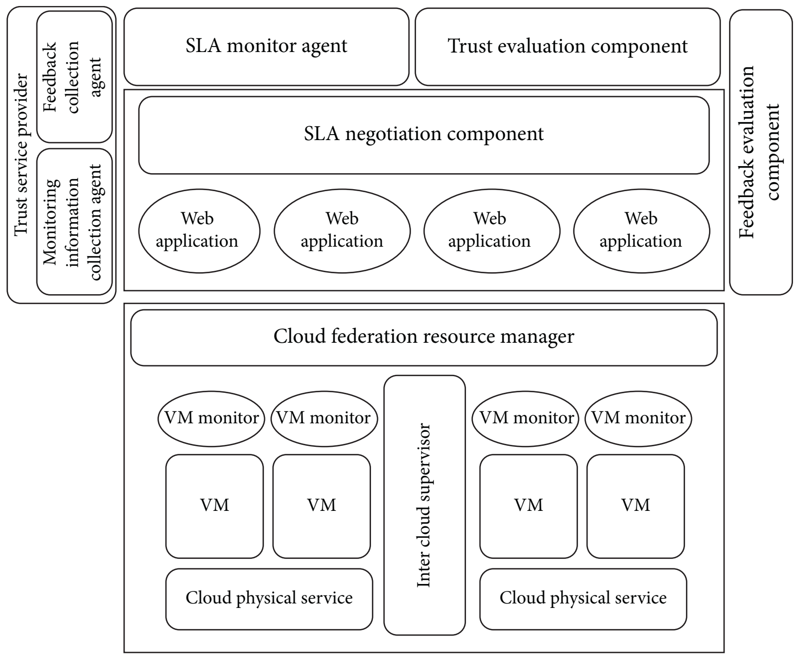


Figure 1: Our proposed trust management framework in multicloud environments.

3.3. SLA Negotiation Component. SLA negotiation component negotiates between CSUs and CSPs. SLA with multiple CSPs negotiation is a complicated work. Usually, the third part does this task. In this paper, the name of this third part is the SLA negotiation component. Figure 2 illustrates the SLA negotiation component.

SLA negotiation component includes the following agents:

(i) Registry agent
(ii) Provider agent
(iii) Negotiation agent
(iv) Trust assessment agent to requests submitted by users
(v) Mediator agent
(vi) Client agent

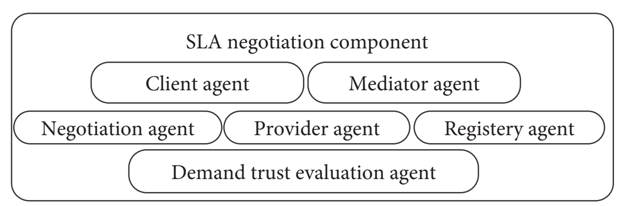


Figure 2: SLA negotiation component.

3.3.1. Registry Agent. The task of REA is to register CSPs information together with their services.

3.3.2. Provider Agent. PA represents CSPs. All negotiations between CSUs and CSPs do by their representatives. Negotiations such as SLA agreement and cost agreement are done by the provider agent.

3.3.3. Negotiation Agent. NA task is SLA production and SLA optimization. CSPs register their services by NA. CSUs provide CSPs SLA details by NA. After successful negotiation between CSP and CSU, they register the negotiation process to investigate and future validation. Negotiation process registers if negotiation between CSP and CSU is finished or not. NA registers negotiation process at specific period of time.

Stored information by NA is as follows:

(i) One directory of CSPs services (CSP may provide multiple services on multiple clouds but for simplicity assumed that CSP provides one service in multiple clouds)
(ii) A catalog of service request from CSP
(iii) In each CSU, service negotiation includes negotiation requester, negotiation start and end time, negotiation content, and negotiation result

3.3.4. Demand Trust Evaluation Agent. This agent is responsible for receiving requests sent from users and sends these requests to the best available server based on the type of service requested. To achieve this goal, the parameters related to the evaluation of the quality of services (QoS) provided to users in the platform are initialized. This agent selects the server that best matches the services requested by the user to respond to the reliability assessment factor to the requests submitted by the user in batches. This factor performs a two-step evaluation of the reliability of the requests submitted by users:

(i) Trust evaluation using the parameters provided in the user-submitted request

(ii) Select the best server to respond to requests and provide the requested services optimally and assign the request to that server

The various parts of the reliability assessment agent to the requests submitted by users are seen in Figure 3.

(1) Trust Evaluation with Demand Parameter Level. In this section, to evaluate the level of trust in the user-submitted requests, several servers are selected as candidates to provide the user requested services. Several parameters affect the quality of services (QoS) provided, which are as follows:

(i) Duration of delay in providing requested services

(ii) Duration of response to the request

(iii) Accuracy of providing service

(iv) The amount of cost requested to provide the requested services of users

Each user sets an initial value for each of the mentioned parameters based on their preferences. To evaluate the amount of trust using the mentioned parameters, it is calculated by accessing the trust repository. For each calculation, the cloud service providers denoted by p have the maximum value defined for reliability and are selected as candidates. Requests submitted by users will be sent to these candidates. Formulas (1) to (5) include all of the steps mentioned above.

$$\mathrm{DP} = (\mathrm{dp}_1, \ldots, \mathrm{dp}_m), \tag{1}$$

$$w_i = \frac{\mathrm{dp}_i}{\sum_{i=1}^{m} \mathrm{dp}_i}, \quad \sum w_{i=1}, \tag{2}$$

$$\mathrm{dtv}_i = \sum_{i=1}^{n} \sum_{j=1}^{m} w_j, p_{j,i}, \tag{3}$$

$$\mathrm{DTV}_j = (\mathrm{dtv}_1, \mathrm{dtv}_2, \mathrm{dtv}_3, \ldots, \mathrm{dtv}_n). \tag{4}$$

$$\mathrm{DTV} = \begin{bmatrix} \mathrm{DTV}_1 \\ \mathrm{DTV}_2 \\ \mathrm{DTV}_3 \\ \cdot \\ \cdot \\ \mathrm{DTV}_k \end{bmatrix}. \tag{5}$$

In formula (1), the DP shows a list of parameters that have been initialized by the user. In formula (2), w_i represents the weight assigned to each parameter by the user. In formula (3), dtv_i shows the amount of trust in the requests provided by the service providers and m indicates the number of parameters used. Formula (4) uses a dtv_i storage for DTV_j for each request and n the number of source nodes. In formula (5), DTV is an array of DTV_j while k is the size of each batch of requests. We calculate and save DTVs in batches.

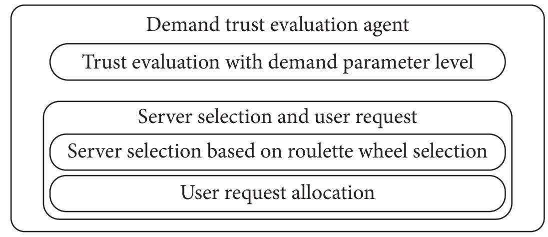


Figure 3: Trust assessment agent to requests submitted by users.

(2) *Server Selection and User Request.* In this section, the most appropriate cloud service provider is selected and requests are sent from users to the selected service provider. In this part, two series of operations are performed as follows:

(i) *Selecting the Most Suitable Server Based on the Roulette Wheel Mechanism.* In this research, the roulette wheel mechanism is used to select the best and most appropriate service provider. The reason for using this mechanism is to create a load balance among all cloud service providers. Formulas (6) to (10) calculate the number of user requests (in percentage) allocated to each cloud service provider. In formula (6), m represents the number of parameters used, w represents the weight assigned to each parameter, and P_i represents the value of the parameter stored in the trust repository. In formula (7), w represents the weight assigned to each parameter. In formula (8), the value of dtv_i is stored in the T.V array. Sp_i is the percentage of user requests assigned to the i server in formula (9). Finally, in formula (10), the value of Sp_i is stored in the SP array.

$$w = \frac{1}{m}, \tag{6}$$

$$\mathrm{t.v}_i = \sum_{i=1}^{n} \sum_{j=1}^{m} w \times p_{j,i}, \tag{7}$$

$$\mathrm{T.V} = (\mathrm{tv}_1, \mathrm{tv}_2, \ldots, \mathrm{tv}_n), \tag{8}$$

$$\mathrm{sp}_i = \frac{\mathrm{tv}_i}{\sum_{i=1}^{n} \mathrm{tv}_i}, \tag{9}$$

$$\mathrm{SP} = (\mathrm{sp}_1, \mathrm{sp}_2, \ldots, \mathrm{sp}_n), \quad \sum_{i=1}^{n} \mathrm{sp}_i = 1. \tag{10}$$

(ii) *Assigning the User Request to the Selected Server.* In this section, using SP and roulette wheel mechanism, we will select the best server from the candidate servers selected in the previous step so that we can assign the request submitted by the user to it.

3.3.5. Mediator Agent. Mediator agent (MA) extracts available CSP from RA. MA sends available CSPs list to demand trust evaluation agent (DTEA). MA contacts between client agent and provider agent. MA determines services price by this contact. MA transfers selected CSPs by CSP selecting agent and load balancing agent with prices to the client agent. The client agent selects one of CSP among CSPs. MA sends selected CSP to the negotiation agent.

3.3.6. Client Agent. The client agent receives the service request of CSU. The service request of CSU has contained QOS parameters. Client agent sends service request of CSU to MA.

3.4. Feedback Evaluation Component. A new feedback evaluation component has been presented in this paper. This component evaluates and updates the received feedback from CSU after receiving service. It qualitatively identifies and rectifies fake feedback. This new component also prevents circumvention, collusion, latency, and impersonation attacks.

(i) *Feedback*. Feedback means the text or reaction that the user sends to the server after receiving the requested service. This feedback can be used to predict the reliability of CSPs. Some attacks are carried out by hackers to change the content of the feedback, which makes the CSPs less reliable. Some of these threats against feedback include circumvention, collusion, delaying, and impersonation.

(ii) *Circumvention*. When CSU requests a regular service from the cloud service provider and after receiving the service, it sends a negative feedback to the server. In this case, the CSU reduces the amount of trust in the CSP with this feedback. To avoid reducing trust in CSP, we identify this type of feedback as fake feedback.

(iii) *Collusion*. Collusion has occurred when several unauthorized and malicious users unite and attack a single cloud service provider. The method of attack of these users is malicious, by sending fake feedback to service providers. This type of attack is very difficult to detect using traditional methods.

(iv) *Delay*. When some CSUs repeatedly send a request to receive a service that is very time consuming to respond, it is said that the attack has been delayed. CSU sends negative feedback based on the delay. CSUs are forced to waste a lot of time servicing those requested services. Therefore, they will not be able to serve other CSUs, and eventually there will be a delay. Identifying this type of malicious user using classical methods is very difficult or sometimes impossible.

(v) *Impersonation*. In several cases, some users may impersonate other users and send fake feedback instead. To combat these threats, CSU checks users' identities and performs authentication operations for them. In addition, the role of each user in the interactions performed and the requested or received services must be confirmed.

The components used to evaluate the feedback are shown in Figure 4, which are as follows:

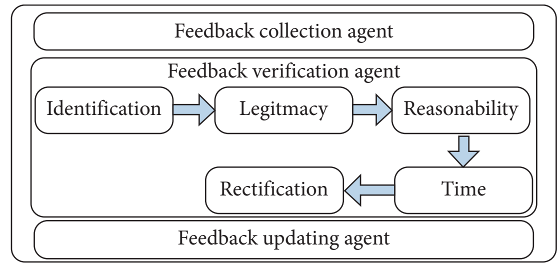


Figure 4: Feedbacks evaluation component.

(i) An agent for collecting feedback
(ii) An agent to confirm the feedback received
(iii) The agent of updating the feedback received

3.4.1. Feedback Collection Agent. The agent collects feedbacks and sends them to the verification of received feedback agent.

3.4.2. Feedback Verification Agent. This agent is responsible for identifying and correcting negative feedback. Operations in this agent include identifying negative feedback, legitimizing that feedback, reasoning, timing, and correcting that feedback.

(1) Identification. To identify negative feedback, we must first identify the credibility of the user who requested the cloud service. All CSUs have a unique identifier and must register their actual details when submitting a request. The feedback agent must verify this ID. In this case, we first check the CSU ID that sends the feedback. If the CSU ID is verified, the feedback will be labeled "valid"; otherwise, it will be identified as "fake feedback." Fake feedback is immediately discarded or removed.

(2) Legitimacy. The term legitimacy refers to the validity or value of the feedback received. To check the legitimacy, we need to check the unique CSU ID and the CSP ID during the transaction. Legitimacy prevents unauthorized users from infiltrating the cloud and prevents these unauthorized users from being able to send feedback over the network.

(3) Time Threshold. An evaluation and verification factor considers a threshold period after the transaction to receive feedback from the client. If feedback is received before or after the defined time limit, it will be ignored or deleted immediately.

(4) Reasonability. One of the most complex aspects of feedback validation is that it is logical as there is no set point for evaluating it. To identify and validate the logic of the feedback received, we compare the current feedback with previous feedback or feedback. If the difference between these feedbacks is less than Delta, the feedbacks are considered reasonable; otherwise, they are either ignored or need to be corrected.

(5) Rectify. If the previous methods (detection of negative feedback, use of legitimacy, or time limit) confirm that the feedback is fake, that feedback will be ignored and discarded. However, if the review of feedback shows that it is logically problematic, we should try to correct that feedback. Formulas (11) to (14) are used to correct fake feedback.

$$a_i = \frac{1}{m}\sum_{k=1}^{m} f_{\mathrm{CSP}_{i,k}}, \quad (11)$$

$$m = \min\left(m, \mathrm{feedbackcountCSP}_i\right), \quad (12)$$

$$F'_{\mathrm{CSP}_i} = a = \left(f_{\mathrm{CSP}_i} - a\right) \times r_{\mathrm{CSU}_0}, \quad (13)$$

$$r_{\mathrm{CSU}_0} = \frac{r_{\mathrm{CSU}_j}}{\sum_{k=1}^{n} r_{\mathrm{CSU}_k}}. \quad (14)$$

In formula (11), a_i presents the previous m feedback average, $f_{\mathrm{CSP}_{i,k}}$ is the CSP_i feedback, and the k index represents the previous k feedback of CSP_i. m is obtained from formula (12). Feedback count CSP_i is the number of received feedback from CSP_i

In formula (13), $F_{\mathrm{CSP}_i}'$is the rectified feedback of CSP_i and r_{CSU_0} can be obtained from formula (14). In formula (14), r_{CSU_j} is the *jth* CSU reliability and n represents the number of cloud service users. When CSU_j sends true feedback, r_{CSU_j} increases, and when it sends fake feedback, r_{CSU_j} decreases. The algorithm of the feedback verification agent is shown in Figure 5.

3.4.3. Feedback Updating Agent. The role of this agent is to update the feedback received from the feedback confirmation agent.

3.5. Trust Evaluation Component. The responsibility of the trust assessment component is to calculate the amount of subjective trust and the amount of objective trust. To calculate the amount of subjective trust and the amount of objective trust from the information collected, we use the supervised data collection agent and the feedback evaluation component. After calculating the amount of trust, another task of the trust assessment component is to update the amount of trust stored in the trust repository. The component of trust assessment includes the following factors:

(i) An agent to evaluate the amount of subjective trust

(ii) An agent for evaluating the amount of objective trust

(iii) An agent for updating the amount of trust

Figure 6 shows the trust evaluation component.

Trust repository includes CSPs trust values that they calculated and stored. Trust values will be used as a trust evidence for future decisions.

Trust modeling methodology assumptions are as follows:

(i) Each CSP only can present one type of service

(ii) Each CSU uses the terms Trust (T), Distrust ($-T$), and Uncertain (T, $-T$) for trust values that they obtain by fuzzy rules after calculating trust values

1 feedback verification agent (receives feedback from feedback collection agent)
2 Begin
3 for i = 1 to m do // $m = \min(m, \text{feedback\_count\_CSP})$
4 $a = \frac{1}{m}\sum_{i=1}^{l} f_{\mathrm{CSP}_i}$
5 If $(f_{\mathrm{CSP}_i} - a > \delta)$ then //else don't do anything
6 rectify ()
7 **End**

Figure 5: Feedbacks verification agent.

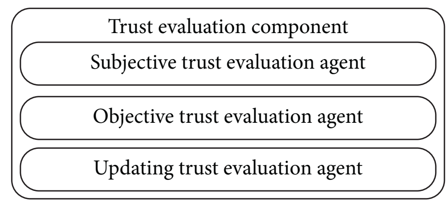


Figure 6: Trust evaluation component.

3.5.1. Subjective Trust Evaluation Agent. Subjective trust values are calculated based on revived feedbacks. The responsibility of subjective trust values agent is receiving CSPs feedbacks that exist in their domain. Subjective trust values evaluation agent receives the feedbacks from the feedback evaluation component, and also it receives the parameters of requested service with their weights from the trust negotiation component. Subjective trust values evaluation agent calculates CSU service satisfaction by using received feedbacks and requested service parameters. CSU service satisfaction is calculated by using formulas (15) to (20).

$$w = (w_1, w_2, \ldots, w_m), \quad (15)$$

$$F' = \sum_{i=1}^{m} F_{p_i} \times w_i. \quad (16)$$

In formula (15), we have trust negotiation component. In formula (16), F_{pi} is received from feedbacks evaluation component. F_{p_i} are feedbacks of each parameter. W_i in formula (16) is obtained from formula (2) and m is the number of parameters.

$$\forall i, \quad i = 1, \ldots, m, P_{dei} = F_{pi}. \quad (17)$$

Formula (18), calculate CSU service satisfaction w_i and p_{dmi} get from trust negotiation component and p_{dei} obtains from formula (17).

$$\overline{f}_{i,j} = \sum_{i=1}^{k} w_i \frac{\left|p_{dmi} - p_{dei}\right|}{p_{dmi}}. \quad (18)$$

In formula (19), $\mathrm{st}_{i,j}^{t}\{T\}$ is subjective trust value of ith service requested related to CSP_j in time t. $\mathrm{st}_{i,j}^{t}\{T\}$ is subjective trust value in trust state, $\mathrm{st}_{i,j}^{t}\{-T\}$ is subjective trust value in distrust state, and $\mathrm{st}_{i,j}^{t}\{\mathrm{U}\}$ is subjective trust value in uncertain state.

$$\mathrm{ST}_{i,j}^{t}=\begin{cases}\mathrm{st}_{i,j}^{t}\{T\}=\dfrac{\bar{f}_{i,j}^{t}-0.5}{0.5}, & \text{if } \bar{f}_{i,j}^{t}>0.5,\\ \mathrm{st}_{i,j}^{t}\{-T\}=\dfrac{0.5-\bar{f}_{i,j}^{t}}{0.5}, & \text{if } \bar{f}_{i,j}^{t}<0.5,\\ \mathrm{st}_{i,j}^{t}\{U\}=1-\mathrm{st}_{i,j}^{t}\{T\}-\mathrm{st}_{i,j}^{t}\{-T\} & \text{else,}\end{cases} \tag{19}$$

$$\mathrm{LST}_{i,j}^{t}=\begin{cases}\left\{\mathrm{lst}_{i,j}^{t}\{T\}=\mu\times \mathrm{st}_{i,j}^{t-1}(T)+(1-\mu)\times\dfrac{\bar{f}_{i,j}^{t}-0.5}{0.5},\right.\\ \mathrm{lst}_{i,j}^{t}\{-T\}=\mu\times \mathrm{st}_{i,j}^{t-1}(-T)+(1-\mu)\times\dfrac{\bar{f}_{i,j}^{t}-0.5}{0.5},\\ \mathrm{lst}_{i,j}^{t}\{U\}=1-\mathrm{lst}_{i,j}^{t}\{T\}-\mathrm{lst}_{i,j}^{t}\{-T\},\end{cases} \quad 0<=\mu<=1. \tag{20}$$

In formula (20), LST is the local subjective trust value. $\mathrm{lst}_{i,j}^{t}\{T\}$ is local subjective trust value in trust state, $\mathrm{lst}_{i,j}^{t}\{-T\}$ is local subjective trust value in distrust, state and $\mathrm{lst}_{i,j}^{t}\{U\}$ is subjective trust value in an uncertain state. In formula (20), μ is the weight factor. If $t=0$, then the subjective trust value of CSU_i related to CSP_j is zero as follows:

$$\mathrm{LST}_{i,j}^{0}=(0,0,1). \tag{21}$$

(1) Calculating Subjective Trust Values. Using Formula (22), global subjective trust (GST) value is calculated; GST_j^t is CSPj subjective trust value of in time t. $\mathrm{gst}_j^t\{T\}$ is the global subjective trust value in trust state, $\mathrm{gst}_j^t\{-T\}$ is the global subjective trust value in distrust state, and $\mathrm{gst}_{i,j}^t\{U\}$ is the global subjective trust value in an uncertain state.

$$\mathrm{GST}_{j}^{t}=\begin{cases}\mathrm{gst}_{j}^{t}\{T\}=\dfrac{\sum_{i=1}^{n}\mathrm{lst}_{i,j}^{t}(T)}{n},\\ \mathrm{gst}_{j}^{t}\{-T\}=\dfrac{\sum_{i=1}^{n}\mathrm{lst}_{i,j}^{t}(-T)}{n},\\ \mathrm{gst}_{j}^{t}\{U\}=1-\mathrm{gst}_{i,j}^{t}\{T\}-\mathrm{gst}_{i,j}^{t}\{-T\}.\end{cases} \tag{22}$$

3.5.2. Objective Trust Values Calculating Agent. CSP ensures a certain level of service performance for CSU that the deal was agreed earlier in the SLA. Services performance will be measured by parameters set. This parameter was proposed by Habib et al. [38].

Based on the study by Habib et al. and different requirements in the industry, many parameters should be investigated (objective trust values parameters). These parameters are availability, reliability, response time, security, privacy, transparency, and consumer protection [39].

Objective trust values evaluate based on measured parameters that whether they met SLA or not. That process is done by an SLA monitoring agent.

Assumptions are as follows:

M is the parameter set used in SLA between CSP and CSU.

$n_{i,j}$ is the number of transactions in w window that t is between 1 and n ($1\le T<=N$).

For each transaction between CSP and CSU, TSP receives collected records for all parameters of SLA. TSP decides the statuses of those records. The statuses of received service are T, $-T$, and U.

$n_{i,j,k}^{\mathrm{succ}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been satisfied in SLA.

$n_{i,j,k}^{\mathrm{failled}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been dissatisfied in SLA.

$n_{i,j,k}^{\mathrm{UN}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been located uncertain state in SLA.

In time $t=0$, there does not exist transaction between CSP and CSU. So, $n_{i,j,k}^{\mathrm{succ}}=0$, $n_{i,j,k}^{\mathrm{failled}}=0$, and $n_{i,j,k}^{\mathrm{UN}}=0$.

Local objective trust value is calculated by using formulas (23) and (24). Local objective trust value is calculated for window T, and it is calculated based on k parameters from SLA.

In formulas (24) and (25), $\mathrm{LOT}_{i,j,k}^{t}$ is objective trust value of CSUi related to CSPj in time t based on k parameter. μ is weight factor in formulas (24) and (25).

$$\mathrm{LOT}_{i,j,k}^{t} = \left(\mathrm{lot}_{i,j,k}^{t}\{T\}, \mathrm{lot}_{i,j,k}^{t}\{-T\}, \mathrm{lot}_{i,j,k}^{t}\{U\}\right), \tag{23}$$

$$0 < = \mu < = 1,$$

$$\mathrm{LOT}_{i,j,k}^{t} = \begin{cases} \mathrm{lot}_{i,j,k}^{t}\{T\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(T) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{succ}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}, \\ \mathrm{lot}_{i,j,k}^{t}\{-T\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(-T) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{failled}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}, \\ \mathrm{lot}_{i,j,k}^{t}\{U\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(U) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{UN}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}. \end{cases} \tag{24}$$

In time $t=0$, $\mathrm{LOT}_{i,j,k}^{0} = (0,0,1)$.

In formula (25), $\mathrm{LOT}_{i,j,k}^{t}$ is CSUi objective trust value related to CSPj in time t.

$$\mathrm{LOT}_{i,j}^{t} = \begin{cases} \mathrm{lot}_{i,j}^{t}\{T\} = \sum_{k-1}^{m} \omega_k \mathrm{lot}_{i,j,k}^{t}(T), \\ \mathrm{lot}_{i,j}^{t}\{-T\} = \sum_{k-1}^{m} \omega_k \mathrm{lot}_{i,j,k}^{t}(-T), \\ \mathrm{lot}_{i,j,k}^{t}\{U\} = 1 - \mathrm{lot}_{i,j}^{t}\{T\} - \mathrm{lot}_{i,j}^{t}\{-T\}, \end{cases} \tag{25}$$

$$\sum_{k=1}^{m} \omega_k = 1, \tag{26}$$

where $\mathrm{lot}_{i,j}^{t}\{T\}$ is objective trust value probability in trust status from CSUi related to CSPj in time t. $\mathrm{lot}_{i,j}^{t}\{-T\}$ is objective trust value probability in distrust status from CSUi related to CSPj in time t. $\mathrm{lot}_{i,j}^{t}\{U\}$ is objective trust value probability in uncertain status from CSUi related to CSPj in time t.

(2) Calculating Global Objective Trust Values. Global objective trust values for CSPj are obtained by local objective trust values combination. Global objective trust values are obtained from the local objective trust values average that is shown in formula (27). got_{j}^{t} is global objective trust values in time t [21].

$$\mathrm{GOT}_{j}^{i} = \begin{cases} \mathrm{got}_{j}^{t}\{T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lot}_{i,j}^{t}(T)}{n}, \\ \mathrm{got}_{j}^{t}\{-T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lot}_{i,j}^{t}(-T)}{n}, \\ \mathrm{got}_{j}^{t}\{U\} = 1 - \mathrm{got}_{i,j}^{t}\{T\} - \mathrm{got}_{i,j}^{t}\{-T\}. \end{cases} \tag{27}$$

3.5.3. Calculating Trust Values. Trust values will be calculated in the proposed framework by using fuzzy rules that fuzzy inputs are objective trust values and subjective trust values and fuzzy output is trust values.

(i) Fuzzy inputs contain three states: low, medium, and high, and their ranges are between 0 and 1

(ii) Fuzzy outputs contain three states: low, medium, and high, and their ranges are between 0 and 1

Some fuzzy rules are as follows:

$$\begin{aligned} &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{low}\right) \mathrm{and}\ \mathrm{GST}_{j}^{t}(T) = \mathrm{lo}\ \mathrm{then}\ \mathrm{TV}(T) = \mathrm{low}, \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{low}\right) \mathrm{and} \left(\mathrm{GST}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{then}\ \mathrm{TV}(T) = \mathrm{low}, \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{and} \left(\mathrm{GST}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{then}\ \mathrm{TV}(T) = \mathrm{medium}, \\ &\qquad \cdots \\ &\qquad \cdots \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{high}\right) \mathrm{and} \left(\mathrm{GST}_{j}^{t}(T) = \mathrm{high}\right) \mathrm{then}\ \mathrm{TV}(T) = \mathrm{high}. \end{aligned} \tag{28}$$

Trust Values Updating Agent. Trust values updating agent task is updating trust repository. Trust value is obtained by fuzzy rules map in three states: trust, distrust, and uncertain. If obtained trust value is low, CSPj trust value maps to distrust. If obtained trust value is high, CSPj trust value maps to trust and otherwise trust value maps to uncertain.

4. Simulation Results

In this section, the results of two components of cloud trust management frameworks are explained. One component is the feedback evaluation component, and another is the SLA negotiation component.

4.1. SLA Negotiation Component Results. In this section, the simulation results of the demand trust evaluation agent are illustrated. As mentioned in Section 4, the trust evaluation agent is one of the most important agents of the **SLA** negotiation component. In this section, the simulation results of DTEA are described especially.

The task of DTEA is described in Section 4. Simulation parameters are described in Table 1. Also, the batch size is 10, and the rate of entry request follows the Poisson process. The simulation results are shown in Figures 7 to 9.

User service satisfaction with DTEA and without DTEA is shown in Figure 7. In Figure 7, the horizontal axis is arrival time and the vertical axis is service satisfaction. Using DTEA enhances user service satisfaction. Also, using DTEA increases trust values shown in Figure 8. In Figure 8, the horizontal axis is arrival time and the vertical axis is trust value. Due to load balance, DTEA uses a roulette wheel mechanism on resource nodes. Figure 9 illustrates service satisfaction in DTEA with a roulette wheel and without a roulette wheel. In Figure 9, the horizontal axis is arrival time and the vertical axis is service satisfaction. Rolette wheel has a small effect on service satisfaction. It should be mentioned that DTEA with a roulette wheel not only decreases service satisfaction but also makes good load balance on resource nods.

As shown in Figures 7–9, using the demand trust evaluation component and assigning the nearest cloud service provider to the request of the cloud service users, the satisfaction of cloud service users from receiving the service has increased, which is indicated in Figure 7. And, since the increase in satisfaction is directly related to the increase in the degree of trust of cloud service providers, this leads to higher trust values as well as the average values of trust in two cases using DTEA and without using it as shown in Figure 8. As shown in Figures 7 and 8, both cloud service user satisfaction and the trust value of cloud service providers have increased by about 0.2. As stated in the description of the proposed framework, since the use of DTE may lead all requests to a number of cloud service providers and create a bottleneck because the balance of load between cloud service providers is used, the revolving mechanism in demand trust evaluation is used, which Figure 9 shows the percentage of satisfaction of the audience using/without using the revolving mechanism. And as shown in Figure 9, the use of this mechanism of about 0.9 has resulted in higher audience satisfaction from the received services.

4.2. Evaluation of Simulation Result. In this research, a ranked data set called epinion has been used to simulate the component of evaluating the feedback received. This data set is selected from mass and Avesani and contains 6194 items from CSP, 55197 items from CSU, and 394691 items from feedback trust values. The epinion dataset has two modes (trust and distrust). The trust mode is denoted by 1, and the distrust mode is denoted by minus 1. Based on the explanations provided in Section 3, the feedback evaluation

Table 1: DTEA simulation parameters assumption.

Parameters	Description	Numbers
N	The number of a user request for a separate domain	1000
M	The number of resource nodes	10
PN	The number of QOS parameter	4

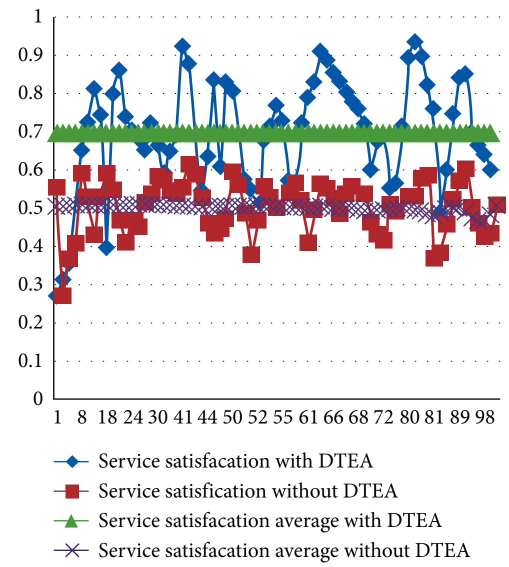


Figure 7: Service satisfaction with DTEA and without DTEA.

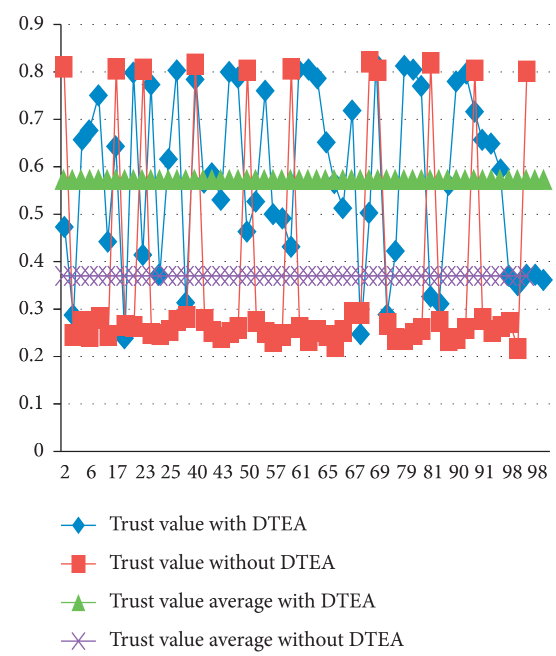


Figure 8: Trust value with DTEA and without DTEA.

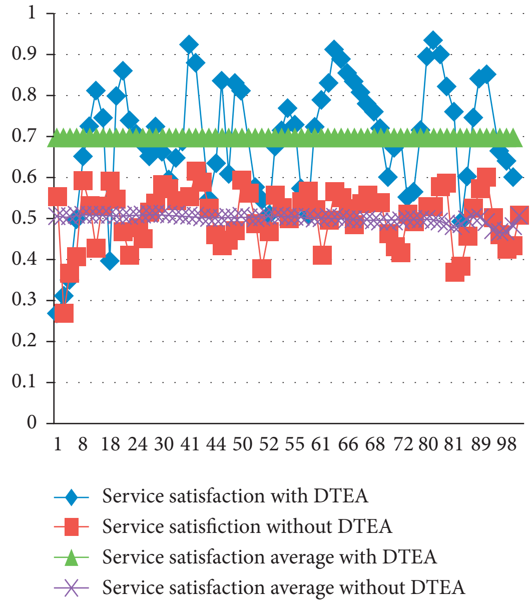


Figure 9: Service satisfaction in DTEA with a roulette wheel and without roulette wheel.

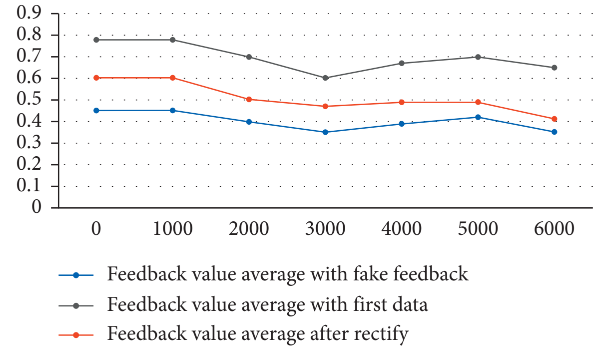


Figure 10: Feedback trust value average.

Table 2: Feedback trust value average (FTVA) based on 1000 CSP.

	1000	2000	3000	4000	5000	6000
FTVA before rectifying with fake feedback	0.45	0.42	0.37	0.38	0.41	0.38
FTVA after rectifying with fake feedback	0.58	0.52	0.47	0.47	0.47	0.39
FTVA without fake feedback (first data)	0.76	0.7	0.61	0.67	0.71	0.65

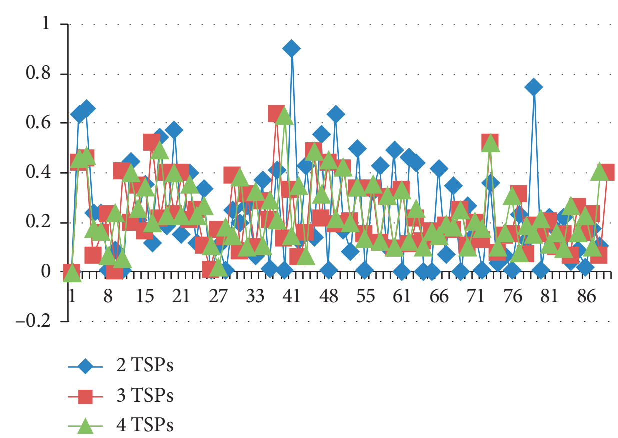


Figure 11: Trust values in multiclouds.

component is able to detect fake feedback. In the simulation performed in this study, we injected fake feedback into the epinion data set. A random injection rate of 20% was considered.

The degree of trust in the fake feedback is corrected using the feedback retrieval component. Figure 10 shows the average amount of feedback trust in three modes (including preinjection, postinjection, and postcorrection). In this figure, the horizontal axis represents the CSP and the vertical axis represents the average values of trust in feedback. To obtain a true approximation to the values of feedback trust, we examine an average of 1000 CSP.

The reliability of feedback based on every 1000 CSPs is shown in Table 2. Based on the results in this table and Figure 10, it can be argued that the average amount of trust in the feedback sent by client users after injection with the amount of confidence in the initial data sets is up to 41% differences. Therefore, by using the feedback component, the proposed method is used to identify and make correction in fake feedback. In this way, the percentage of confidence is reduced to 20%. It can be concluded that the use of this new component increases the efficiency of the trust management framework and ultimately reduces the impact of attacks by malicious users on the trust percentage of feedback received.

As we can see in Figure 10, the average confidence and trust level are greatly reduced (approximately 29%) by injecting fake feedback without the feedback evaluation component. However, after using the feedback evaluation component, the average value of feedback trust increases by 10% and its value is closer to the initial data.

4.3. The Trust Values in Multicloud Computing. In this section, the simulation results of trust values in multicloud computing are presented. Figure 11 shows trust values for three states 2TSPs, 3TSPs, and 4TSPs. In Figure 11, the horizon axis represents the time and perpendicular axis indicating the average trust values, which indicates the conformity of the proposed framework with multicloud environments; for example, the simulation results for 2TSPs, 3TSPs, and 4TSPs are presented.

4.4. Comparing Proposed Multiclouds Trust Management Framework with Other Frameworks. In this section, the simulation results of four models (feedback-based model, SLA-based model, multicloud model, and new proposed model) are presented in Figures 12 and 13. Trust is one of the most important parameters that can be used to compare models with each other in the field of trust management. The higher the average trust values, i.e., the framework or method proposed has performed better, the higher the satisfaction of cloud service users from receiving the service, which means that a suitable cloud service provider is

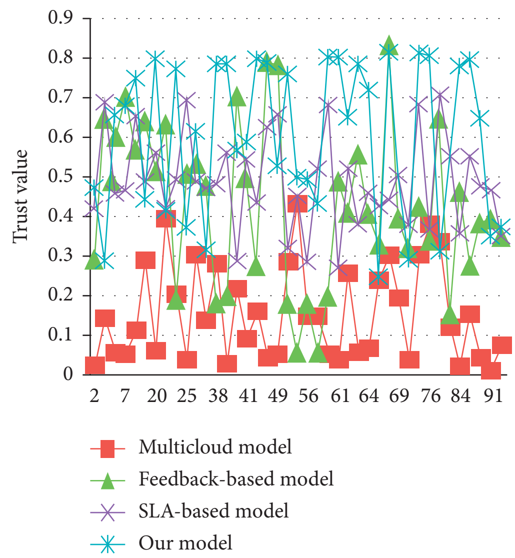


Figure 12: Comparison of the proposed model with other models.

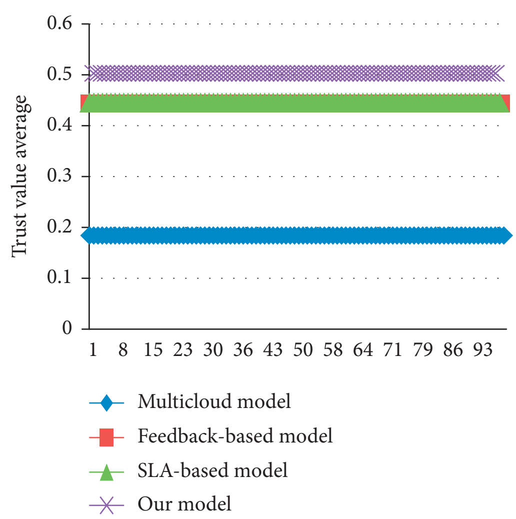


Figure 13: The comparison of trust values means between the new proposed model and other models.

assigned to their request. Therefore, in this study, trust values, as well as average trust values in the same conditions and the same dataset, were used to evaluate the proposed framework and compare it with other methods and models. In the proposed framework, because the assignment of the nearest cloud service provider to the request of the cloud service user is done using the request trust evaluation components, this will lead to higher cloud service user satisfaction and a consequently higher level of trust of cloud service providers. The number of articles and studies that have provided a trust management framework in multicloud environments is limited. The most complete and closest model that is presented in multicloud environments for trust management is the multicloud model that the proposed framework presented in this study has advantages such as detection fake feedback compared with that in the results section of the proposed framework.

Figure 12 shows the comparison of the proposed model with other models. Also, in Figure 13, the comparison of trust mean values between the new proposed model and other models has been presented. In Figures 12 and 13, horizontal axis is time and the vertical axis is the trust values.

As shown in Figures 12 and 13, the proposed framework increased trust values rather than other models. Also compared with other models, our framework gives better mean trust values.

It can be concluded that this new framework is suitable for multicloud environments and gives reliable trust values. Moreover, it is clear that the nearest trust values of other models with the proposed model are obtained in the SLA-based model.

As you can see in Figure 12, the average value of trust in the proposed model is higher than that in the others because in the proposed model, we have a new component named as "demand trust evaluation component". This component selects the service provider which closely matches with other components. The request of CSUs helps a lot, and also the presence of components such as fake feedback detection also helps in having a higher average value of trust than other models, which has not been addressed by a model so far and is one of the advantages of the proposed model.

As can be seen in Figures 12 and 13, in the proposed method, due to the use of the fake feedback evaluation component and also because of the use of the demand evaluation trust component, the mean trust values are at a higher level than the other three models, which indicates better performance of the proposed framework than other frameworks.

The proposed model in this study is compared and simulated with three other models because of the number of proposed models in the field of trust management in multicloud environments that look at trust values as a multidimensional parameter and not a limited one. The proposed model is compared with the three models that have been presented in this field and have such features.

5. Conclusion

In this paper, a new trust management framework for multicloud environments has been proposed.

The advantages of this framework are as follows:

(i) Subjective trust value and objective trust value applied to calculate trust values.

(ii) Objective trust value and subjective trust value are multidimensional parameters.

(iii) Feedback evaluation component was applied in this framework. The Feedback evaluation component task is identifying and rectifying fake feedbacks that any framework does not apply to this component yet. Simulation results had shown the performance of this component, and it shows the effect of this component on trust values.

(iv) Trust negotiation component has used the platform that the output of its component is SLA contract.

One agent of the SLA negotiation component is the demand trust evaluation component. This component selects the CSPs that have the nearest adoption with CSU request, and finally this component causes increase in the service satisfaction and trust values average. The simulation results confirm it.

(v) The proposed framework increased trust values rather than other models (SLA-based model, feedback-based model, and multicloud model). As future work, the trust management model can be proposed along with the detection of fake feedback in other applications such as fog computing and the Internet of Things.

In the case of failures, it can be noted that if several cloud service users colluded with each other and attacked a cloud service provider for a period of time, the proposed feedback evaluation component cannot detect fake feedback from other feedbacks.

As a future work, it is planned to introduce a trust management model with the feature of detecting fake feedback in IoT networks and fog computing. Game theory can also be used to detect fake feedback in the feedback evaluation component of trust management models.

Data Availability

Data are available on request through contacting with safieh.siadat@gmail.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] N. Jafari Navimipour, A. M. Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Expert cloud: a cloud-based framework to share the knowledge and skills of human resources," *Computers in Human Behavior*, vol. 46, pp. 57–74, 2015.

[2] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities," in *Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications*, Dalian, China, September 2008.

[3] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[4] N. Jafari Navimipour, A. Masoud Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Resource discovery mechanisms in grid systems: a survey," *Journal of Network and Computer Applications*, vol. 41, pp. 389–410, 2014.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, Busan, Korea, May 2010.

[6] P. Xiao, Z.-G. Hu, and Y.-P. Zhang, "An energy-aware heuristic scheduling for data-intensive workflows in virtualized datacenters," *Journal of Computer Science and Technology*, vol. 28, no. 6, pp. 948–961, 2013.

[7] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *Proceedings of the International Conference on Web Information Systems Engineering*, Sydney, NSW, Australia, October 2011.

[8] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, Indianapolis, IN, USA, November 2010.

[9] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," *International Journal of Information Security*, vol. 13, no. 2, pp. 113–170, 2014.

[10] I. M. Abbadi, "A framework for establishing trust in Cloud provenance," *International Journal of Information Security*, vol. 12, no. 2, pp. 111–128, 2013.

[11] C. Tang and J. Liu, "Selecting a trusted cloud service provider for your SaaS program," *Computers & Security*, vol. 50, pp. 60–73, 2015.

[12] I. M. Abbadi and A. Martin, "Trust in the cloud," *Information Security Technical Report*, vol. 16, no. 3-4, pp. 108–114, 2011.

[13] I. U. Haq, I. Brandic, and E. Schikuta, "Sla validation in layered cloud infrastructures," in *Proceedings of the International Workshop on Grid Economics and Business Models*, Ischia, Italy, August 2010.

[14] W. Conner, "A trust management framework for service-oriented environments," in *Proceedings of the 18th International Conference on World Wide Web*, Madrid, Spain, April 2009.

[15] Z. Malik and A. Bouguettaya, "Rateweb: reputation assessment for trust establishment among web services," *The VLDB Journal*, vol. 18, no. 4, pp. 885–911, 2009.

[16] W. Fan, S. Yang, and J. Pei, "A novel two-stage model for cloud service trustworthiness evaluation," *Expert Systems*, vol. 31, no. 2, pp. 136–153, 2014.

[17] J. Y.-j. Hsu, K.-J. Lin, T.-H. Chang, C.-j. Ho, H.-S. Huang, and W.-r. Jih, "Parameter learning of personalized trust models in broker-based distributed trust management," *Information Systems Frontiers*, vol. 8, no. 4, pp. 321–333, 2006.

[18] C. Uikey and D. Bhilare, "A broker based trust model for cloud computing environment," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 11, pp. 247–252, 2013.

[19] S. Chakraborty and K. Roy, "An SLA-based framework for estimating trustworthiness of a cloud," in *Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, UK, June 2012.

[20] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems*, vol. 70, pp. 392–406, 2014.

[21] S. Song, K. Hwang, and M. Macwan, "Fuzzy trust integration for security enforcement in grid computing," in *Proceedings of the IFIP International Conference on Network and Parallel Computing*, Wuhan, China, October 2004.

[22] H. Liao, Q. Wang, and G. Li, "A fuzzy logic-based trust model in grid," in *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, Wuhan, China, April 2009.

[23] S. Siadat, A. M. Rahmani, and M. Mohsenzadeh, "Proposed Platform for improving grid security by trust management system," 2009, https://arxiv.org/abs/0911.0498.

[24] S. Siadat, A. M. Rahmani, and H. Navid, "Identifying fake feedback in cloud trust management systems using feedback

evaluation component and Bayesian game model," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2682–2704, 2017.
[25] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: a trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.
[26] J. Guo, I.-R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.
[27] I. U. Din, "Trust management techniques for the Internet of Things: a survey," *IEEE Access*, vol. 7, pp. 29763–29787, 2018.
[28] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 232–238, 2010.
[29] C. Ngo, Y. Demchenko, and C. De Laat, "Toward a dynamic trust establishment approach for multi-provider intercloud environment," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science*, Taipei, Taiwan, December 2012.
[30] A. K. Sangaiah, A. S. Rostami, A. A. R. Hosseinabadi et al., "Energy-aware geographic routing for real time workforce monitoring in industrial informatics," *IEEE Internet of Things Journal*, 2021.
[31] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.
[32] A. K. Sangaiah, D. V. Medhane, G.-B. Bian, A. Ghoneim, M. Alrashoud, and M. S. Hossain, "Energy-aware green adversary model for cyberphysical security in industrial system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3322–3329, 2020.
[33] H. M. Mousa and G. F. Elhady, "Trust model development for cloud environment using fuzzy mamdani and simulators," *Journal: Internation Journal of Computers and Technology*, vol. 13, p. 11, 2014.
[34] M. J. Sule, M. Li, G. Taylor, and C. Onime, "Fuzzy logic approach to modelling trust in cloud computing," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 2, pp. 84–89, 2017.
[35] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems*, vol. 70, pp. 392–406, 2014.
[36] S. Kumar, S. Mittal, and M. Singh, "Fuzzy based trust management system for cloud environment," *Advances in Science and Technology Research Journal*, vol. 10, no. 30, pp. 32–37, 2016.
[37] M. Nielsen, K. Krukow, and V. Sassone, "A Bayesian model for event-based trust," *Electronic Notes in Theoretical Computer Science*, vol. 172, pp. 499–521, 2007.
[38] S. M. Habib, S. Ries, and M. Muhlhauser, "Towards a trust management system for cloud computing," in *Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, Washington, DC, USA, November 2011.
[39] A. Lakshminarayanan, "Can CRLs provide bandwidth-efficient online certificate status?" in *Proceedings of the 2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, USA, November 2006.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5575129, 15 pages
https://doi.org/10.1155/2021/5575129

Research Article

LBPSGORA: Create Load Balancing with Particle Swarm Genetic Optimization Algorithm to Improve Resource Allocation and Energy Consumption in Clouds Networks

Seyedeh Maedeh Mirmohseni, Amir Javadpour, and Chunming Tang

School of Mathematics and Computer Science, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Amir Javadpour; a_javadpour@e.gzhu.edu.cn and Chunming Tang; ctang@gzhu.edu.cn

Received 21 January 2021; Revised 17 March 2021; Accepted 20 May 2021; Published 7 June 2021

Academic Editor: Nazrul Islam

Due to the purpose of this study that reducing power consumption in the cloud network is based on load balancing, the fitness function measures the load balance between cloud network and servers (the hosts). This technique is appropriate for handling the resource optimization challenges, due to the ability to convert the load balancing problem into an optimization problem (reducing imbalance cost). In this research, combining the results of the particle swarm genetic optimization (PSGO) algorithm and using a combination of advantages of these two algorithms lead to the improvement of the results and introducing a suitable solution for load balancing operation, because in the proposed approach (LBPSGORA), instead of randomly assigning the initial population in the genetic algorithm, the best result is procured by putting the initial population. The LBPSGORA method is compared with PSO, GA, and hybrid GA-PSO. The execution cost, load balancing, and makespan have been evaluated and our method has performed better than similar methods.

1. Introduction

Recent advances in computer networks, devices, Internet speeds, and so forth have led many companies and users to use cloud services. Resource sharing, virtualization of various services, and scheduling techniques have increased the efficiency of data centers. All of this has led users to use cloud services to meet their needs. Users need strong processors and fast networks to set up cloud data centers in different parts of the world. One of the most important concerns in this area is related to energy [1, 2]. For this purpose, in recent years, attention has been paid to enhanced techniques for providing power management for cloud servers. Therefore, many researchers have worked on different algorithms to reduce costs by reducing energy consumption and also reduce greenhouse gas emissions. To improve the performance of several algorithms to affect the hardware or the software has been tested. With the help of virtualization techniques, several virtual machines can be created on a physical machine, so we need less hardware and can increase performance. Cloud computing uses virtualization technology based on user's needs to provide resources, which reduces energy consumption. One of the most popular ways to reduce energy consumption is to use metaheuristic techniques [3, 4]. This technique could attract attention due to its simplicity of implementation. The cloud services providers, because of the benefits they gain from serving customers as well as cloud consumers, such as organizations and economic organizations, tend to cloud computing because of the benefits they receive from reducing or eliminating infrastructure maintenance costs for these services [1]. However, since applications for business operations of users running in the cloud may be highly sensitive, consumers need to have a guarantee by the cloud providers when the service is delivering. Usually, such a guarantee is supplied between the consumer and the provider through a service level agreement. Recently, the use of virtualization to improve the performance of cloud networks has flourished [2, 5]. So virtual machines can be moved from one host to another to enhance the efficiency of users' applications without interrupting service. Also, it is possible that the number of resources allocated to a customer increases or decreases dynamically [6–9].

In cloud computing, the users pay the cost for the amount of use and services they receive, and the workspace is transmitted to the cloud (the Internet) from data and information on the personal computer [10] so users can access their data and evidence anytime, anywhere in the world. Users also do not need hardware systems with high processing capacity and storage capacity, as all cloud computing operations and storage are handled in the cloud service providers section by equipped and advanced servers [11]. Currently, cloud computing services are offered in three different forms: SaaS_Software as a Service, PaaS_-Platform as a Service, and IaaS_Infrastructure as a Service [1]. Organizations and companies can select and use one or more service strategies depending on their needs. Cloud accounting is one of the most famous words in today's economic organization. An intrinsic cloud feature that distinguishes it from traditional and old hosting services. The amount of unlimited resource capacity (such as CPUs, memory, input-output network, and disk) is provided at a competitive rate [12]. This subject provides opportunities for startup companies to host their applications in the cloud, so it eliminates the overflow of prepared old ultrastructural resources that lasted several months [1, 13, 14].

Cloud service providers need to reassure customers that their needs will be fulfilled completely [15–17].

1.1. Motivations and Goals. Using virtualization-based networks and storing bits for cloud computing can be a way to balance the load on networks; just as in the phase map and radius created in the big data, it can upset the balance. Using virtualization methods and load balancing methods can help a lot. In this paper, we present an algorithm PSGO considering scheduling. The main challenge about earlier researches during past years was that although resource scheduling and request management were taken into place, there was not considerable attention for energy optimization. In this research, we considered energy optimization as the main objective and tried to present a solution based on this. The energy-performance tradeoff can be considered for a physical machine that has computing ability in its peak usage. For optimizing energy consumption, we used the PSO optimization algorithm [14]. In this approach, user applications are mapping to virtual machines through the autonomous domain layer. Therefore, resource scheduling has been done with the assistance of an autonomous domain and virtual machine layer. The advantage of this heuristic algorithm is that it starts searching from some areas of space and extends the exploration domain and develops a stronger algorithm. In this work, we do the following to improve results: the final result of the PSO is to initiate population for GA. Thus, the first population of the GA is the optimal solution of the last step (Figure 1). Bestowing to the goal of this study is reducing load power in the cloud network based on load balancing, and the fitness function measures the load balance between the cloud network (host) servers.

In this study, we have dealt with the issue of virtualized reserve controlling of cloud computing records middles and, in particular, the load balancing among the servers, and we have done this by considering and implementing particle swarm optimization algorithms and genetic procedures [3, 18]. The sequel displays that the combination of these two can be effective in reducing the cost of resource allocation and load balancing. On the other hand, considering the combination of two optimization algorithms, the program execution time is increased compared to the implementation of an optimization algorithm. In this study, to improve the results, we will do the following: the final result of the PSO algorithm will be the initial value used in the genetic algorithm. Therefore, the initial population used in the genetic algorithm is the optimal solution obtained from the previous step. This method is very suitable for answering the main question of the current research due to its ability to convert the load balancing problem to the optimization problem (reduction of the imbalance cost function) and has been used in studies in this field as well.

In the first part of this article, some of the definitions related to this field and the introduction are discussed. The second section discusses the research background. In the third part, LBPSGORA based on the heuristic algorithms is presented [19], and, in the fourth part, an implementation of the proposed method and the result of the power consumption reduction are presented. In Section 5, we present future work and conclusions.

2. Literature Review

The load balancing algorithms are in three categories: (1) sender initiated load balancing algorithms, (2) receiver initialed load balancing algorithms, and (3) symmetric load balancing algorithms, a combination of both modes. The Round Robin algorithm is a load balancing method with an optimization energy consumption methodology. This issues tasks to all secondary processors. All tasks are assigned to secondary processors based on Round Robin (RR-Order) (Figure 2). Resources are selected with the machine to run in the alone direction. Selecting processors locally and independently of the allocation of processors is performed on each processor. The RR-Order algorithm generally is recycled in web servers where generally the WEB applications have the same nature and, as a result, they are evenly distributed. This defined pattern has not been under the principle of SLA [20].

With increasing the number of intelligent communication monitoring system devices such as WSN, WBANs, Big Data, SDN, and mCloud (mobile-Clouds), these devices have become moveable, proficient, and personal accounting infrastructure for the end handler in various modes [2, 8, 17, 21–25]. These devices come with cloud network intermediaries and IoT and SDNs, which cause them to enter the global Internet resources [12, 17]. Many end-users tend to use these devices instead of desktops or clients. In this case, CloudIoT and fog computing have appeared as a practical explanation, and their tasks can be offloaded on the cloud substructure [1]. As another example, network

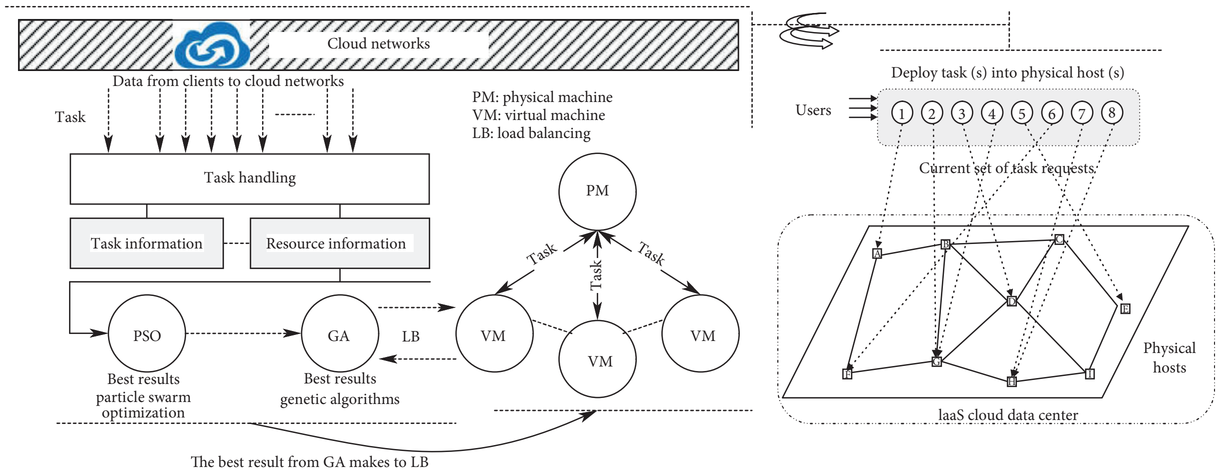


Figure 1: The framework of an algorithm based on load balancing to optimize energy consumption.

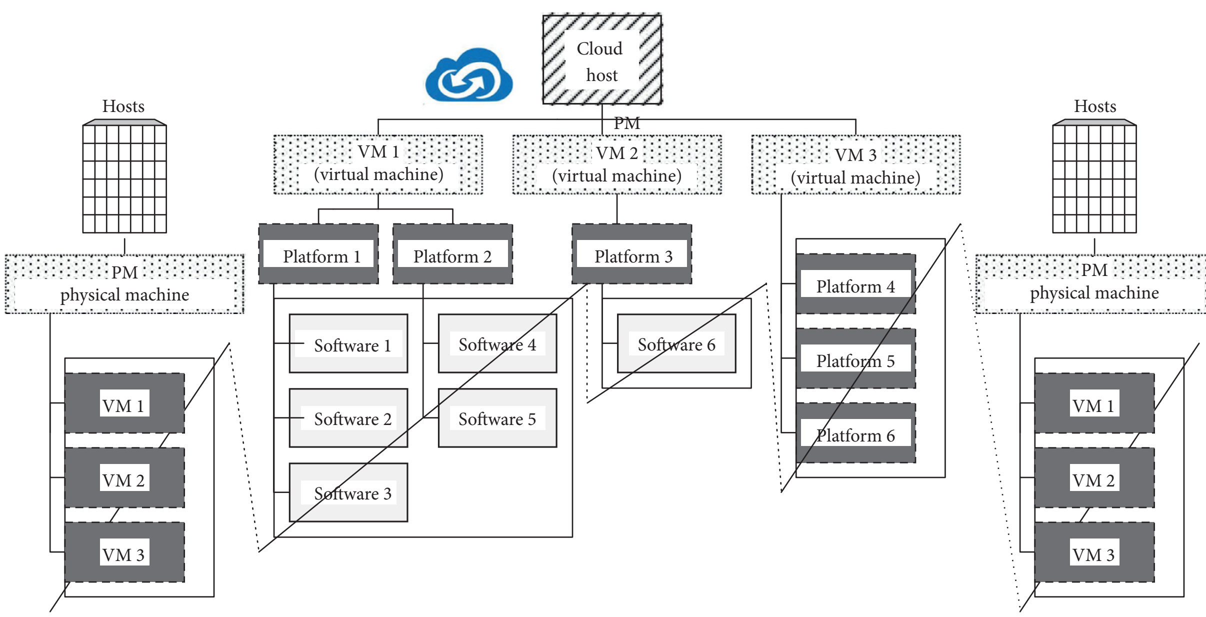


Figure 2: The diagram of cloud base structure.

bandwidth may change over time, and the performance levels of data rate cannot be guaranteed for task scheduling [18]. This is especially regarded in the area of real-time applications, where minimum latency is important. Resource management allocated to the mobile cloud must be considered in the cloud infrastructure. Because load-free applications and tasks run on the cloud substructure, fog resources such as storing are recycled to complement mobile device home-grown resources [9, 26].

In this regard [2], power consumption in CDC (CloudDataCenters) has been used. In this paper, the goal is to achieve an efficient resource management system to reduce operating costs and ensure service quality. The model of reducing energy consumption in this paper is based on continuous control of virtual machines using efficient methods. The main tool is the use of live migration. In this case, they can transfer virtual machines to physical machines without any interruption and with the highest flexibility, so that they can use the highest capacity of physical machines.

In [16], genetic algorithms for allocating virtual machines to reduce energy consumption in data centers are proposed, which shows that evolutionary algorithms are useful for allocating virtual machines because they have been able to reduce energy consumption by up to 10%.

In [27], a method of load balancing using PSO was provided. The authors have used many mobile cloud solutions to conserve many properties resources such as power and energy in mobile-portable devices. They optimized the grid energy by examining the SLA and placing the virtual machines using the PSO method.

In [28], in PSOSA, the PSO is used to minimize the energy. In experiments using cellular communications ad hoc mobile, the load balancer decision-maker used cellular convergence by PSO by load balancing on the network. The authors tried to evaluate their mechanism on a cloud-based cellular network.

Another method of load balancing the MOPSO multi-objective code transfer interval is proposed: the Energy Management and Service Quality Awareness (QOS) Framework [29]. The authors have used the Internet of Things for their inputs. They have considered the infrastructure for the network. Their approach focuses on portable mob-cloud service transfer created on QOS as well as power constrictions. If using the mobile device remote service is costly or the latency time is so high that QoS is not available, the fog service can be transmitted by communication and locally at a lower cost. It proposes algorithms for migration and transfers alongside the decision process for fog-grown performance against inaccessible performance.

In [30], a solution was presented based on the migration of virtual machines using PSO. The authors have used mobile devices to deliver cloud-based services. As part of their method, they supported fuzzy to apply multiple services; for example, a service output can be used as another service input. Their goal is to reduce energy consumption in the network, and resources and the potential of this platform in data are shared by online services. They have analyzed this potential and mentioned the difficulty of this way.

In [31], a C-RAN application parsing solution was presented; it works by splitting Apps to different mechanisms; then these modules are activated. The authors also discussed how to avoid the middleware solution from allocating further mob-resources to the fog server by using the OFDMA and PSO facilities instead of allocating all virtual machines. In the results section, several experimentations were achieved with m-devices to reach a DSS model to place the load on the network and reduce energy consumption.

In [32], a method with PSO considers reducing the consumption of available fog resources without the need for additional overflow in the m-device. In the obtained method, the implementation of PSO algorithm has been used to find suitable places in the network. Cloudlets and their modifications to find the suitable virtual machine location are another method for mobile clouds which are used like VM by placing load on the cloud infrastructure that may be nearby.

In [33], a scheduling method called NP-Hard is presented for the cloud. The authors have used PSO to reduce costs. How the cloudlet approach to scaling is used according to their resources, how m-devices network efficiency will cooperate with multiple users, and just how fog cloud will manage the stimulus ability need to be explored. The authors have introduced the CSOS method and optimized network energy by reducing makespan. In their method, the calculated cost has a low value.

In [34], the analysis of energy consumption by PSO on mobile devices has been examined to some extent in articles and writings. As mentioned, most of power and costs connected with using an m-device come from "DPRA" energy, known as the network interface. When the connection is made, the net edge remains active for each outstanding packet from the fog server, and it is costly for net operation. The work done in this paper is to implement a system called VM-DPRA, which groups network needs and requests together to reduce the number of tail energy occurrences. Their tail-ender method similarly attempts to cluster network requirements to minimize the power. In this research, K-means clustering method has been used for mapping tasks and microgenetic algorithm for dynamic aggregation of virtual machines. It has also been able to reduce the execution time.

In [28], a dynamics algorithm is presented. This algorithm is proposed to create load balance in cloud computing environments using the ant colony algorithm. This algorithm is one of the biological algorithms and almost simulates the behavior of ants.

The authors in [29] provide a method for allocating virtual machines using a linear search to optimize resources. The architecture used in this paper is that data centers include several physical machines and physical machines have a fixed size of RAM, CPU, and memory. The broker is responsible for managing user's requests; then the requests are transferred to the data center via the cloud, where the center manager reserves a certain number of virtual machines to execute the users' request and provides different classes of virtual machines by cloud maker is suggested, and the user can choose from these classes that have special features such as RAM, CPU, and memory. In this way, it has been able to reduce energy consumption to some extent.

The issue of proper placement of virtual machines in physical machines has a great impact on optimizing energy consumption. In [30], the ant colony system has been used to place virtual machines. In this research, the dynamic placement system of virtual machines has been considered. In this way, the request for several virtual machines changes in each time interval; at the beginning of the time interval, the list of virtual machines needs to be updated, and this happens by deleting the expired virtual machines before the previous interrupt. Then the resources of these expired machines are released, and new virtual machines are given to them. Updated virtual machines receive the right information and resources, which improves energy efficiency.

The study in [7] develops new VM scheduling to reduce energy costs for cloud services providers. In this study, it is assumed that there is an optimal frequency for one PM for VM processing. The optimal rate is also defined for the performance-power ratio based on which the heterogeneous PMs are weighted in the cloud environment. This problem is defined for the time-bound networks and has achieved good results. The workflow process is divided into several periods, where in each one the virtual machines have been dedicated to the proper PM and each core is operating at its optimal frequency. After each period, the cloud reconfigures to consolidate computing resources, which further reduces energy consumption.

Table 1 illustrates some related works with meta-heuristics algorithms for solving time scheduling problems and improving energy consumptions in cloud networks.

3. The Proposed Method (LBPSGORA)

3.1. Problem Statement. Here, $ADS = AD_1, AD_2, \ldots, AD_r$ represents the autonomous domains set and each AD represents an autonomous domain. Each autonomous domain has several virtual machines, as shown in Figure 3 [18, 35]. The cloud is mapped user applications to the virtual machine by running user applications and through a layer of autonomous domains. Therefore, the source environment scheduling of the cloud should be done by the layer of autonomous domains and the layer of virtual machines.

So here we have used two layers for scheduling; in this way, first, we can find autonomous domains that have more fitness and then search for the best virtual machines for assigning tasks among virtual machines at those domains. In addition, we considered some autonomous tasks as an application and attempted to assign them to the virtual machines by the proposed scheduling method. Thus, a hierarchical scheduling framework is used for the proposed algorithm. To illustrate this problem, we present a two-dimensional form in Figure 4, in which X_{axis} shows speed of CPUs and Y_{axis} shows the amount of RAM (Table 2 and Table 3).

3.2. Proposed Method LBPSGORA. In this study, a load balancing algorithm is presented for optimizing energy consumption by considering scheduling. The problem with research that has been done over the last years is that, despite addressing resource allocation and requests management, there has not been much attention paid to energy efficiency in it. But this research aims to provide a solution based on this vital problem, considering energy efficiency as the primary goal. The energy efficiency ratio for a physical machine (PM) that contains the computational power ratio at the peak of energy can be used for this purpose. The algorithm used to optimize energy consumption is the particle swarm optimization algorithm. In this method, users' applications are mapped to virtual machines through a layer of autonomous domains. Therefore, the scheduling of cloud environment resources is done by the layer of autonomous domains and the layer of virtual machines. The advantages of using metaheuristic algorithms are that one can start searching from several areas of the problem space and increase the level of exploration and have a stronger algorithm. In this study, to improve the results, the final achieved result of the PSO algorithm will be used as an initial value in the genetic algorithm. Therefore, the initial population used in the genetic algorithm yields is the optimal response obtained from the previous step.

3.3. Particle Swarm Optimization Algorithm. In this part, we describe the several specifics of the LBPSGORA. As virtual machines are the subset of physical machines that are processing tasks, the PSO algorithm can work in a complex problem to find suitable tasks. The formulas that are presented in this section are related to the particle swarm optimization method. This method is based on using metaheuristic algorithms, in such a way that the metaheuristic methods have been used for assigning VM to PMs. The PSO algorithm is a global optimization method that can solve problems with solutions in a point or surface in n-dimensional space. Therefore, in addition to the initial velocity of particles, the channels of communication between the particles are identified. By moving these particles into the solution space, the results are evaluated based on a merit criterion. A large number of swarm particles make the method flexible against the local optimal solution problem. Each particle has a location. $x_i(t)$ specifies the position of particle i at time t. Each particle also needs a swiftness to move through space. $v_i(t)$ specifies the velocity of particle (i) at time (t). The equations for updating the particle positions are given below. The particles are talented at memorizing the best position in their lives. The best individual experience of a particle or the best position met by a particle is called y_i. The particles know the best position met in the whole group. This is called the $\hat{y}$ position. The two values of c_1 and c_2 are constant and w is the initial weight that is decreased linearly from $w_{\max}$ to $w_{\min}$. $t_{\max}$ is the maximum number of repetitions intended for algorithm (Table 4). Therefore, it can be expected that, by utilizing metaheuristics in the initial allocation of VM to PMs, it would be closer to the optimal solution and reduce the number of migrations (Figure 5, part A).

$$\begin{aligned}
v_i^{t+1} &= w * v_i^t + c_1 * \text{rand}_1 * \left(y_i - x_i^t\right) + c_2 * \text{rand}_2 * \left(\hat{y} - x_i^t\right),\\
x_i^{t+1} &= x_i^t + v_i^{t+1},\\
w &= w_{\max} - t * \frac{w_{\max} - w_{\min}}{t_{\max}},\\
v_{\min} &\leq v_i^t \leq v_{\max}, \quad i = 1, \ldots, n,\\
x_{\min} &\leq x_i^t < x_{\max}, \quad i = 1, \ldots, n.
\end{aligned} \tag{1}$$

3.4. Genetic Algorithm. The steps used in the genetic algorithm are as follows: at first, building the initial population is based on the best solution obtained from the PSO algorithm. Selecting and combining initial samples (arrays) to generate a new sample or a crossover are done in this algorithm. In the crossover phase, a random combination of two fathers was used to generate a new child. If the generated children contain repeated genes, they will be replaced by the number of unused genes. The whole crossover phase is performed in ten iterations with a combination of 10 randomly selected pairs. Modifying existent samples and building a new sample prevent local optimization because similar samples may be used for the combination in the crossover phase. Selecting a sample with the best fitness among the produced samples in this generation continues to produce the next generation of samples until the end condition is met (Figure 5, part B).

On the path to the aim and on the way to achieving the optimal solution for the problem, it is necessary at each step to measure the proximity of the current solution to the aim using a criterion. This is done by the fitness function. In this paper, the fitness function is carried out by using the study in [17]. To achieve an optimal solution where there is load balancing between cloud (hosts) network servers, we need to minimize the mean deviation between the loads allocated to

Table 1: Comparison of load balancing methods for cloud networks.

Reference	Description	Method	Advantages	Disadvantages	Simulations
[27]	Load balancing for optimization problem	Using modified PSO for load balancing	To maximize user-centric energy efficiency in the uplink communication	High capacity, low throughput, time complexity	MATLAB
[28]	Cloud provider load balancing PSOSA	Using PSO for having high performance LB in fog	VMs is presented to decrease LB on fog cloud	Has a multiplex physical resource and high time complexity	CloudSim
[29]	Load balancing with placement optimization MOPSO	PSO algorithm for time scheduling and improving energy	Low energy consumption	High delay and low throughput, time complexity	MATLAB
[30]	Load balancing decentralized particle swarm optimization	PSO-GA-based is presented to decrease LB on CloudNets	Low energy consumption, solve the optimization problem of maximizing the profit	High total execution time	MATLAB
[31]	Task scheduling problem on cloud is a load balancing for clouds with load balancing C-RAN	Scheduling constitutes one of the crucial aspects of resources management system in cloud computing	Improve energy consumption and resource utilization	High total execution time (makespan), bad throughput, and low quality of service	CloudSim
[32]	Cloud task scheduling problem with the aim of decreasing power	PSO-based for minimization energy and LB	Low energy consumption	Have a scheduling problem between virtual machines	CloudSim
[33]	CSOS energy-saving research of virtualization of the cloud computing platform with load balancing	PSO algorithm for solving time scheduling	Reduces energy with migration and high quality of service	Time complexity, high delay	MATLAB
[34]	DPSRA resource allocation VM-DPRA	PSO algorithm for solving time scheduling	Low energy consumption	Time complexity, high delay	MATLAB
[18]	Hybrid load balancing task scheduling problem PSO and GA	Novel load balancing symbiotic organisms search PSO + GA	Low cost, solving task scheduling	Time complexity and low quality of service	CloudSim

the servers. Suppose that T_i denotes all computing powers of a server and L_i is the computational power left by the server after assigning it to requests. The computing power of a server is calculated by the speed of CPU and the number of storage resources and its memory.

If the number of servers or hosts is m (the m physical hosts are heterogeneous),

$$T_i = \alpha T_c^i + \beta T_r^i,\quad L_i = \alpha L_c^i + \beta L_r^i,\quad i = 1, 2, \ldots, m. \tag{2}$$

Therefore, according to the standard deviation formula to optimize the load balancing in the cloud network, we use the following fitness function, where $E(X)$ is a mathematical expectation (Figure 6):

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i - E(X)\right)^2}. \tag{3}$$

The remaining load ratio in a server is defined as (residual load rate)

$$E_i = \frac{L_i}{T_i},\qquad i\epsilon\{1.2.3,\ldots,m\}. \tag{4}$$

Therefore, the fitness function is defined as the standard deviation function:

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\alpha L_c^i + \beta L_r^i}{\alpha T_c^i + \beta T_r^i} - \frac{\sum_{k=1}^{m}\left(\alpha L_c^k + \beta L_r^k\right)/\left(\alpha T_c^k + \beta T_r^k\right)}{m}\right)^2}. \tag{5}$$

4. Simulation and Results

In the cloud computing environment, some slaves are the central interfaces between users and the cloud infrastructure. This slave is responsible for resources management policy and allocating requests for machines. In addition to the slave that manages the cloud with its policies, due to the distributed nature of the cloud computing network, distributed resource management algorithms can also be provided. The purpose of this research is to manage resources in the cloud environment and to present a new method for examining the current status and history of semicentralized cloud-level machines. The cloud environment includes the physical infrastructure and specific control functions that the cloud, with their collaboration, will be able to provide and manage services. These infrastructures and resources are integrated and configured by the infrastructure provider in the data centers. The cloud provider

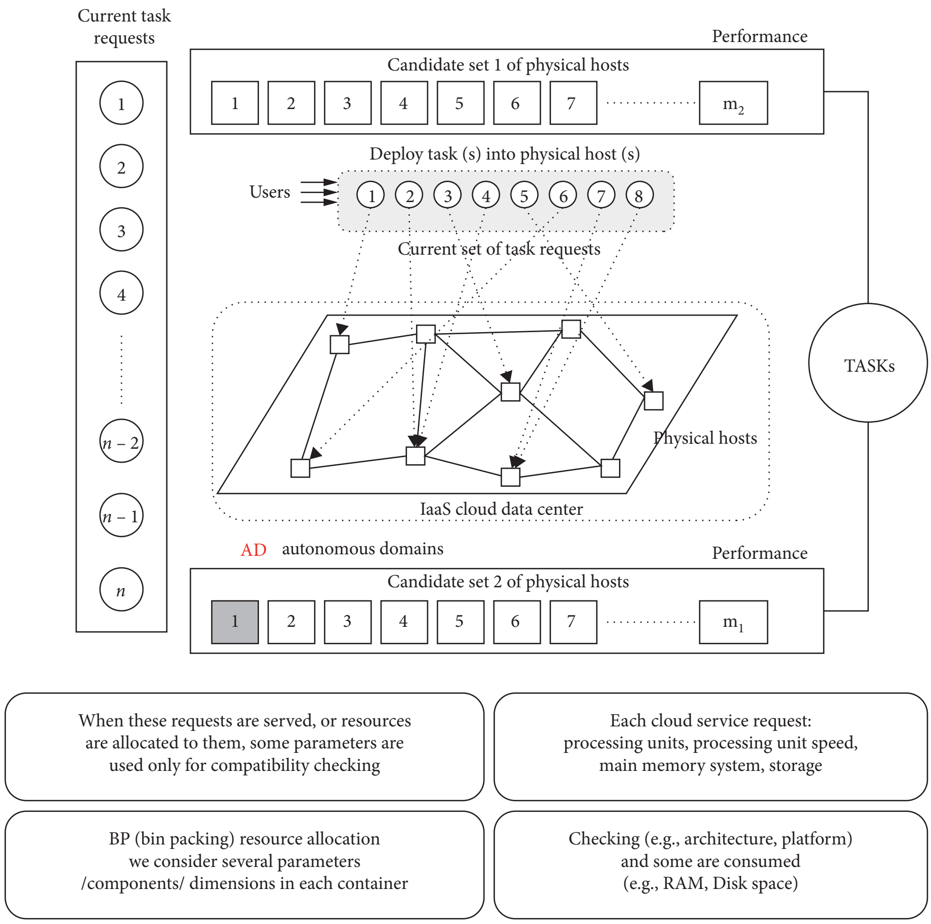


Figure 3: The application scheduling framework on the cloud.

provides resources to developers with the help of virtualization techniques and resource management algorithms (Figure 7). We used MATLAB and CloudSim to evaluate the algorithm. The simulations have been run in an environment with an Intel processor and 6 GB of RAM. The operating system used is 64-bit Windows 7 Ultimate (Tables 5 and 6).

MATLAB's programming language is used to simulate the peer-to-peer system, connect between records, and construct the data distribution. Figure 8 illustrates the implementation steps of this research. After performing this initialization, Figure 9 shows the number of resources and registered requests in the program when the LBPSGORA is evaluating.

4.1. Applying PSO to Data Resources Matrix. In the process of implementing LBPSGORA after identifying the resource matrix and also the submissions matrix, the PSO_Algorithm has been applied to data. So we can allocate sent requests appropriately so that the load balancing between the servers is maintained and the cost function defined for this research will be minimized. Table 7 shows the parameters used in the particle swarm optimization algorithm.

After the implementation of the PSO algorithm according to the initialization, the manner of initial and final allocation to some of the requests is given in the following table as an example. Comparing Table 8 and Table 9, it can be seen that the final cost considered to the resource allocation is much lower than the initial cost. Figure 10 shows a comparison of the initial cost and the final cost in resource allocation to the requests for a better understanding of how the algorithm works. Also, Figure 11 shows the amount of resource allocation cost per replicate. As can be seen, this algorithm can find the optimal path correctly and after 100 iterations can propose the desired results at the lowest cost. Table 7 and Table 8 represent the details of how to allocate a resource to received requests.

The number of resources is considered to be 100, where, for each of these resources, the parameters related to CPU capacity, memory, and so forth are given values according to the initial values in Table 9.

4.2. Applying Genetic Algorithm. After applying the PSO to data related to the defined requests and resources in the system, it is time to implement the genetic algorithm. Thus, after performing the PSO and achieving the best cost in allocating resources for the received requests to balance genetic servers, the GA_algorithm is applied to the best results of the particle swarm optimization algorithm. Thus, we believe that performing this method can reduce the cost of resource allocation for received requests.

In this research, we also intend to minimize the cost defined. By comparing Table 10 with the final output of the

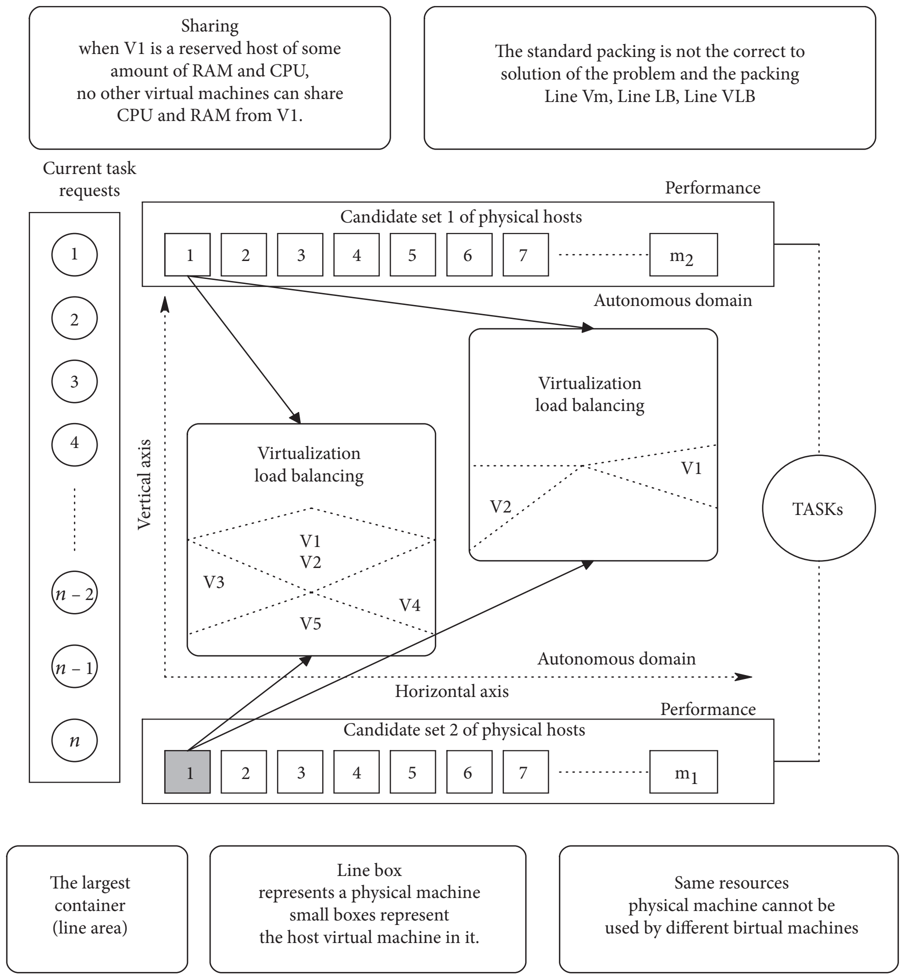


Figure 4: A hierarchical scheduling framework is used for AD.

Table 2: An example of available resources.

Application number	CPU (GHz)	RAM (GB)	HDD (T)
1.1	7.88	32	2.
2.1	7.44	32	2.
3.1	6.48	16	2.
4.1	6.48	16	2.

Table 3: Example of requests received by the cloud.

Application number	CPU (GHz)	RAM (GB)	Memory (GB)
1.1	3.03	12	100
2.1	2.04	12	200
3.1	3.18	32	200
4.1	2.15	32	200
1.1	4.12	32	400
2.1	4.24	8	100
3.1	7.22	16	400

PSO algorithm and Table 11 showing the parameters of the genetic algorithm, it can be concluded that the final cost considered to allocate resources and load balancing has been reduced desirably in comparison to the cost obtained in the PSO algorithm. Therefore, the LBPSGORA can propose appropriate allocations that maintain the load balancing between the servers. Figure 12 shows a comparison of the initial cost and the final cost in allocating resources for the requests for a better understanding of how the algorithm works. Also, this method reduces the cost of resources to requests, and Figure 12 shows the amount of allocation costs of resources in each iteration. This algorithm can find an optimal path and can propose the desired results with the minimum cost after 100 iterations.

4.3. Runtime Calculation. In this sector, we investigate the runtime calculation for the LBPSGORA, PSO algorithm, and genetic algorithm. To be fair to the obtained results, we assume that the numbers of rings are equal, as depicted in Figure 13. The numerical value of the LBPSGORA runtime in different repetitions is shown in Table 12.

The variable of the cost of consumption is taken as a threshold to load balancing in the cloud network and optimizing energy consumption. By spending more time and developing a hybrid algorithm, we have improved energy consumption and cost. Without load balancing, when clients send requests to the cloud network, data transmission operations are complicated, and retransmission occurs. As a result, it needs to spend money and extra energy (Table 12).

Table 4: Resource parameters and their initialization range.

Components	Application	Description
v	The velocity of each particle	Each particle has a velocity vector such that $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$
x	The location of each particle	Each particle has a position vector such that $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$
$\mathbf{C}_1, \mathbf{C}_2$	Two random coefficients	c_1 , c_2 are acceleration coefficients. The acceleration of each particle is directed to $\mathbf{P}_{\text{best}}$ and $\mathbf{G}_{\text{best}}$
$\mathbf{p}_{\text{best}}$	Best value	The particle's best known position
$\mathbf{g}_{\text{best}}$	The best cost considered	Moving to the best position that has been in the neighborhood of the swarm from the beginning until now

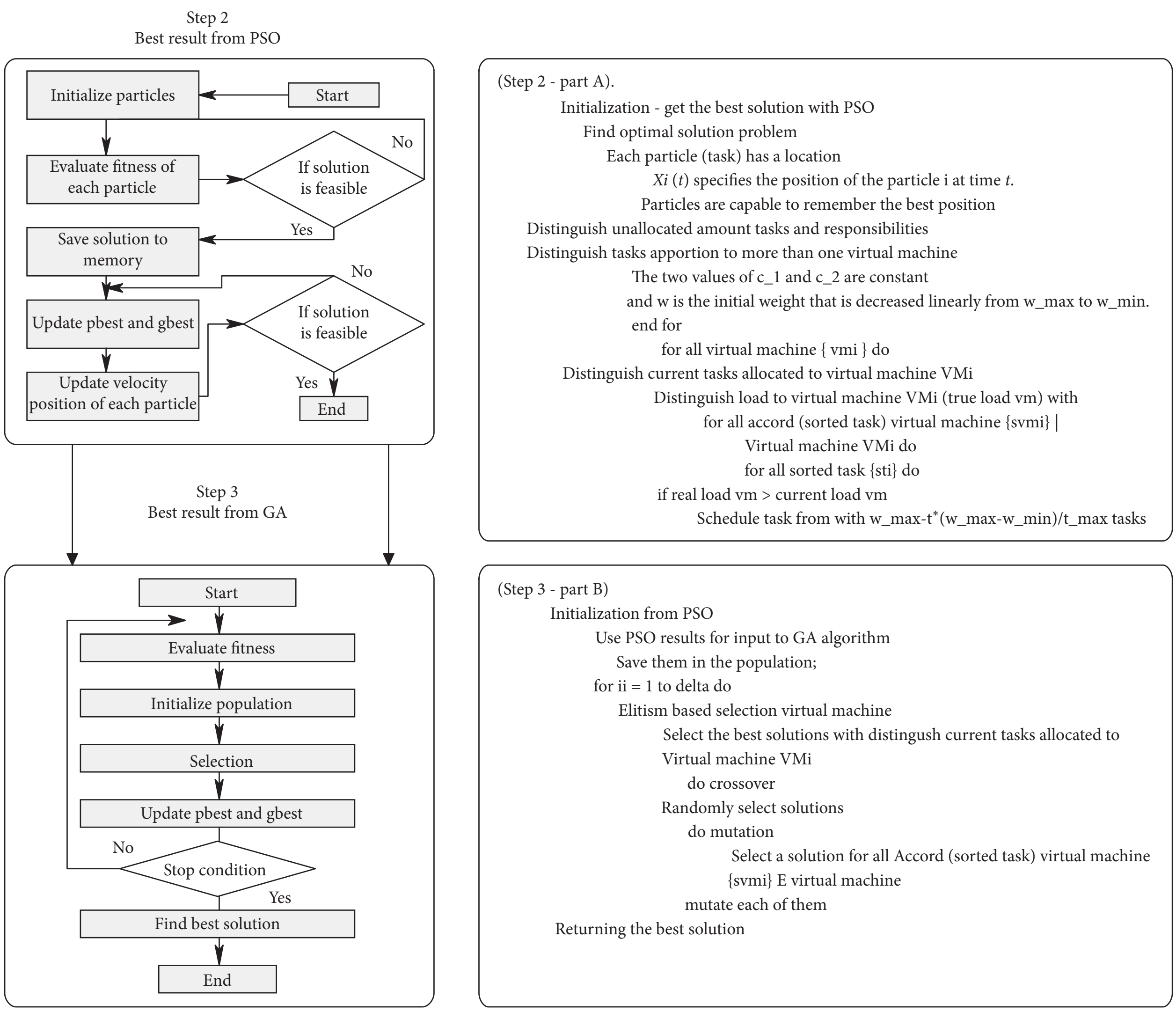


Figure 5: The proposed algorithm. Step 2 is related to PSO for distinguishing tasks apportion to more than one virtual machine and step 3 is related to GA for returning the best solution.

4.4. Check the Load Balance against Tasks. In this section, the details of LB in exchange for increasing tasks in the physical machine are discussed. In the proposed method, the LBPSGORA combination is intended to handle tasks.

The proposed method is compared with the method in [16], hybrid GA-PSO, and GA algorithms. In hybrid GA-PSO method, due to having higher complexity compared to our proposed method, it has a weaker load balancing compared to our proposed method. The difference between our work and the article is that the authors therein used a combination of PSO + GA for their method and the two methods depend on internal steps to create the initial population. In LBPSGORA method, two separate algorithms work, and the optimal PSO response for the initial population of GA is considered.

As can be seen in Figure 14, the proposed method has better conditions for increasing tasks. This makes it less

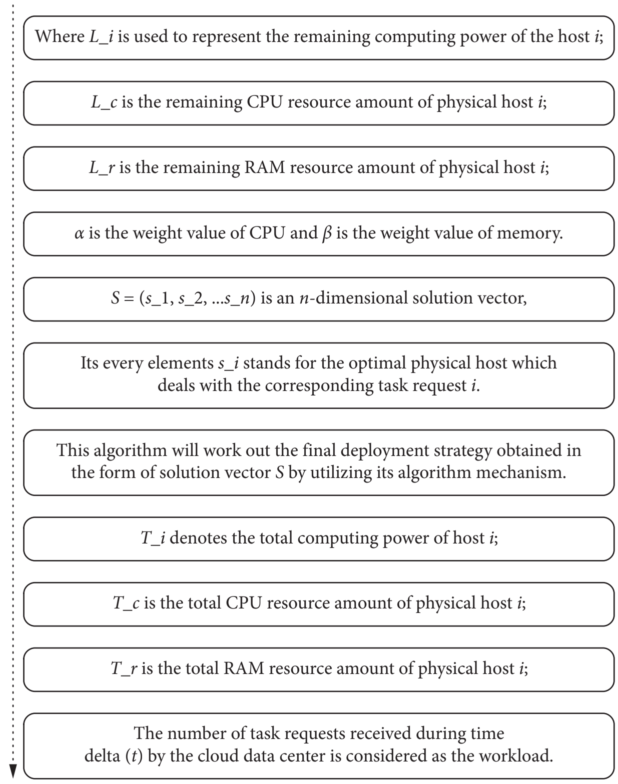


Figure 6: Each step to measure the proximity of the current solution to the aim using a criterion.

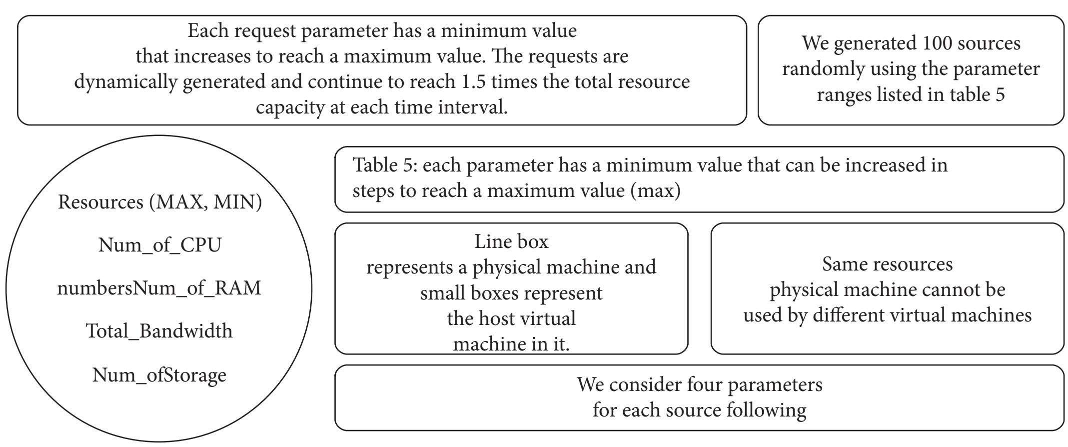


Figure 7: Resources management policy and allocating requests.

Table 5: Resources and initialization range.

Resources (MAX, MIN)	Num_of_CPU numbers	Num_of_RAM	Total_Bandwidth	Num_of Storage
MinRe	8 (2.4 GHz)	32.2	10	1
MaxRe	7 (3.8 GHz)	32.4	8	200
StepsRe	4 (2.5 GHz)	32.7	4	200

Table 6: An example of requests and their range.

Resources (MAX, MIN)	Num_of_CPU numbers	Num_of_RAM (GB)	Total_Bandwidth (GB)	Num_of Storage (TB)
Maximum	7.00 (4.2 GHz)	16	10	1
Minimum	11.00 (4.4 GHz)	256	1	100
Steps	5.00 (7.2 GHz)	256	1	100

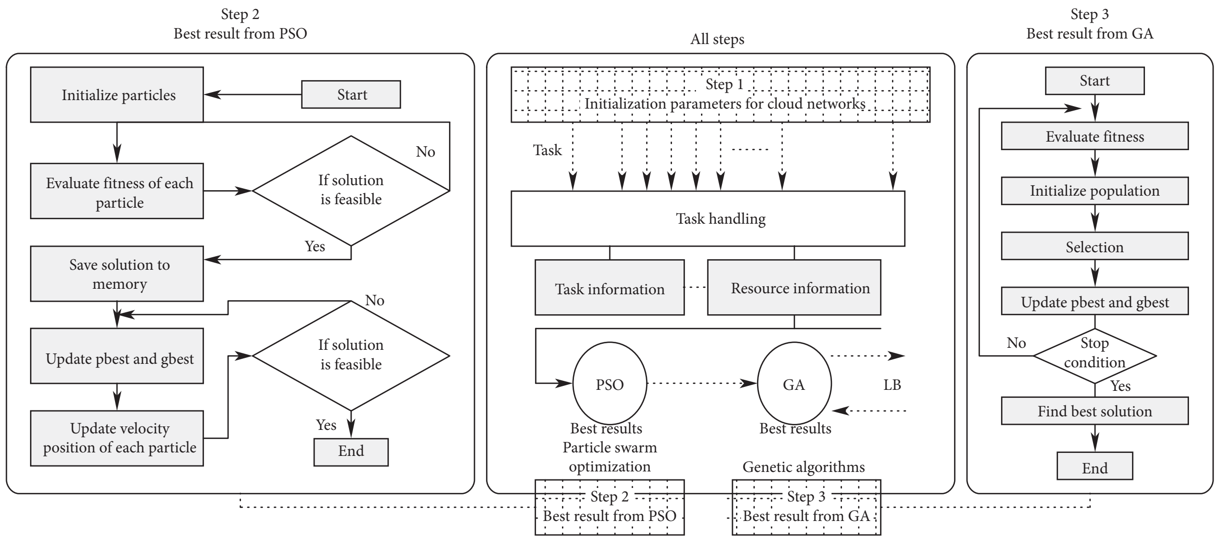


Figure 8: The steps of this research (LBPSGORA algorithm).

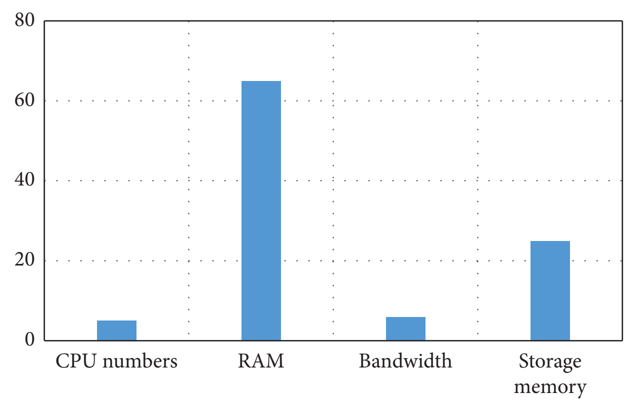


Figure 9: The average number of each virtual machine in the resource parameters.

Table 7: The parameters used in the PSO algorithm.

Parameter	Description	Value
nVar	Number of decision parameters	1
MaxIt	Maximum number of rings	1000
W	Initial weight	1
Wdamp	Weight change coefficient	0.99
c1	Personal learning rate	1.5
c2	Basic learning rates	2

Table 8: Initial values attributed to requests from 1 to 3.

Request ID	Resource ID assigned	Request value in parameter order	The number of resources in the parameter order	Cost
1	88	[3, 11, 6, 2]	[6, 64, 8, 35]	0.0618
2	94	[3, 35, 5, 9]	[3, 96, 10, 21]	0.0617
3	17	[2, 51, 6, 3]	[7, 128, 8, 7]	0.06204

Table 9: Final values attributed to requests from 1 to 3.

Request ID	Resource ID assigned	Request value in parameter order	Source value in parameter order	Cost
1	35	[3, 11, 6, 2]	[8, 112, 10, 23]	0.0615
2	26	[3, 35, 5, 9]	[12, 141, 5, 10]	0.0616
3	45	[2, 51, 6, 3]	[25, 150, 20, 20]	0.0616

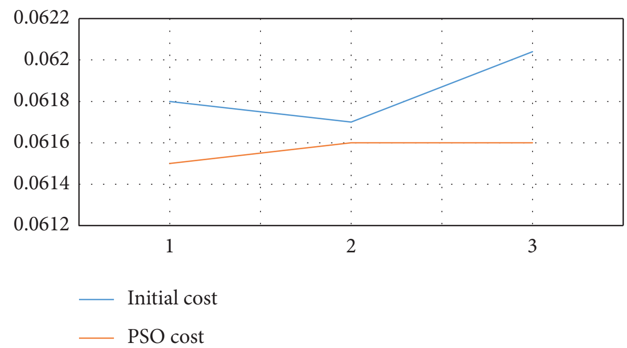


FIGURE 10: Comparison of the initial cost and final cost of assigning the source to requests from 1 to 3.

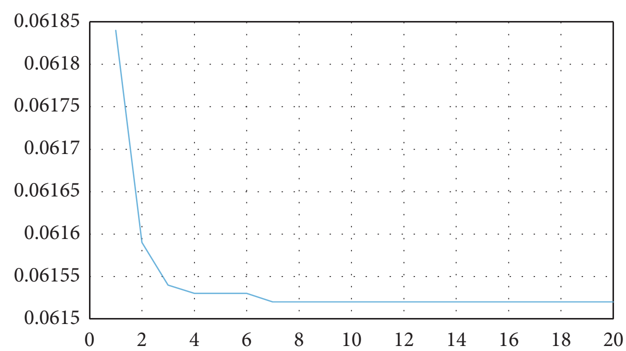


FIGURE 11: The cost of resource allocation in different replays of implementation of the PSO algorithm.

TABLE 10: Parameters used in the genetic algorithm.

Parameter	Description	Value
nPop	Population size (number of request parameters)	4
n_gen	Number of generations	1000
Rp	Crossover percentage or crossover	0.8
Mu	Percentage of mutation or mutation	0.5

TABLE 11: Final values attributed to requests from 1 to 3.

Request ID	Resource ID assigned	Request value in parameter order	The amount of resources in the parameter order	Cost
1	35	[3, 11, 6, 2]	[25, 308, 52, 71]	0.0491
2	26	[3, 35, 5, 9]	[28, 303, 58, 64]	0.0608
3	45	[2, 51, 6, 3]	[25, 332, 51, 78]	0.0600

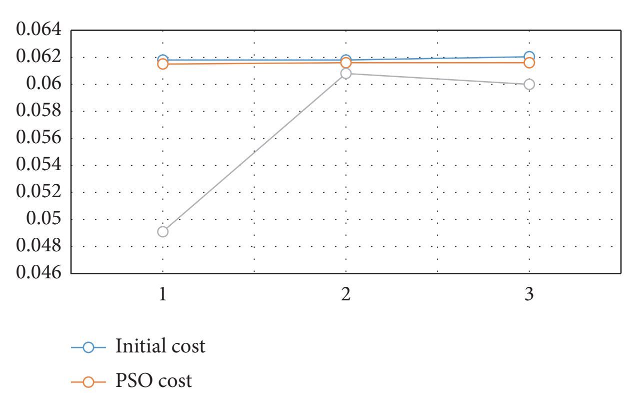


FIGURE 12: Comparison of the initial cost and final cost in resource allocation to requests 1 to 3.

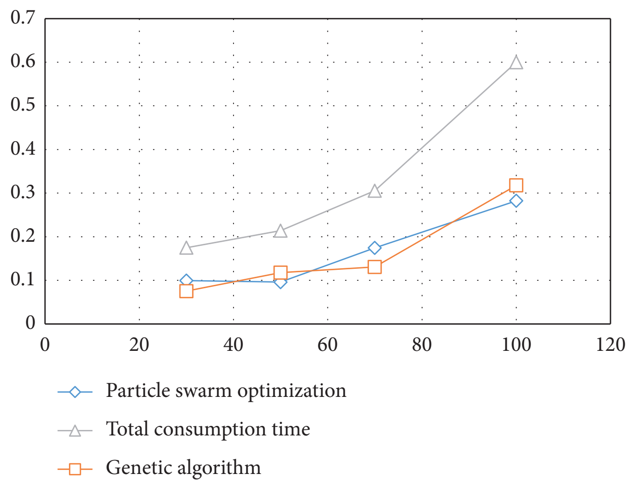


Figure 13: Comparison of the runtime of the LBPSGORA in different repetitions in seconds.

Table 12: The numerical value of the LBPSGORA runtime in different repetitions.

Number of iterations	PSO algorithm/t (s)	GA algorithm/ t (s)	Total time spent/t (s)
30	0.099411	0.075396	0.174807
50	0.096196	0.117715	0.213911
70	0.174365	0.130853	0.305218
100	0.282268	0.317631	0.599899

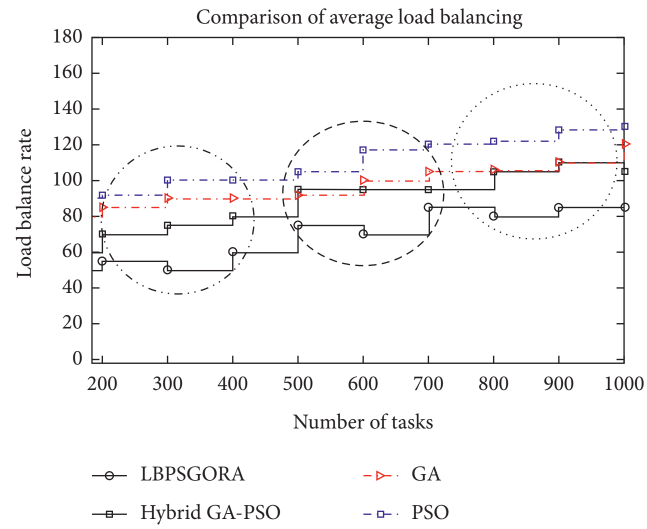


Figure 14: Load balancing rates in LBPSGORA and hybrid GA-PSO.

energy-consuming. The PSO and GA methods are also unfavorable conditions for observing the LB of the other methods. The circles in Figure 15 show the effect of increasing tasks on LB. With the increase of tasks, we see that the load created on the network has increased. As shown in Figure 13, the proposed method is similar to the compared methods in different cases and the improvement rate of the proposed method compared to hybrid GA-PSO is 8.42%, and it is 10.61% compared to GA and 11.71% compared to PSO.

Also, the simulation results for tasks 100 to 1000 are examined in detail in Figure 15. The improvement rate of LBPSGORA method compared to hybrid GA-PSO, respectively, is as follows: 5.12% for 100 tasks, 3.67% for 200 tasks, 4.18% for 400 tasks, 2.28% for 600 tasks, 5.12% for 800 tasks, and 3.23% for 1000 tasks.

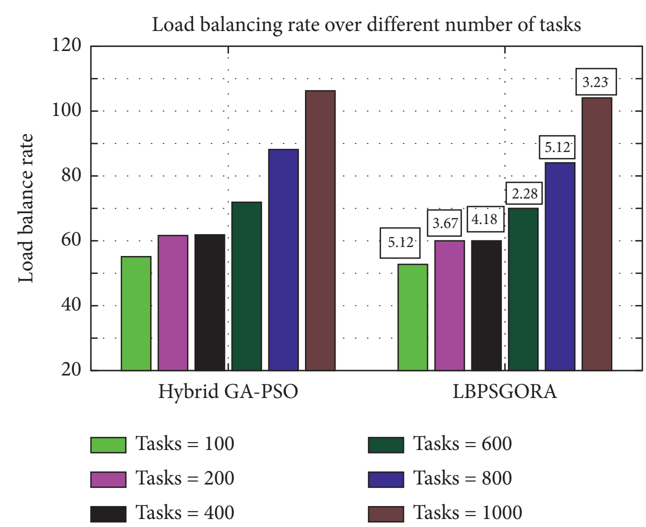


Figure 15: Load balancing changes in LBPSGORA and hybrid GA-PSO.

Also, the simulation results for the 100 to 1000 tasks are discussed in detail in Figure 15. The improvement rate of LBPSGORA method compared to hybrid GA-PSO in makespan, respectively, is as follows: 2.1% for 100 tasks, 1.39% for 200 tasks, 1.36% for 400 tasks, 2.74% for 600 tasks, 1.06% for 800 tasks, and 2.60% for 1000 tasks.

4.5. Check Makespan against Task Execution. This section compares the proposed LBPSGORA method with the hybrid GA-PSO method and the number of makespan changes against the number of tasks. As shown in Figure 16, as the number of tasks increases in different cases, the proposed method and other methods will be changed. Because the tasks in the proposed method are more efficient than the hybrid GA-PSO method, makespan changes in various tests are due to the use of two-stage PSO and GA and network updates. In some incremental modes, load balancing changes are also observed. The proposed method is similar to other compared methods in different cases and the improvement rate of the proposed method is 7.32%.

4.6. Task Execution Cost. Figure 17 examines the costs of task execution changes in different implementation of tasks, and, as can be seen, the proposed method has a lower implementation cost task execution compared to other methods. In the hybrid GA-PSO method, the cost changes are similar to the performance of the proposed method and the LBPSGORA is 6.87% more efficient. As can be seen, with the increase in the number of tasks, the cost task execution for both methods of implementation costs is increasing. Also, the simulation results for 100 to 1000 tasks are reviewed in detail. The

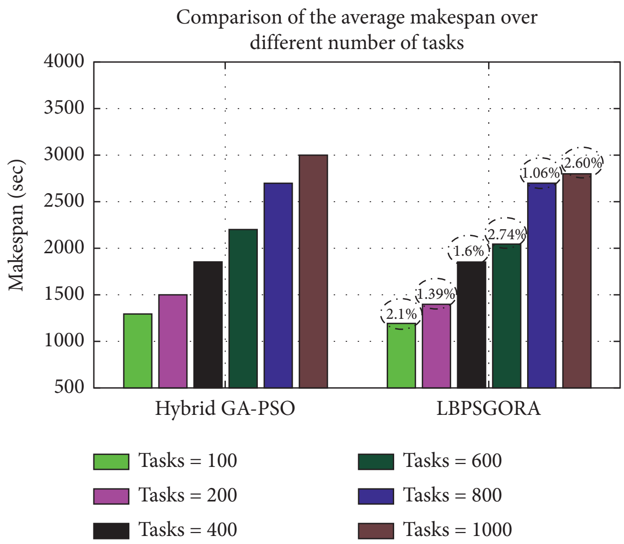


FIGURE 16: Makespan implementation changes in LBPSGORA and hybrid GA-PSO.

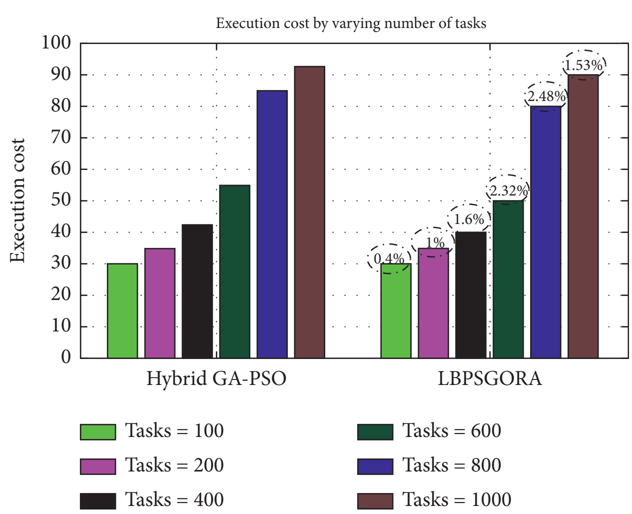


FIGURE 17: Execution cost in the LBPSGORA and hybrid GA-PSO.

improvement cost task execution rate of LBPSGORA compared to hybrid GA-PSO, respectively, is as follows: 0.4% for 100 tasks, 1% for 200 tasks, 1.6% for 400 tasks, 2.32% for 600 tasks, 2.48% for 800 tasks, and 1.53% for 1000 tasks.

5. Conclusions

The most significant challenges that cloud service providers face are managing physical resources and scheduling tasks by taking into consideration the reduction of energy consumption throughout the network. In the LBPSGORA approach, instead of randomly assigning the initial population to the genetic algorithm, the best result obtained from the PSO algorithm is considered as the initial population in GA. The hybrid GA-PSO method is similar in performance to the proposed method with tasks changes. The LBPSGORA method is 6.87% more efficient in execution cost and 7.32% more efficient in makespan compared to hybrid GA-PSO. In load balancing, LBPSGORA was 8.42% superior to the hybrid GA-PSO, 10.61% superior to GA, and 11.71% superior to PSO. For future work of this research, the use of other heuristic methods with load balancing and scheduling problems will be considered. Besides, reliability and response time considerations are other areas of future research.

Data Availability

The data underlying this article will be shared on reasonable request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. M. Mirmohseni, C. Tang, and A. Javadpour, "Using markov learning utilization model for resource allocation in cloud of thing network," *Wireless Personal Communcations.*vol. 115, pp. 653–677, 2020.
[2] A. Javadpour, G. Wang, S. Rezaei, and K.-C. Li, "Detecting straggler MapReduce tasks in big data processing infrastructure by neural network," *Journal of Super Computing*, vol. 76, pp. 6969–6993, 2020.
[3] M. Farid, R. Latip, M. Hussin, and N. A. W. Abdul Hamid, "A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing," *Symmetry (Basel)*, vol. 12, no. 4, 2020.
[4] S. M. Bozorgi, M. R. Hajiabadi, A. A. R. Hosseinabadi, and A. K. Sangaiah, "Clustering based on whale optimization algorithm for IoT over wireless nodes," *Soft Computing*, vol. 25, no. 7, pp. 5663–5682, 2021.
[5] A. Mosa and N. W. Paton, "Optimizing virtual machine placement for energy and SLA in clouds using utility functions," *Journal of Cloud Computing*, vol. 5, no. 1, p. 17, 2016.
[6] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.
[7] Y. Ding, X. Qin, L. Liu, and T. Wang, "Energy efficient scheduling of virtual machines in cloud with deadline constraint," *Future Generation Computer Systems*, vol. 50, pp. 62–74, 2015.
[8] A. Javadpour, G. Wang, S. Rezaei, and S. Chend, "Power curtailment in cloud environment utilising load balancing machine allocation," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1364–1370, IEEE, Guangzhou, China, October 2018.
[9] A. K. Sangaiah, M. Y. Suraki, M. Sadeghilalimi, S. M. Bozorgi, A. A. R. Hosseinabadi, and J. Wang, "A new meta-heuristic algorithm for solving the flexible dynamic job-shop problem with parallel machines," *Symmetry (Basel)*, vol. 11, no. 2, 2019.

[10] A. K. Sangaiah, S. R. Ali, S. R. H. Ali et al., "Energy-aware geographic routing for real time workforce monitoring in industrial informatics," in *Proceedings of the IEEE Internet Things Journal*, p. 1, NewYork, NY, USA, February 2021.
[11] S. G. Domanal and G. R. M. Reddy, "Optimal load balancing in cloud computing by efficient utilization of virtual machines," in *Proceedings of the Communication Systems and Networks (COMSNETS), 2014 Sixth International*, pp. 1–4, Bangalore, India, January 2014.
[12] A. Javadpour, "Providing a way to create balance between reliability and delays in SDN networks by using the appropriate placement of controllers," *Wireless Personal Communications*, vol. 110, pp. 1057–1071, 2019.
[13] A. Javadpour, G. Wang, and S. Rezaei, "Resource management in a peer to peer cloud network for IoT," *Wireless Personal Communications*, vol. 115, pp. 2471–2488, 2020.
[14] A. Javadpour, N. Adelpour, G. Wang, and T. Peng, "Combing fuzzy clustering and PSO algorithms to optimize energy consumption in WSN networks," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing Advanced Trusted Computing Scalable computing. Communications Cloud Big Data Computing Internet People Smart City Innovations*, pp. 1371–1377, Guangzhou, China, October 2018.
[15] A. Priyadharshini and T. Nadu, "A survey on security issues and countermeasures in cloud computing storage and a tour towards multi-clouds," *International Journal of Engineering and Technology*, vol. 1, no. 2, pp. 1–10, 2013.
[16] A. Javadpour, K. Saedifar, G. Wang, and K.-C. Li, "Optimal execution strategy for large orders in big data: order type using Q-learning considerations," *Wireless Personal Communications*, vol. 112, pp. 123–148, 2020.
[17] A. Javadpour, "Improving resources management in network virtualization by utilizing a software-based network," *Wireless Personal Communications*, vol. 106, no. 2, pp. 505–519, 2019.
[18] A. M. Manasrah and H. Ba Ali, "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1934784, 16 pages, 2018.
[19] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 305–316, 2016.
[20] S. Banerjee, M. Adhikari, and U. Biswas, "Development of a smart job allocation model for a Cloud Service Provider," in *Proceedings of the Business and Information Management (ICBIM), 2014 2nd International*, pp. 114–119, Durgapur, India, January 2014.
[21] A. Javadpour, "An optimize-aware target tracking method combining MAC layer and active nodes in wireless sensor networks," *Wireless Personal Communications*, vol. 108, pp. 711–728, 2019.
[22] A. Javadpour and H. Memarzadeh-Tehran, "A wearable medical sensor for provisional healthcare," in *Proceedings of the ISPTS 2015 - 2nd International. Symposium on Physics.and Technologyof Sensors Dive Deep Into Sensors*, pp. 293–296, Pune; India, March 2015.
[23] A. Javadpour, H. Memarzadeh-Tehran, and F. Saghafi, "A temperature monitoring system incorporating an array of precision wireless thermometers," in *Proceedings of the International Conference on Smart Sensors and Application (ICSSA)*, pp. 155–160, Kuala Lumpur, Malaysia, May 2015.
[24] A. Javadpour, G. Wang, and X. Xing, "Managing heterogeneous substrate resources by mapping and visualization based on software-defined network," in *Proceedings of the IEEE International Conference on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 316–321, December 2018.
[25] F. Ja'fari, S. Mostafavi, K. Mizanian, and E. Jafari, "An intelligent botnet blocking approach in software defined networks using honeypots," *Journal of Ambient Intelligence Humanized Computing*, vol. 12, no. 2, pp. 2993–3016, 2021.
[26] A. K. Sangaiah, M. Sadeghilalimi, A. A. R. Hosseinabadi, and W. Zhang, "Energy consumption in point-coverage wireless sensor networks via bat algorithm," *IEEE Access*, vol. 7, pp. 180258–180269, 2019.
[27] P. Jain and S. K. Sharma, "A Systematic Review of Nature inspired Load Balancing Algorithm in Heterogeneous Cloud Computing Environment," in *Proceedings of the 2017 Conference on Information and Communication Technology (CICT)*, pp. 1–7, Ghaziabad, India, November 2017.
[28] B. Huang, W. Wei, Y. Zhang et al., "A task assignment algorithm based on particle swarm optimization and simulated annealing in Ad-hoc mobile cloud," in *Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, October 2017.
[29] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404–413, 2017.
[30] F. Ramezani, M. Naderpour, and J. Lu, "A multi-objective optimization model for virtual machine mapping in cloud data centres,," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2016*, pp. 1259–1265, Vancouver, BC, Canada, July 2016.
[31] Y. Sun, C. Li, Y. Huang, and L. Yang, "Energy-efficient resource allocation in C-RAN with fronthaul rate constraints," in *Proceedings of the 2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, pp. 1–6, Yangzhou, China, October 2016.
[32] S. Wang, Z. Liu, Z. Zheng, Q. Sun, and F. Yang, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in *Proceedings of the 2013 International Conference on Parallel and Distributed Systems*, pp. 102–109, Seoul, Korea, December 2013.
[33] M. Abdullahi, M. A. Ngadi, and S. I. Dishing, "Chaotic symbiotic organisms search for task scheduling optimization on cloud computing environment," in *Proceedings of the2017 6th ICT International Student Project Conference (ICT-ISPC)*, pp. 1–4, Johor, Malaysia, May 2017.
[34] L.-D. Chou, H.-F. Chen, F.-H. Tseng, H.-C. Chao, and Y.-J. Chang, "DPRA: dynamic power-saving resource allocation for cloud data center using particle swarm optimization," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1554–1565, 2018.
[35] D. Pandit, S. Chattopadhyay, M. Chattopadhyay, and N. Chaki, "Resource allocation in cloud using simulated annealing," in *Proceedings of the Applications and Innovations in Mobile Computing (AIMoC), 2014*, pp. 21–27, Kolkata, India, February 2014.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6675759, 19 pages
https://doi.org/10.1155/2021/6675759

Research Article

TCDABCF: A Trust-Based Community Detection Using Artificial Bee Colony by Feature Fusion

Zhihao Peng,[1] **Mohsen Rastgari**,[2] **Yahya Dorostkar Navaei**,[3] **Raziyeh Daraei**,[4] **Rozita Jamili Oskouei**,[4] **Poria Pirozmand**,[1] **and Seyed Saeid Mirkamali**[5]

[1]*School of Computer and Software, Dalian Neusoft University of Information, Dalian 116023, China*
[2]*Department of Computer Engineering Germi Branch, Islamic Azad University, Germi, Iran*
[3]*Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[4]*Department of Computer Science and Information Technology, Mahdishahr Branch, Islamic Azad University, Mahdishahr, Iran*
[5]*Department of IT and Computer Engineering, Payame Noor University (PNU), Tehran, Iran*

Correspondence should be addressed to Seyed Saeid Mirkamali; s.mirkamali@pnu.ac.ir

Received 26 December 2020; Revised 16 January 2021; Accepted 2 March 2021; Published 28 March 2021

Academic Editor: Petr Hájek

Social network aims to extend a widespread framework to communicate users and find alike people with common features, easier and faster. As people usually experience in everyday life, social communication can be formed from common groups with almost identical properties. Detecting such groups or communities is a challenging task in various fields of social network analysis. Many researchers intend to develop algorithms that work effectively and efficiently on social networks. It is believed that the most influential user in a community that had been followed by similar users could be a central point of a community or cluster, and the similar user would be members of the community. Research studies tend to increase intracommunity similarity and decrease intercommunity similarity to improve the performance of the community detection methods by finding such influential users accurately. In this paper, a hybrid metaheuristic method is proposed. In the proposed method called trust-based community detection using artificial bee colony by feature fusion (TCDABCF), we use a fusion approach combined with artificial bee colony (ABC) to improve the accuracy of the community detection task. In this approach, not only the social features of users are considered but also the relationship of trust between users in a community is also calculated. So, the proposed method can lead to finding more precise clusters of similar users with influential users in the center of each cluster. The proposed method uses the artificial bee colony (ABC) to find the influential users and the relation of their followers accurately. We compare this algorithm with nine state-of-the-art methods on the Facebook dataset. Experimental results show that the proposed method has obtained values of 0.9662 and 0.9533 for NMI and accuracy, respectively, which has improved in comparison with state-of-the-art community detection methods.

1. Introduction

The expansion of social network users and interaction between them indicates the complex structure of these networks that needs to be analyzed [1, 2]. Humans have always preferred to live in a society by forming groups consisting of small or large communities. Communities can be formed based on different characteristics that have been distributed over long distances around the world. This is possible due to the available social networks [3]. A network is composed of nodes and edges where the edges represent the relationships between the nodes. Complex networks can be displayed graphically. This show facilitates an understanding of its structure and reveals some important information. The structure of such a network can be shown as a graph in which nodes associate individuals and links compress friendships between them [4]. Social networks can be divided into social groups based on individuals' preferences or features, and social groups refer to a group of socially connected individuals with a closer relationship [5, 6]. It is clear that people in social networks are self-organized in communities where the nodes of the same one are very dense and close in comparison to the rest of the network.

The growth of social networks is the main way of communication between people. But, trust is the foundation of these connections. On the contrary, with the expansion of social networks in the electronic environment and since this environment is less tangible than the external one and its dark spots are more, the concept of security and trust has become crucial [7, 8]. In many computing systems, information is generated and processed by many people. Trusted agents can be a very useful resource for collecting, filtering, and ordering information. Also, if the trust is used to support decision-making, it is important to have an accurate estimate of that, especially, if it is not directly available. Influential users who are trusted can be good agents for interacting and sharing preferences. They can also be a good head cluster candidate node on a social network since users are looking to connect with influential and trusted similar agents [9].

Social networks are aggregated from communities that, in a community, nodes inside a community are more related, whereas nodes between other communities are less related and more separated [10]. Determining the overall features of communities is a challenging concern because some of the people in a community have different features but they are common in a few ones. So, finding common features in a community leads to finding similar mutual users in a specific community. In general, community detection can be considered as clustering [11, 12]. The main concept of clustering is to use local similar features as similarity measures to detect clusters and find candidate points representative of these measures as cluster head nodes. In the case of social network analysis, community detection plays the same role as finding influential users as a community head node. Also, similarity measures can be considered as social features of users consisting of membership in a specific group, and it could also express the same ideas, follow reciprocal users, and so on.

In social networks, usually, the users are forced to communicate together, and it is one of the inevitable certain rules of this association. Otherwise, individuals and communities have to observe basic rules to achieve success targeted in social media. Trust is the basic requirement of social communication, for a healthy social network, and the main motivation of this is to take into account the sense of belonging and loyalty necessary for success in all areas. Social trust reflects the confidence to each of the individuals in a community for sufficient utilization of resources and license to reduce formal barriers to reduce the cost of security extension in all aspects of a network [13]. Therefore, to gain an integrated community, it is required to establish a healthy social structure in that users trust each other and influential agents, respectively.

In this paper, a trust-based community detection using artificial bee colony by features fusion (TCDABCF) is proposed. The purpose of this method is to find influential and reliable users in the network that are considered as cluster head nodes in communities. Such users can be a good feed source for other users and have a strong connection with them, so they can be a good central point for an integrated community. By searching among users surrounded by candidate agents, similar users can be found who have a trust-based connection with the agents, as members of the community. Finding such users is considered as an optimization issue in this paper. We use a metaheuristic approach to trade-off a balance between the goals of community discovery and building trust-based communities. The main contribution of this paper is given in the following:

(i) Adapting the metaheuristic approach of artificial bee colony (ABC) as a clustering algorithm based on the trust between people in a community is the novelty and the main contribution of this work. In this approach, we use artificial bee colony (ABC) to find influential users as candidate agents for the central points of communities, based on the centrality criterion.

(ii) Defining a constrain-based optimization problem based on the fusion of trust and social similarity and related constraints in the social network as the fitness function of the bee colony optimization algorithm is the second contribution of the proposed method.

(iii) Defining intraclusters' distance and interclusters' distances to find the level of trust between users and agents is another contribution of the proposed method. In this method, we find similar users as candidate agents by the metaheuristic exploration due to the result of the proposed constraint-based optimization fitness function.

The rest of this paper is organized as follows: in the second section, related works will be discussed. In the third section, the preliminaries and algorithm of the proposed method will be detailed. In the fourth section, the implementation and result of the experiments will be shown. Finally, we will discuss and conclude this paper in Section 5.

2. Related Work

In recent years, due to the importance of community detection in social networks and effectiveness of the research area for analysis of it, many community detection algorithms have been developed to reveal community structures in complex social networks [14]. In this section, some existing community detection algorithms are briefly introduced.

Artificial bee colony (ABC) is a swarm intelligence-based evolutionary algorithm that is inspired by exploring the food source behavior of honey bees. Similar to many other metaheuristic algorithms such as differential evolution (DE) algorithm, ant colony (ACO) algorithm, particle swarm optimization (PSO), and gravitational emulation that are adapted to solve many different problems such as routing [15, 16], image segmentation [17–22], and many other clustering problems [23–25], ABC is also used in solving many different problems [26, 27]. In the study by Hafez et al.

[28], artificial bee colony optimization was used for community detection problems for the first time. It had motivated to determine the number of communities automatically. It also is based on this hypothesis that the performance metaheuristics-based community detection algorithms are influenced directly by the quality function used in the optimization process. In the work by Aung et al. [29, 30], ABC community detection is used by introducing exclusive mutation and crossover operations. By creating diversity in the new population, these operators try to find optimal similar users to the influential user in each community. Aung and Nyunt [31] proposed a new community detection method based on modularity ABC that normalizes mutated information to use for the fitness function to increase the performance of the specific community structure. In the work by Saoud [24], an ABC-based approach is used to split communities into high-quality communities by modularity and select one node at each community as a representative concept. Then, each user is assigned to community similarity to a representative node. Moreover, this approach could determine the number of communities, automatically.

Ding et al. [32] examined trust model-based detection schemes, and based on these schemes, they modeled user interactions as a transfer of trust. They focused on the cold start problem, which is one of the most important problems in trust model-based diagnostic schemes. The researchers proposed a new algorithm called TLCDA to identify a community based on the trust model. This algorithm uses the *K*-mediods clustering method to identify communities. The researchers said that the results of the implementation and evaluation of their proposed algorithm showed that this algorithm has the ability to identify coherent communities and also support topological coherence.

Lingam et al. [33] focused on identifying social botnets in online social networks. The researchers first created a weight chart based on behavioral similarities (such as the similarity in interests, similarity in social interactions, and similarity in content tweeted by users) and participants' trust in each other on Twitter. The researchers then developed two algorithms, Social Botnet Community Detection (SBCD) to identify social botnet communities that behave similarly to malicious behavior and Deep Autoencoder-based SBCD (DA-SBCD) to reconstruct and identify social botnet communities. The researchers used the Twitter dataset to implement and evaluate the accuracy of the algorithms proposed by the researchers. The results of their evaluation showed that the DA-SBCD algorithm has 90% accuracy in detecting communities, which when compared to other methods (such as normalized mutual information (NMI)) showed an improvement of 8%.

Sheng et al. [34] studied how to discover the structure of the society in the network in order to extract information. These researchers suggested a new algorithm for identifying communities' novel community detection algorithm NTCD (Community Detection based on Node Trust). This algorithm is an algorithm for identifying stable communities that do not need to set any parameters, and its temporal complexity is almost linear. The proposed algorithm for determining the probability of a node in the current community examines the relationship between the node and its neighboring communities (this relationship is called the Node Trust). Node Trust uses the maximum search method to identify communities. To implement the proposed algorithm in this research, real-time and hybrid networks were used, and the results showed that the proposed algorithm has high accuracy for identifying communities and belonging to a node to a community.

Zhu et al. [35] focused on how various social networks, including Foursquare and Instagram, work. These networks often use two methods, including Point-of-Interest (POI) discovery and friendly advice to make suggestions based on where people live. In other words, social trust among individuals has no effect on the performance of social networks and the suggestions they make to users. In this research, a new algorithm for clustering users based on the level of trust in them has been proposed, which then, using these clusters, a method to predict the level of trust in users was presented. The researchers used the level of trust in people and the similarities in their behaviors to provide advice. To improve the quality of the recommendations made to the user, the researchers proposed a framework for providing POI recommendations based on the user's preferences, the impact of the geographical coordinates of the user's location, and the level of trust in the user. The results of implementation and evaluation of the proposed algorithms showed that the clustering method based on the trust proposed in this research has high efficiency and accuracy in identifying people with similar and reliable desires and interests.

Chen et al. [36] focused on the analysis of social networks, and from the discovery of communities between individuals on social networks to better understand the topological features of real social networks, to monitor public opinion, and to offer personalized suggestions, ideas used by leaders were identified. The researchers considered the characteristic of trusting users in creating links between them. In this study, they proposed a new algorithm for identifying communities between users on social networks. The proposed algorithm has two steps. First, the trust relationship between users was divided into three categories: direct, indirect, and mutual trust. A method was provided to calculate the level of trust between users. Secondly, they used edge fit and fitness between users to identify nonoverlapping communities and to integrate these communities and build a relationship of trust. Then, they proposed a new trust-based algorithm to apply trust between nodes in a community. The results of the implementation of the proposed algorithm using Lesmis and Gemo datasets showed that it can be very effective for deep understanding and discovery of communities among social network users.

Wu et al. [37] investigated the role of the trust factor in communities created on social networks. For this purpose, they reviewed all the research conducted in two databases, SSCI and SCIE. The results of this study showed that most of the research in this field has been done by researchers in the United States, and in this research, more emphasis is placed on creating computational models and proposing systems. Finally, some of the most important limitations in the development of research in this field were analyzed in this research.

Beigi et al. [38] focused on interactions in e-commerce and emphasized on the need to create a factor called trust between users so that they can conduct their interactions in the world of e-commerce with ease. In this research, a new method for identifying communities in the network has been proposed, which has used the relationships created based on trust and similarities in user rankings based on the amount of credibility and trust in them to build relationships in communities. The main purpose of the proposed framework and method in this research is that the value of trust between different people in the society can be used as a predictor of relationships between other people in the society. In this study, two proposed recognition algorithms were used (one to validate people in dealing with people inside the community and the other to validate the level of trust in people based on their behavior inside and outside the network community) to assess the level of trust between people. The results of the implementation of the proposed method showed that this method, compared to other existing methods of predicting trust, in the dataset of the famous websites reviewing the products of Epinions and Ciao, works better.

Kou et al. [39] studied how social media delivers content to different people on a signed social network (SSN) and modeled some of the more complex systems in the real world using these networks. These networks provide content to users by creating two groups of users (including trusted and untrusted). One of the major challenges in using these networks is that some users on these networks are unknown, so you cannot attach a trusted or untrusted label to them. For this purpose, these researchers proposed a new method called the trust-based missing link prediction (TMLP) method. In this method, first, they use the Simhash method to create a hash index for each user, and then, to determine whether it can create a social relationship between two users or not, they calculate the Hamming distance between two users. Finally, they use the fuzzy computing model to determine the type of new social relationships between users (including trustworthy or untrustworthy).

Baek et al. [40] focused on online social networks (OSN). These networks can facilitate the establishment of new relationships with previously unknown individuals who have similar opinions and interests to the user, and they can also have many security problems (abuse, disclosure of uncontrollable content, dissemination of incorrect information, etc.) to create for them. For this purpose, we need a dynamic model that can evaluate the reliability of users dynamically. In this study, they proposed a dynamic access control method based on trust in online social networks. In this method, various sociological methods were used to understand negative behaviors and limit the level of access dynamically in the communities created by users in online social networks. The evaluation results of this method showed that it can have a high ability to prevent uncontrolled disclosure of information.

Li et al. [41] focused on using data mining techniques to analyze users' behaviors and feelings on mobile social networks. They provided an overview of the user emotion estimation scheme (including chain of trust between users, the meaning of different users' emotions, etc.). Further, they proposed a model for building a chain of trust between users and finally proposed a new way to model emotions by presenting modeling rules. The results of implementing their proposed method showed that their method has been very effective in modeling the trust chain.

Ma et al. [42] examined two important factors in creating communities on social networks (including the quality of grouping and the cost of grouping). To optimize these two factors, they proposed a multiobjective group recognition algorithm based on user relationship analysis that can optimize the two factors, while normalizing the information. In the proposed algorithm, the local search strategy, which is related to the specific knowledge of the problem, is used to improve the effectiveness of the new algorithm. Experiments performed by these researchers on various local and social networks showed that their proposed algorithm can find group structures with high accuracy in social networks.

All the methods mentioned in this section used various algorithms in different ways for community detection, but none of them has paid attention to the importance of trust. The method proposed in this paper, in addition to being inspired by previous methods, has also added the trust factor to the parameters of the fitness function of the ABC optimization algorithm in order to effectively perform community detection in social networks.

3. Methodology

As mentioned earlier, in this paper, the trust-based ABC optimization algorithm by feature fusion is used for community detection in social networks. This method finds influential users in the network. Influential users are the ones who have the most communication on the network. After calculating the degree of trust for every influential user, then the characteristics of influential users and their trust degree are used as parameters of the fitness function in the ABS optimization algorithm to obtain similar users to each influential user and later to form communities. Algorithm 1 shows the algorithm of the proposed method.

3.1. Problem Formulation. The complex networks in the presented method are considered with an adjacency graph, denoted by $G = (V, E)$, where V and E are the vertex and edge sets, respectively [2]. Each vertex in G represents a user in the network, and each edge is a communication between a pair of users. The size of the network is defined by $n = |V|$ which is the number of users and $m = |E|$ which is the number of links. The network structure is demonstrated as a square adjacency matrix $A = (a_{ij})_{n\times n}$, where each element can have one of the values of the set $\{0, 1\}$, that is, $a_{ij} = 1$ if user$_i$ is connected to user$_j$;otherwise, $a_{ij} = 0$.

The main steps in this method include the influential users' identification, the trust level calculation, and the closet user detection using the metaheuristic ABC algorithm. We

Input: M ∗ M adjacency matrix
Output: Communities
Begin
For all users in the social network.
Calculate all user's communication rates.
Calculate all user's trust levels.
End of for
Select influential users with maximum communication rate and trust level as a community center.
Merge overlap influential users due to the direct link.
Extract community centers.
Consider influential users as food sources in ABC.
Search for a similar user to community centers based on fitness function to find employee, onlooker, and scout bees.
Form communities based on fitness values greater than the threshold.
Integrate communities based on overlap users.
End

Algorithm 1: ABC-based community detection algorithm of the proposed method.

describe the proposed algorithm in detail in the following subsections.

3.2. Influential Users' Identification. At the first step in TCDABCF, the main assumption is that the influential user in a social network might have followed by many users, while followers of the influential user may not be familiar with each other. The number of followers of a user is considered as a communication degree, and a user with a larger degree would have a lot of users surrounding it in the local neighborhood. The influential user with a high communication degree is more likely to be a central node of the community. Figure 1 shows an example of a synthetic network that nodes related to influential users, according to their communication degree, have a bigger circle than other nodes.

Considering a case like in Figure 1, if the distance between influential users is too short, they cannot be considered as central points of discrete communities. Therefore, we assumed that the distance of influential users is not less than the average distance of the network. The average network distance is defined as the average of the shortest paths between each two separate node pairs in the network. Hence, we identify the influential users by their communication degree and the distance of other influential users. If the distance between two influential users is less than the threshold, the lower one is ignored and is considered as a follower of the main influential user. So,

$$\tau = \frac{\sum_{i,j=1}^{n} D\left(n_i, n_j\right)}{n},$$
$$D\left(I_i, I_j\right) \geq \tau, \tag{1}$$

where $n_{i,j}$ is the user in the network, τ is threshold, $I_{i,j}$ is the influential user, and D is the shortest distance between two nodes.

3.3. Trust Level Calculation. In online social networks, trust is defined as the belief of one user to another user in a mutual relationship [32]. In other words, the expectation of a user from the other hand user in a cyberspace relationship is considered as trust. Therefore, in social networks, users who have a higher level of trust are considered by other network users and can be present as the main node in the network. Hence, to calculate the trust of the user in social relations in a social network, scholars measured user relation strength as an individual weight that indicates trust level and trust evolution, which are practically useful for improving the accuracy in social network analysis.

Let us consider two users i and j as connected neighbors in a social network G, and their direct trust can be calculated as follows:

$$\text{trust}(i, j) = \frac{w(i, j)}{v(i)}, \tag{2}$$

where $w(i, j)$ indicates all paths between user i and user j and $v(i)$ is the sum of all direct paths between node i and its adjacent nodes.

3.4. ABC Algorithm. ABC is a swarm intelligence-based evolutionary algorithm that is inspired by exploring the food source behavior of honey bees. The ABC algorithm has three types of bees consisting of employee, onlooker, and scout bees which search the fly in the problem space to find the optimum solution. At the initial stage, the bees are grouped by their fitness. An employee bee saves a food source in her mind when she searches in the space, and she demonstrates the quality of food source with onlookers by dancing around that. Onlookers look carefully at the dances of the employee bees and select the best source due to it. Onlookers tend to this source to exploit it. After emptying the food source, it may be released, in which case the scout bees will randomly try to find a new food source. The percentage of scout, employee, and onlookers is usually determined manually.

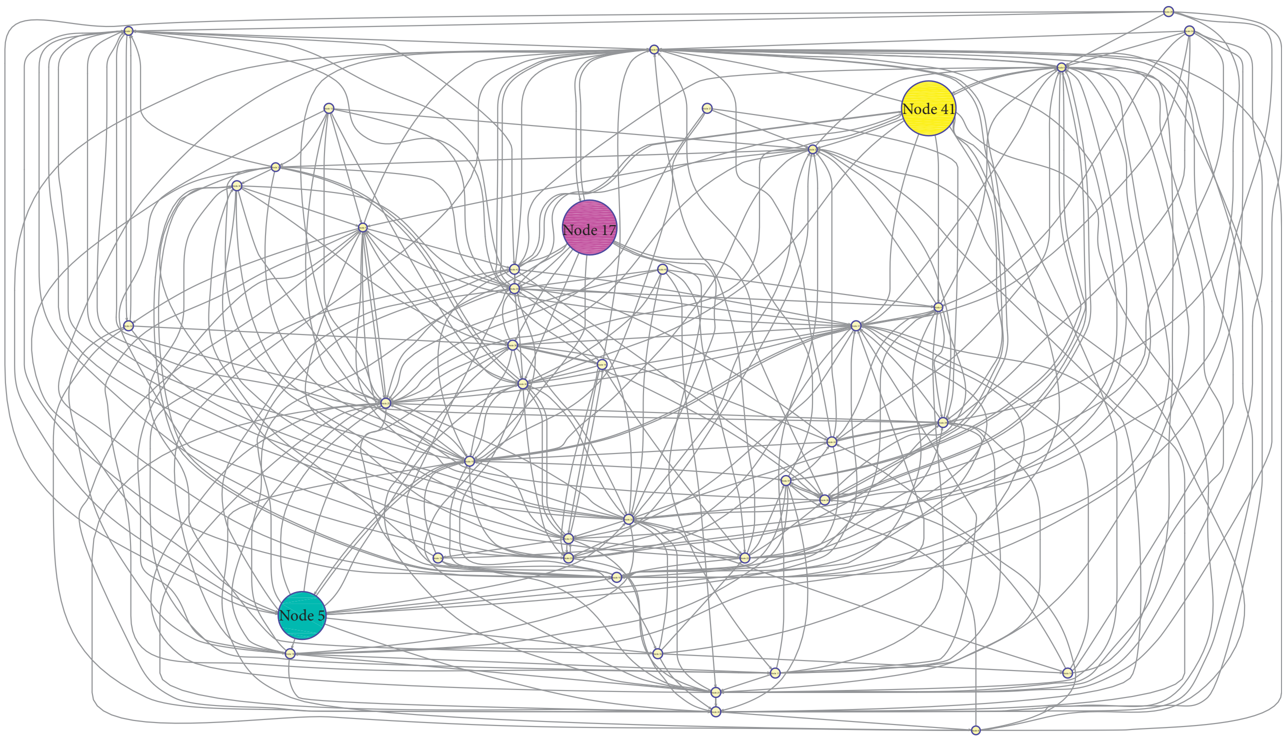


Figure 1: Example of a synthetic network and influential users.

In the ABC algorithm, the position of a food source represents a solution to the optimization problem, and the amount of nectar from the food source is associated with the suitability of the solution. The number of employee bees or onlooker bees is equal to the number of solutions in the community. In the first step, the ABC distributes the initial population randomly ($P = 0$, G solutions of SN food source positions, and N number of bees), where SN represents the population size.

Each solution (food source SN_i, $i = 1, 2, \ldots$) is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of situations is subject to a repetition of the cycle $C = 1, 2, \ldots, C_{\max}$ where C is the process of searching for employee and scout bees of an artificial bee or onlooker which is likely to generate a position change (solution) in their memory to find a new food source and test the amount of nectar (merit value) from the new source (new solution) [33].

The onlooker bee selects a food source according to the probability value associated with that food source, P_i, which is calculated by

$$P_i = \frac{\text{fit}_i}{\sum_{n=1}^{\text{SN}} \text{fit}_n}, \tag{3}$$

where fit_i is the degree of suitability of solution i evaluated by its employee bee, which is proportional to the amount of nectar in the food source at position i, and SN is the number of food sources equal to the number of employee bees. In this process, employee bees exchange information with onlooker bees. In order to generate a preselected food position, ABC uses the following expression:

$$x_{ij} = \varphi_{ij}\left(x_{ij} + x_{kj}\right), \tag{4}$$

where x_{ij} is the previous position of the bee x, $k \in \{1, 2, \ldots, N\}$, and $j \in \{1, 2, \ldots, D\}$, *and* their indexes are randomly selected. Although k is randomly assigned, it is different from φ_{ij} which is a random number between [−1, 1]. It controls the production of the neighbor's food source position around x_{ij} and provides comparative variations of the neighbor's food situation visually. Equation (4) shows different parameters between x_{ij} and x_{kj} and also changes in the position x_{ij}. Therefore, the search approaches of the optimal solution are reduced periodically. The bee moves according to the following equation:

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} + x_{kj}\right), \tag{5}$$

where v_{ij} is the new position of bee $\underline{x}$, $(x_{ij} + x_{kj})$ is the distance between two bees x_i and x_k, i is the current bee, k is the neighboring bee, and j is the next move of bee x_i towards bee x_k.

Here, a dimension is considered. So, a variable (j) is selected. If the new position (new food area) has a better quality (more nectar or more of the same function as the target), the bee stays in the new area; otherwise, it returns to its previous area, and a unit is added to the bee's trial index.

In fact, the trial index is a counter to the number of scout bee movements with no improvement. If the trial index value of a bee exceeds the predefined value, this means that the food area no longer has martyrs and must have left [32, 43].

The second group is scout bees. Employee bees are spread in the food area. Depending on the quality of the nectar and the amount of nectar in that area, they show a movement above their food area, and the guard scout bees find out which area has a better quality according to the movements of the onlooker bees. Therefore, scout bees select these areas accordingly. In other words, those areas that have more nectar have a better chance of being selected by scout bees. Assigning scout bees gives the bees in the area another chance to move. Assigning scout bees means that bees in the area are given a second chance to move. How to calculate the odds is done according to the following equation:

$$P_i = \frac{\text{fit}_i}{\sum_{n=1}^{\text{SN}} \text{fit}_n},$$

$$\text{fit}_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if } f_i > 0, \\ 1 + \text{abs}(f_i) & \text{if } f_i < 0. \end{cases} \quad (6)$$

The third group is the search engines. The bees leave the areas that have been identified as unfavorable for nectar and randomly select other areas. If the stop condition is met, the algorithm pattern stops. Otherwise, we go to the beginning of the ring, which is the movement of the employee bees [44].

3.5. Closet User Detection Using ABC. In the proposed method, the closet user is a user who, in addition to having a direct relationship with influential users in the network, has social characteristics and trust degree that are similar to influential users in the social network. As mentioned earlier, influential users in the network are the users who have the most communication and the highest trust degree. Influential users are surrounded by a multitude of users. In the proposed method, the purpose of determining closet users is to find those followers who are more similar to influential users both in terms of social characteristics and the trust degree. Such users, in conjunction with influential users, can form integrated communities with the same features that are surrounded by a multitude of other users.

In this paper, we use the ABC algorithm in order to find similar users to influential users. In this algorithm, social similarity and trust are selected as evaluation criteria in the fitness function. In the case of community detection, each possible solution defines a set of users that are similar together and linked to an influential user in trusted relation. So, a food source is considered a community due to the coding [28]. In this way, influential users are considered as employee bees, which are the core of the community in the social network. The onlooker bees are looking for the employee bee that performs the best dance. The quality of the dance is based on a combination of similarity and trust as the fitness function. The onlooker bee follows the employee bee, which is more similar to her and less different in their level of trust. Scout bees look for the new community due to the influential users in a social network. Since the social features of a user are represented as a vector in which each element of the vector is considered as an attribute, to calculate the similarity between the two users, the Jacquard criterion can be used as follows:

$$\text{Sim}(i, j) \frac{2|A(i) \cap A(j)|}{A(i) + A(j)}, \quad (7)$$

where $A(i)$ and $A(j)$ are the connected users in the social network [45].

3.6. Fitness Function. The fitness function in the proposed method is a combination of the trust degree of users in the social network and the similarity between users and influential users in the network. As mentioned, the degree of trust of a user is defined as the proportion of the input communications of each user divided by the total network users. The higher trust degree of a user shows that many users among all users in the network know this user and trust this user and have communicated with him. In online social networks, the principle of communication between two users is considered as knowledge and trust. On the contrary, since, in the proposed method, influential users are selected as the central points in each community, the degree of similarity of users to influential users can be an important factor in determining the community. Hence, the proposed fitness function is defined as the fusion of users' trust degree and their similarity to influential users as the central points of any community. So, the fitness function of the proposed method is defined as an optimization problem as follows:

$$\min(f) = \sum_{i=1}^{n}\sum_{j=1}^{m} \text{trust}(i, j) - \text{sim}(i, j),$$

$$\text{s.t.}$$

$$\sum_{i=1}^{n} w(i, j) \leq \frac{n(n-1)}{2} - 1, \quad (8)$$

$$\sum_{i=1}^{n} v(i) \leq \frac{n}{2} - 1,$$

$$E(i, j) \neq 0, \quad i, j > 0,$$

where the total paths between an influential user and its neighbors should not be more than half of the total edges in the network, and the number of neighbors of an influential user should not be more than half of the network users because in that case, the whole social network will be considered as one community.

4. Experimental Results

In this paper, the Facebook dataset stored in the Stanford University Governor's Data Repository is used [37]. This dataset is part of the data related to social media on Facebook. This dataset is used to test many different clustering methods [46]. This dataset consists of "circles" (or "friends' lists") from Facebook that were collected from survey participants using this Facebook app. The Facebook dataset includes node features (profiles), circles, and ego networks. In this dataset, a social media review tool for collecting, analyzing, and visualizing the level of its use and disseminating it in social media has been published. Also, the latest articles that the data collection has used for accurate analysis of community detection can be found in this data collection. Table 1 shows the details of the database.

The complete dataset cannot be distributed due to Facebook's privacy policies and the copyright of the news publisher. Therefore, in this dataset, specific jobs and user information are not disclosed due to Facebook's policy. This dataset can be used to check Facebook's connection to work and data related to social media.

This dataset is constantly being updated due to the development of the social network Facebook and the increase in the amount of work on it and the network content. The latest version of this data collection (located in the data collection folder) includes networks related to schools and colleges. Each school file has a sparse adjacency matrix and a "local_info" variable of feature information.

4.1. Influential Users. As mentioned in the previous sections, in this study, we use the ABC algorithm to validate the work on the social network. In the proposed method, the target community is those users who are connected in a trust chain. The proposed method is under the assumption that influential users in the social network can play an important role in the formation of communities and other users who surround influential users have similar characteristics to the influential users, and these users act as cluster centers or community representations. Therefore, the first step of community detection is to find influential users in the social network. Influential users in the network are extracted according to their communication rate with other users in the social network. Users whose normalized communication exceeds the average of the network are selected as influential users on the network. Figure 2 shows the influential users on the social network in the Facebook dataset.

As shown in Figure 2, influential users on the social network are extracted according to their communication and threshold. Now, according to the figure, it can be seen that some of its influential users may be in direct contact with another influential user. In such a case, if the distance between the two influential users is short and as a result, their similarity to each other can be high. In such cases, we measure the similarity of two effective users, and if there is a similarity between these two users, we merge the two users. In fact, in such a case, a user with a lower communication rate will be considered as a member of the community of the influential user with a higher communication rate. Figure 3 shows the integration of the effective users extracted in the previous step.

As shown in Figure 3, from 10 influential users extracted in the previous step, 5 influential users are merged due to the similarity, and only 5 influential users who are not similar to each other remain. These users are considered as the central point of a cluster related to a community, and the ABC algorithm starts its search and optimization by considering this user as the target.

4.2. Implementation of Community Detection Using ABC. In this article, influential users are considered as food resources. According to the bee optimization algorithm, in the proposed method, first, the employee bees are found, which are the direct neighbors of the influential user. These users are the users who are most likely to be influential users and can belong to the community associated with influential users. Figure 4 shows an example of a total of worker bees associated with an influential user on a social network.

As mentioned in Figure 4, total neighbors of an influential user with index 6 have been selected as employee bees in the initial population. Now, according to the fitness function, among these users, only users who have a fit value higher than the average value are selected as employee bees specified by ABC. Given that the fitness function is a combination of trust and similarity between the target user and other users, the users selected by ABC as the trusted users are who have the most similarity to the influential user. Selecting such users as members of the community, not only reduces intracluster distances in community discovery but will also increase the accuracy of the selected community.

Employee bees are evaluated according to the fitness function, and each of them has taken a value according to the degree of trust and similarity to the relevant influential user. Figure 5 shows the selected employee bees in the ABC algorithm.

As shown in Figure 5, all neighbors of the influential user with index 6 are extracted, but they do not have the qualifications to become an employee user, and only some of these users with a high level of trust and similarity due to the fitness function are selected as the employee user. Thus, according to Figure 5, it can be seen that these users are in one effective step of the influential user and have the most similarity and the least distance from the center of the cluster. In the next step of the proposed method, onlooker bees are considered as close neighbors of employee bees whose fitness function is greater than the average of the total values of the obtained fitness function. These users have a direct relationship with employee bees and are one step away from influential users as a food source. Onlooker users are selected similar to employee bees and therefore can be placed in the relevant community due to the similarity of employee bees and influential users. Figure 6 shows an example of onlooker bees based on the fitness function.

In the next step of the proposed method, scout bees are considered as close neighbors of onlooker bees. These users have a direct relationship with onlooker users and are two steps away from influential users as a food source. Scout users are selected similar to onlooker users and therefore can be placed in the relevant community due to the similarity of onlooker, employee, and influential users. Figure 7 shows an example of scout bees based on the fit function.

Table 1: Details of the Facebook database.

Dataset details	
Nodes	4039
Edges	88234
Nodes in the largest WCC	4039
Edges in the largest WCC	88234
Nodes in the largest SCC	4039
Edges in the largest SCC	88234

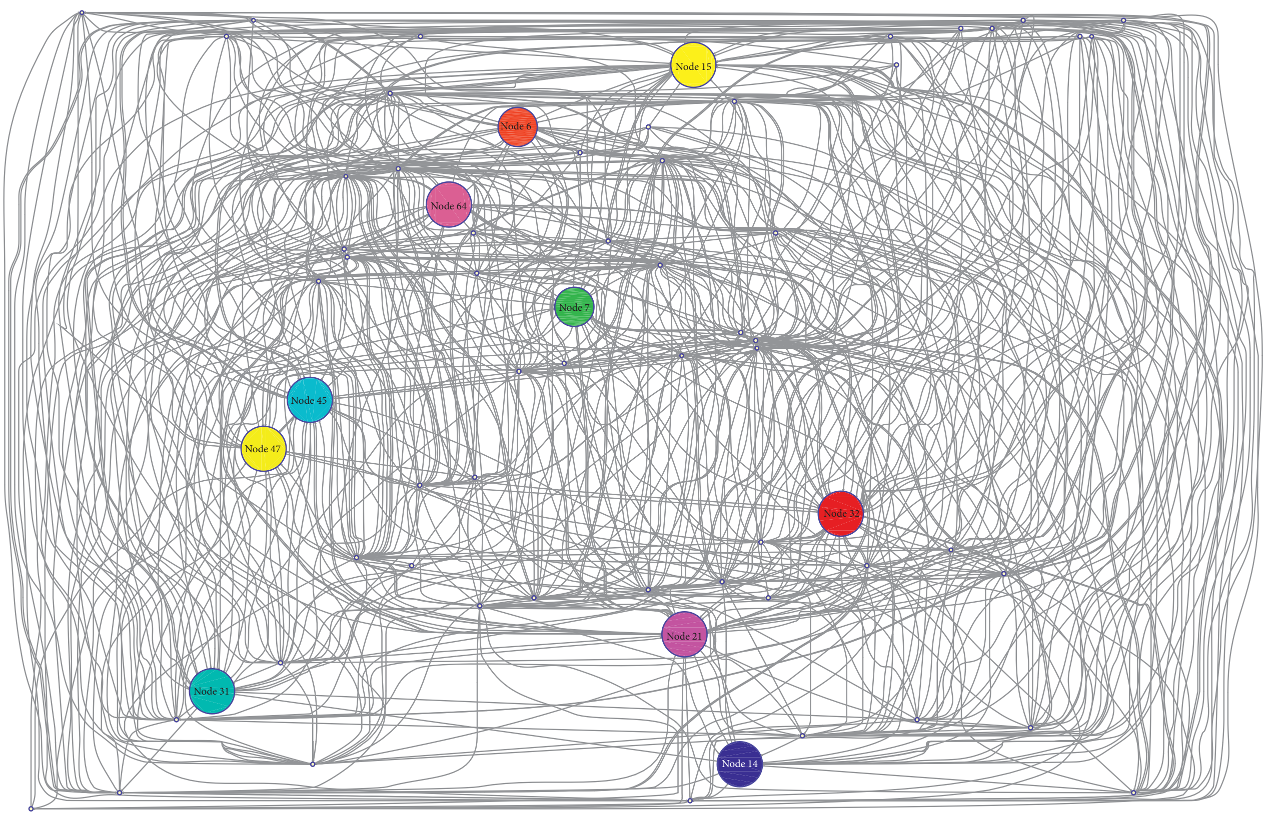


Figure 2: Influential users in the social network.

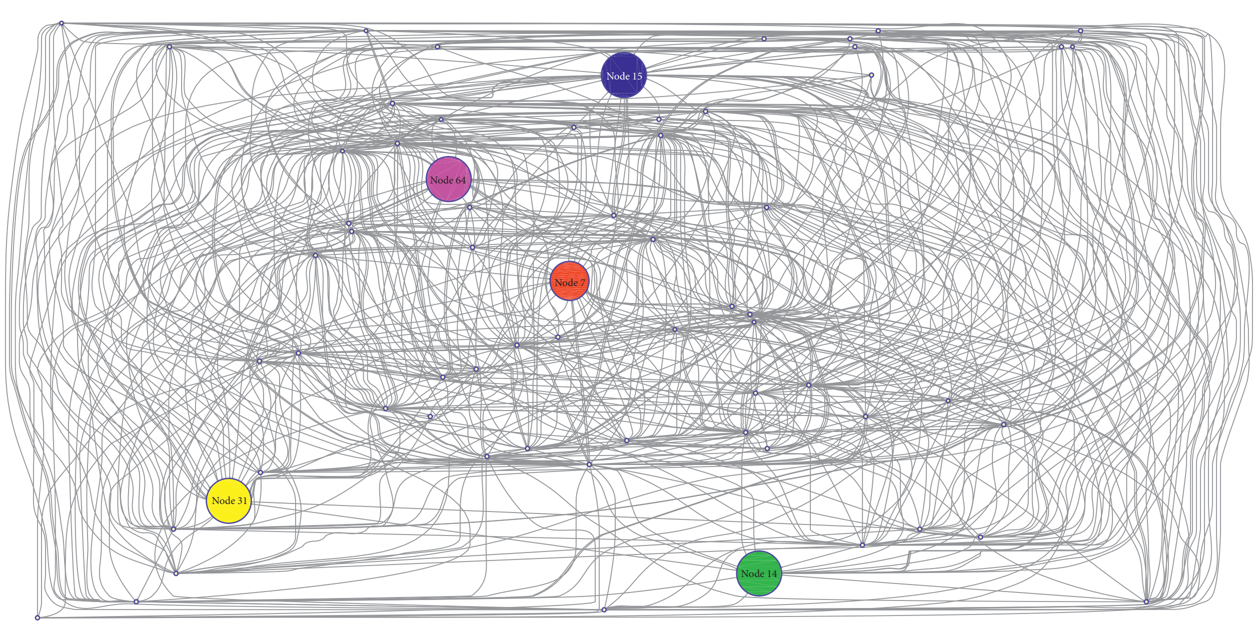


Figure 3: Integrated influential users in the social network.

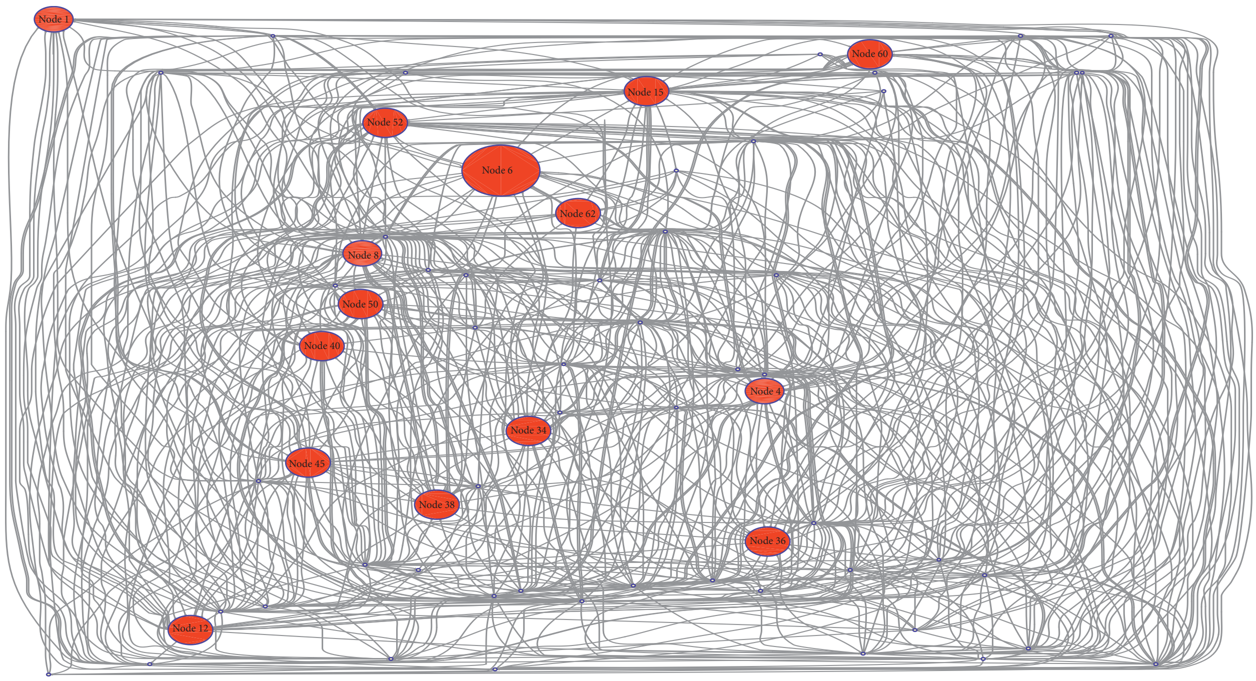


Figure 4: Total employee bees associated with one influential user.

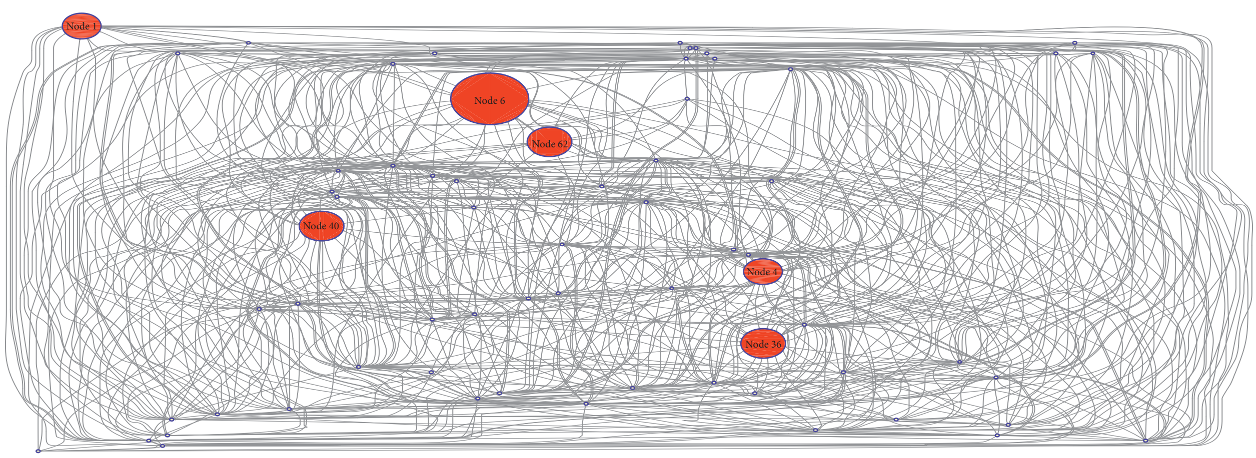


Figure 5: Example of employee bees in ABC.

4.3. Results. The ABC steps are repeated, and the employee, onlooker, and scout bees are updated in each step. Finally, after 20 generations of repetition, the community of each influential user is detected. Figure 8 shows the final clustering of influential user communities.

As shown in Figure 8, users are extracted from communities centered on influential users. According to the figure, it can be seen that due to the more number of users around some influential users, two influential users without any similar user alone have formed a community. Of course, if a more complete dataset is available, these users may also have their communities. ABC converges and stops globally after 20 generations of repetition. Figure 9 shows the convergence of ABC to the optimal point.

As shown in Figure 9, ABC has been able to find the target users in each community and explore the communities of each target user due to the users' adaptation to the problem parameters. This algorithm directs the repetition of the defined goal values in each step to the optimal one and minimizes the community discovery error.

4.4. Performance Evaluation. The quality of community discovery performance in social networks can be measured by many criteria, and the type of criteria used depends on the method used. In our proposed method, the two criteria of Normalized Mutual Information (NMI) and the accuracy of community discovery performance based on the accuracy of user clustering were adopted. The normalized common information criterion is used to compare the modular structure of communities discovered in the social network, which is inspired by information theory [47]. This criterion

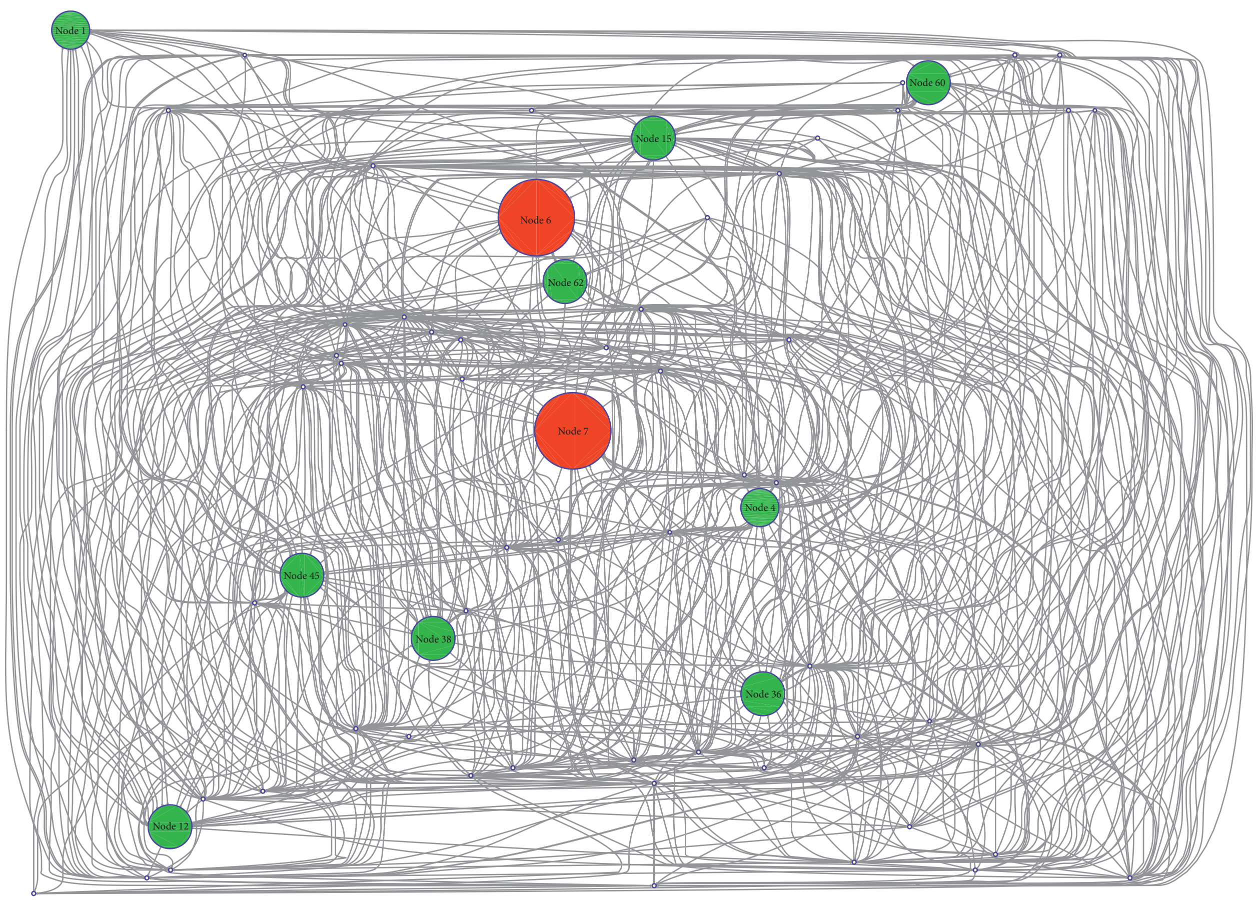


Figure 6: Example of onlooker bees in the ABC.

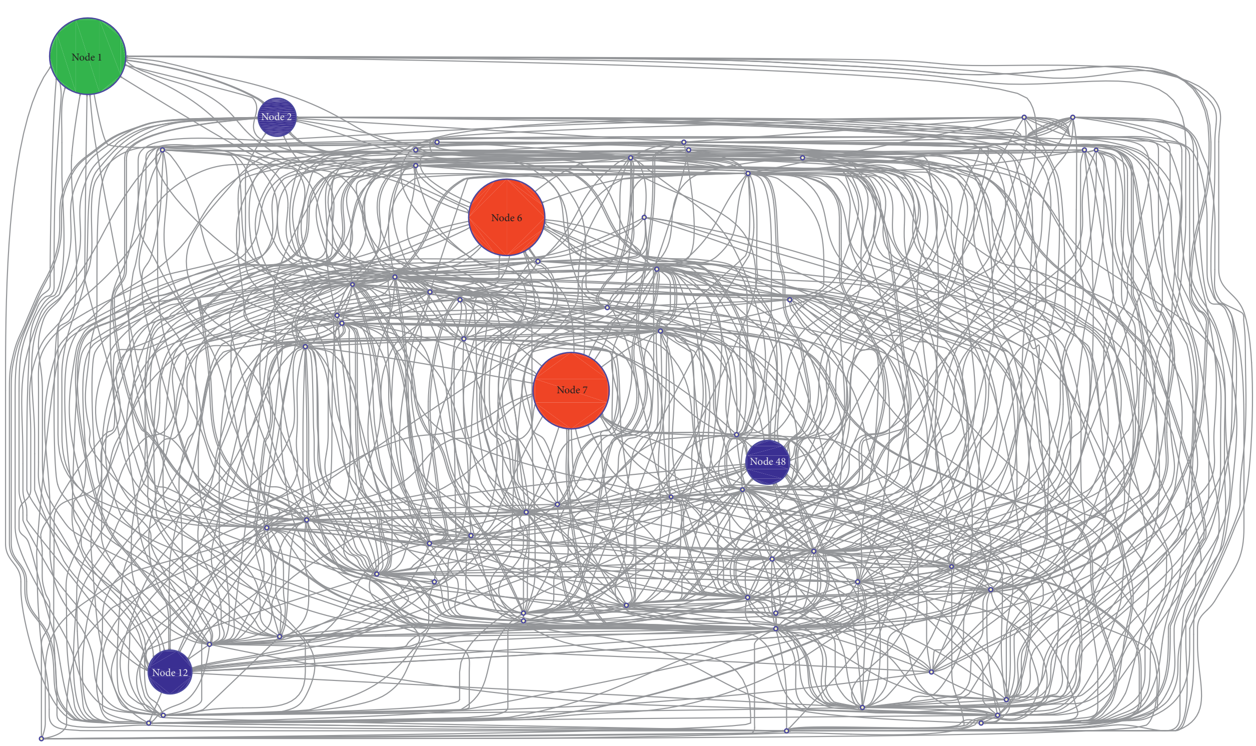


Figure 7: Example of scout bees in the ABC.

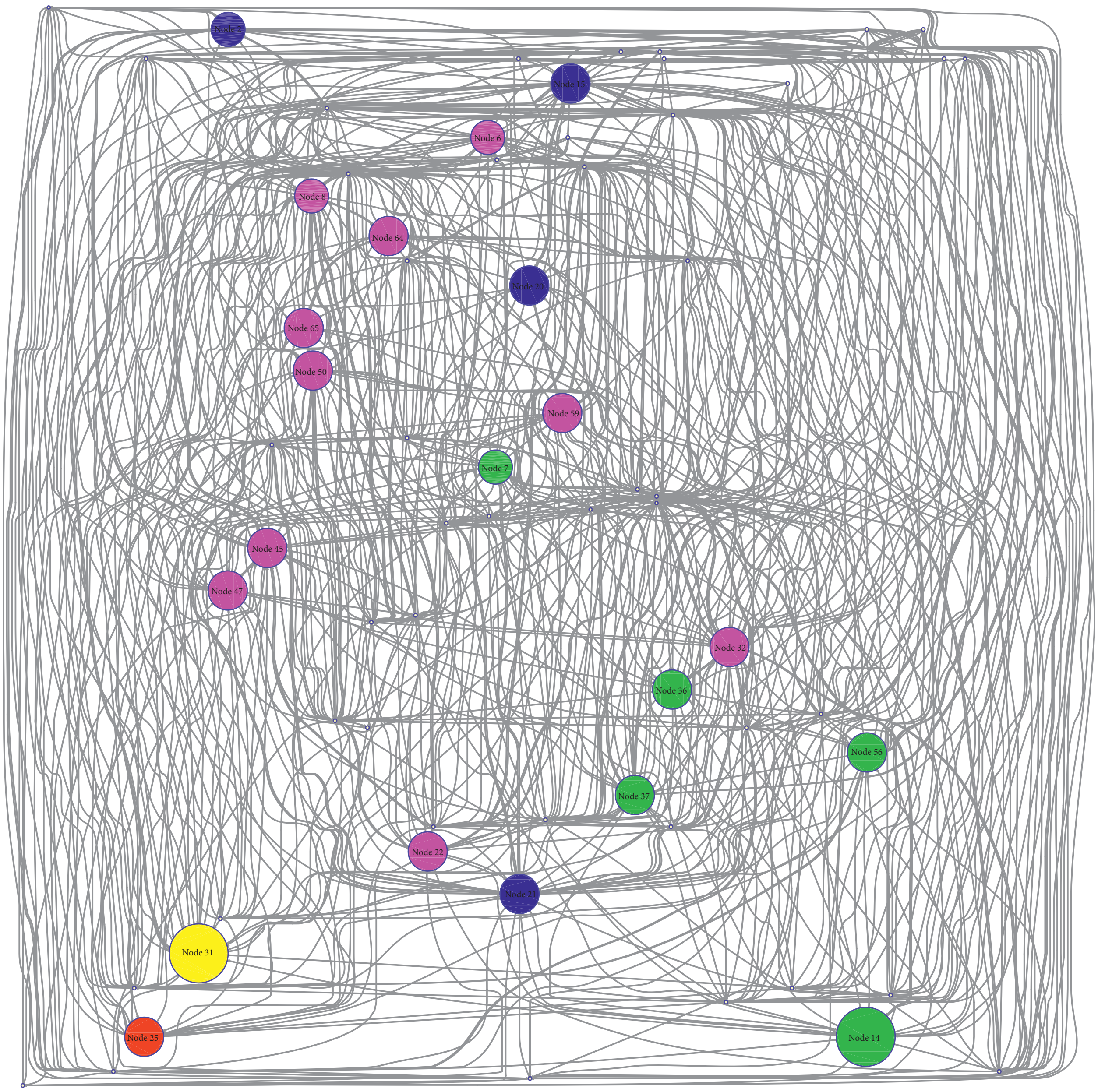


FIGURE 8: Community detection according to ABC.

measures the common users between the discovered communities and similar users based on the features mentioned in the dataset. The NMI criterion is calculated in equation (5):

$$\mathrm{NMI}(S, C) = \frac{2 * I(S, C)}{|H(S) + H(C)|}, \tag{9}$$

where S is the set of similar users, C is the set of users in the communities, $I(S, C)$ is the set of mutual users between these two sets, $H(S)$ is the number of similar users, and $H(C)$ is the number of users discovered in the communities. Figure 10 shows the NMI criteria for each of the communities discovered in the proposed method.

As shown in Figure 10, the NMI criterion in the proposed method is high for the discovered communities. Figure 10 shows that most of the users extracted for each community are similar users on the social network.

Another criterion used in this study to evaluate the performance of the proposed method is the criteria related to the confusion matrix for clustering and community discovery in social networks. This matrix, which is one of the most well-known criteria for measuring clustering performance, is obtained by comparing the class of clustered samples with the class of original samples. In this paper, the clustering class of the samples is extracted based on users in the communities, and the main class includes similar users in the social network based on the published features in the dataset. This matrix consists of four elements: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN):

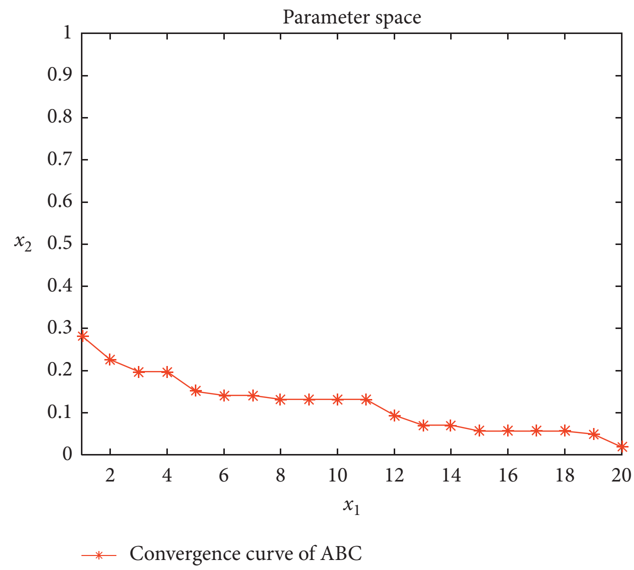


Figure 9: The convergence of ABC to its optimal point.

(i) TP: users who have been selected as members of the community and whose features in the dataset are similar to each other

(ii) TN: users who have been selected as community members but whose features in the dataset are not similar to each other

(iii) FP: users who are not selected as members of the community and whose features in the dataset are not similar to each other

(iv) FN: users who are not selected as members of the community but whose features in the dataset are similar to each other

Criteria for assessing the accuracy of community discovery performance are determined based on similar features in users on social networks. Therefore, the evaluation criteria based on the confusion matrix obtained from the mentioned factors are given in the following equations:

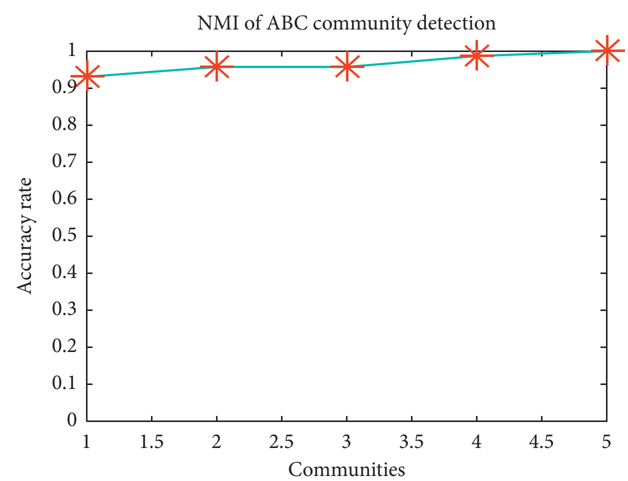


Figure 10: NMI measure in the proposed method.

$$\text{Accuracy} = \frac{\text{Correctly detected users}}{\text{Total users}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (10)$$

$$\text{Precision} = \frac{\text{Correctly detected users in communities}}{\text{Total detected users}} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (11)$$

$$\text{Recall} = \frac{\text{Correctly detected users in communities}}{\text{Correctly detected and incorrectly not detected users}} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (12)$$

In general, the Precision criterion is defined as the ratio of correctly detected users in communities to the total number of users detected in communities. The Recall criterion is defined as the ratio of correctly detected users to the total number of correctly detected users and members of the community who are not detected.

The *F*-measure performance evaluation criterion is defined to combine the two criteria, Precision and Recall, into a single criterion. The resulting value makes comparisons between algorithms, and the entire dataset is very simple and easy. The F-measure criterion is defined as follows:

$$F - \text{measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (13)$$

Algorithm 1 shows the values of the evaluation criteria for the proposed method. These values are obtained from the average values obtained for each community.

According to Table 2, it can be seen that the proposed method has good average values for evaluation criteria, and this shows the good performance of ABC in community detection in social networks. Figure 11 shows a curve of the accuracy criterion for the communities discovered in ABC.

As shown in Figure 11, the accuracy of ABC for community discovery is a high value with an average of 95.33% due to the agreement of ABC parameters with the similarity structure of users in the social network. This high value for the accuracy of the community detection method indicates the high similarity of users in terms of the level of trust and social characteristics with influential users in the network.

As shown in Figure 12, the Recall criterion has a high value in ABC-based community discovery concerning the proper use of it, with an average of 98.62%. Based on the calculated Recall, it can be said that the method could achieve a low intracluster distance and a high intercluster distance between users in the social network.

Figure 13 shows the curve for the Precision criterion for the communities discovered in the proposed method. As shown in this figure, the Precision criterion in ABC for community detection is 87.9% on average. This value for the Precision of the community discovery method indicates a small percentage of users that the proposed method has not been able to discover and cluster them in the form of communities. In other words, the percentage of undetected users who were members of a community is minimum.

Finally, the calculated F-measure of the proposed method based on the number of communities with an average of 92.62% is presented in Figure 14. This criterion balances the sensitivity and specificity of the proposed method for discovering users on a social network. This figure emphasizes the sensitivity and specificity of the proposed method in finding the most similar users in terms of the level of trust and social characteristics with influential users in the social network that are highly reliable.

Table 2: Evaluation criterion values.

Evaluation criteria	Value
NMI	0.9662
Accuracy	0.9533
Recall	0.9862
Precision	0.8790
F-measure	0.9282

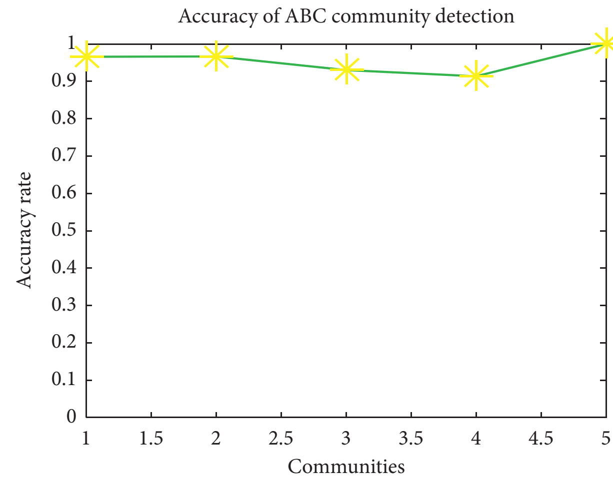


Figure 11: Accuracy of ABC community detection.

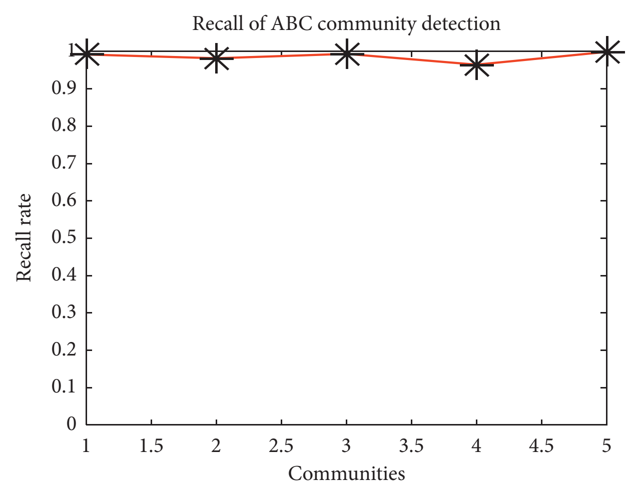


Figure 12: Recall of ABC community detection.

4.5. Comparison with Previous Methods. After implementing and evaluating the proposed method, to validate the proposed method and to show the improvement made in this method, we compared our method with other methods. Due to the importance of community detection in social networks and finding similar users in order to divide them into systematic groups in the form of similar social structures, many researchers in this field have done a lot of research studies. For this purpose, the proposed method is compared with some previous methods [3, 48–55] in the aspect of NMI. This criterion is one of the most important evaluation criteria in the field of community recognition in social networks. Figure 15 shows a comparison of the proposed method with other methods. In order to make the comparison as fair as possible, we use the same experimental setting as given in [46].

As shown in Figure 15, the proposed method, in comparison to other previous methods and even from the ABC-based method without considering the degree of trust and the influential users, has achieved better results in terms of NMI criteria.

In Figure 16, the proposed method is compared with previous methods in the aspect of accuracy. This criterion has shown the quality of the proposed method in detecting community users among all users in the social network

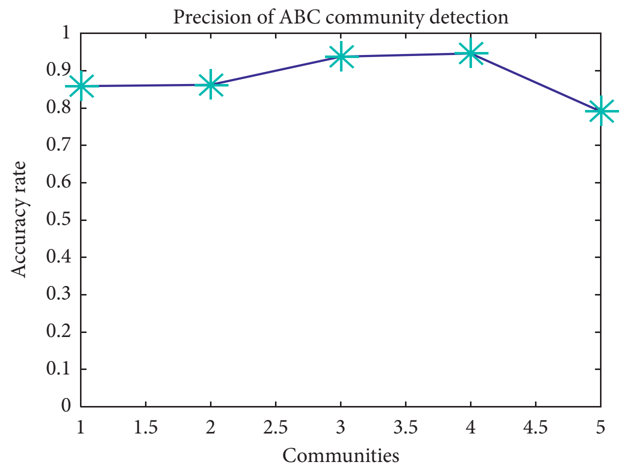


Figure 13: Precision of ABC community detection.

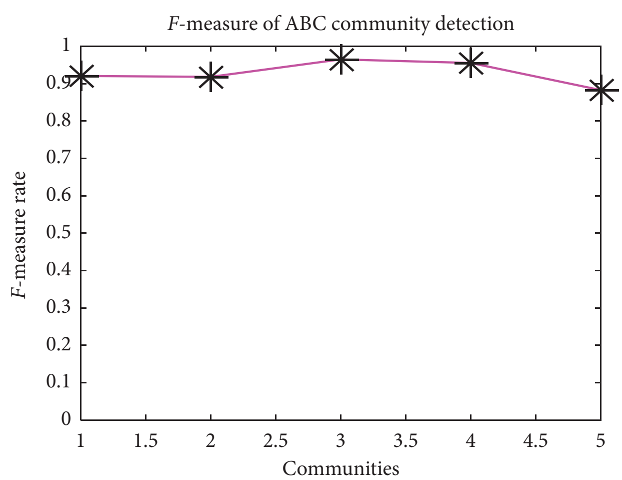


Figure 14: *F*-measure of ABC community detection.

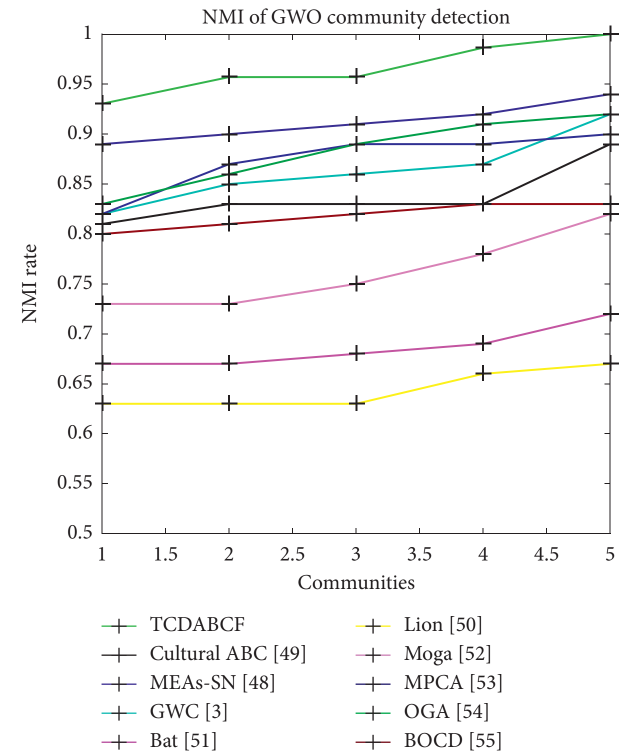


Figure 15: Comparison of NMI with previous methods.

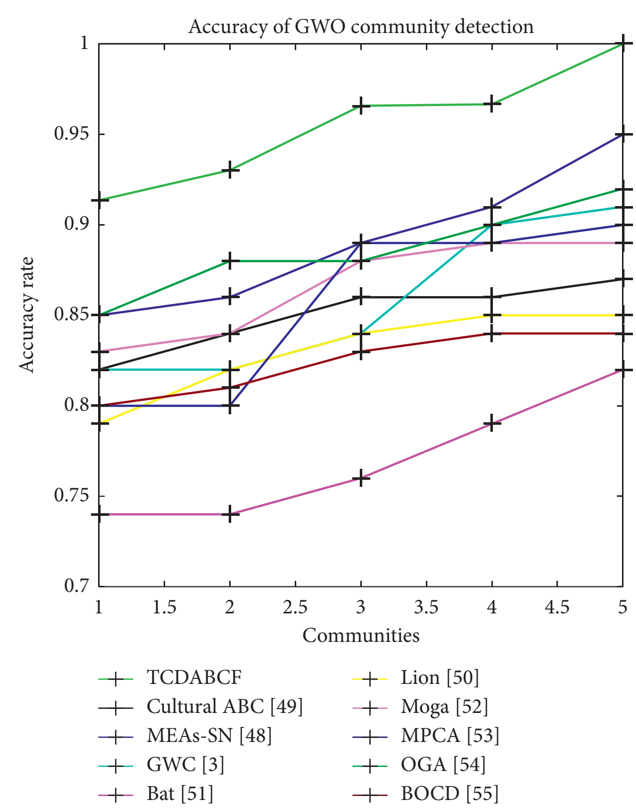


Figure 16: Comparison of accuracy with previous methods.

concerning other methods. It refers to the effect of selecting trusted and influential users as the central points of any community in the proposed method.

The sensitivity of detected users among all users discovered as members of communities is presented in Figure 17. This figure compares the point-to-point Recall value of the proposed method with previous methods. In this figure, the calculated Recall of the proposed method is better than other methods in all the cases.

Figure 18 presents a point-to-point comparison of the proposed method with previous methods in the aspect of Precision that shows the specificity of correctly detected users among all detected users of communities. As shown in Figure 18, the proposed TCDABCF method achieved better results in terms of Precision compared to most of the previous methods especially when the number of communities increases.

Finally, the proposed method is compared with previous methods in the aspect of *F*-measure in Figure 19. As shown in this figure, the point-to-point comparison of the proposed

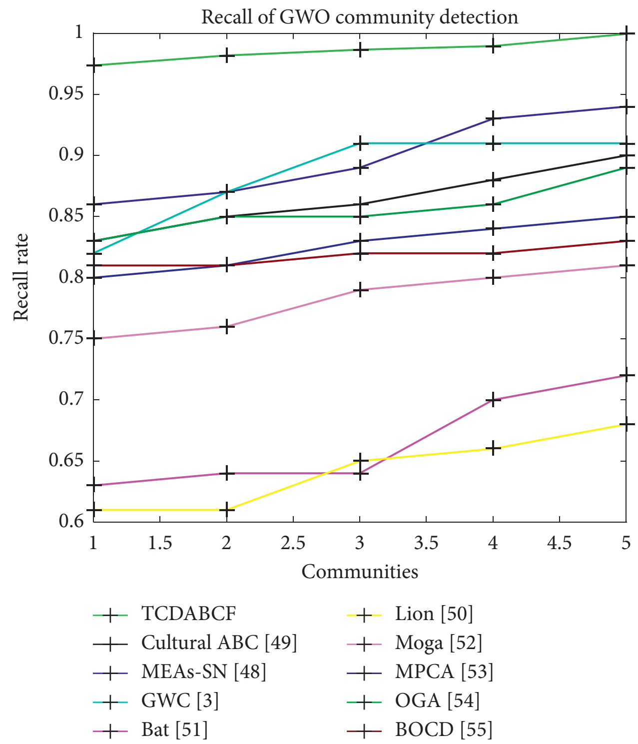


Figure 17: Comparison of Recall with previous methods.

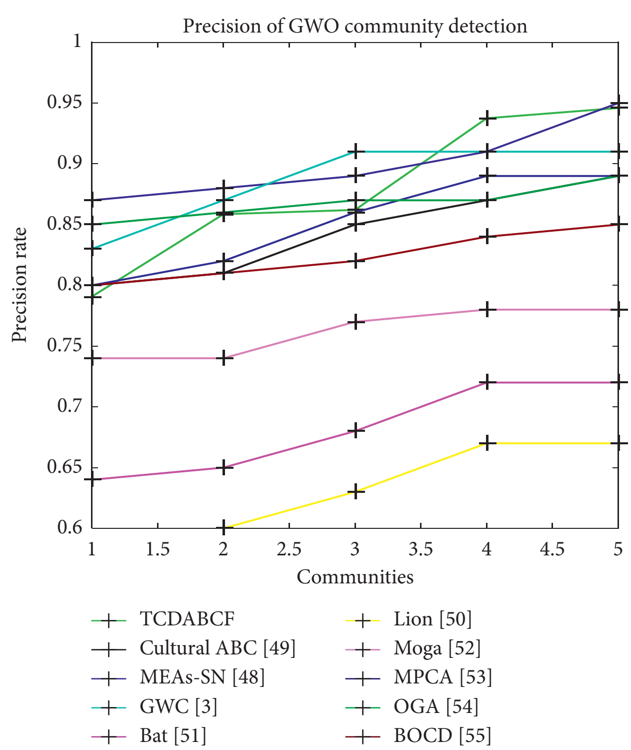


Figure 18: Comparison of Precision with previous methods.

method with previous methods shows that sensitivity and specificity of the proposed method for discovering users on a social network has achieved better results in all the cases.

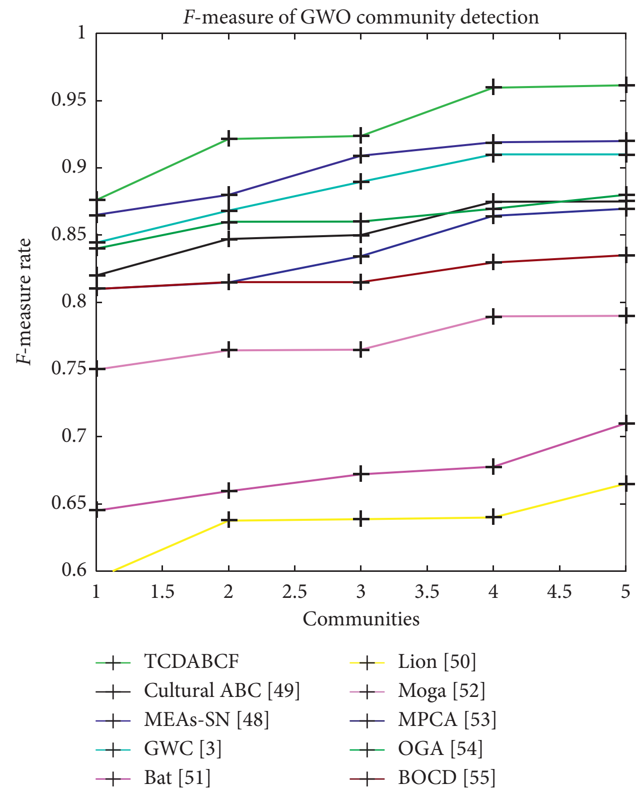


Figure 19: Comparison of *F*-measure with previous methods.

That proves the superiority of the proposed TCDABCF method compared with state-of-the-art methods.

5. Conclusion

Today, social networks play an effective role in all aspects of life, and the use of social networks in all aspects of daily life is clearly palpable. Many complex phenomena can be modeled on social networks and have now become a hot research area. Relationships can be presented as social networks in many areas of real life, such as social media, biology, and scientific communication. Therefore, finding similar users in the form of communities and organizations in social networks can bring very useful applications for all researchers. Discovering the community on social media, not only systematizes the search for users participating in social media but also can bring together similar users with almost identical interests, and the results of which can be used in many areas. On the contrary, due to the dynamic nature of social networks, searching for similar users who reject the same level of trust requires the use of metaheuristic search algorithms. Hence, in order to explore a society in social networks, an approach based on the artificial bee colony algorithm is presented. In the proposed method, an influential user is selected as the target in the network, and other users in the network are known as bees. According to the social characteristics and the level of trust and the similarity of the user to the target, the hierarchy of users is formed, and thus, a community of similar users is identified.

Using trust and similarity of users combined with the ABC algorithm makes the proposed method superior to the previous works in this area. As mentioned in the Experimental Results section, we compare this algorithm with nine similar methods on the Facebook dataset, and results show that the proposed method has obtained values of 0.9662 and 0.9533 for NMI and accuracy, respectively, which has improvements in comparison with state-of-the-art community detection methods.

In order to make future suggestions for the proposed method, a multiobjective artificial bee colony algorithm can be proposed by aggregating the objectives of increasing trust, increasing similarity, reducing intracluster distances, and increasing intercluster distances in order to perform community detection in social networks.

Data Availability

In this paper, the Facebook dataset stored in the Stanford University Governor's Data Repository was used Kilroy, Ref: D.S., and Chelsea Hejny, Facebook. Internet resource, 2017.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

ZhihaoPeng developed the theory and performed the computations, developed theoretical formalism, performed the experiments using the Facebook database, and analyzed the results with the support from Poria Pirozmand. Mohsen Rastgari implemented the method using the Matlab compiler with the support from Zhihao Peng and wrote the manuscript with the support from Yahya Dorostkar Navaei. Yahya Dorostkar Navaei helped Mohsen Rastgari to write the manuscript and conceived and planned the experiments. Raziyeh Daraei took the lead in revising the manuscript, inserted more related works, while revising the manuscript, and proofread the final manuscript before passing it to a native proofreader. Rozita Jamili Oskouei supervised Mohsen Rastgari in implementing codes, helped Mohsen Rastgari in writing the manuscript, submitted the manuscript to the journal, and was the Iranian team leader. Poria Pirozmand contributed to sample preparation, helped Zhihao Peng in developing theoretical formalism, processed the experimental data, and was the Chinese team leader. Seyed Saeid Mirkamali supervised the project and Sestablished the connection between Chinese and Iranian teams.

References

[1] I. Messaoudi and N. Kamel, "A multi-objective bat algorithm for community detection on dynamic social networks," *Applied Intelligence*, vol. 49, no. 6, pp. 2119–2136, 2019.

[2] S. Tavakoli, A. Hajibagheri, and G. Sukthankar, "Learning social graph topologies using generative adversarial neural networks," in *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling & Prediction*, São Paulo, Brazil, May 2017.

[3] S. Chand and S. Mehta, "Community detection using nature inspired algorithm," in *Hybrid Intelligence for Social Networks*, pp. 47–76, Springer, Berlin, Germany, 2017.

[4] A. Said, R. A. Abbasi, O. Maqbool, A. Daud, and N. R. Aljohani, "CC-GA: a clustering coefficient based genetic algorithm for detecting communities in social networks," *Applied Soft Computing*, vol. 63, pp. 59–70, 2018.

[5] F. Zou, D. Chen, D.-S. Huang, R. Lu, and X. Wang, "Inverse modelling-based multi-objective evolutionary algorithm with decomposition for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 513, pp. 662–674, 2019.

[6] L. Pournajaf, F. Tahmasebian, L. Xiong, V. Sunderam, and C. Shahabi, "Privacy preserving reverse k-nearest neighbor queries," in *Proceedings of the 2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pp. 177–186, IEEE Computer Society, Aalborg, Denmark, June 2018.

[7] J. Khodadoust, A. M. Khodadoust, S. S. Mirkamali, and S. Ayat, "Fingerprint indexing for wrinkled fingertips immersed in liquids," *Expert Systems with Applications*, vol. 146, Article ID 113153, 2020.

[8] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," 2020, https://arxiv.org/abs/2010.03978.

[9] X. Li, Q. Tian, M. Tang, X. Chen, and X. Yang, "Local community detection for multi-layer mobile network based on the trust relation," *Wireless Networks*, vol. 26, no. 3, pp. 1–13, 2019.

[10] D. Naik, N. B. Gorojanam, and D. Ramesh, "Community based emotional behaviour using Ekman's emotional scale," in *Proceedings of the International Conference on Innovations for Community Services*, pp. 63–82, Bhubaneswar, India, January 2020.

[11] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific Reports*, vol. 2, p. 336, 2012.

[12] S. Tavakoli and S. Yooseph, "Learning a mixture of microbial networks using minorization-maximization," *Bioinformatics*, vol. 35, no. 14, pp. 23–30, 2019.

[13] F. Sabatini and F. Sarracino, "Online social networks and trust," *Social Indicators Research*, vol. 142, no. 1, pp. 229–260, 2019.

[14] E. J. Neystadt, R. Karidi, Y. T. Weisfeild, R. Varshavsky, A. Oron, and K. Radinsky, "Social network based contextual ranking," Google Patents, 2018.

[15] A. A. R. Hosseinabadi, N. S. H. Rostami, M. Kardgar, S. Mirkamali, and A. Abraham, "A new efficient approach for solving the capacitated vehicle routing problem using the gravitational emulation local search algorithm," *Applied Mathematical Modelling*, vol. 49, pp. 663–679, 2017.

[16] A. A. R. Hosseinabadi, J. Vahidi, V. E. Balas, and S. S. Mirkamali, "OVRP_GELS: solving open vehicle routing problem using the gravitational emulation local search algorithm," *Neural Computing and Applications*, vol. 29, no. 10, pp. 955–968, 2018.

[17] S. Mirkamali and P. Nagabhushan, "Depth-wise image inpainting," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 141–144, Tsukuba, Japan, November 2012.

[18] D. Zhao, L. Liu, F. Yu et al., "Ant colony optimization with horizontal and vertical crossover search: fundamental visions for multi-threshold image segmentation," *Expert Systems with Applications*, vol. 167, pp. 114–122, 2020.

[19] S. S. Mirkamali and P. Nagabhushan, “Object removal by depth-wise image inpainting,” *Signal, Image and Video Processing*, vol. 9, no. 8, pp. 1785–1794, 2015.

[20] T. Y. Tan, L. Zhang, C. P. Lim, B. Fielding, Y. Yu, and E. Anderson, “Evolving ensemble models for image segmentation using enhanced particle swarm optimization,” *IEEE Access*, vol. 7, pp. 34004–34019, 2019.

[21] S. S. Mirkamali and P. Nagabhushan, “RGBD image segmentation,” in *Proceedings of the 2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 41–44, Tehran, Iran, May 2015.

[22] W. Gtifa, F. Hamdaoui, and A. Sakly, “3D brain tumor segmentation in MRI images based on a modified PSO technique,” *International Journal of Imaging Systems and Technology*, vol. 29, no. 4, pp. 501–509, 2019.

[23] J. Cai, H. Wei, H. Yang, and X. Zhao, “A novel clustering algorithm based on DPC and PSO,” *IEEE Access*, vol. 8, pp. 88200–88214, 2020.

[24] B. Saoud, “Networks clustering with bee colony,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1297–1309, 2019.

[25] A. Ramtin, V. Hakami, and M. Dehghan, “A self-stabilizing clustering algorithm with fault-containment feature for wireless sensor networks,” in *Proceedings of the 7th International Symposium on Telecommunications (IST’2014)*, pp. 735–739, Tehran, Iran, September 2014.

[26] Z. Gu, Y. Zhu, Y. Wang, X. Du, M. Guizani, and Z. Tian, “Applying artificial bee colony algorithm to the multidepot vehicle routing problem,” *Software: Practice and Experience*, 2020.

[27] Y. Marinakis, “An artificial bee colony algorithm for the multiobjective energy reduction multi-depot vehicle routing problem,” in *Proceedings of the Learning and Intelligent Optimization: 13th International Conference*, vol. 11968, p. 208, Chania, Greece, May 2019.

[28] A. I. Hafez, H. M. Zawbaa, A. E. Hassanien, and A. A. Fahmy, “Networks community detection using artificial bee colony swarm optimization,” in *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA*, pp. 229–239, Springer, Cham, Switzerland, 2014.

[29] T. T. Aung, T. T. S. Nyunt, and P. P. W. Cho, “Community detection in social graph using nature-inspired based artificial bee colony algorithm with crossover and mutation,” in *Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pp. 213–217, Singapore, 2019.

[30] T. T. Aung and T. T. S. Nyunt, “Community detection in social network using artificial bee colony with genetic operator,” in *Proceedings of the 2nd International Conference on Advanced Information Technologies (ICAIT)*, pp. 45–50, Yangon, Myanmar, November 2018.

[31] T. T. Aung and T. T. S. Nyunt, “Modularity based ABC algorithm for detecting communities in complex networks,” *International Journal of Machine Learning and Computing*, vol. 10, no. 2, 2020.

[32] S. Ding, Z. Yue, S. Yang, F. Niu, and Y. Zhang, “A novel trust model based overlapping community detection algorithm for social networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2101–2114, 2020.

[33] G. Lingam, S. Rashmi Ranjan Rout, and S. Das, “Social botnet community detection: a novel approach based on behavioral similarity in twitter network using deep learning,” in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pp. 708–718, Taipei, Taiwan, October 2020.

[34] J. Sheng, Q. Li, B. Wang et al., “Node trust: an effective method to detect non-overlapping community in social networks,” *Modern Physics Letters B*, vol. 351 page, 2021.

[35] J. Zhu, C. Wang, Xu Guo, M. Qian, J. Li, and Y. Liu, “Friend and POI recommendation based on social trust cluster in location-based social networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 89, pp. 1–12, 2019.

[36] X. Chen, C. Xia, and J. Wang, “Chaos solitons & fractals,” in *A Novel Trust-Based Community Detection Algorithm Used in Social Networks*, pp. 57–65, Elsevier, Amsterdam, Netherlands, 2018.

[37] B. Wu, X. Wen, and J. Ge, “Trust analysis in social network community research,” *Applied Mechanics and Materials*, vol. 55, pp. 1578–1583, 2020.

[38] G. Beigi, M. Jalili, H. Alvari, and G. Sukthankar, “Leveraging community detection for accurate trust prediction,” in *Proceedings of the 2014 ASE Bigdata/Socialcom/Cybersecurity Conference*, Stanford University, Stanford, CA, USA, May 2014.

[39] H. Kou, F. Wang, C. Lv et al., “Trust-based missing link prediction in signed social networks with privacy preservation,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8849536, 10 pages, 2020.

[40] S. Baek and S. Kim, “Trust-based access control model from sociological approach in dynamic online social network environment,” *Scientific World Journal*, vol. 2014, Article ID 936319, 8 pages, 2014.

[41] M. Li, X. Yang, B. Zhang, and Z. Huang, “A sentiment delivering estimate scheme based on trust chain in mobile social network,” *Mobile Information Systems*, vol. 2015, Article ID 745095, 20 pages, 2015.

[42] J. Ma, J. Liu, M. Wenping, M. Gong, and L. Jiao, “Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic social networks,” *Scientific World Journal*, vol. 2014, Article ID 402345, 22 pages, 2014.

[43] D. Prangle, “Adapting the ABC distance function,” *Bayesian Analysis*, vol. 12, no. 1, pp. 289–309, 2017.

[44] M. R. C. Qazani, H. Asadi, and S. Nahavandi, “High-fidelity hexarot simulation-based motion platform using fuzzy incremental controller and model predictive control-based motion cueing algorithm,” *IEEE Systems Journal*, vol. 14, no. 4, pp. 5073–5083, 2019.

[45] S. S. Jadon, J. C. Bansal, R. Tiwari, and H. Sharma, “Artificial bee colony algorithm with global and local neighborhoods,” *International Journal of System Assurance Engineering and Management*, vol. 9, no. 3, pp. 589–601, 2018.

[46] X. You, Y. Ma, and Z. Liu, “A three-stage algorithm on community detection in social networks,” *Knowledge-Based Systems*, vol. 187, Article ID 104822, 2020.

[47] E. Osaba, Javier Del Ser, D. Camacho, and X.-S. Yang, “Community detection in networks using bio-inspired optimization: latest developments, new results and perspectives with a selection of recent meta-heuristics,” *Applied Soft Computing Journal*, vol. 87, pp. 1–58, 2019.

[48] C. Liu, J. Liu, and Z. Jiang, “A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks,” *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2274–2287, 2014.

[49] H. Baofang, “A cultural algorithm based on artificial bee colony optimization for community detection in signed social networks,” in *Proceedings of the 2015 10th International Conference on Broadband and Wireless Computing,*

Communication and Applications (BWCCA), pp. 404–409, Krakow, Poland, 2015.

[50] R. Babers, A. E. Hassanien, and N. I. Ghali, "A nature-inspired metaheuristic Lion Optimization Algorithm for community detection," in *Proceedings of the 2015 11th International Computer Engineering Conference (ICENCO)*, pp. 217–222, Cairo, Egypt, December 2015.

[51] W. Chunyu and P. Yun, "Discrete bat algorithm and application in community detection," *The Open Cybernetics & Systemics Journal*, vol. 9, no. 1, 2015.

[52] C. Pizzuti, "A multi-objective genetic algorithm for community detection in networks," in *Proceedings of the 2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pp. 379–386, Newark, NJ, USA, November 2009.

[53] P. M. Zadeh and Z. Kobti, "A multi-population cultural algorithm for community detection in social networks," *Procedia Computer Science*, vol. 52, pp. 342–349, 2015.

[54] B. Dickinson, B. Valyou, and W. Hu, "A genetic algorithm for identifying overlapping communities in social networks using an optimized search space," *Social Networking*, vol. 2, no. 4, pp. 193–201.

[55] R. Agrawal, "Bi-objective community detection (bocd) in networks using genetic algorithm," in *Proceedings of the International Conference on Contemporary Computing*, pp. 5–15, Springer, Noida, India, August 2011.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6653056, 21 pages
https://doi.org/10.1155/2021/6653056

Research Article

A Hybrid Method Integrating a Discrete Differential Evolution Algorithm with Tabu Search Algorithm for the Quadratic Assignment Problem: A New Approach for Locating Hospital Departments

Asaad Shakir Hameed,[1,2] **Modhi Lafta Mutar**,[1,2] **Haiffa Muhsan B. Alrikabi**,[3] **Zakir Hussain Ahmed**,[4] **Abeer A. Abdul–Razaq**,[5] **and Huda Karem Nasser**[2]

[1]*Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka Hang Tuah Jaya, Durian Tunggal, Melaka, Malaysia*
[2]*Department of Mathematics, General Directorate of Thi-Qar Education, Ministry of Education, Thi-Qar, Iraq*
[3]*Department of Mathematics, College of Education for Pure Sciences, University of Thi-Qar, Nasiriyah, Iraq*
[4]*Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia*
[5]*Department of Mathematics, College of Computer Science and Mathematics, Thi-Qar University, Nasiriyah, Iraq*

Correspondence should be addressed to Asaad Shakir Hameed; asaadutem@yahoo.com and Zakir Hussain Ahmed; zaahmed@imamu.edu.sa

Received 12 December 2020; Revised 25 January 2021; Accepted 30 January 2021; Published 17 March 2021

Academic Editor: Ali Asghar Rahmani Hosseinabadi

The facility layout problem (FLP) is a very important class of NP-hard problems in operations research that deals with the optimal assignment of facilities to minimize transportation costs. The quadratic assignment problem (QAP) can model the FLP effectively. One of the FLPs is the hospital facility layout problem that aims to place comprehensive clinics, laboratories, and radiology units within predefined boundaries in a way that minimizes the cost of movement of patients and healthcare personnel. We are going to develop a hybrid method based on discrete differential evolution (DDE) algorithm for solving the QAP. In the existing DDE algorithms, certain issues such as premature convergence, stagnation, and exploitation mechanism have not been properly addressed. In this study, we first aim to discover the issues that make the current problem worse and to identify the best solution to the problem, and then we propose to develop a hybrid algorithm (HDDETS) by combining the DDE and tabu search (TS) algorithms to enhance the exploitation mechanism in the DDE algorithm. Then, the performance of the proposed HDDETS algorithm is evaluated by implementing on the benchmark instances from the QAPLIB website and by comparing with DDE and TS algorithms on the benchmark instances. It is found that the HDDETS algorithm has better performance than both the DDE and TS algorithms where the HDDETS has obtained 42 optimal and best-known solutions from 56 instances, while the DDE and TS algorithms have obtained 15 and 18 optimal and best-known solutions out of 56 instances, respectively. Finally, we propose to apply the proposed algorithm to find the optimal distributions of the advisory clinics inside the Azadi Hospital in Iraq that minimizes the total travel distance for patients when they move among these clinics. Our application shows that the proposed algorithm could find the best distribution of the hospital's rooms, which are modeled as a QAP, with reduced total distance traveled by the patients.

1. Introduction

The demand for public facilities has been on increase due to the ever-growing population. These facilities include hospitals, schools, universities, train stations, and airports. As such, it has become pertinent to urgently find the best way to design and manage these facilities, as a lack of design leads to financial losses alongside unnecessary waste of time. The facility layout problem (FLP) is an important class of operations research problems that have been studied for several decades [1]. Most facility layout variants are NP-hard; therefore, global optimal solutions are difficult or impossible to compute in a reasonable time [2]. The FLPs include ensuring that the departments are not given similar assignments and that customization and material handling costs are reduced to a minimum. Over the years, few studies have focused on hospital layout problems, except for those involving different facilities.

One of the NP-hard problems is the hospital facility layout problem, which aims to place comprehensive clinics, laboratories, and radiology units within predefined boundaries in a way that minimizes the cost of movement of patients and healthcare personnel [3]. Several models have been proposed to design and facilitate the best layout for the distribution of the facilities. Recent studies proved that the use of the quadratic assignment problem (QAP) to model the FLP leads to identifying the best facilities for the sites in order to minimize the cost. The QAP has been successfully applied in fields such as economics, engineering, and computing because it uses the best distribution of several facilities to same number of locations to minimize computational cost.

The QAP, which is a combinatorial optimization problem (COP), was first introduced in [4] as a mathematical model related to economic activities which can be defined as follows: there are n facilities and n locations. Let f_{ij} be an information flow between facilities i and j (for example, the number of materials transported between the facility pairs) and d_{kl} be the distance between locations k and l. Also, let $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$ be an assignment (or permutation or one-to-one mapping between all facilities to locations), where $\pi(i)$ represents the location of the facility i. The aim of the problem is to fix the facilities to different locations such that the sum of the distances multiplied by the corresponding flows is minimized. That is, it aims to minimize the total assignment cost (Z_π), where

$$Z_\pi = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)}. \quad (1)$$

In this problem, n facilities and n locations define the problem size; the distance matrix (D) consists of the distances between every location pairs; the flow matrix (F) is the amount of traffic between every facility pairs; and a solution (π) is an assignment of facilities to locations and an objective function defined by total cost of a solution (Z_π). Based on the above, we can summarize the QAP model steps in Figure 1.

Different viewpoints have attracted researchers' attention to study the QAP. From the theoretical viewpoint, the QAP is an NP-hard problem that cannot be solved in polynomial time when the problem size increases [5, 6]. From the practical viewpoint, many real-life applications can be modeled by the QAP, such as backboard wiring problem in the design of computer and other electronic equipment, hospital, and campus layouts [7]. From the application domain viewpoint, the cost of building hospitals and universities is very high, so renovating them is a better option because it is cheaper [8]. In addition, poorly designed layouts lead to financial losses and unnecessary waste of time [9]. From the technique viewpoint, open and active research is recommended for the enhancement of the QAP optimization techniques [10, 11].

Two classes of approaches have been presented to solve the QAP: the exact approaches and the heuristic approaches. The approaches that deliver the exact solution to the problem are called exact approaches. Many exact approaches have been suggested in many studies, but they cannot find the optimal solution for large sized problem instances within a reasonable computational time [12, 13]. Therefore, an effective heuristic approach that can find the best solutions of the QAP within an acceptable computational time for large sized problem instances is needed. Unlike the exact approaches, heuristic approaches seek good solutions (i.e., near-optimal solutions) within a reasonable computational time without giving guarantee to the optimality of the solutions. Metaheuristics are the most recent heuristic approaches which are applied to various combinatorial optimization problems (COPs) as well as the QAP [14, 15].

Some well-known metaheuristic approaches applied to the QAP are discrete bat algorithm (DBA) [16], genetic algorithm (GA) [17], simulated annealing (SA) [18], tabu search [19], ant colony optimization [20], memetic algorithm (MA) [21], migrating birds optimization (MBO) algorithm [22], differential evolution [23], etc. Out of the metaheuristic approaches, differential evolution (DE) algorithms have been proven to be very efficient in finding solutions that are optimal or nearly optimal within a reasonable computational time for large sized problem instances. However, they are characterized by the slow exploitation of solutions [24]. To solve COPs, a few studies, such as [23, 25], have suggested discrete differential evolution (DDE) algorithms.

In this study, we first aim to discover the issues that make the current problem worse and to identify the best solution to the problem, and then we propose to develop a hybrid algorithm (HDDETS) by combining the DDE and tabu search (TS) algorithms to enhance the exploitation mechanism in the DDE algorithm. Then, the performance of the proposed HDDETS algorithm is evaluated by implementing on the benchmark instances from QAPLIB website and by comparing with DDE and TS algorithms on the benchmark instances. Our experimental investigation shows that the proposed HDDETS algorithm is the best algorithm that could find very good quality solutions to the benchmark instances. Finally, we propose to apply the proposed algorithm to find the optimal distributions of the advisory clinics inside the Azadi Hospital in Iraq that minimizes the total travel distance for patients when they move among these

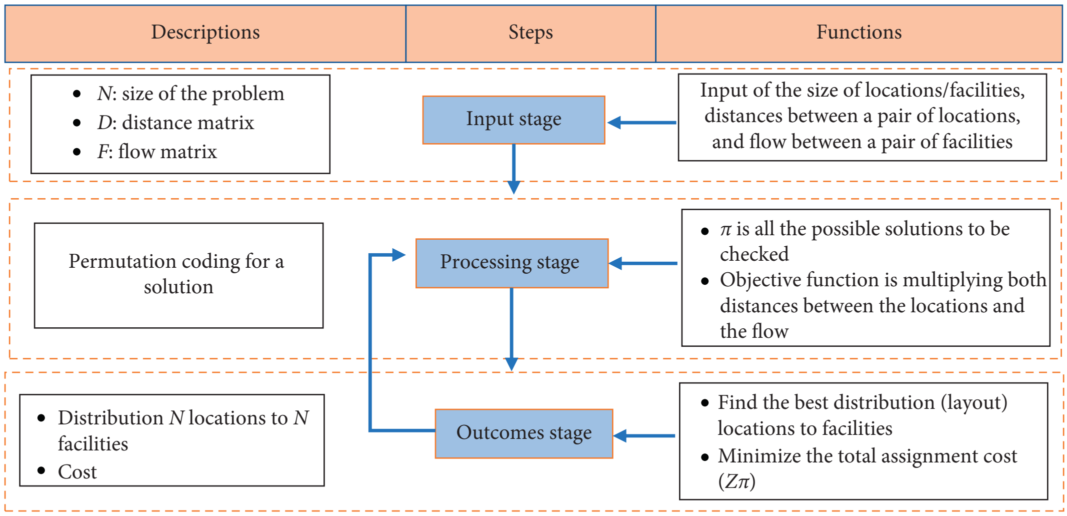


Figure 1: QAP model steps.

clinics. Our application shows that the proposed algorithm could find the best distribution of the hospital's rooms, which are modeled as a QAP, with reduced total distance traveled by the patients when they move among these clinics.

The rest of the paper is organized as follows: Section 2 introduces the research problems; Section 3 includes related works; Section 4 lays out the research methodology; computational results are introduced in Section 5; Section 6 discusses theoretical analysis and limitation of the proposed algorithm; and Section 7 presents the conclusions.

2. Research Problems

Recently, metaheuristic algorithms have been proven successful in several fields such as technology, engineering, and science. However, most of the existing research could not achieve all the aims and objectives due to certain gaps that needed to be filled, especially for large-scale complex optimization problems. Another open area of research in these algorithms is the balance between exploration and exploitation mechanisms, premature convergence and stagnation issues, etc. [11].

The evolutionary algorithms (EAs) are one of the most important types of metaheuristics algorithms, which are very useful in solving complex optimization problems [26, 27]. It has a global exploration search, but sufferings from slow exploitation search [28]. Premature convergence and stagnation issues that occur in the iterative operation are current problems faced by all EAs [29]. In the same context, the DE algorithm is a robust EA. It has a good convergence feature, which makes it necessary to solve global optimization problems, in addition to its high efficiency that has proven to be successful in various real-world applications [29, 30]. The DE algorithm was used in [23, 25] to solve problems related to discrete values such as the QAP. DDE is a modified version of the DE algorithm. Finding solutions of the QAP is very difficult within a polynomial time as the size of the problem increases. Besides, there is no guarantee that a near-optimal solution would be found within polynomial time. This can be regarded as the problem associated with allocating a set of facilities to a set of locations and considering the distance between any two locations as well as the flows between any two facilities to minimize the cost [16].

The QAP is as an NP-hard problem [5]. An NP-hard problem is very complex, and it requires a method that can be used to simplify it. Regarding the applications of the QAP in the real world, the increasing interest of governments in the service sectors (such as hospitals) and educational institutions (such as universities) has led to an increase in the number of these buildings. These buildings are considered suitable for the application of the QAP model, which in turn has led to an increase in the number of these buildings. In the same context, the cost of building hospitals and universities is quite high, so renovating them is often a better option as it is less expensive [8].

In conclusion, there are two main problems worth to be highlighted in this proposed study.

(i) In the view of the fundamental technique, the EAs have a global exploration advantage; however, these kinds of algorithms have various well-known limitations. Suffering from the mechanism of exploitation is among these limitations [30]. Based on a critical reading of the literature, we found that this issue has not been studied in the DDE algorithm. As such, for a DDE algorithm to achieve good performance in terms of solving complex optimization problems, the QAP is considered. The use of local research can enhance the exploitation mechanism by increasing the speed of convergence to optimal solutions [29, 31]. Therefore, there is a need to integrate the DDE algorithm with another algorithm that uses the local search feature.

(ii) In the view of application domain of the QAP, the lack of layout of the clinics' locations inside the hospital causes considerable distance for patients [32]. Therefore, it is important for clinics within a hospital to be meticulously organized because disorganization can result in too much movement for patients, thereby leading to waste of time.

3. Related Works

We have mentioned that there are two classes of approaches presented to solve the QAP. Since we are going to use metaheuristic, we are presenting a brief review of the metaheuristic approaches for the QAP. We carried out studies to identify better algorithms to find optimal solutions for the QAP. Some of these studies suggest that some algorithms may be improved using parallel approaches [33–36], while other studies suggest some algorithms based on the hybrid approaches [37–39], which have proven its superiority over other types of algorithms because it has addressed the issues found in other algorithms by using their advantages.

The study [40] has been presented a hybrid algorithm between the memetic algorithm (MA) and tabu search (TS) algorithm that was used to improve the local improvement strategy. The MA uses the concept of regular "restarts" of a population. A restart mechanism is employed in times when population diversity is low; in this case, the TS algorithm is applied. The results have shown the performance of the proposed algorithm competitively with some of the state-of-the-art algorithms for the QAP from the literature. The usage of TS algorithm has increased the accuracy of the solutions obtained for QAP problems.

Another study [41] proposed hybrid algorithm between biogeography-based optimization (BBO) algorithm and tabu search (TS) algorithm to overcome the weakness of the classical BBO algorithm in the mutation stage by replacing a mutation operator with a tabu search procedure. The ability to find the best solutions for 36 instances out of a total of 37 instances from QAP data was demonstrated by the hybrid proposed.

Recently, in the study [8] an algorithm is proposed by integrating the whale algorithm (WA) with the tabu search (TS) algorithm to solve the QAP. Since the WA suffers from a slow exploitation mechanism, so this integration is proposed to deal with this issue. According to the results, the proposed algorithm has obtained near-optimal results in an acceptable time, is well-suited for QAP, and outperforms the current algorithms in the literature when solving QAP. Furthermore, this study also aimed to collect hospital departments that have been modeled as a QAP model to reduce the average distance traveled by patients when moving between these departments. The results show that the performance of WAITS is better than the other algorithms that have a comparison with it, the reason is due to the improved local search using TS.

In the paper written by [42], discrete particle swarm optimization (DPSO) algorithm was modified to enable it to solve the QAP. The PSO is an efficient tool for solving combinatorial optimization problems. In the proposed algorithm, the authors made efforts to solve five sets of QAPs, and the results show slight deviations from the best-known solutions.

On the other hand, many studies recommend a comprehensive survey of metaheuristic algorithms; though these algorithms have been used successfully in various fields, some lacunae still exist in some of these algorithms, and further research is needed to resolve this [9]. For this purpose, this study aims to address some issues that are the DDE algorithm suffers from it which modified to solve QAP by the studies [23, 25]. The following two subsections have presented the critical analysis of relevant literature studies to identify the main weaknesses of these studies and the research's conceptual framework to identify the direction of this work.

3.1. Critical Analysis of Relevant Literature Studies [23, 25]. The main aim of constructive criticism is to analyze previous studies. This activity involves analyzing the strengths and weaknesses of a given study using a wide range of available frameworks and methods. Such frameworks and methods have been proposed to identify research materials (including theses and articles), the subject of the research, and the purpose of the study. The literature reviewed in this study has been constructively criticized. Table 1 shows the critical analysis of relevant literature studies.

3.2. Research Conceptual Framework. The metaheuristics approach has been divided into two groups: single-based metaheuristic and population-based metaheuristic algorithms. The advantage of the single-based metaheuristic is its local exploitation, whereas the population-based metaheuristic has global exploration and local exploitation. Our conceptual research framework has been presented in Figure 2.

On the other hand, this study has been selected by the DDE algorithm that belongs to the category of EAs from a population-based metaheuristic algorithm and the TS algorithm that belongs to the category of single-based metaheuristic algorithms and has been proved successful in integrating with other algorithm processes [39, 41]. The hybrid algorithm by combining the DDE that has global exploration and the TS algorithm that has local exploitation has been proposed to balance between the exploration mechanism and the exploitation mechanism.

4. Research Methodology

The research methodology in this study has been covered by five scenarios. Figure 3 shows these scenarios.

4.1. Proposed Hybrid Algorithm. This section proposes a hybrid algorithm, named HDDETS, by combining the DDE and TS algorithms. The basic steps of the HDDETS are as follows:

(a) Initialization: initialize population matrix $\pi = \{\pi_1, \pi_2, \pi_3, \ldots, \pi_{PS}\}$ randomly whose size is $P_s \times N_d$, where P_s is the size of population and N_d is the dimension of problem size. All population individuals should be unique. Initialize the set of solution and check the solutions stagnation $W =$ array of P_s with zeros and maximum wait and ht (iteration of tabu search).

(b) Evaluate fitness: evaluate solutions to find the best solution π_{best}^{t-1} from the population π by using equation (1).

Table 1: Critical analysis of relevant literature studies.

Reference	Algorithm	Critical analysis
[23]	Discrete differential evolution algorithm	(1) This study focused on solving discrete optimization problems using a modified differential evolution algorithm. (2) In this study, a restart strategy was recommended by the authors as a technique for generating solutions, rather than using crossover operators which have been used in many studies reviewed in the literature. However, the authors did not compare the result produced by the strategy with that of other techniques in which crossover operators were used. (3) In the study, the authors only used the tabu list which is just a component of the tabu search algorithm, which in previous studies, have demonstrated the ability to enhance the exploitation mechanism in the evolutionary algorithms. (4) Only QAPs of small size were focused on in this study. (5) Results showed that the proposed algorithm did not perform better than the max-min ant system combined with random selection and local search. (6) As for a direction for future work, the authors suggested the integration of the proposed method with another method such as a local search algorithm so that the efficiency of the proposed method can be improved. (7) This search did not take into account the effect of parameters tuning.
[25]	Discrete differential evolution algorithm	(1) The focus of this study was on the modification of the differential evolution algorithm which was modified to discrete DE (DEE) algorithm. (2) In this research, the mutation stage was improved through the inclusion of the proposed swap that uses a local search. (3) The method proposed in this study was applied to just a small portion of the QAP dataset. (4) The authors did not identify the kind of crossover which was used in the crossover stage although there is a wide range of crossover operators that have been recommended to be used together with algorithms for the purpose of solving QAP. (5) In this study, consideration was not given to the problems of premature convergence and stagnation. (6) Despite the numerous studies that have been carried out on the DDE algorithm, there are still inadequate exploitation mechanisms for it, although it has global exploration. This issue was not addressed in the study. (7) No method of parameter testing was used in the study.

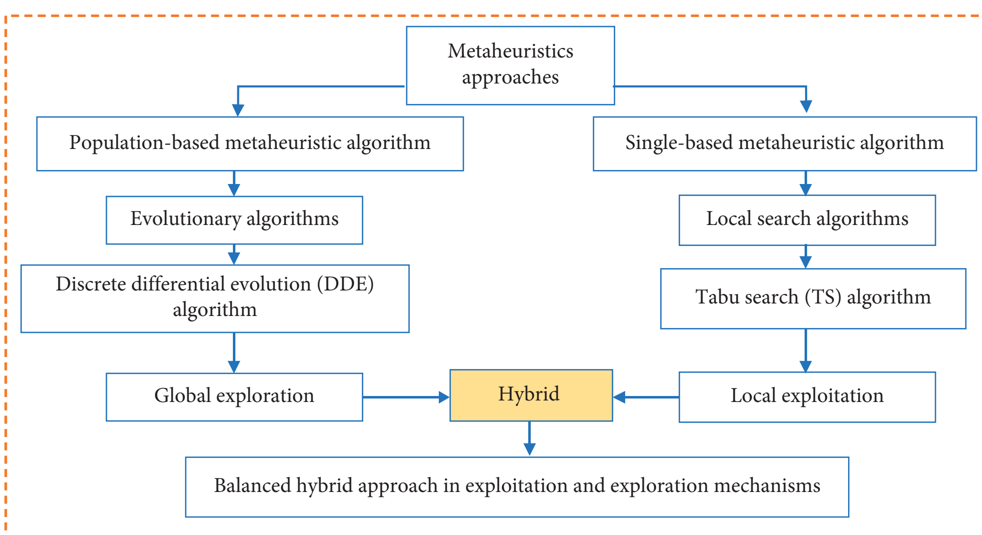


Figure 2: Our research conceptual framework.

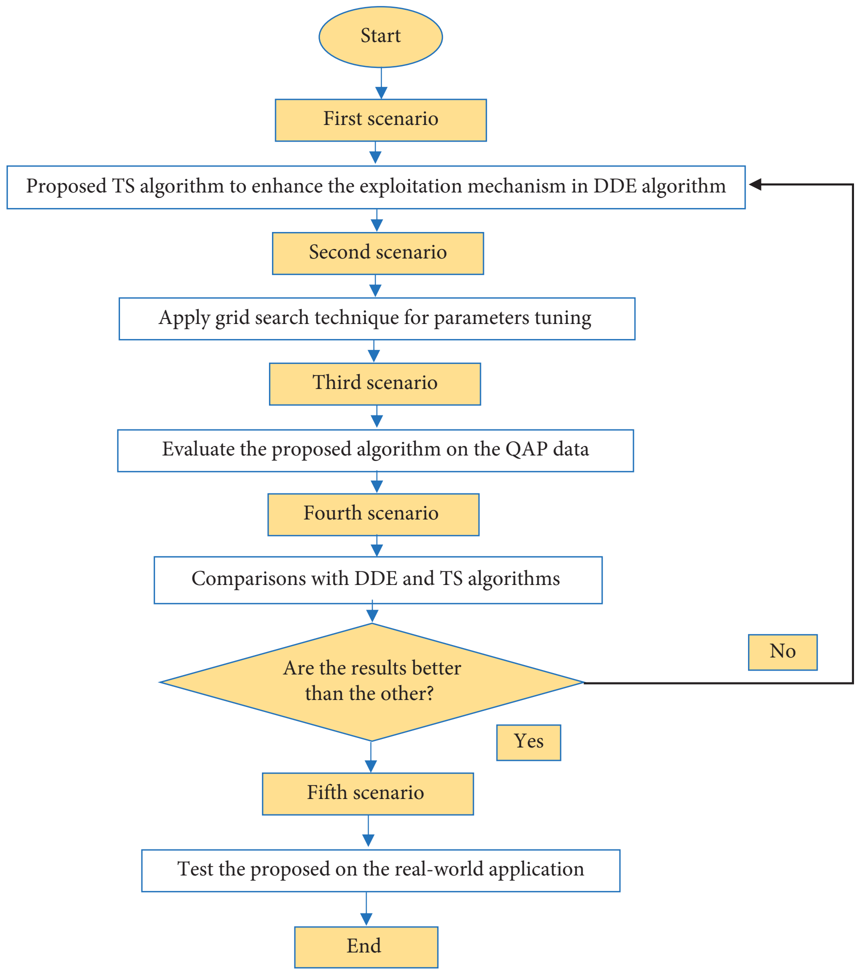


Figure 3: Research methodology flow.

(c) Mutation stage: the following equation shows this stage:

$$v_i^t = \begin{cases} \text{insert}\left(\pi_b^{t-1}\right), & \text{if } (r < P_m), \\ \text{swap}\left(\pi_b^{t-1}\right), & \text{otherwise.} \end{cases} \quad (2)$$

(d) Crossover stage: the type of crossover operator that was used in this stage called uniform-like crossover (ULX) was proposed by [43]. It works as follows: beginning, the similar locations in both parents are checked and then copied to the child (new solution). The second step involves selecting an item randomly and uniformly from both parents that have not yet been selected for the child after checking the unassigned locations from left to right. Finally, the rest of the items are randomly assigned to the locations. This stage is represented by the following equation:

$$u_i^t = \begin{cases} \text{CR}, & \text{if } (r < P_c), \\ v_i^t, & \text{otherwise.} \end{cases} \quad (3)$$

(e) Apply the TS for a hybrid algorithm:

(i) Initial solution: the solution from crossover stage is the initial solution in TS.

(ii) Great tabu list: the task of the tabu list is to avoid the swap move between the ith and jth elements in the current solution π that visited in the past to generate the neighborhood.

(iii) Generation neighborhood: the neighborhood has been obtained from π (solution) by applying the swap to move between the ith and jth elements in the current solution π. The new solutions π_n^* are obtained by generating the neighborhoods based on the swap movement made on the current solution π. Figure 4 shows the process of generated neighborhoods.

(iv) Evaluate neighborhood: the objective function in (1) is used to evaluate neighborhood permutation (exchanging two facilities).

(v) Selection of best neighborhood: after evaluating the neighborhood permutations, the lowest value among them is the best permutation.

(vi) Best candidate solution: best solution in neighborhoods and not existing in tabu list or better than best solution.

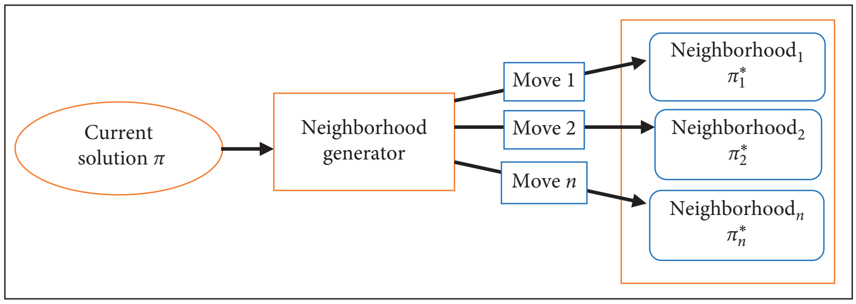


Figure 4: Process of generated neighborhoods [44].

(vii) Update tabu list: push the best candidate solution to the tabu list.

(f) Selection stage: the selection of the solutions depends on their values on the objective function. This stage is shown by the following equation:

$$\pi_i^t = \begin{cases} u_i^t, & \text{if } \left(f\left(u_i^t\right) \leq f\left(\pi_i^{t-1}\right)\right), \\ \pi_i^{t-1}, & \text{otherwise.} \end{cases} \quad (4)$$

(g) Checking the solutions stagnation: sometimes, the algorithm is not able to generate a new solution, which is the stagnation issue. To handle this issue, the term W is suggested as a wait for improvement of the solution within a certain number of times. If there is any improvement, the solution is deleted and regenerated randomly. The following equation shows this stage:

$$W_i = \begin{cases} 0, & \pi_i^t = u_i^t, \\ W_i + 1 & \pi_i^t = \pi_i^{t-1}. \end{cases} \quad (5)$$

The pseudocode of this hybrid algorithm is presented in Algorithm 1:

The flowchart of the proposed hybrid approach HDDETS is presented in Figure 5.

4.2. Parameters Tuning. The quality of the solutions obtained by using the proposed HDDETS algorithm can be influenced by the set of parameters. To identify the most suitable set of parameters that produce desirable results, the grid search technique is applied. The grid search technique is a simple optimization technique based on an exhaustive search through a manually defined set of classification technique parameter space. The advantage of grid search technique is the systematic creation of the candidate parameter settings; therefore, the same candidate parameter settings will be used to test each dataset [45]. Grid search is described as follows:

(i) Consistently generate candidate parameter settings supported by a given budget threshold. For instance, a budget threshold of 5 can limit the number of candidate settings for every parameter to 5. Thus, for an absolute classifier with three parameters and a budget threshold of five, the grid search technique generates $5 \times 5 \times 5 = 125$ mixtures of parameter settings.

(ii) Evaluate every candidate parameter setting.

(iii) Establish the optimal parameter settings.

The parameters that have been found by using grid search technique are shown in Table 2.

5. Results and Discussion

This section elucidates the efficiency of the proposed HDDETS algorithm. In order to encode the proposed algorithm, MATLAB was employed and run on a PC with Intel(*R*) Core(TM) i7-3770 CPU @ 3.40 GHz. Additionally, the PC was used which ran under MS Windows 10 with 8 GB RAM. This section comprises two parts: the first part highlights the parameters used for the proposed algorithm, whereas the second part discusses the results of the study.

5.1. Implementations of HDDETS Algorithm on QAP Instances. This section reports and discusses the results by the proposed HDDETS algorithm on three types of benchmark QAP instances from QAPLIB website, namely, "Esc," "Sko," and "Tai." These results include some statistical metrics such as the best-known solution (BKS) reported in QAPLIB website, best solution (BS), worst solution (WS), average solution (AS), best gap (BG), worst gap (WG), average gap (AG), standard deviation (SD), and average time (AT) over 10 runs. The following equation is used to find the percentage of gap (G):

$$G = 100 \times \frac{(\text{BS} - \text{BKS})}{\text{BKS}}. \quad (6)$$

Table 3 reports the results of 18 "Esc" type instances obtained by the proposed hybrid HDDETS algorithm. It is found from the table that the algorithm has ability to provide optimal solutions for all 18 instances in all 10 runs within a reasonable computational time. Table 4 reports the results of 13 "Sko" type instances by the HDDETS algorithm. These instances are considered as a challenge to all algorithms because no heuristic algorithm could find the best-known solutions (supposed to be optimal solutions) for all instances within reasonable computational time. It is found from Table 4 that the proposed algorithm could find the optimal

t_max: maximum number of iterations
ht_max: maximum number of tabu search iterations
w_max: maximum number of wait for improvement of the solution
NP: number of solutions S in population
Initial Population
For $t = 1$ **to** t_max
  Find the best solution π_{best}^{t-1}
  For each solution **in** the population
    Mutation stage
    Crossover stage
    Apply the TS for a hybrid
    Initial solution
    For $ht = 1$ **to** ht_max
      Create tabu list
      Create neighborhood
      Select the best neighbor and is not in the tabu list
      Update tabu list
    End for
    Selection stage
    Check solution stagnation W
    If keep same the solution
      $w = w + 1$
    Else
      $w = 0$
    End If
    If $w > w\_max$
      Regenerated
      $w = 0$
    End If
  End For
End For

Algorithm 1: Hybrid algorithm by combining DDE and TS algorithms.

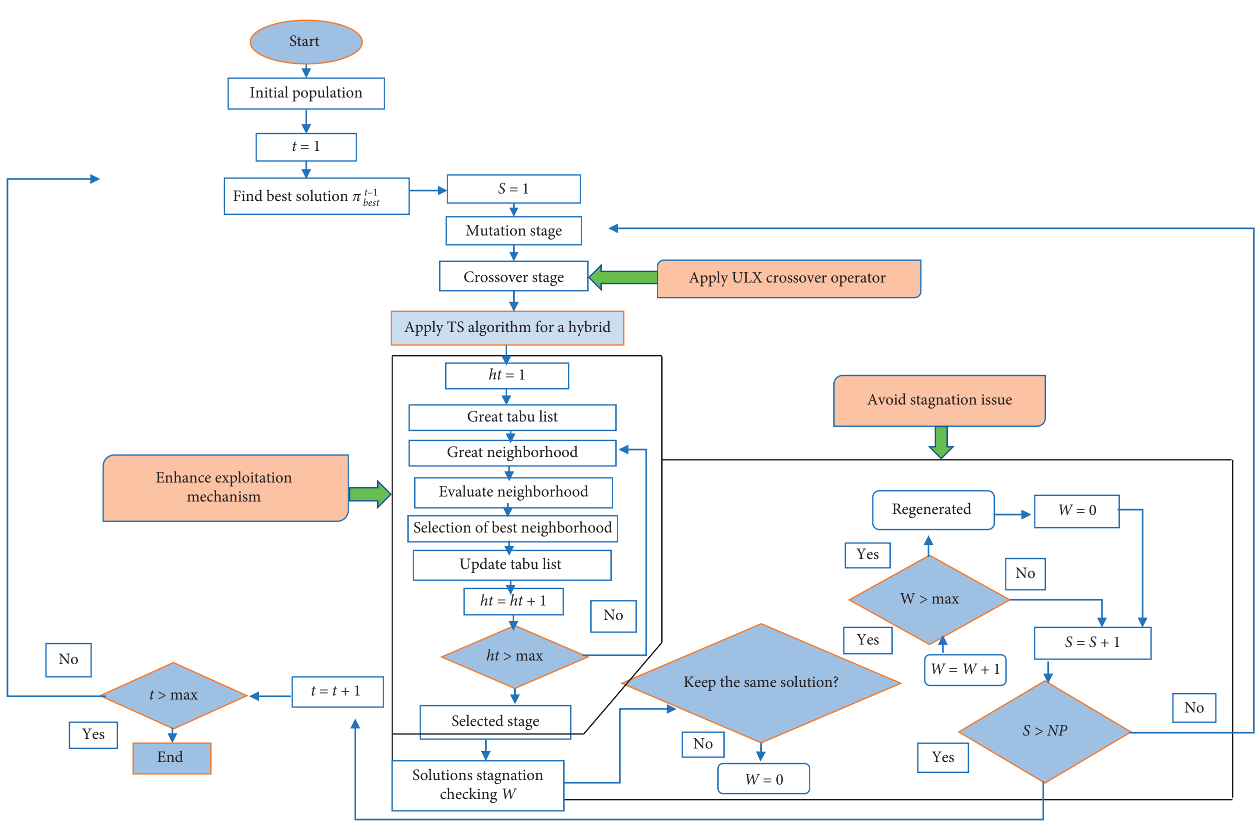


Figure 5: Flowchart of HDDETS algorithm.

Table 2: Parameters tuning.

Parameter	Value
Size of population, P_s	200
Maximum iterations	300
Probability of mutation, P_m	0.7
Probability of crossover, P_c	0.8
Maximum waiting for solutions updates	10
Tabu list length	7
Maximum iterations in TS	50
Number of runs	10

Table 3: Results by HDDETS algorithm on "Esc" type instances.

No.	Instance	BKS	HDDETS algorithm							
			BS	WS	AS	BG	WG	AG	SD	AT
1	Esc16a	68	**68**	68	68	**0**	0	0	0	0.533
2	Esc16b	292	**292**	292	292	**0**	0	0	0	0.634
3	Esc16c	160	**160**	160	160	**0**	0	0	0	0.577
4	Esc16d	16	**16**	16	16	**0**	0	0	0	0.532
5	Esc16e	28	**28**	28	28	**0**	0	0	0	6.496
6	Esc16g	68	**68**	68	68	**0**	0	0	0	6.578
7	Esc16h	26	**26**	26	26	**0**	0	0	0	0.472
8	Esc16i	996	**996**	996	996	**0**	0	0	0	0.473
9	Esc16j	14	**14**	14	14	**0**	0	0	0	0.629
10	Esc32a	8	**8**	8	8	**0**	0	0	0	0.737
11	Esc32b	130	**130**	130	130	**0**	0	0	0	7.953
12	Esc32c	168	**168**	168	168	**0**	0	0	0	1.924
13	Esc32d	642	**642**	642	642	**0**	0	0	0	2.276
14	Esc32e	200	**200**	200	200	**0**	0	0	0	2.184
15	Esc32g	6	**6**	6	6	**0**	0	0	0	1.835
16	Esc32h	438	**438**	438	438	**0**	0	0	0	1.907
17	Esc64a	116	**116**	116	116	**0**	0	0	0	1.896
18	Esc128a	64	**64**	64	64	**0**	0	0	0	9.927

The boldface denote that the best solutions (BS) of these instances have been obtained through the lowest value obtained by the objective function of the QAP model during the process of repetition using our HDDETS algorithm. Moreover, these solutions have achieved the best-known solution (BKS) reported in the QAPLIB dataset. Thus, obtained the values of the best gaps (BG) of the solutions of these instances are 0% according to equation (6).

Table 4: Results by HDDETS algorithm on "Sko" type instances.

No.	Instance	BKS	HDDETS algorithm							
			BS	WS	AS	BG	WG	AG	SD	AT
1	Sko42	15812	**15812**	15812	15812	**0**	0	0	0	146.959
2	Sko49	23386	**23386**	23402	23394	**0**	0.068	0.032	0.063	27.001
3	Sko56	34458	**34458**	34472	34461	**0**	0.040	0.0079	0.132	623.89
4	Sko64	48498	**48498**	48604	48524	**0**	0.218	0.052	0.241	858.975
5	Sko72	66256	**66256**	66422	66328	**0**	0.250	0.108	0.136	368.841
6	Sko81	90998	91008	91156	91042	0.068	0.578	0.346	0.18	1624.424
7	Sko90	115534	115578	115898	115766	0.192	0.834	0.443	0.197	1727.96
8	Sko100a	152002	152252	153366	152646	0.164	0.897	0.423	0.382	2969.776
9	Sko100b	153890	**153890**	154276	154128	**0**	0.250	0.154	0.189	1490.758
10	Sko100c	147862	147868	147920	147894	0.004	0.039	0.021	0.325	2930.167
11	Sko100d	149576	149666	150092	149862	0.060	0.344	0.190	0.15	1472.923
12	Sko100e	149150	**149150**	149742	149382	**0**	0.396	0.155	0.341	6488.738
13	Sko100f	149036	149070	149470	149291	0.022	0.291	0.171	0.144	1265.821

The boldface denote that the best solutions (BS) of these instances have been obtained through the lowest value obtained by the objective function of the QAP model during the process of repetition using our HDDETS algorithm. Moreover, these solutions have achieved the best-known solution (BKS) reported in the QAPLIB dataset. Thus, obtained the values of the best gaps (BG) of the solutions of these instances are 0% according to equation (6).

Table 5: Results by HDDETS algorithm on "Tai" type instances.

No.	Instance	BKS	HDDETS algorithm							
			BS	WS	AS	BG	WG	AG	SD	AT
1	Tai12a	224416	**224416**	224416	224416	**0**	0	0	0	0.508
2	Tai12b	39464925	**39464925**	39464925	39464925	**0**	0	0	0	0.684
3	Tai15a	388214	**388214**	388214	388214	**0**	0	0	0	0.847
4	Tai15b	51765268	**51765268**	51765268	51765268	**0**	0	0	0	0.81
5	Tai20a	703482	**703482**	703482	703482	**0**	0	0	0	5.653
6	Tai20b	122455319	**122455319**	122455319	122455319	**0**	0	0	0	0.511
7	Tai25a	1167256	**1167256**	1167256	1167256	**0**	0	0	0	7.848
8	Tai25b	344355646	**344355646**	344355646	344355646	**0**	0	0	0	13.46
9	Tai30a	1818146	**1818146**	1818146	1818146	**0**	0	0	0	57.4
10	Tai30b	637117113	**637117113**	637117113	637117113	**0**	0	0	0	29.794
11	Tai35a	2422002	**2422002**	2435108	2428275	**0**	0.541	0.259	0.066	29.589
12	Tai35b	283315445	**283315445**	283315445	283315445	**0**	0	0	0	57.4
13	Tai40a	3139370	3141431	3151727	3148060	0.065	0.393	0.276	0.087	138.357
14	Tai40b	637250948	**637250948**	637250948	637250948	**0**	0	0	0	416.445
15	Tai50a	4938796	4965748	4991782	4981218	0.545	1.072	0.858	1.297	768.214
16	Tai50b	458821517	**458821517**	458821517	458821517	**0**	0	0	0	48.479
17	Tai60a	7205962	7259430	7321212	7296755	0.742	1.599	1.26	0.294	921.089
18	Tai60b	608215054	**608215054**	608215054	608215054	**0**	0	0	0	123.702
19	Tai64c	1855928	**1855928**	1855928	1855928	**0**	0	0	0	8.308
20	Tai80a	13499184	13616880	13678302	13654388	0.871	1.326	1.149	0.215	1195.736
21	Tai80b	818415043	**818415043**	818415043	818415043	**0**	0	0	0	1399.883
22	Tai100a	21044752	21256606	21318876	21304805	0.621	0.916	0.768	0.202	2740.755
23	Tai100b	1185996137	1187179912	1212182931	1191632007	0.099	2.208	0.475	0.624	1553.481
24	Tai150b	498896643	501892435	508173332	505261057	0.600	1.859	1.275	0.442	9402.760
25	Tai256c	44759294	44786418	44838798	44813276	0.060	0.120	0.120	0.041	41014.570

The boldface denote that the best solutions (BS) of these instances have been obtained through the lowest value obtained by the objective function of the QAP model during the process of repetition using our HDDETS algorithm. Moreover, these solutions have achieved the best-known solution (BKS) reported in the QAPLIB dataset. Thus, obtained the values of the best gaps (BG) of the solutions of these instances are 0% according to equation (6).

Table 6: Best gap by HDDETS algorithm on "Esc," "Sko," and "Tai" instances.

Best gap (%)	Frequency	Percentage	Cumulative percentage
0.000	42	75.0	75.0
0.004	1	1.8	76.8
0.022	1	1.8	78.6
0.060	2	3.6	82.1
0.065	1	1.8	83.9
0.068	1	1.8	85.7
0.099	1	1.8	87.5
0.164	1	1.8	89.3
0.192	1	1.8	91.1
0.545	1	1.8	92.9
0.600	1	1.8	94.6
0.621	1	1.8	96.4
0.742	1	1.8	98.2
0.871	1	1.8	100.0
Total	56	100.0	

solutions for 7 instances at least once in 10 runs. Also, the results of the remaining 6 instances are very close to optimal solutions, within a gap of less than 1% within a reasonable time.

Table 5 reports the results of 25 "Tai" type instances by our proposed algorithm. It is found that the algorithm could find the optimal solution for 16 instances at least once in 10 runs. Looking at the average solutions, the algorithm could find the optimal solution in all 10 runs for 10 instances, namely, Tai12a, Tai12b, Tai15a, Tai15b, Tai20a, Tai20b, Tai25a, Tai25b, Tai30b, and Tai64c, while for the rest of 15 instances, percentage of average gap is found to be very small, within 1.30%.

Further, the results of the proposed HDDETS algorithm on all three types of instances are discussed and analyzed statistically by using the SPSS software. Table 6 reports the results of statistical analysis of the proposed algorithm on 56 instances. The results showed that the best value is achieved for 42 of 56 instances (75%) with a 0% gap, while for the remaining 14 instances, results are very close to optimal solutions, within a gap of less than 1%.

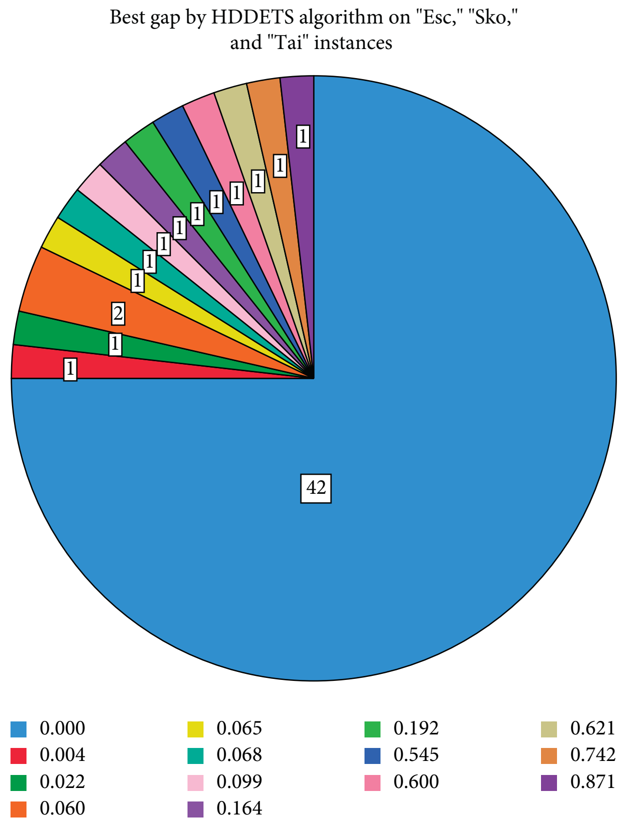


Figure 6: A graphical representation of the best gap in Table 6.

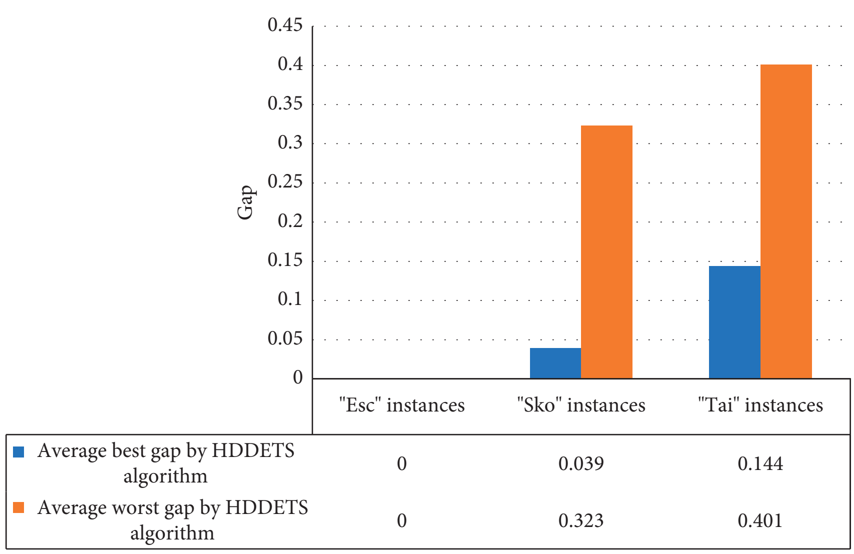


Figure 7: Comparison between an average best gap and average worst gap obtained by HDDETS algorithm on “Esc,” “Sko,” and “Tai” instances.

A graphical representation of the best gap from Table 6 has been presented in Figure 6.

Figure 7 shows the average of the best gap and the average of the worst gap of the solutions of "Esc," "Sko," and "Tai" instances using our proposed HDDETS algorithm.

Statistical analysis presented in Table 7 proved that the performance of the proposed algorithm is efficient and effective, as the mean of gaps is 0.073 for all the 56 instances and the maximum value of the gaps was observed to be 0.871. The variance of the gap is found to be 0.039, which indicates that the

Table 7: Statistical analysis on best gap by HDDETS algorithm on "Esc," "Sko," and "Tai" instances.

Description	Value
No. of instances	56
Mean	0.073
Std. deviation	0.197
Variance	0.039
Range	0.871
Minimum	0.000
Maximum	0.871

Table 8: Comparisons on "Esc" type instances.

No.	Instance	Best gap (DDE)	Best gap (TS)	Best gap (HDDETS)
1	Esc16a	0	0	0
2	Esc16b	0	0	0
3	Esc16c	0	0	0
4	Esc16d	0	0	0
5	Esc16e	0	0	0
6	Esc16g	0	0	0
7	Esc16h	0	0	0
8	Esc16i	0	15.384	0
9	Esc16j	0	14.285	0
10	Esc32a	20.000	0	0
11	Esc32b	19.047	0	0
12	Esc32c	0	0	0
13	Esc32d	0	0	0
14	Esc32e	0	0.913	0
15	Esc32g	0	0	0
16	Esc32h	0.913	0	0
17	Esc64a	0	0	0
18	Esc128a	34.375	0	0

Table 9: Comparisons on "Sko" instances.

No.	Instance	Best gap (DDE)	Best gap (TS)	Best gap (HDDETS)
1	Sko42	2.567	2.203	0
2	Sko49	1.599	1.941	0
3	Sko56	2.704	1.967	0
4	Sko64	3.365	2.581	0
5	Sko72	3.595	3.420	0.090
6	Sko81	3.356	2.960	0.068
7	Sko90	3.661	3.368	0.192
8	Sko100a	3.326	3.018	0.206
9	Sko100b	3.184	3.082	0.18
10	Sko100c	3.907	3.872	0.193
11	Sko100d	3.866	3.403	0.124
12	Sko100e	3.886	3.297	0.244
13	Sko100f	3.616	3.016	0.454

solutions obtained are to some extent identical to the solutions when compared with the standard QAP.

5.2. Comparisons between DDE, TS, and HDDETS Algorithms. This section has been covered by two scenarios. The first relates to the comparisons between the DDE, TS,

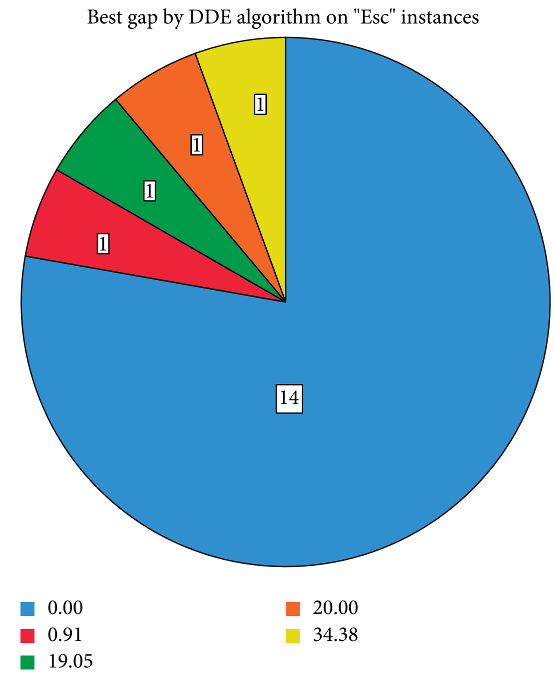


Figure 8: Best gap by DDE algorithm on "Esc" instances.

and HDDETS algorithms, while the second presents and discusses the results statistically. The results of the comparisons between the DDE, TS, and HDDETS algorithms are given in Tables 8–10 on the instances belonging to the three types ("Esc," "Sko," and "Tai") of the QAP instances.

There are 18 "Esc" type instances in the QAPLIB website that have optimal solutions. The results in Table 8 show that the DDE and TS algorithms have optimal solutions for 14 and 15 instances, respectively, while the HDDETS algorithm obtained optimal solutions for all the 18 instances. Figures 8–10 show graphical representations of the best gap from Table 8.

Table 11 reports statistical measures for DDE, TS, and HDDETS algorithms on "Esc" type instances. According to the results reported in this table, the means of the gaps are 4.129, 1.699, and 0 by the DDE, TS, and HDDETS algorithms, respectively. Variances of gaps obtained by the DEE, TS, and HDDETS algorithms are 96.369, 22.918, and 0, respectively. Additionally, the maximum gaps obtained are 34.375, 15.384, and 0 using DEE, TS, and HDDETS algorithms, respectively. In conclusion, performance evaluation for the proposed hybrid algorithm, HDDETS, indicates that it outperforms both DDE and TS on the "Esc" type instances.

Another comparison was made among DDE, TS, and HDDETS algorithms on 13 "Sko" type instances. Neither DDE nor TS could reach any best-known solution out of 13 instances, while the proposed algorithm managed to obtain the best-known solution of 4 instances. Table 9 reports these results.

Graphical representations of the best gaps reported in Table 9 are presented in Figures 11–13. Further, the results are discussed according to the statistical measures that are presented in Table 12.

The means of the gaps are obtained as 3.279, 2.932, and 0.039 by DDE, TS, and HDDETS algorithms, respectively.

Table 10: Comparisons on "Tai" instances.

No.	Instance	Best gap (DDE)	Best gap (TS)	Best gap (HDDETS)
1	Tai12a	0	0	0
2	Tai12b	2.849	0	0
3	Tai15a	2.043	0.17	0
4	Tai15b	0.339	0	0
5	Tai20a	2.983	1.40	0
6	Tai20b	4.592	2.46	0
7	Tai25a	1.743	0	0
8	Tai25b	4.216	2.81	0
9	Tai30a	2.039	1.93	0
10	Tai30b	4.548	4.47	0
11	Tai35a	3.502	2.70	0
12	Tai35b	4.777	2.70	0
13	Tai40a	2.157	2.56	0
14	Tai40b	4.748	5.85	0.065
15	Tai50a	0.076	3.18	0
16	Tai50b	5.246	2.93	0.019
17	Tai60a	3.388	3.51	0
18	Tai60b	4.609	4.20	0.047
19	Tai64c	0.417	0.09	0
20	Tai80a	5.665	3.76	1.059
21	Tai80b	7.486	4.35	0
22	Tai100a	5.743	10.85	1.146
23	Tai100b	7.116	5.20	0.099
24	Tai150b	8.607	6.22	0.6
25	Tai256c	1.564	0.30	0.06

Table 11: Statistical measures for DDE, TS, and HDDETS algorithms on "Esc" instances.

Description	DDE algorithm	TS algorithm	HDDETS algorithm
No. of instances	18	18	18
Mean	4.129	1.699	0
Std. deviation	9.816	4.787	0
Variance	96.369	22.918	0
Range	34.375	15.384	0
Minimum	0	0	0
Maximum	34.375	15.384	0

Variances among the gaps are converging, and they are 0.655, 0596, and 0.066 by DDE, TS, and HDDETS algorithms, respectively. In addition, the recorded maximum gaps are 3.907, 3.872, and 0.192 derived by DDE, TS, and HDDETS algorithms, respectively. Based on the above-mentioned study, one can say that the performance of the proposed algorithm, HDDETS, is found to be better than the performance of DDE and TS algorithms on the "Sko" type instances. Further, 25 "Tai" type instances have been used for comparison among the proposed hybrid algorithm (HDDETS) and DDE and TS algorithms. Table 10 shows the results of this comparison.

From Table 10, it is found that the DDE algorithm hits the optimal solution for only one instance at least once in 10 runs, whereas TS algorithm obtained four optimal solutions. The proposed algorithm, HDDETS, could hit the optimal solutions for 17 instances at least once in 10 runs. For the remaining 8 instances, the gap is very small, within less than 1.15%. Figures 14–16 show graphical representations of the best gaps reported in Table 10.

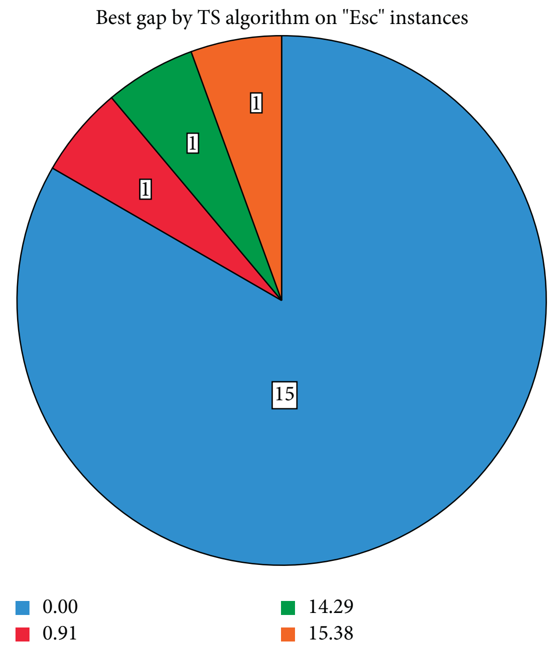


Figure 9: Best gap by TS algorithm on "Esc" instances.

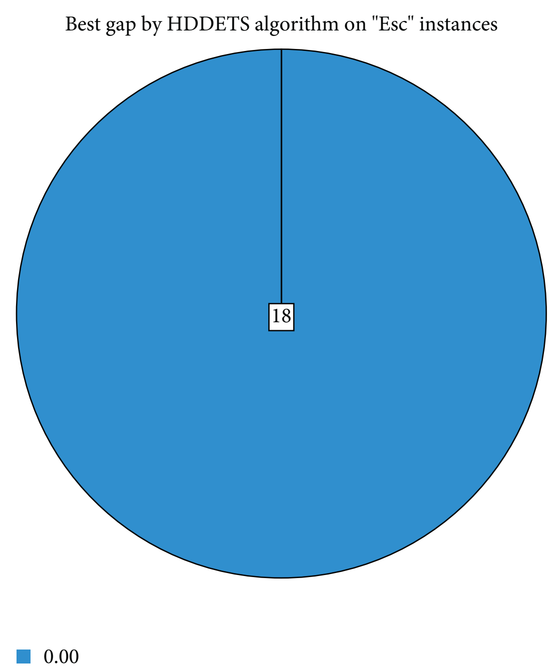


Figure 10: Best gap by HDDETS algorithm on "Esc" instances.

Table 13 shows the statistical measures that are used to compare these algorithms for "Tai" type instances.

The means of the gaps by DDE and TS algorithms are 3.618 and 2.864, respectively. By the proposed algorithm, the mean of the best gaps is 0.144. It is worth noting that

Table 12: Statistical measures for DDE, TS, and HDDETS algorithms on "Sko" instances.

Description	DDE algorithm	TS algorithm	HDDETS algorithm
Instance	13	13	13
Mean	3.279	2.932	0.039
Std. deviation	0.655	0.596	0.066
Variance	0.429	0.356	0.004
Range	2.308	1.931	0.192
Minimum	1.599	1.941	0.00
Maximum	3.907	3.872	0.192

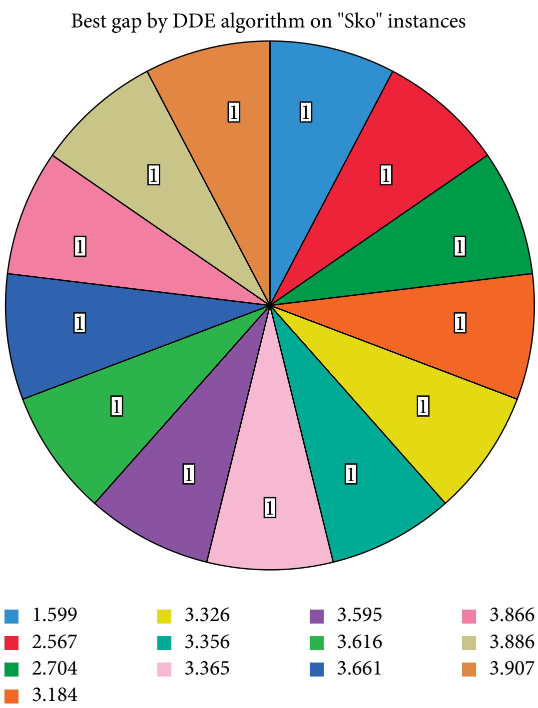


Figure 11: Best gap by DDE algorithm on "Sko" instances.

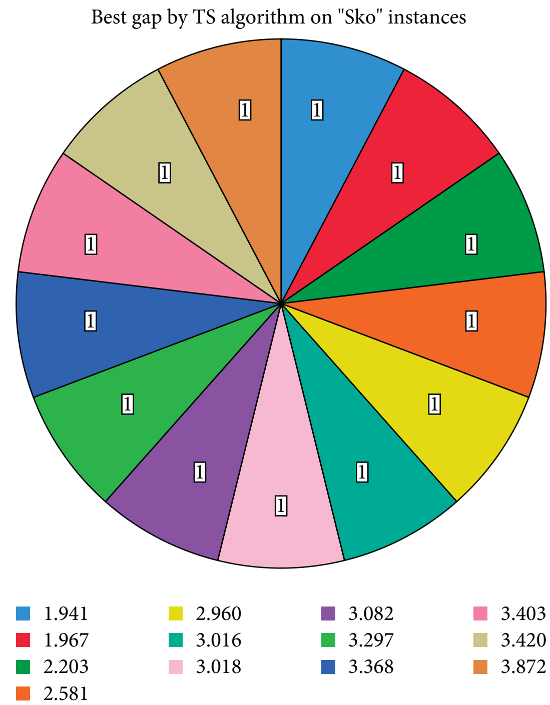


Figure 12: Best gap by TS algorithm on "Sko" instances.

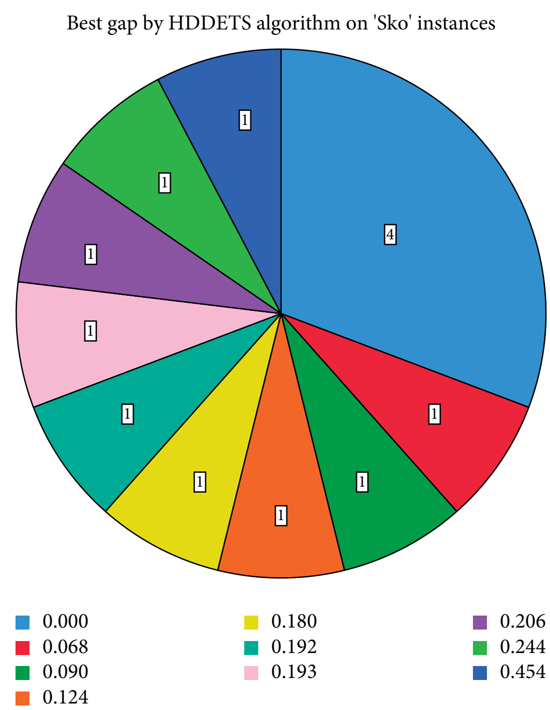


Figure 13: Best gap by HDDETS algorithm on "Sko" instances.

the variance among the gaps is 2.343 by the DDE algorithm, while the variance by the TS algorithm is 3.209. In the same context, the HDDETS algorithm has got the best variance among the gaps (0.077). Moreover, the maximum relative gap value was 8.61%, 10.85%, and 0.87% obtained by the DDE, TS, and HDDETS algorithms, respectively.

5.3. Real-World Application. There are many studies that have proven the successful use of the QAP in real applications, because of its ability to find the best allocation of locations for the available facilities. The artificial bee colony algorithm was used in [32] which aimed at solving a QAP within the context of Azadi Hospital so that a better distribution of advisory services can be found within the hospital building. This way, the efforts required when moving patients from one advisory service to another are minimized, which in turn also reduces the total distance required for transmitting the patients. As an evaluation of the efficiency of the performance of the proposed algorithm in this research, it was applied to the data of this case study.

5.3.1. Data Description of the Azadi Hospital Layout Problem. The hospital layout problem aims for the distribution of departments inside the hospitals to minimize the total travel distance by patients. Azadi Hospital, which was used for this study, was founded in 1985. Recent study [32] dealt with this problem and has discussed the description of the data as follows.

Table 13: Statistical measures for DDE, TS, and HDDETS algorithms on 'Tai' instances.

Description	DDE algorithm	TS algorithm	HDDETS algorithm
Instance	25	25	25
Mean	3.618	2.864	0.144
Std. deviation	2.343	1.791	0.277
Variance	5.490	3.209	0.077
Range	8.61	10.85	0.87
Minimum	0.00	0.00	0.00
Maximum	8.61	10.85	0.87

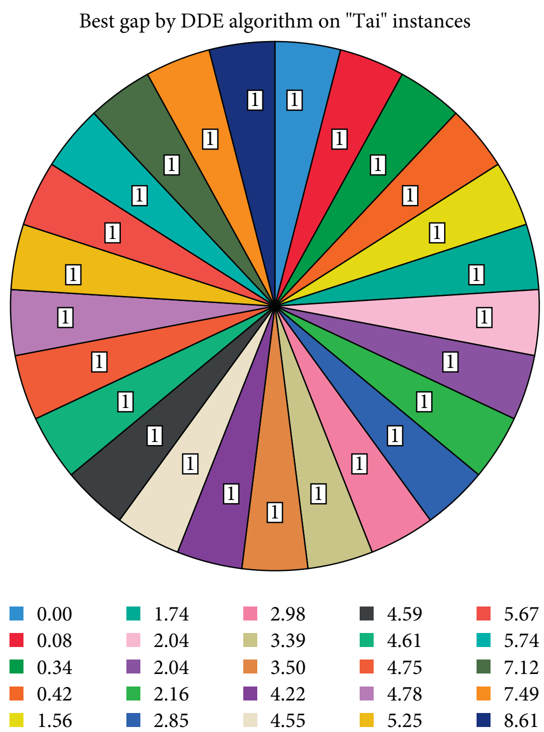


Figure 14: Best gap by DDE algorithm on "Tai" instances.

The hospital offers a variety of quality healthcare services and is operated by the most specialized doctors with sufficient experience in the field of medicine. The hospital, which houses approximately 398 beds, is made up of 7 floors and is well-equipped with state-of-the-art facilities and equipment. In addition, the hospital is also equipped with operation theatres for minor and major procedures, an emergency room, and a variety of laboratories in which different kinds of ailments can be diagnosed. Despite the fact that the hospital is made up of different floors, sections, and units, this study only focused on 30 facilities located on the ground floor, because this floor witnesses the influx of patients on a daily basis, and for this reason, this specific floor is used for the study.

The problem was divided into three categories, the first category includes fixed consulting services, which must be installed for the lack of other similar places such as entry, laboratory, pharmacy, rays, computerized axial tomography, magnetic resonance imaging, electrocardiography, and physical therapy. The second category represents consulting departments such as a dental department, ENT department, and ophthalmology department, while the third category represents consulting rooms such as echo, straining the heart, neurological, lung functions, ultrasound, urologist, pediatric, vaccine, general surgery, trauma room, gynecology, the internists1, the internists2, minor operations, interactions skin, dermatologic, and hematological.

In order to obtain the flow matrix entries, the flow between each pair of clinics was averaged. The paths which were taken by patients during their movement from one location to another were traced by measuring the distances between locations. Based on observation, patients are required to go and mark off their cards in the pharmacy after they had visited several clinics. It was noticed that a patient, after being through a sequence of visits to more than one clinic, must go to the pharmacy to mark off his card. Table 14 shows the distribution of clinics to rooms by using the study [32].

5.3.2. Justifications of the Case Study. Given the applications domain of the QAP, the lack of proper layout of the clinic's locations inside the hospital requires patients to cover a lot of distance [32]; therefore, the clinics within a hospital need to be meticulously organized, as disorganization can result in too much movement for patients, thereby leading to wastage of time. In the hospital, the layout can be formulated as a QAP for locating the clinics within the hospital which can substantially reduce the total distance that patients travel as they move from one clinic to another in the hospital building. On the other hand, modelling the hospital's departments on QAP minimizes the travel distance for patients [8]. This helps to

(i) Reduce the traveling efforts of patients between departments, thus saving their time.

(ii) Improve the service efficiency of the hospital, and

(iii) The hospital would be capable to serve a greater number of patients daily.

5.3.3. Location of the Study. This study was conducted at the Azadi Hospital in Kirkuk Governorate, Iraq. It involved the distribution of thirty advisory clinics inside the hospital, based on the QAP model, and found the minimized cost through solving the QAP model by using the proposed algorithm. Figure 17 shows the locations of the case study in this study.

5.3.4. Applying the Proposed Algorithm HDDETS on Azadi Hospital Layout Problem. The proposed hybrid algorithm HDDETS has given a clear picture of the effective performance of the algorithm in addition to obtaining optimal solutions for some instances of QAP data with reasonable computational time, unlike exact algorithms. Moreover, the results have proven the ability of the HDDETS algorithm to find the optimal solution if the size of the problem is more

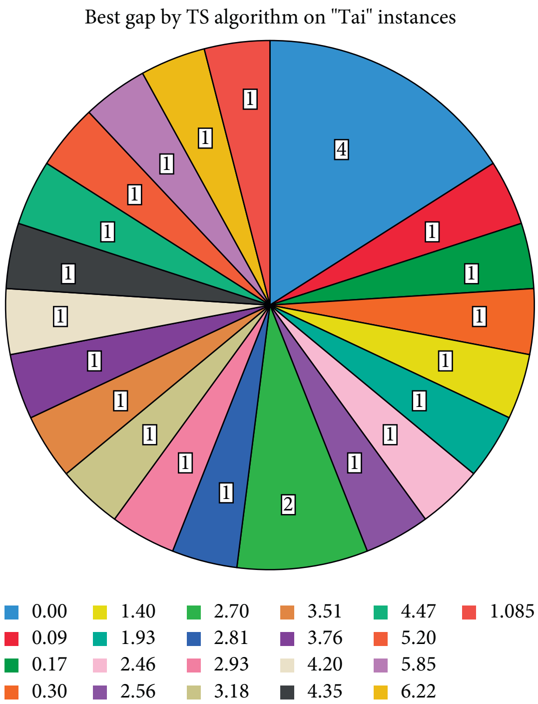


Figure 15: Best gap by TS algorithm on "Tai" instances.

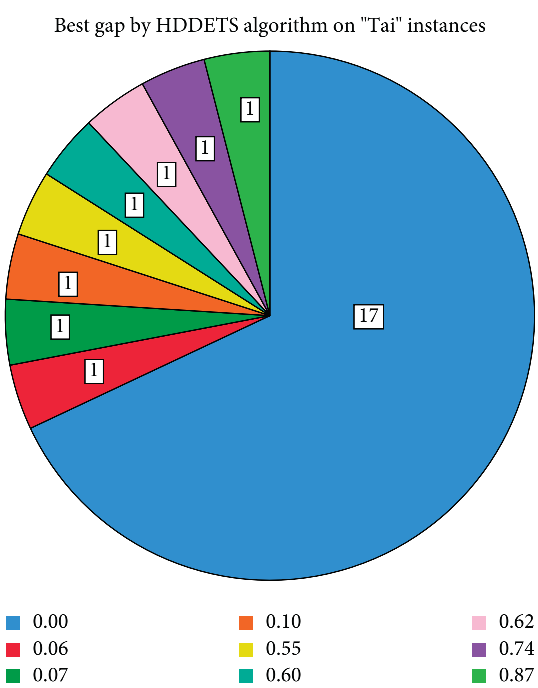


Figure 16: Best gap by HDDETS algorithm on "Tai" instances.

than 30 (locations/facilities), such as Tai30b, Esc32a, Esc32b, Esc32c, Esc32d, Esc32e, Esc32g, Esc32h, Esc64a, and Esc128a.

Modelling the hospital's departments on QAP will optimize the distribution of the departments within the hospital which helps to obtain a minimum total cost. On one hand, the cost of transferring patients when they move from one advisory clinic to another was 8, 974, 071 meters per month. This

Table 14: Distribution of the advisory clinics by [32].

Location	Facility (advisory clinic)
1	Entry
2	Pharmacy
3	Laboratory
4	Rays
5	Computerized axial
6	Magnetic resonance imaging
7	Electrocardiography
8	Physical therapy
9	Dentistry
10	Ear nose and throat
11	Ophthalmologic
12	Allergy and asthma
13	Lung functions
14	Endoscopy
15	Straining the heart
16	Neurological
17	Echo
18	Gynecology
19	Urologist
20	Dermatologic
21	Pediatric
22	Ultrasound
23	Interactions skin
24	The internists1
25	Hematological
26	Trauma room
27	The internists2
28	Minor operations
29	General surgery
30	Vaccine

cost has been reduced to 8, 312, 830 meters per month by using the study [32] based on the distribution of the advisory clinics in Table 14. The minimum cost value of the QAP problem is obtained when the optimal distribution for those facilities' locations is found. Table 15 presents the permutation of the optimal distribution of the advisory clinics inside Azadi Hospital by using the HDDETS algorithm.

The Azadi Hospital layout problem consists of 30 advisory clinics and the best cost result of transferring patients inside the hospital was 8, 312, 830 meters per month obtained by the study [32] which has been optimizing this value to 4, 559, 262 meters per month when using the proposed HDDETS algorithm.

6. Theoretical Analysis and Limitations of the Proposed Hybrid Algorithm HDDETS

The DDE and TS algorithms used in this study are metaheuristic algorithms. The literature studies have proven that the metaheuristic algorithms are one of the best techniques for handling NP-hard problems. According to the no free lunch (NFL) theorem [46], there is no algorithm that has better performance for all problems, but in some cases, it is poor. So, we aim to develop an algorithm that has good performance for a specific problem only, but not a comprehensive solution to all problems. Therefore, proposing a

Table 15: New distribution of the advisory clinics.

Location	Facilities (advisory clinics)
1	Entry
2	Pharmacy
3	Laboratory
4	Rays
5	Computerized axial
6	Magnetic resonance imaging
7	Electrocardiography
8	Physical therapy
9	Dentistry
10	Trauma room
11	Echo
12	Neurological
13	Allergy and asthma
14	Hematological
15	Lung functions
16	Straining the heart
17	Interactions skin
18	Minor operations
19	Endoscopy
20	The internists 1
21	The internists 2
22	General surgery
23	Vaccine
24	Gynecology
25	Urologist
26	Ultrasound
27	Ear nose and throat
28	Dermatologic
29	Pediatric
30	Ophthalmologic

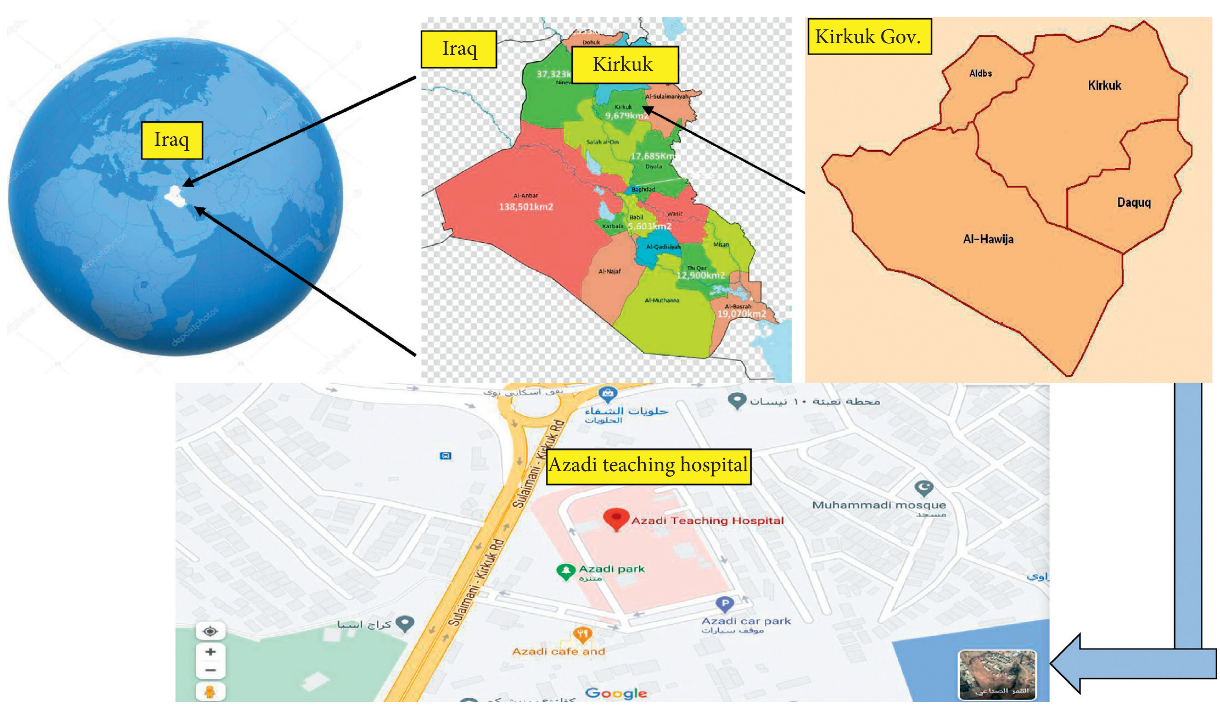


Figure 17: Locations of the study area, Azadi Hospital in Kirkuk Governorate, Iraq.

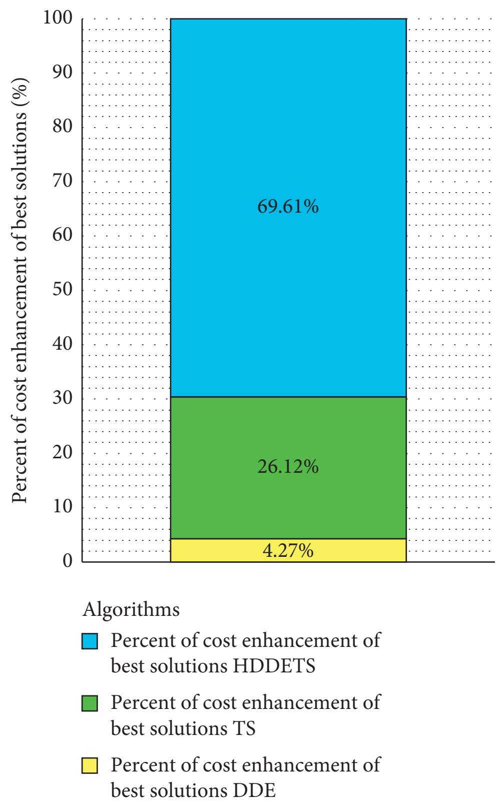


FIGURE 18: Percent of cost enhancement of best solutions by DDE, TS, and HDDETS algorithms.

hybrid algorithm will increase the performance and thus increase the probability of obtaining the best results [47].

Our study addresses some of the issues that meta-algorithms face, in particular, evolutionary algorithms that are not covered in the DDE algorithm, such as the stagnations, lack of the exploitation mechanism, and parameter tuning.

Sometimes, the algorithm is not able to generate a new solution through the evolutionary process, which means it has lost its capability to improve the solutions. This issue is called stagnation, and Equation (5) has suggested for avoiding this issue, while the issue of the lack of the mechanism of exploitation in the DDE algorithm is addressed by hybridizing the algorithm with a TS algorithm that is characterized by having the feature of local exploitation based on searching in the neighborhoods of the solution in addition to the tabu list that is used to prevent visiting solutions that have been visited in the past. In this way, it prevents falling into local optimal. On the other hand, a grid search technique has been applied for parameter tuning, which is another issue in the optimization algorithms because of its prominent role in improving the performance of the algorithm.

The results obtained by the HDDETS algorithm proved that the performance of the hybrid algorithm is better than the performance of both DDE and TS algorithms. The explanation of this outstanding and efficient performance has been discussed as follows: issues of stagnation and parameters tuning in addition to the lack of the mechanism of exploitation in the DDE algorithm have a negative and highly evident impact on the performance of this algorithm, as indicated in Figure 18.

The first task of the HDDETS algorithm is to generate a random population and then determine the best existing solution, thus calculating the cost of this solution by applying all stages of the algorithm. After that, we investigate the percentage of cost improvement of this solution in order to find the reasons for the difference in cost. This difference was made for three reasons. The first was related to the improvement of the solution found through the DDE algorithm part. The second reason was the improvement achieved using the TS algorithm part. The final reason of this improvement is the use of the entire HDDETS algorithm.

It is noted that the percent of cost enhancement of the best solutions of the DDE algorithm part is small (4.27%) because it suffers from the abovementioned issues. The TS algorithm part is characterized by using local exploitation, by searching in the areas of the solution. This algorithm resulted in an improved cost enhancement of the best solutions (26.12%). Finally, we proposed the hybrid algorithm HDDETS, using global exploration and local exploitation mechanisms that produce a more balanced algorithm. This balance is positively reflected by the increase in cost enhancement percent of best solutions (69.61%) using the entire HDDETS algorithm within a reasonable time.

In general, the proposed hybrid algorithm HDDETS has proven to be efficient in solving problems that have a large-scale implication which is considered a challenge for all algorithms. However, there are still some limitations that must be addressed. This study had applied crossover operators in the stage of crossover in the DDE algorithm called uniform-like crossover (ULX) although there are other types of crossover operators such as multiple parent crossover operator (MPX) and cycle crossover (CX) that have been suggested for the QAP with other algorithms. It is difficult for one study to cover all the operators that have been proposed in the literature with other algorithms.

The time for metaheuristic algorithms to find the solutions is a reasonable time, in contrast to exact algorithms. In general, hybridization takes a longer time for some instances; for these instances, initially the solution accuracy was low, but it was increased when the iteration was increasing; there is a trade-off between the quality of solutions and their computational times. In other words, this study focused to solve single objectives related to optimization solutions of QAP within a reasonable computing time.

7. Conclusion and Discussion

The quadratic assignment problem (QAP) has attracted researchers' attention, and they continue to be interested in developing methods and techniques to find a solution for the problem. Their interest is due to the QAP's practical importance, which can be used in many real-world applications that can be modeled as a QAP. These include the layouts of hospital facilities and campuses. This study focused on proposing a hybrid algorithm based on algorithms after having critical analysis of studies and the conceptual framework.

The purpose of this process is to balance the two main components in metaheuristic approaches: exploitation and exploration. On the other hand, certain issues, such as premature convergence, stagnation, and exploitation mechanism, have not been studied in the discrete differential evolution (DDE) algorithm to solve the QAP. Therefore, we introduced an enhanced algorithm that addressed existing issues. So, we developed a new hybrid algorithm, HDDETS, by combining the DDE and tabu search (TS) algorithms.

From a computational perspective, our algorithm has proven its efficiency by finding the optimal solution within a reasonable computational time. It is found that the HDDETS algorithm is better than both the DDE and TS algorithms where the HDDETS has obtained 42 optimal and best-known solutions from 56, while the DDE and TS algorithms have obtained 15 and 18 optimal and best-known solutions out of 56, respectively. In the application domain of the QAP, the proposed HDDETS algorithm is used in Azadi Hospital in Iraq to find the best distribution of the hospital's rooms. The rooms are modeled as a QAP to reduce the total distance traveled by patients. During this study, several directions emerged that can be considered good seeds for future research. According to this study, the following research directions are promising:

(i) Applying our hybrid metaheuristic algorithm, HDDETS, to multiobjective quadratic assignment problems

(ii) Hybridizing other local search algorithms with the DDE algorithm, such as the simulated annealing algorithm

(iii) Solving other combinatorial optimization problems, such as the scheduling problem and the vehicle routing problem by using our hybrid metaheuristic algorithm, HDDETS

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Jannat, A. A. Khaled, and S. K. Paul, "Optimal solution for multi-objective facility layout problem using genetic algorithm," in *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management*, Dhaka, Bangladesh, January 2010.

[2] M. F. Anjos and M. V. C. Vieira, "Mathematical optimization approaches for facility layout problems: the state-of-the-art and future research directions," *European Journal of Operational Research*, vol. 261, no. 1, pp. 1–16, 2017.

[3] V. Tongur, M. Hacibeyoglu, and E. Ulker, "Solving a big-scaled hospital facility layout problem with meta-heuristics algorithms," *Engineering Science and Technology, an International Journal*, vol. 23, no. 4, pp. 951–959, 2020.

[4] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, 1957.

[5] S. Samanta, D. Philip, and S. Chakraborty, "A quick convergent artificial bee colony algorithm for solving quadratic assignment problems," *Computers & Industrial Engineering*, vol. 137, Article ID 106070, 2019.

[6] A. S. Hameed, B. M. Aboobaider, N. H. Choon, M. L. Mutar, and W. H. Bilal, "Review on the methods to solve combinatorial optimization problems particularly: quadratic assignment model," *International Journal of Engineering & Technology*, vol. 7, pp. 15–20, 2018.

[7] R. K. Bhati and A. Rasool, "Quadratic assignment problem and its relevance to the real world: a survey,," *International Journal of Computer Applications*, vol. 96, no. 9, pp. 42–47, 2014.

[8] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, "Integrating the whale algorithm with Tabu search for quadratic assignment problem: a new approach for locating hospital departments," *Applied Soft Computing*, vol. 73, pp. 530–546, 2018.

[9] L.-S. Nurul Nadia, "Heuristics and metaheuristics approaches for facility layout problems: a survey," *Pertanika Journal Of Scholarly Research Reviews*, vol. 2, no. 3, pp. 62–76, 2016.

[10] A. Misevicius, "An implementation of the iterated tabu search algorithm for the quadratic assignment problem," *OR Spectrum*, vol. 34, no. 3, pp. 665–690, 2012.

[11] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2243, 2018.

[12] ZH. Ahmed, “A new reformulation and an exact algorithm for the quadratic assignment problem,” *Indian Journal of Science and Technology*, vol. 6, no. 4, pp. 4368–4377, 2013.

[13] A. S. Hameed, B. M. Aboobaider, N. H. Choon, M. L. Mutar, and W. H. Bilal, “A comparative study between the branch and cut algorithm and ant colony algorithm to solve the electric meter reader problem in rural areas,” *Opcion*, vol. 34, no. 86, pp. 1525–1539, 2018.

[14] ZH. Ahmed, *Algorithms for the Quadratic Assignment Problem*, LAP LAMBERT Academic Publishing, Latvia, Mauritius, 2019.

[15] M. Abdel-Basset, G. Manogaran, H. Rashad, and A. N. H. Zaied, “A comprehensive review of quadratic assignment problem: variants, hybrids and applications,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–24, 2018.

[16] M. E. Riffi, Y. Saji, and M. Barkatou, “Incorporating a modified uniform crossover and 2-exchange neighborhood mechanism in a discrete bat algorithm to solve the quadratic assignment problem,” *Egyptian Informatics Journal*, vol. 18, no. 3, pp. 221–232, 2017.

[17] Z. H. Ahmed, “A simple genetic algorithm using sequential constructive crossover for the quadratic assignment problem,” *Journal of Scientific & Industrial Research*, vol. 73, pp. 763–766, 2014.

[18] D. T. Connolly, “An improved annealing scheme for the QAP,” *European Journal of Operational Research*, vol. 46, no. 1, pp. 93–100, 1990.

[19] N. Fescioglu-Unver and M. M. Kokar, “Self controlling tabu search algorithm for the quadratic assignment problem,” *Computers & Industrial Engineering*, vol. 60, no. 2, pp. 310–319, 2011.

[20] N. Ç. Demirel and M. D. Toksarı, “Optimization of the quadratic assignment problem using an ant colony algorithm,” *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 427–435, 2006.

[21] Z. Drezner, “A new genetic algorithm for the quadratic assignment problem,” *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 320–330, 2003.

[22] E. Duman, M. Uysal, and A. F. Alkaya, “Migrating Birds Optimization: a new metaheuristic approach and its performance on quadratic assignment problem,” *Information Sciences*, vol. 217, pp. 65–77, 2012.

[23] J.-i. Kushida, K. Oba, A. Hara, and T. Takahama, “Solving quadratic assignment problems by differential evolution,” in *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pp. 639–644, Kobe, Japan, November 2012.

[24] A. W. Mohamed and A. K. Mohamed, “Adaptive guided differential evolution algorithm with novel mutation for numerical optimization,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253–277, 2019.

[25] M. F. Tasgetiren, Q.-K. Pan, P. N. Suganthan, and I. E. Dizbay, “Metaheuristic algorithms for the quadratic assignment problem,” in *Procedings of the 2013 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS)*, pp. 131–137, Singapore, April 2013.

[26] S. Jain, S. Kumar, V. K. Sharma, and H. Sharma, “Improved differential evolution algorithm,” in *Proceedings of the 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pp. 627–632, Dubai, UAE, September 2017.

[27] S. Prabha and R. Yadav, “Differential evolution with biological-based mutation operator,” *Engineering Science and Technology, an International Journal*, vol. 23, no. 2, pp. 253–263, 2020.

[28] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and Y. Deng, “Improving exploration and exploitation via a hyperbolic gravitational search algorithm,” *Knowledge-Based Systems*, vol. 193, Article ID 105404, 2020.

[29] K. Zhang, P. Mu, Y. Zhang, Z. Jin, and Q. Huang, “Enhancing differential evolution algorithm with repulsive behavior,” *Soft Computing*, vol. 24, no. 12, pp. 9279–9305, 2020.

[30] M. Yang, C. Li, Z. Cai, and J. Guan, “Differential evolution with auto-enhanced population diversity,” *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 302–315, 2015.

[31] M. Leon, *Improving Differential Evolution with Adaptive and Local Search Methods*, Mälardalen University, Västerås, Sweden, 2019.

[32] J. A. Sultan, D. A. Matrood, and Z. M. Khaleel, “Artificial bee colony for quadratic assignment problem: a hospital case study,” *Journal of University of Human Development*, vol. 2, no. 3, pp. 502–508, 2016.

[33] C. Sanhueza, F. Jimenez, R. Berretta, and P. Moscato, “PasMoQAP: a parallel asynchronous memetic algorithm for solving the multi-objective quadratic assignment problem,” in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1103–1110, IEEE, San Sebastian, Spain, June 2017.

[34] E. Sonuc, B. Sen, and S. Bayir, “A cooperative GPU-based parallel multistart simulated annealing algorithm for quadratic assignment problem,” *Engineering Science and Technology, an International Journal*, vol. 21, no. 5, pp. 843–849, 2018.

[35] S. C. B. Semlali, M. E. Riffi, and F. Chebihi, “Parallel hybrid chicken swarm optimization for solving the quadratic assignment problem,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 2064–2074, 2019.

[36] A. Sangaiah, M. Suraki, M. Sadeghilalimi, S. Bozorgi, A. Hosseinabadi, and J. Wang, “A new meta-heuristic algorithm for solving the flexible dynamic job-shop problem with parallel machines,” *Symmetry*, vol. 11, no. 2, pp. 165–217, 2019.

[37] A. Acan and A. Ünveren, “A great deluge and tabu search hybrid with two-stage memory support for quadratic assignment problem,” *Applied Soft Computing*, vol. 36, pp. 185–203, 2015.

[38] Z. H. Ahmed and A. Alexandridis, “A hybrid algorithm combining lexisearch and genetic algorithms for the quadratic assignment problem,” *Cogent Engineering*, vol. 5, no. 1, pp. 1423743–1423815, 2018.

[39] M. A. Kaviani, M. Abbasi, B. Rahpeyma, and M. M. Yusefi, “A hybrid Tabu search-simulated annealing method to solve quadratic assignment problem,” *Decision Science Letters*, vol. 3, no. 3, pp. 391–396, 2014.

[40] M. Harris, R. Berretta, M. Inostroza-Ponta, and P. Moscato, “A memetic algorithm for the quadratic assignment problem with parallel local search,” in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 838–845, Sendai, Japan, March 2015.

[41] W. L. Lim, A. Wibowo, M. I. Desa, and H. Haron, “A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem,”

Computational Intelligence and Neuroscience, vol. 2016, Article ID 5803893, , 2016.

[42] T. G. Pradeepmon, R. Sridharan, and V. V. Panicker, "Development of modified discrete particle swarm optimization algorithm for quadratic assignment problems," *International Journal of Industrial Engineering Computations*, vol. 9, pp. 491–508, 2018.

[43] D. M. Tate and A. E. Smith, "A genetic approach to the quadratic assignment problem," *Computers & Operations Research*, vol. 22, no. 1, pp. 73–83, 1995.

[44] G. Zäpfel, R. Braune, and M. Bögl, *Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics*, Springer, Berlin, Germany, 2010.

[45] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "The impact of automated parameter optimization on defect prediction models," *IEEE Transactions on Software Engineering*, vol. 45, no. 7, pp. 683–711, 2019.

[46] W. G. Wolpert, D. H. Wolpert, and Macready, "No free Lunch theorems for optimization," in *Proceedings of the 10th Annual Conference Genetic and Evolutionary Computation*, pp. 811–818, Atlanta, Georgia, July 2008.

[47] T. Dokeroglu, E. Sevinc, and A. Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Applied Soft Computing*, vol. 76, pp. 595–606, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6639714, 10 pages
https://doi.org/10.1155/2021/6639714

Research Article

The Anomaly- and Signature-Based IDS for Network Security Using Hybrid Inference Systems

Sajad Einy[1], **Cemil Oz**[1], **and Yahya Dorostkar Navaei**[2]

[1]*Computer Engineering Department, Sakarya University, Serdivan, Turkey*
[2]*Computer and Information Technology Engineering, Qazvin Islamic Azad University, Qazvin, Iran*

Correspondence should be addressed to Yahya Dorostkar Navaei; y.dorostkar@qiau.ac.ir

Received 30 December 2020; Revised 15 February 2021; Accepted 1 March 2021; Published 13 March 2021

Academic Editor: Ali Asghar Rahmani Hosseinabadi

With the expansion of communication in today's world and the possibility of creating interactions between people through communication networks regardless of the distance dimension, the issue of creating security for the data and information exchanged has received much attention from researchers. Various methods have been proposed for this purpose; one of the most important methods is intrusion detection systems to quickly detect intrusions into the network and inform the manager or responsible people to carry out an operational set to reduce the amount of damage caused by these intruders. The main challenge of the proposed intrusion detection systems is the number of erroneous warning messages generated and the low percentage of accurate detection of intrusions in them. In this research, the Suricata IDS/IPS is deployed along with the NN model for the metaheuristic's manual detection of malicious traffic in the targeted network. For the metaheuristic-based feature selection, the neural network, and the anomaly-based detection, the fuzzy logic is used in this research paper. The latest stable version of Kali Linux 2020.3 is used as an attacking system for web applications and different types of operating systems. The proposed method has achieved 96.111% accuracy for detecting network intrusion.

1. Introduction

In the last decade, the use of different types of networks (such as communication networks, social networks, mobile internet networks, and Internet of Things networks) has been welcomed by people all over the world, and many life affairs (such as buying and selling, medical consulting, business, and education) are done through these networks.

Providing security is one of the most important challenges for all of these networks, and different researchers have proposed different methods (such as firewalls, cryptography, and restricting access to the network) to overcome this challenge. The biggest weakness of all these methods is that they are not able to detect intrusions and attacks inside the network.

To eliminate this weakness, intrusion detection systems have been proposed that can detect internal and external attacks and intrusions. These systems monitor inbound network traffic to detect any abnormal behavior or abuse by users. These systems can play an important role in reducing the damage done by these intruders to the network structure or the data exchanged through this network due to the early detection of intrusions.

Intrusion detection systems can detect intrusions in two ways, including the following:

(a) Signature recognition: data related to the history of operations and transactions which are performed by authorized users and hackers in the networks are used, and patterns for normal behaviors and abnormal behaviors are created. By matching these patterns or signatures with the pattern of behavior of existing users, it is possible to identify unauthorized users or hackers from authorized users. The most important disadvantage of this method is that if the user's signature or pattern is new and there is no similar pattern or signature in the database before, we cannot determine whether it is an authorized or unauthorized user.

(b) Identify anomalies from machine learning methods (such as neural networks and SVM) to create a classification of users based on their behavior in the network (such as the amount of traffic generated, the content of files shared or downloaded, and network usage time). Two classes, normal and abnormal, are created to identify and classify users, based on which the system is trained. One of the advantages of this method is that it is possible to identify hackers who infiltrate the system with new methods.

The whole process performed in intrusion detection systems can be divided into two main parts. These parts are as follows:

(i) Selecting specific features from all the features registered and contained in the collected dataset: because the collected dataset about users' behaviors and interaction with different networks may include different types of features, therefore, using all of these features for identifying patterns or performing analyses is time-consuming and can be made complex or confuse the application of different techniques to detect the pattern of attackers and ordinary users, so to expedite the detection of hackers' abnormal behaviors in different networks, using selection techniques, a subset of the entire set of collected features will be created. For this matter, we select useful and effective features. In other words, in feature selection, we select a subset from the total features in the available datasets, and this subset can not only help reduce the time which is required to analyze these data and identify normal and abnormal behaviors but also greatly reduce the complexity of these data.

(ii) Using two techniques (including classification and anomaly detection) to separate normal users from attackers: all network users can be generally divided into two groups (normal users and attackers). Therefore, because the type and number of classes are known, it is possible to create patterns to identify the type of class of each user in the network environment by using different classification techniques (such as SVM and decision tree) and finally identify normal and abnormal behaviors of users in the network environment (such as creating abnormal traffic in the network).

This article is organized into six sections. In the second part, we will review some related works which are done by other researchers. In the third section, we will explain the proposed method. In the fourth section, we will show the implementation and evaluation results. In the fifth section, we will have more discussion about the proposed method and the results obtained. In the sixth section, we will bring the conclusion.

2. Related Works

Several kinds of research studies have been conducted by authors about the proposed techniques or methods for improving the accuracy of intrusion detection by IDSs. Furthermore, many authors have used feature selection methods for increasing precision in IDSs. In this work, we will be reviewing some of these research studies' goals and results.

Farha and Singh [1] studied about feature selection. They proposed an intelligent hybrid technique using the concept of metaheuristic optimization. The proposed technique combines IWD (intelligent water drops) and ACO (ant colony optimization) for selecting favorite features from the data. The main goal of this paper is to concentrate on reducing the training time and creating an optimal feature subset. In other words, these authors attempted to optimize the process of feature selection from the available data. Finally, these authors applied their proposed technique for the selection of features from KDDCUP 99 data, and the results showed that their proposed technique achieved its aim.

Azam Davahli and Abaei [2] proposed a new model using concepts of GA (genetic algorithm) and mathematical equations of GWO (grey wolf optimizer) for developing SVM- (support vector machine-) based LIDS (lightweight IDS). This new model is called GABGWO. The new model to increase the performance of the LIDS, by applying two new crossover and mutation operators, attempts to find the most relevant traffic features and eliminates irrelevant ones. The performance of GABGWO is evaluated, and the results show that, by using this model, we can choose optimal traffic, decrease computation cost, and obtain high accuracy for LIDS. Furthermore, the results show that the composition of GA and GWO had better performance in comparison with other recent methods and existing FS algorithms.

Al-Yaseen [3] used the support vector machine (SVM) and firefly algorithm (FA) for proposing a new wrapper feature selection method. The aim of this research was removing irrelevant and repetitive features to reduce the dimension of the data and reduce the time of classification and, finally, improving the performance of the IDS. FA was inapplicable in diverse combination problems, and this research was used to produce feature subsets. The SVM model was used for evaluating these feature subsets. The main advantage of the proposed method is the improvement of the ability of FA to produce an optimal feature subset. These authors used the NSL-KDD dataset for the evaluation of their proposed method. Experimental results showed that the proposed method had 78.89% overall accuracy of the IDS. Furthermore, these results showed that the proposed method (FA-SVM) can reduce the number of features, improve the accuracy of identifying intruders, and reduce the number of false alarm rates.

Hadeel Alazzam [4] proposed a new wrapper selection algorithm and a new method for binarizing a continuous pigeon optimizer for the IDS. The new algorithm aims to utilize the selection process by using the pigeon-inspired optimizer. This author compared the new method with the traditional methods for binarizing continuous swarm intelligent algorithms. In this research, a new method of discretization of a continuous algorithm was proposed that was based on the usage of cosine similarity. For evaluating this proposed algorithm, the author used three popular datasets including UNSW-NB15, NLS-KDD, and KDDCUP 99. The results showed that the proposed algorithm can successfully reduce the number of features needed to build the IDS, and it had high TPR, FPR, accuracy, and *F*-score. Furthermore, the proposed algorithm reduced the required time for building the decision.

Yukang Liu [5] proposed a novel feature selection algorithm called MH_SFS (metaheuristic-based sequential forward selection). The main goal of this algorithm was to reduce the dimension of anomaly detection methods. The proposed algorithm can improve the performance of the SFS (sequential forward selection) feature selection algorithm. For decreasing the computational cost in addition to the metaheuristic search process, this author added a filter selection function. The author used the KDDCUP 99 dataset for testing their proposed algorithm and compared the performance of this proposed algorithm with traditional SFS and IFFS algorithms over three anomaly detection models. Results showed that the proposed algorithm had high performance and a smaller number of features selected in comparison with other algorithms.

Calvet et al. [6] studied the researches which are done by other researchers related to the combination of machine learning methods and metaheuristics. They proposed a novel type of hybrid algorithm with the help of the concept of learning heuristics. Constraint Optimization Problem Dynamic Inputs (COPDIs) attempt to solve combinatorial optimization problems. Usually, in these COPDIs, the problem inputs are not fixed; furthermore, they vary predictably. Therefore, the solution is made according to some heuristic-based iterative processes. These authors said that the coordination between the metaheuristic algorithm and the learning mechanism may be required for solving problems that occur due to variations in the input. The results of testing their proposed novel type of hybrid algorithms showed that the learning methods updated the input models iterationaly which were used by the metaheuristic.

Lin et al. [7] studied big data which were created from various IoT applications. They said that, for finding an optimal feature subset from these big data for applying classification methods, we need to have a metaheuristic search algorithm. They proposed a modified version of CSO (cat swarm optimization) and called it MCSO. The main goal of the proposed algorithm was to improve search efficiency within the problem space. Experimental results showed that MCSO, with a decreasing number of features in subfeatures in UCI datasets, improves the classification accuracy in comparison to the original CSO. Furthermore, MCSO increased training time and had higher accuracy in comparison with the original CSO. Therefore, MCSO can be applied to real-time IoT applications.

Yuyang Zhou [8] proposed a new IDS framework which was based on feature selection and ensemble learning techniques. For selecting an optimal subset of features, the author proposed a heuristic algorithm called CFS-BA. This algorithm used the correlation between features for selecting the optimal subset of features. Furthermore, the author combined forest, RF (random forest), and C4.5 by forest PA (penalizing attribute) algorithms. For combining the probability distribution of the base learners for attack detection, voting techniques were used. The algorithm was tested with the help of 3 different IDS datasets (including CIC-IDS-2017, AWID, and NSL-KDD datasets). The results showed that the proposed algorithm had better performance (accuracy: 99.81%) compared with the other approaches.

Opeyemi Osanaiye [9] combined three different filter methods (including ReliefF, chi-squared, and gain ratio) to propose a new feature selection method. Their main aim was reducing the system complexity and increasing the classification accuracy (using decision tree algorithm and J48). The proposed method made a subset with 14 important features out of 41 original features in the NSL-KDD dataset. The performance of the proposed method was evaluated by considering accuracy, detection rate, and false alarm rate. Experimental results showed that the proposed method can reduce effective features and have high classification accuracy and detection rate and less power consumption in comparison with other filtering methods.

Melike Günay [10] proposed the modification of FA-based feature selection. The author improved the performance of traditional FA by using a KNN classifier and adding an extra feature selection step. Four different datasets of IDS were used, various subfeatures were made for each dataset, and the performance of classification methods that are applied to these subsets was compared. The experimental results showed that the proposed FA can decrease the dimension of features, decrease memory usage (approximately 50%), and save the time of computations. Furthermore, the proposed FA improved the accuracy of classification.

To optimize the performance of intrusion detection systems, in [11], a new integrated learning approach has been proposed that is scalable and adaptive in nature. In this approach, a random forest machine learning algorithm was used to build the classified model. The proposed method was implemented with the NSL-KDD 40558 dataset, and its evaluation results showed that the accuracy of intrusion detection by applying this method was 99.91%.

Li et al. [12] proposed an improved krill swarm algorithm based on the linear nearest neighbor lasso step (LNNLS-KH) to solve problems related to the low performance and high number of false positives in intrusion detection systems. In this method, the linear nearest neighbor lasso step optimization was performed, which increased the accuracy of intrusion detection. The results of the implementations performed by these researchers showed that the LNNLS-KH algorithm selected an average of 7 features in the NSL-KDD dataset and 10.2 features in the CICIDS2017 dataset, and features that were not applicable were easily removed. This algorithm also performed well in the optimal fitness iteration curve and convergence speed and showed a false positive rate.

To improve the classification accuracy and speed of the NNGE algorithm and reduce computational resource consumption, Li et al. [12] introduced a method of vertical and horizontal data reduction named as VHDRA (Vertical and Horizontal Intelligent Dataset Reduction Approach). VHDRA provides the following features:

(i) Reducing the properties in the dataset vertically by selecting the most important properties and over-reducing the NNGE rectangles

(ii) Tracking the mode and method of data extraction horizontally using a method called STEM, which is a method for monitoring states and data extraction

(iii) The results of the implementation of the proposed method showed that this method can reduce the number of features in the dataset and increase the accuracy of intrusion detection compared to other methods

Using the Unified Intrusion Anomaly Detection (UIAD) method, Kamarudin et al. [13] presented a new way to develop the function of intrusion detection systems to detect unknown attacks on web servers. The proposed method used three components (preprocessing, statistical analysis, and classification). Implementation of the proposed method using DARPA 1999 and ISCX 2012 datasets showed that it has 95% accuracy in detecting unknown attacks, and the detection rate of false alarms is about 1%.

Maple [14] conducted extensive studies on the use of intrusion detection systems in communication networks used in the Internet of Things (such as WSN, MANET, and CPS). The researcher said that most intrusion detection systems are not quite suitable due to the limitations of the resources used in the Internet of Things. Therefore, we need to develop appropriate intrusion detection systems for the Internet of Things.

Guoquan Li at al. [15] has reviewed some of the most important problems in using intrusion detection systems, creating a large number of false alarms and the accuracy of accurate intrusion detection. In this research, the methods of dealing with this problem have been researched. To this end, they focused on data fusion (DF) techniques to detect network intrusions and examined a set of benchmarks to compare the performance of DF techniques. In this study, the efficiency of many classification techniques (such as RF, C4.5, NN, and SVM) in creating more efficient detection systems was demonstrated.

Khraisat et al. [16] proposed a hybrid HIDS (combining the C5 decision tree classification and a support vector machine (SVM)). The purpose of this intrusion detection system was to overcome the problems of traditional intrusion detection systems in detecting known and unknown threats. They also proposed a framework aimed at identifying known intrusions, increasing the accuracy of intrusion detection, and reducing false alarm rates. In this framework, two methods were used to identify anomalies (to identify unknowns) and to identify signatures (for known threats). The proposed HIDS was evaluated using different datasets, and the evaluation results showed that it had better performance compared to SIDS and AIDS in terms of accuracy of diagnosis and low false alarm rate.

Snehi [17] discussed about different signature-based influence systems and their benefits. The results of their studies showed that a set of signatures and basic patterns show the relative importance of each feature of the intrusion detection system and help system administrators identify cyber attacks and network and computer system threats. Therefore, 80% of intrusions can be easily and quickly identified using signature-based detection methods.

David Mudzingwa [18] tried to study the methods used to create intrusion detection systems and intrusion prevention systems. The researcher focused more on anomaly-based, signature-based, state-of-the-art, and composition-based protocol analyses. The anomaly-based method was involved in detecting new threats without any updates or input for users, but most IDPSs on the market used a combination of several basic methods. This study also provided methods for easy comparison and evaluation of IDPS methods used by IDPS products in the market.

Liu Hua Yeo [19] provided an overview of different types of intrusion detection systems, how they are classified, and different algorithms used to identify unusual activities to operate in intrusion detection systems. The main focus of this research was on anomaly-based and signature-based intrusion detection systems. Then, the researcher tried to compare different methods of penetration techniques and also described different methods and the importance of IDSs in information security.

Singh [20] proposed a hybrid IDS by combining two approaches in one system. The hybrid IDS combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) created using misuse-based IDS Snort, an open-source project, is a system based on anomaly detection. The proposed hybrid intrusion detection system has simulated using DARPA intrusion detection evaluation dataset that is available in MIT Lincoln laboratories repository. The results of this evaluation showed that the proposed system had a high efficiency in identifying intrusions created by creating anomalies and abuses.

3. Proposed Solution for Intrusion Detection Systems

In this research paper, the proposed solution for the network security of any organization against various types of attacks is given. In these attacks, the web application, operating system vulnerabilities, or any software vulnerabilities are considered. Structured Query Language (SQL) injection attack, Cross-Site Scripting (XSS) attack, broken authentication, and any payload for ransomware attacks are mostly focused. For this, the FreeBSD 12 Unix operating system is used as a gateway or firewall along with Suricata as IDS/IPS. The two types of machine learning methods are used for the creation of Suricata signatures to block the malicious traffic on the targeted network. For the metaheuristic-based feature selection, the neural network, and the anomaly-based detection, the fuzzy logic is used in this research paper. The latest stable version of Kali Linux 2020.3 is used as an attacking system for web applications and different types of operating systems. In these client systems, Windows 10/7 and Ubuntu 20.04 are being used. To access web applications, Chrome, Firefox, and Edge browsers are utilized. This lab environment is deployed in VirtualBox for the proof of concept. The basic overview of any organization for its network security is given in Figure 1.

3.1. The Neural Fuzzy Logic Inference System. Here, in the proposed solution, the Suricata IDS/IPS is deployed along with the NN model for the metaheuristic's manual detection of malicious traffic in the targeted network. By utilizing this method for the detection of malicious traffic, the different attacks with higher identification proportion and lower false alert have been recognized. The NN model is one of the widely utilized methods and has been fruitful in tackling various composite reasonable inconveniences and different metaheuristic streamlining algorithms for advancement since firstly, it relies upon a somewhat direct start and is not hard to recognize. Secondly, it does not require exact information, and thirdly, it has the ability to avoid neighborhood optima. Fourthly, it can be utilized in an expansive extent of issues covering contrasting disciplines. Nature-pushed metaheuristic figuring settles upgrade issues by copying natural or physical miracles. They can be accumulated in three main functions: headway-based, material science-based, and swarm-based procedures. Progression-based procedures are impelled by the laws of trademark advancement. This method begins with indiscriminately made people which is categorized by ages. The quality reason for these systems is that the individuals are joined to the groups with same age and the manner of individuals ar changed by time. This empowers the general population to be redesigned in the wealth of the strategy for a very long time. ANN-based IDS exist in two points of view: I) lower revelation exactness, particularly for low-visit attacks, e.g., distant to the neighborhood and client to root; II) more fragile identification steadiness [21]. The fuzzy logic is used for the detection of anomaly-based attacks. The traffic with attachment files or unknown protocols will be processed by the fuzzy logic-based traffic anomaly detection. The detailed internal working of the proposed model is explained in Figure 2.

The patterns or signatures of various types of attacks related to web applications, operating systems (OSs), and any mobile apps will be trained in the NN model. After the processing of the trained data from the NN method, the new rule sets will be created for various types of malicious traffic such as SQL injection attacks, XSS attacks, broken authentication, insecure traffic, old versions of any application or framework, vulnerable OS, and any known vulnerability of mobile applications. The output of the NN model will be input to the Suricata IDS/IPS for preventing attacks on the targeted network. For the more efficient performance of the proposed solution, the open-source blacklist Internet Protocol (IP) address sources are used. The blacklist IP address sources are stored in the database at the frequency of updating these IP addresses after every 4 hours. The program has been used for storing blacklist IP which addresses the custom PHP and MySQL community version.

4. Implementation and Results

The lab has been deployed in a virtual environment for the proof of concept regarding the proposed solution for the IDS/IPS using ML methods. This has been deployed on VirtualBox open-source software to avoid damage to any real server or users. Due to this, the privacy of any web application users or operating system users is protected. FreeBSD 12.1 is deployed as a gateway and firewall for a virtual lab environment. On this, the proposed machine learning method for manual neural networks and anomaly fuzzy logic inference systems are installed. Along with this, the Suricata IDS/IPS, FreeBSD built-in packet filtering (PF), and customized program developed in PHP for blacklist IP address databases from different open-source resources are used. Ubuntu 20.04 is deployed for the Damn Vulnerable Web Application (DVWA) [22] for testing the proposed solution in this research paper. This web application has all the top ten vulnerabilities along with advanced SQL injection attacks. It is depicted in Figure 3.

The Windows 10 client system is also installed for securing it against attacks from the hackers through Suricata IDS/IPS by implementing the ML methods. The hacking machine is also deployed with the installation of the Kali Linux 2020.3 version. This will be used for exploiting OS and web application vulnerabilities.

To launch an attack on the DVWA website and on Windows 10 OS, Kali Linux is used. With the help of Kali Linux, the security signature of Suricata IDS/IPS is created by using ML methods. In this process, two kinds of techniques are used. One is a neural network for well-known attacks such as SQL injection, XSS, broken authentication, and OS system vulnerabilities. The SQL injection attack is launched with the help of the SQLMAP open-source tool in Kali Linux. In the first command, the database name is extracted from the target web application, but due to the Suricata IDS/IPS as a firewall, it is not possible to obtain the database name or enumerate the targeted web application regarding database information as shown in Figure 4. Secondly, the usernames and passwords are extracted from this database of the web application. The result of advanced queries of SQL injection by using –temper or– random-agent has not been included here because it is unable to get required information due to repetition of the same type of result in a paper image of that result.

The XSS attack is launched with the help of the XSSer tool of pen-testing in Kali. Due to the proposed firewall system, this attack is also blocked as an error of 302 or 301 as is highlighted in Figure 5. Because of the failed attack, it

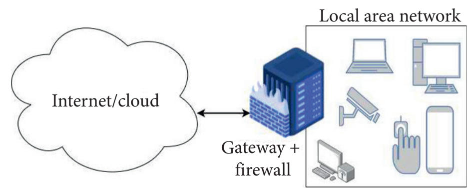


Figure 1: The basic overview of network security.

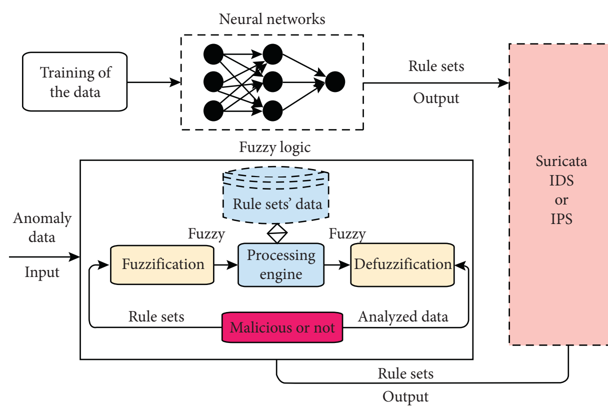


Figure 2: Internal working of the proposed solution for the IDS/IPS.

has generated a few errors also. The Metasploit tool has been used for exploiting the Windows 10 client systems. In this paper, the main target is to exploit the Simple Message Block (SMB) CVE-2018-0749 vulnerability. The Common Vulnerabilities and Exposures (CVE) (SMB) [23] is utilized to look into common vulnerabilities of Windows client systems. The target system is installed in a virtual environment to avoid harm to any real client. The attack launch on the target system is shown in Figure 6. This attack also fails due to the deployed firewall as security with the utilization of ML methods to generate the Suricata signatures.

The core work in this research paper is the implementation of Suricata IDS/IPS for the security of any organization network. This security will be from layer 4 to layer 7 applications. The novelty in this paper is that it is deployed on FreeBSD 12.1, the best Unix operating system for firewalls, which itself has the best packet filter (PF) firewall functionality. This PF will work for network-level filtering of malicious traffic or which administrator wants to block. The known attack's signatures for Suricata are generated with the help of this method of metaheuristic neural network. In this method, the data regarding web application and operating system attacks will be trained into the NN for the signatures, and these will be input into Suricata. Secondly, the fuzzy logic is used for the anomaly-based traffic on the network. In this process, unknown attacks or any attached files will be analyzed by the fuzzy logic systems. After analysis, this system will decide whether the traffic is malicious or legitimate. If it is malicious, the signature for the IDS/IPS will be generated to protect the clients of the targeted organization. The few malicious types of data are highlighted in red color in this raw logs JavaScript Object Notation (JSON) format as shown in Figure 7.

These JSON format logs are converted into XML, and from this format, these are converted into a tabular form. In Table 1, the success of full login or access to other resources at the network is shown.

In Table 2, the data are presented from the Suricata IDS/IPS logs. These logs are related to many types of events that have generated at a firewall in which remote login on Windows client system and web application. In these attacks, there are SMB exploits, SQL injection attack, XSS, brute force attack on web application login.

4.1. Confusion Matrix of the Proposed Hybrid Inference System. The confusion matrix is known as the error matrix for the analysis of statistical data. As in this paper, the hybrid inference system is used for the signature generation and anomaly-based traffic analysis for the Suricata intrusion detection system or intrusion prevention systems. Various types of attacks have been launched in the locally developed lab such as SQLi, XSS, Exploitation of Windows OS vulnerabilities, and distributed denial-of-service (DDoS) attack on the targeted network. These mentioned attacks are prevented by the proposed system for the targeted network. For the evaluation of results, the confusion matrix has been used as defined in Table 3.

The system accuracy of 96.11% has not been achieved due to the false positive ratio of network traffic detection by the Suricata IDS/IPS. As the results of the proposed solution As the results of the proposed solution in this paper, it shows overall good accuracy of detection or prevention of attacks on the targeted network. The limitations of this proposed system can be mentioned as per limited type of datasets. Furthermore, this proposed system is deployed within a

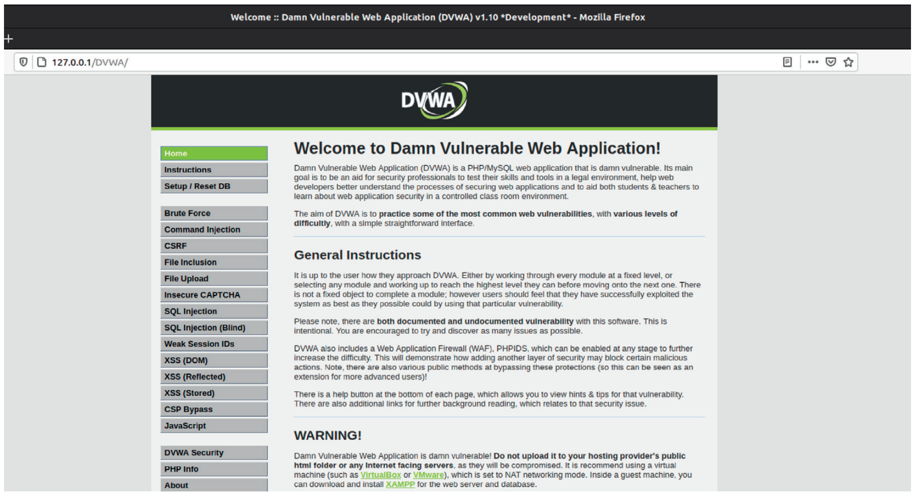


Figure 3: The vulnerable web application deployed in the virtual lab.

Figure 4: The SQL injection attack by using SQLMAP.

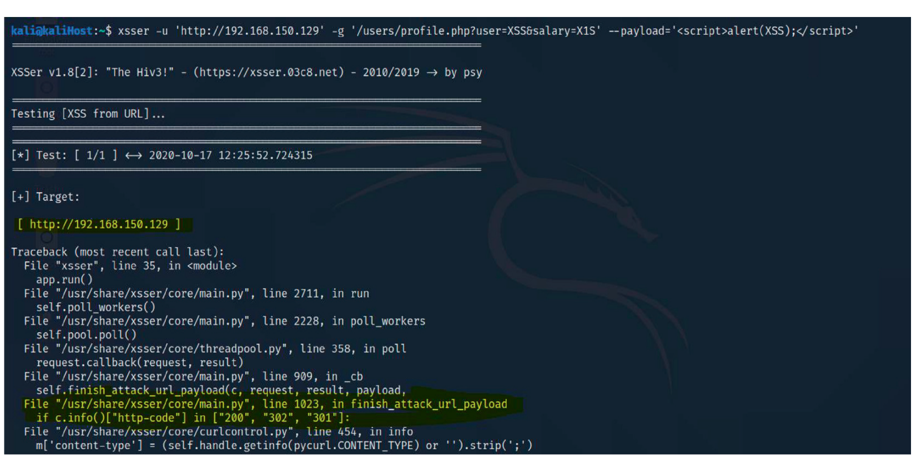


Figure 5: The XSSer tools for the XSS attack on the target web application.

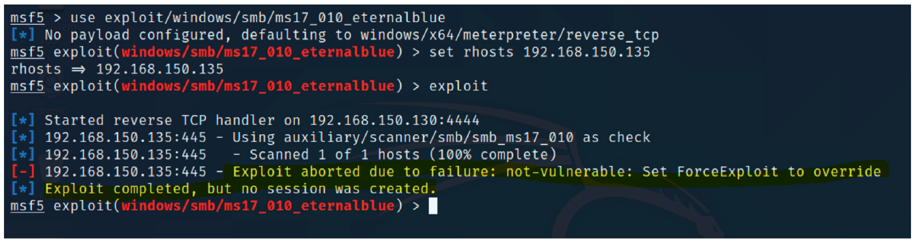


Figure 6: Attack launched on the Windows client.

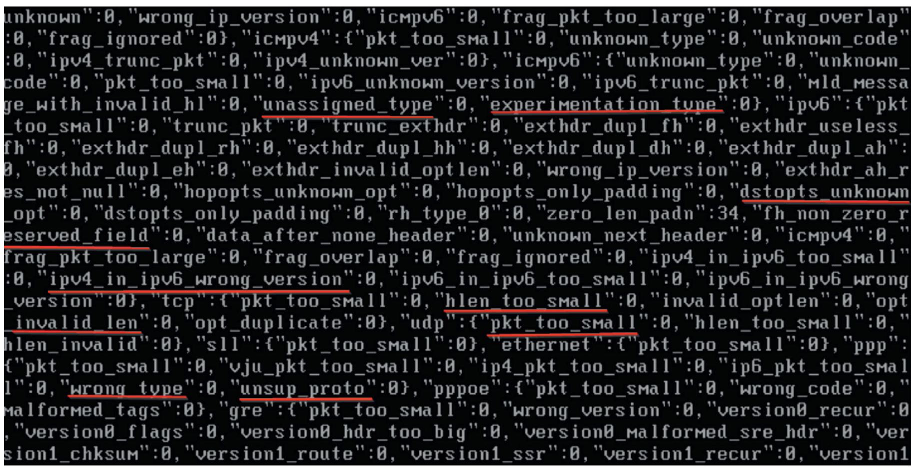


Figure 7: JSON format logs of Suricata IDS/IPS.

Table 1: Successful user login on Windows 10 client.

Action	Date and time	Status	Source IP	Destination IP	Username	Risk level
Logged in	Oct 01, 2020, 1:24:43 AM	User login success	192.168.150.128	192.168.150.135	N/A	5
Logged in	Oct 01, 2020, 1:20:03 AM	User login success	192.168.150.128	192.168.150.135	N/A	5
Logged in	Oct 01, 2020, 1:13:13 AM	User login success	192.168.150.128	192.168.150.135	N/A	5
Logged in	Oct 01, 2020, 1:07:13 AM	User login success	192.168.150.128	192.168.150.129	N/A	5
Logged in	Oct 01, 2020, 1:00:53 AM	User login success	192.168.150.128	192.168.150.129	N/A	5

Table 2: User login fail attempts of Windows client and web application.

Action	Suricata IDS/IPS	Date and time	Status	Source IP	Destination IP	Username or attack type	Risk level
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:53 AM	Failure	89.46.223.240	192.168.150.135	POSTGRES	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	118.107.76.23	192.168.150.135	TEST1	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	192.168.150.128	192.168.150.135	ADMINISTRATOR	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	212.42.214.3	192.168.150.135	LENOVO	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	124.109.54.218	192.168.150.129	SQLi	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	192.168.150.128	192.168.150.129	SQLi	4
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:43 AM	Failure	212.42.214.3	192.168.150.129	XSS	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	192.168.150.128	192.168.150.129	XSS	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	212.42.214.3	192.168.150.129	SQLi	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	112.161.27.203	192.168.150.129	SERVER	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	192.168.150.128	192.168.150.129	ADMINISTRATOR	4
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	212.42.214.3	192.168.150.129	MASTER	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	192.168.150.128	192.168.150.129	ACER	3
Failed to log in	192.168.150.131	Oct 01, 2020, 1:29:34 AM	Failure	89.46.223.240	192.168.150.129	Web applicatiin	3

Table 3: The confusion matrix of the hybrid inference system for the IDS.

	Normal	SQLi	XSS	Windows OS attacks	DDoS	Classification overall	Precision (%)
Normal	500	0	0	0	0	500	100
SQLi	0	900	50	0	50	1000	90
XSS	0	0	925	25	50	1000	92.5
Windows OS attacks	0	0	0	1000	0	1000	100
DDoS	0	0	0	0	1000	1000	100
Truth overall	500	900	975	1025	1100	4500	
User accuracy (%)	100	100	94.872	97.561	90.909		

Overall accuracy = 96.111% is obtained.

virtual system lab created for weak operating systems and applications on which these tests were applied. The run time of the algorithm is within an expected response from the proposed system. For big datasets, the delay may be faced by this system, but it can optimize in future to improve the response time within the required time limits. As per the strategy proposed in the past areas of this paper, the initially produced fuzzy model on the extrema data of each solutions. To accomplish a decent harmony between the framework execution and rule intricacy, distinctive fix sizes and similarity limits are attempted. The subsequent fluffy frameworks utilizing diverse fix sizes are assessed by rule intricacy and the nearby guess mistake. This research has different types of datasets and attack types which were tested on the proposed system. To the best of authors' knowledge, this type of work has not been done before.

5. Conclusions

Nowadays, with the advancement of science and technology and the increase in the level of awareness and knowledge of human beings, as seen day by day, there are widespread attacks on various commercial, communication, information, and other networks. Some of these attacks target important and sensitive data and information contained in these networks, and some other attacks try to generate extra traffic in the communication routes with the resources and servers which are providing different services for users; thus, they block the access of authorized users to those services either making the servers unavailable or disrupting the network performance for several minutes or hours. One way to identify hackers is to use intrusion detection systems (IDSs) that try to detect or predict the possibility of attacks in various ways. One of these ways is to check the network traffic and try to identify the attacks by creating a lot of traffic in the network. In this paper, we considered this type of attack and proposed a new method for deploying Suricata IDS/IPS with the help of the NN model for the meta-heuristic's manual detection of malicious traffic in the targeted network. In the proposed solution, the traffic with attachment files or unknown protocols will be processed by the fuzzy logic as based on detected as anomaly traffic. For implantation of the proposed solution for the IDS/IPS using ML methods, a virtual environment is deployed. The results showed that our proposed method can detect different types of attacks (including in these attacks, SMB exploits, SQL injection attack, XSS, and brute force) on web application login. The overall accuracy of our proposed method is about 96.11% when experimented on the particular dataset.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H. Farha and S. Singh, "Feature selection technique for intrusion detection system based on iwd and aco," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 9, pp. 270–275, 2017.

[2] M. S. Azam Davahli and G. Abaei, "A lightweight Anomaly detection model using SVM for WSNs in IoT through a hybrid feature selection algorithm based on GA and GWO," *Journal of Computing and Security*, vol. 7, no. 1, pp. 63–79, 2020.

[3] W. L. Al-Yaseen, "Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 1–7, 2019.

[4] A. S. Hadeel Alazzam, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Systems With Applications*, vol. 148, pp. 1–14, 2020.

[5] Z. X. Yukang Liu, "A novel meta-heuristic-based sequential forward feature selection approach for anomaly detection systems," in *Proceedings of the 2016 International Conference on Network and Information Systems for Computers*, pp. 218–227, Wuhan, China, April 2016.

[6] L. Calvet, J. D. Armas, D. Masip, and A. A. Juan, "Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs," *Open Mathematics*, vol. 15, no. 1, pp. 261–280, 2017.

[7] K.-C. Lin, Y.-H. Huang, J. C. Hung, and Y.-T. Lin, "Feature selection and parameter optimization of support vector machines based on modified cat swarm optimization," *International Journal of Distributed Sensor Networks*, pp. 1–9, 2014.

[8] G. C. Yuyang Zhou, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, pp. 1–21, 2020.

[9] O. O. Opeyemi Osanaiye, "Feature selection for intrusion detection system IN a cluster-based heterogeneous wireless sensor network," *Facta Universitatis*, vol. 32, no. No 2, pp. 315–330, 2019.

[10] Z. O. Melike Günay, “A modified firefly algorithm-based feature selection method and artificial immune system for intrusion detection,” *Uludağ University Journal of The Faculty of Engineering*, vol. 25, no. 1, pp. 269–287, 2020.
[11] A. Chiche and M. Meshesha, “Towards a scalable and adaptive learning approach for network intrusion detection,” *Journal of Computer Networks and Communications*, vol. 2021, Article ID 8845540, 9 pages, 2021.
[12] X. Li, P. Yi, W. Wei, Y. Jiang, and L. Tian, “LNNLS-KH: a feature selection method for network intrusion detection,” *Security and Communication Networks*, vol. 2021, Article ID 8830431, 22 pages, 2021.
[13] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, “A new unified intrusion anomaly detection in identifying unseen web attacks,” *Security and Communication Networks*, vol. 2017, Article ID 2539034, 18 pages, 2017.
[14] Z. A. Maple, “recent advancements in intrusion detection systems for the Internet of Things,” *Security and Communication Networks*, vol. 2019, Article ID 4301409, 19 pages, 2019.
[15] Z. Y. Guoquan Li, Z. Yan, Y. Fu, and H. Chen, “Data fusion for network intrusion detection: a review,” *Security Measurements of Cyber Networks Issue*, vol. 2018, p. 18, Article ID 821061, 2018.
[16] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, “Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine,” *Electronics*, vol. 9, no. 1, p. 173, 2020.
[17] J. Snehi, “Diverse methods for signature based intrusion detection schemes adopted,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 2, pp. 44–49, 2020.
[18] R. A. David Mudzingwa, “A study of methodologies used in intrusion detection and prevention systems (IDPS),” *Proceedings of IEEE Southeastcon*, 2012.
[19] X. C. Liu Hua Yeo, “Understanding modern intrusion detection systems: a survey,” *Information & security: An International Journal*, vol. 46, no. 2, pp. 155–167, 2020.
[20] S. Singh, “A hybrid intrusion detection system design for computer network security,” *International Journal of Engineering Sciences & Research Technology*, vol. 7, no. 4, pp. 339–343, 2018.
[21] R. Beghdad, “Critical study of neural networks in detecting intrusions,” *Computers & Security*, vol. 27, no. 5, pp. 168–175, 2008.
[22] DVWA. (2020). Damn Vulnerable Web Application. http://www.dvwa.co.uk/.
[23] SMB T M (2020). https://www.cvedetails.com/cve/CVE-2020-0749/.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6697598, 16 pages
https://doi.org/10.1155/2021/6697598

Research Article

A Hybrid Cellular Genetic Algorithm for the Traveling Salesman Problem

Yanlan Deng, Juxia Xiong, and Qiuhong Wang

School of Mathematics and Physics, Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Juxia Xiong; xiongjuxia@gxun.edu.cn

Received 3 December 2020; Revised 22 January 2021; Accepted 28 January 2021; Published 9 February 2021

Academic Editor: Ali Asghar Rahmani Hosseinabadi

The traveling salesman problem (TSP), a typical non-deterministic polynomial (NP) hard problem, has been used in many engineering applications. Genetic algorithms are useful for NP-hard problems, especially the traveling salesman problem. However, it has some issues for solving TSP, including quickly falling into the local optimum and an insufficient optimization precision. To address TSP effectively, this paper proposes a hybrid Cellular Genetic Algorithm with Simulated Annealing (SA) Algorithm (SCGA). Firstly, SCGA is an improved Genetic Algorithm (GA) based on the Cellular Automata (CA). The selection operation in SCGA is performed according to the state of the cell. Secondly, SCGA, combined with SA, introduces an elitist strategy to improve the speed of the convergence. Finally, the proposed algorithm is tested against 13 standard benchmark instances from the TSPLIB to confirm the performance of the three cellular automata rules. The experimental results show that, in most instances, the results obtained by SCGA using rule 2 are better and more stable than the results of using rule 1 and rule 3. At the same time, we compared the experimental results with GA, SA, and Cellular Genetic Algorithm (CGA) to verify the performance of SCGA. The comparison results show that the distance obtained by the proposed algorithm is shortened by a mean of 7% compared with the other three algorithms, which is closer to the theoretical optimal value and has good robustness.

1. Introduction

Traveling salesman problem (TSP) is one of the most common combinatorial optimization problems, and it has been widely used in this field. At the same time, many problems in the field of combinatorial optimization can be formulated as special TSP instances [1]. For example, resource optimization in the shortest path and shop scheduling problems are the most frequent TSP applications [2–4]. In Word Sense Disambiguation (WSD), problems can also be solved by describing WSD as a TSP variant [5]. In the global positioning system (GPS), TSP can also be used as the basis for finding route problems [6]. It can be seen that the application of TSP is extensive; however, TSP is an NP-hart problem, and the number of feasible solutions increases significantly with the increase of nodes, which brings great difficulty to the solution [7]. If the exact algorithm is used to solve TSP, it will take a long time, so the feasibility of using the exact algorithm to solve TSP is very low. On the other hand, although using an approximate solution to solve TSP problems cannot guarantee an optimal solution, it can reach a satisfactory solution in a very short time [8]. Thus, with the development of heuristic algorithms [9–14], many experts and scholars have begun to apply heuristic algorithms to solve TSP, which provides new ideas for solving TSP [15–21].

Genetic algorithm (GA) is an evolutionary algorithm proposed by Professor John H. Holland of the University of Michigan and his students from the late 1960s to the early 1970s [22]. GA uses biological evolution and genetic principles to imitate the generation and evolution process of all life and intelligence by drawing on the basis of biology. It is a bionic algorithm in a macroscopic sense and is applied to many optimization fields [23–26]. The coding method commonly used in genetic algorithms is binary coding, but most search space sequences do not correspond to feasible travel when binary coding is used. We can use ordinal coding and a partially mapped crossover (PMX) as the cross operator so that all the solutions obtained will be feasible solutions [27]. However, there are some disadvantages in using traditional GA to solve TSP, such as poor searchability and low convergence accuracy. On this basis, the improved GA also developed vigorously. Liu et al. [28] proposed an improved GA with decision cut-off algebra to solve TSP, which improves the convergence of the GA. Besides, Yu et al. [29] proposed an improved GA for solving TSP by a greedy algorithm, which used the greedy algorithm to generate the initial population and reduce the number of iterations. At the same time, the hybrid algorithm has also received the attention of many experts and scholars. For instance, Wang et al. [30] combined GA and SA, and proposed an improved simulated annealing genetic algorithm for solving TSP, improving the local search capabilities. Also, Zhang et al. [31] combined the GA and Firefly algorithm proposing a Firefly GA algorithm to solve TSP and prevent the algorithm from falling into local optimality. Moreover, Tao et al. [32] combined the GA and Ant Colony (AC) algorithm proposing a dynamic Ant Colony genetic algorithm to solve TSP, which improved the reinsertion of offspring to improve stability.

Cellular genetic algorithm (CGA) is a hybrid of cellular automata and genetic algorithms. It has been applied to various decision-making problems [33]. However, the application of the CGA to TSP is very limited at present. To study the application of the CGA in TSP and improve GA's optimization performance, this paper proposes a hybrid Cellular Genetic Algorithm with SA (SCGA). We improve TSP optimization performance by combining the cellular genetic algorithm and the simulated annealing algorithm. The contributions of this paper are as follows:

(i) The paper successfully applies the cellular genetic algorithm to solve TSP and proposed a hybrid Cellular Genetic Algorithm combined with a simulated annealing algorithm. The experimental results show that the hybrid cellular genetic algorithm's optimization performance is better than GA, SA, and CGA.

(ii) The optimization performance of the three cellular automata rules in the SCGA is examined through experiments and concluded that cellular automata rules with a moderate total number of living cells are beneficial to algorithm optimization.

The rest of this article is arranged as follows. Section 2 is a brief introduction to the preliminary knowledge that will be used, including the traveling salesman problem model, the basic content of genetic algorithm (GA), simulated annealing (SA) algorithm, and cellular automata (CA). In Section 3, the proposed algorithm and its steps are explained in detail. Section 4 reveals the simulation results and discussions. Section 5 summarizes the full text and proposes the next step.

2. Preliminaries

2.1. Traveling Salesman Problem (TSP). TSP is a commonly used test benchmark for optimization methods, which can be simply described as follows.

Given the coordinates of n cities, starting from any city, passing through the remaining cities only once, and finally returning to the starting city, there are $n!$ routes. We need to choose the shortest route from these $n!$ routes. In the symmetric Euclidean TSP, city A''s distance to city B is equal to the distance from city B to city A. According to the description of TSP, the following mathematical model can be established [29]:

$$R = (N_1, N_2, \ldots, N_n), \quad \min f(R), \tag{1}$$

$$f(R) = \sum_{i=1}^{n-1} d(N_i, N_{i+1}) + d(N_n, N_1), \tag{2}$$

$$d(N_i, N_{i+1}) = d(N_{i+1}, N_i). \tag{3}$$

We use the city number as the city mark. The solution $(1, 2, 3, \ldots, n)$ represents a trip starting from city 1, along city 2, city 3, until city n, and finally back to city 1. $(N_1, N_2, \ldots, N_n)$ is a permutation of $(1, 2, 3, \ldots, n)$, and $d(N_i, N_{i+1})$ represents the Euclidean distance between city N_i and city N_{i+1}.

2.2. Genetic Algorithm (GA). Genetic algorithm is one of the universal algorithms in the optimization field. The essence of GA is an efficient, parallel, and global search method. It can automatically acquire and accumulate search knowledge during the search process and adaptively control the search process to find the optimal solution [27, 34]. The traditional GA is mainly composed of population and population size, coding method, selection strategy, genetic operator, and stopping criterion [35–37]. The population is composed of chromosomes or individuals, and each individual corresponds to a feasible solution to the problem. The population size refers to the number of individuals in the population, which is a subset of the problem's feasible region. Generally speaking, the larger the number of individuals in the population is, the more feasible solutions it contains, and the better the optimal solution will be. However, as the number of individuals in the population increases, the algorithm's computation time will also increase, which affects the performance of the algorithm. Therefore, it is necessary to select an appropriate population size. The coding method is also called the gene expression method. The chromosomes in the population are composed of genes. Correctly coding the chromosomes is the basic and important work of GA. According to the mathematical model of TSP, we can choose the ordinal encoding as the encoding method, and each chromosome is composed of a nonrepetitive ordinal number $(1, 2, 3, \ldots, n)$. The selection strategy is the process of selecting parents from the current population. We can use the commonly used roulette as the selection operator. The pseudocode of the roulette operator is shown in Algorithm 1, and the steps of the roulette operator are as follows:

(i) *Step 1.* Calculate the number NP of individuals to be selected according to the selection probability P_c

(ii) *Step 2.* Calculate the fitness value of each individual according to the following equation:

$$f_k = \frac{1}{\sum_{i=1}^{n-1} d(N_i, N_{i+1}) + d(N_n, N_1)}. \quad (4)$$

(iii) *Step 3.* Calculate the cumulative probability of an individual P_k:

$$P_k = \frac{\sum_{i=1}^{k} f_i}{\sum_{j=1}^{n} f_j}. \quad (5)$$

(iv) *Step 4.* Randomly generate the number rand in [0, 1]; if $P_{i-1} <$ r and $< P_i$, select the i^{th} individual. Repeat NP times to get the selected NP individuals

Genetic operators mainly include crossover and mutation, which is the most important part of the GA. We choose partially mapped crossover (PMX) and reverse order mutation to hide the infeasible solutions in the execution process. Reverse order mutation refers to reversing the order of the numbers between two tangent points. Therefore, the numbers will not change, and there will be no duplicate numbers. Figure1 is a schematic diagram of PMX, and the steps of PMX are as follows:

Step 1. Randomly generate two cut points.

Step 2. Exchange the part between two individual tangent points.

Step 3. Determine the mapping relationship.

Step 4. Restore the legitimacy of the individual according to the mapping relationship.

The stopping criterion generally uses the maximum number of iterations, and the algorithm stops when the maximum number of iterations is reached. Algorithm 2 shows the pseudocode of the GA for solving TSP.

2.3. Simulated Annealing (SA) Algorithm. In 1953, Metropolis et al. [38] first proposed the simulated annealing algorithm (SA). In 1983, Kirkpatrick et al. [39] successfully used SA in combinatorial optimization problems, especially large-scale problems. SA is a general random search algorithm, which is an extension of the local search algorithm. It is derived from the simulation of the annealing process in thermodynamics [40]. Because modern SA can effectively solve NP complexity problems, avoid falling into local optima, and overcome initial value dependence, it has been widely used in engineering, production scheduling, control engineering, machine learning, and other fields [41–43]. SA mainly includes state expression, neighborhood definition, thermal equilibrium, and cooling control [39]. State expression is to describe the system's energy state in a mathematical form, and one state corresponds to one solution of the problem. State expression is the main work of SA, which directly determines the neighborhood structure and size. A reasonable state expression method can reduce computational complexity and improve algorithm performance. Consistent with genetic algorithms, nonrepeated ordinal numbers are used to represent solutions. The neighborhood definition's starting point is to ensure that the neighborhood's solutions are distributed as much as possible in the entire solution space. The nature of the problem usually determines the definition of the neighborhood. Consistent with the mutation operator in GA, we use reverse order to form neighbor solutions. To replace the neighbor solution with the original solution is determined by the Metropolis criterion. Algorithm 3 shows the pseudocode of the Metropolis criterion. The thermal equilibrium is equivalent to the isothermal process in the physical annealing, which is the internal cycle process in SA. To ensure the equilibrium state, the number of internal cycles should be large enough, leading to an increase in computation time. The number of internal cycles is the length of the Markov chain L. Cooling control is the outer cycle process of SA. Unlike GA, SA uses a temperature drop to control the iteration of the algorithm. The algorithm stops when the temperature drops to the end temperature. The pseudocode of SA for solving TSP is shown in Algorithm 4, and the steps of SA for solving TSP are as follows:

Step 1. Set initial parameters (including initial temperature, Markov chain length, cooling rate, and termination temperature.), and randomly generate an initial solution.

Step 2. Using the reverse order method to generate a neighbor solution.

Step 3. Move the neighbor according to Metropolis criterion.

Step 4. Judge whether the thermal equilibrium is reached; if the thermal equilibrium is not reached, go to Step 2.

Step 5. Lower the temperature. If the temperature reaches the end temperature, the algorithm will stop; otherwise, go to Step 2.

2.4. Cellular Automata (CA). In 1948, Stanislaw M.Ulam and von Neumann proposed cellular automata (CA). CA is a simplified mathematical model for describing complex phenomena in nature [44]. CA reflects the law of self-replication of life through relatively simple rules. The realization of self-replication of life on the computer has brought a new revolution in computing and simulation fields. As a discrete model of a complex system, CA is an important experimental method for studying dynamic interaction and spatiotemporal evolution [45–47]. CA is composed of four parts: cell, cellular space, cellular neighbor' type, and cellular automata rule [44].

The choice of cellular neighbor's type and cellular automata rule is a crucial part that affects CA's performance. Here, the Moore neighborhood is utilized. The Moore neighborhood consists of a central cell (the one which is to be updated) and its eight geographical neighbors, north, west, south, east, north-east, north-west, south-west, and south-east, that is a total of nine cells [48]. Each central cell can be alive ($s^t = 1$) or dead ($s^t = 0$). If permutations and combinations are used to describe neighbors' states, many unnecessary duplicate

Roulette operator
Begin
Input: population, selection probability P_c
Calculate NP, f_k, P_k, $(k = 1, 2, \ldots, n)$
for $i = 1$ to NP
if rand $< P_i$ and rand $> P_{i-1}$
select the i^{th} individual
End if
End for
Output: the selected individual.
End

Algorithm 1: The pseudocode of the roulette operator.

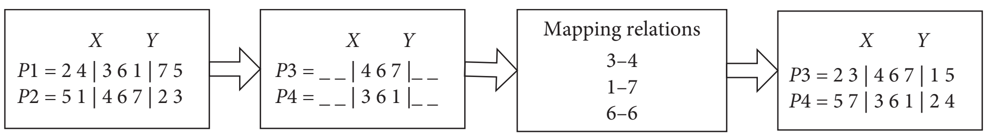


Figure 1: Schematic diagram of PMX.

Genetic algorithm (GA) for solving TSP
Begin
Input: city coordinates: $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$. selection probability P_s, crossover probability P_c, mutation probability P_m, the maximum number of iterations (Maxiter)
Initialization: using ordinal coding to generate the initial population, number of iteration $k = 1$
while $k \leq$ Maxiter
Selection
Crossover
Mutation
New population
Save the optimal result in the iteration
$k = k + 1$
End while
Output: the optimal result: $R = \{N_1, N_2, \ldots, N_n\}$, $f(R)$
End

Algorithm 2: The pseudocode of genetic algorithm (GA) for solving TSP.

Metropolis criterion
Begin
Input: city coordinates: $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$. Solution S_1 and neighbor solution S_2, current temperature T
Calculate $f(S_1)$ and $f(S_2)$ by equation (2)
$\Delta f = f(S_2) - f(S_1)$, $P = \exp(-\Delta f / T)$
if $\Delta f < 0$ or $P >$ rand
$S_1 = S_2$
End if
Output: S_1
End

Algorithm 3: The pseudocode of the Metropolis criterion.

data will be generated. Therefore, there is no need to enumerate the states and positions of specific neighbors; just calculate the total number of cellular states in neighbors [49]. The sum of neighbors' states is K; so, K can be any value between 0 and 8. Cellular automata rules [50] are as follows:

```
Simulated annealing (SA) algorithm for solving TSP
Begin
  Input: city coordinates: X = {x_1, x_2, ..., x_n}, Y = {y_1, y_2, ..., y_n}. Initial temperature T_0, Markov chain length L, cooling rate R,
and termination temperature T_t
  Initialization: using ordinal coding to generate the initial solution S, current temperature T = T_0
  while T ≤ T_t
    for i = 1 to L
      Generate a neighbor solution S_1
      Metropolis criterion
    End for
    Save the optimal solution in the iteration
  T = T* R
  End while
  Output: R = {N_1, N_2, ..., N_n}, f (R)
End
```

Algorithm 4: The pseudocode of simulated annealing (SA) algorithm for solving TSP.

$$\begin{aligned}
&\text{Rule 1:} \begin{cases} s^t = 1, & s^{t+1} = \begin{cases} 1, & K = 2,3, \\ 0, & K \neq 2,3, \end{cases} \\ s^t = 0, & s^{t+1} = \begin{cases} 1, & K = 3, \\ 0, & K \neq 3, \end{cases} \end{cases} \\
&\text{Rule 2:} \begin{cases} s^t = 1, & s^{t+1} = \begin{cases} 1, & K = 1,2,3,4, \\ 0, & K \neq 1,2,3,4, \end{cases} \\ s^t = 0, & s^{t+1} = \begin{cases} 1, & K = 4,5,6,7, \\ 0, & K \neq 4,5,6,7, \end{cases} \end{cases} \\
&\text{Rule 3:} \begin{cases} s^t = 1, & s^{t+1} = \begin{cases} 1, & K = 2,4,6,8, \\ 0, & K \neq 2,4,6,8, \end{cases} \\ s^t = 0, & s^{t+1} = \begin{cases} 1, & K = 1,3,5,7, \\ 0, & K \neq 1,3,5,7. \end{cases} \end{cases}
\end{aligned} \tag{6}$$

Using different cellular automata rules, the total number of living cells in the cellular space is different, and the optimization performance will also be affected.

3. A Hybrid Cellular Genetic Algorithm (SCGA)

This section describes the main contents of SCGA in detail, including the dynamic evolution of cellular space, genetic operations, elitist strategy, and the steps of SCGA.

3.1. Dynamic Evolution of Cellular Space. A state matrix in the cellular space is composed of each cell state in the cellular space. In the iteration, each cell in the cellular space is used as the central cell, and the state of the central cell is updated using the cellular automata rules to form a new state matrix. The pseudocode of the dynamic evolution of the cellular space is shown in Algorithm 5. Given a 10×10 cellular space arbitrarily, each cell state is randomly generated to form an initial state matrix (an n-th order square matrix N composed of 0 and 1), as shown in Figure 2. The cellular automata rule determines each cell's updated state, for example, if the cell in the third row and the third column is the central cell. The central cell state is 1, and there are four cells in the neighboring cells with states equal to 1. According to rule 1, the central cell update state will be 0, while the central cell updated state will be 1, according to rule 2 or rule 3. It can be seen that the updated cellular state changes according to different cellular automata rules. Because dead cells do not perform genetic operations, we only consider the number of living cells. Figure 3 shows that when different cellular automata rules are selected, the total number of living cells will differ. Therefore, if rule 1 is chosen, the total number of living cells will eventually remain at about 10%, while if rule 2 is selected, the total number of living cells will eventually remain at about 38%. On the other hand, if rule 3 is selected, the total number of living cells will eventually remain in a range of 50%. The total number of living cells in the cellular space will affect the optimization performance of the algorithm. Aiming at the TSP, we will compare the optimal solutions obtained by applying these three rules in the hybrid cellular genetic algorithm through experiments.

3.2. Genetic Operations. SCGA puts the individual as a cell into the cellular space, selects the living cell, and forms the parents with the best cell among neighbors. Then, the parents perform crossover and mutation operations. If the cellular state is dead, no crossover and mutation operations are performed. Algorithm 6 shows the pseudocode for the genetic operations.

Assuming that the number of cities is 10 and the number of populations is 100, the corresponding cellular space is a 2D grid of 10×10 units. The algorithm works as follows:

(i) Randomly generate 100 individuals to form the initial population and associate the individuals with the cells one-to-one

(ii) If Figure 2 is used as the state matrix of the cellular space at the initial moment, the individual is represented by the letter. The number after the letter represents the state of the cell, as shown in Figure 4

```
Dynamic evolution of cellular space
Begin
  Input: An n-th order square matrix N composed of 0 and 1, three cellular automata rules
  for i = 1 to n^2
    Calculate the sum K of 8 neighbors' states
    if N (i) = 1
      if K = 2 or K = 3 # rule 1
      if K = 1 or K = 2 or K = 3 or K = 4 # rule 2
      if K = 2 or K = 4 or K = 6 or K = 8 # rule 3
       N (i) = 1
      else
       N (i) = 0
      End if
    else
      if K = 3 # rule 1
      if K = 4 or K = 5 or K = 6 or K = 7 # rule 2
      if K = 1 or K = 3 or K = 5 or K = 7 # rule 3
       N (i) = 1
      else
       N (i) = 0
      End if
    End if
  End for
  Output: New square matrix N
End
```

Algorithm 5: The pseudocode of the dynamic evolution of the cellular space.

0	1	1	0	1	1	1	1	0	0
1	1	0	1	0	0	0	0	1	0
0	1	1	0	1	0	0	0	1	0
0	0	0	1	0	1	0	1	0	0
1	0	1	1	1	1	0	0	1	1
0	0	1	1	0	0	0	0	1	0
0	0	1	1	1	0	0	1	0	0
1	1	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	1	1	1
0	0	0	1	0	0	1	0	1	0

Figure 2: State matrix.

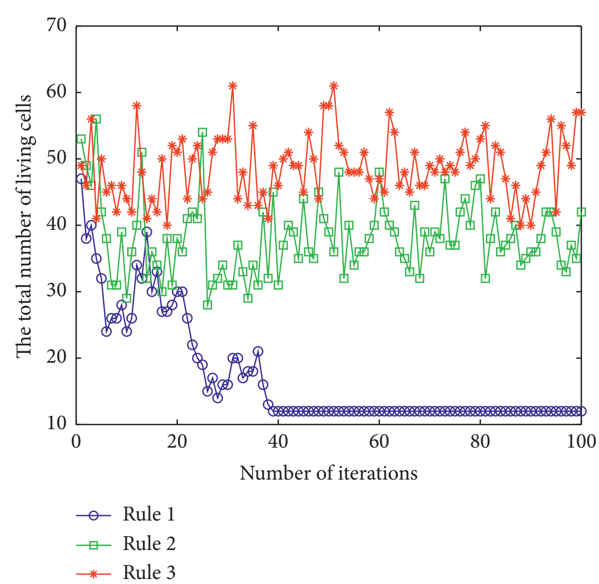


Figure 3: The change of the total number of living cells.

(iii) Take the individual V in the third row and second column as the central cell

(iv) In the current iteration, the cellular state is living (state 1)

(v) Tables 1 and 2 show that the individual with the shortest distance among neighbors is AE, so choose AE and V and perform crossover and mutation operations as parents:

Firstly, the parents can get the offspring individuals O1 and O2 after PMX.

Perform the reverse mutation on O1 and O2 to get the individuals O3 and O4, as shown in Table 3. The new individuals obtained after genetic manipulation are O3 and O4; the best progeny is O3, and O3 is better than V; so O3 replaces V to complete the genetic operation.

3.3. Elitist Strategy. The living central cell and the best neighbor get offspring after crossover and mutation operations. If the distance between the optimal progeny and central cell is not compared, and the optimal progeny directly replaces the central cell, the inferior solution may enter the population. If the central cell's distance is better than the optimal progeny distance, and the central cell is not replaced, it may fall into a local optimum. In order to solve

```
Genetic operations
Begin
  Input: An n-th order square matrix N composed of 0 and 1, cellular space,
  city coordinates: X = {x_1, x_2, ..., x_n}, Y = {y_1, y_2, ..., y_n}
  Calculate the total distance f_i(R) of each individual in the cellular space by Equation (2)
  for i = 1 to n^2
    if N(i) = 1
      Select the i^th individual and the individual with smallest total distance among the 8 neighbors
      Crossover
      Mutation
      New the i^th individual
    End if
  End for
  Output: New cellular space
End
```

Algorithm 6: The pseudocode for genetic operations.

A 0	B 1	C 1	D 0	E 1	F 1	G 1	H 1	I 0	J 0
K 1	L 1	M 0	N 1	O 0	P 0	Q 0	R 0	S 1	T 0
U 0	V 1	W 1	X 0	Y 1	Z 0	AA 0	AB 0	AC 1	AD 0
AE 0	AF 0	AG 0	AH 1	AI 0	AJ 1	AK 0	AL 1	AM 0	AN 0
AO 1	AP 0	AQ 1	AR 1	AS 1	AT 1	AU 0	AV 0	AW 1	AX 1
AY 0	AZ 0	BA 1	BB 1	BC 0	BD 0	BE 0	BF 0	BG 1	BH 0
BI 0	BJ 0	BK 1	BL 1	BM 1	BN 0	BO 0	BP 1	BQ 0	BR 0
BS 1	BT 1	BU 0	BV 0	BW 0	BX 1	BY 0	BZ 0	CA 1	CB 0
CC 0	CD 1	CE 0	CF 0	CG 0	CH 1	CI 0	CJ 1	CK 1	CL 1
CM 0	CN 0	CO 0	CP 1	CQ 0	CR 0	CS 1	CT 0	CU 1	CV 0

Figure 4: Cell and state matrix.

Table 1: Central cell and neighbors.

Cell	Route	Distance	Cell	Route	Distance	Cell	Route	Distance
K	3 4 6 9 7 1 8 10 2 5	316.0711	*U*	6 2 4 8 9 3 1 5 7 10	344.9062	AE	7 8 1 9 3 5 10 4 2 6	244.1511
L	8 9 4 6 10 5 2 7 1 3	262.5791	*V*	4 3 2 7 10 6 8 1 5 9	290.6597	AF	2 3 8 4 6 10 9 7 5 1	267.6944
M	4 8 12 7 3 9 10 6 5	260.2487	*W*	2 4 8 5 7 3 6 1 9 10	299.5755	AG	1 10 4 8 6 5 7 2 9 3	288.2223

Table 2: City coordinates.

Number	1	2	3	4	5	6	7	8	9	10
x	37	49	52	20	40	21	17	31	52	51
y	52	49	64	26	30	47	63	62	33	21

Table 3: Individuals in crossover and mutation process.

Cell	Route	Distance	Cell	Route	Distance
V	4 3 2 7 10 6 8 1 5 9	290.6597	O2	9 3 2 7 10 6 8 4 1 5	296.2696
AE	7 8 1 9 3 5 10 4 2 6	244.1511	O3	1 8 4 9 3 5 10 2 6 7	**258.7903**
O1	4 8 1 9 3 5 10 2 6 7	264.5371	O4	9 4 8 6 10 7 2 3 1 5	286.1712

the above problems, SCGA sets an elite individual retention strategy after the crossover and mutation operations. When the optimal progeny's obtained distance after the crossover and mutation operation is less than the distance of the center cell, the center cell is directly replaced with the optimal progeny. Otherwise, after performing the simulated annealing operation on the optimal progeny, new offspring are obtained, and the central cell is replaced with the new offspring. Algorithm 7 shows the pseudocode of the elitist strategy. Through the elitist strategy, the living central cell can be optimized as much as possible to avoid falling into the local optimum.

3.4. Process of SCGA. SCGA, as explained before, has three main contents: the dynamic evolution of cellular space, genetic operations, and elitist strategy. In this section, we will introduce the main steps of applying SCGA to solve TSP. Algorithm 8 shows the pseudocode of the proposed hybrid Cellular Genetic Algorithm with SA (SCGA) and its required steps for the TSP solution.

(i) *Step 1.* Population initialization:

 (a) Initialize the following parameters:
 (i) Crossover probability P_c,
 (ii) Mutation probability P_m,
 (iii) Cooling rate r,
 (iv) Initial temperature T_0,
 (v) Population size Num,
 (vi) The size of cellular space Num. C,
 (vii) Maximum iteration times Maxiter,
 (viii) Markov chain length L,
 (ix) Termination temperature T_t;
 (b) Randomly generate the initial population as cellular space and state matrix (an n-th order square matrix N composed of 0 and 1).

(ii) *Step 2.* Calculate the city distance matrix.

(iii) *Step 3.* Calculate the total distance of individuals in the population.

(iii) *Step 4.* Take each cell as the central cell in turn.

 (c) If the central cell state is 1, then it is the central cell and the best cell from neighbors to form the parents. Otherwise, skip to Step 7.

(v) *Step 5.* Perform crossover and mutation operations on the parents to get the offspring.

(vi) *Step 6.* If the distance of the optimal progeny is longer than the distance of the central cell, the simulated annealing algorithm is used to optimize the optimal progeny and replace the central cell after optimization. Otherwise, the optimal progeny will directly replace the central cell.

(vii) *Step 7.* Update the state matrix according to the cellular automata rule.

(viii) *Step 8.* Calculate the distance of individuals in the current population and store the best individuals.

(ix) *Step 9.* Judge the stop condition.

 (d) Stop when the maximum number of iterations is reached. Otherwise, skip to Step 4.

4. Experimental Results and Discussion

Here, instances in the TSPLIB test library are selected, and GA, SA, CGA, and SCGA (cellular automata rules 1–3) are used to solve these instances. All of the proposed algorithms are implemented using MATLAB R2018b using a laptop equipped with Windows 10 Ultimate 64 bit operating system, processor Intel Core i7-10510U CPU, 12 GB of RAM, and 1.80 GHz processor speed.

4.1. Experiments Settings. 13 instances from TSPLIB are selected for simulation experiments. To avoid the influence of randomness, the results of the experiments on each algorithm for each instance were taken independently over 10 repeated times. The result is expressed by the route distance obtained from the experiment. The basic parameters of SCGA include crossover probability $P_c = 0.9$, mutation probability $P_m = 0.05$, cooling rate $r = 0.9$, and initial temperature $T_0 = 100$. The other parameters are shown in Table 4, where Num, Num. C, Maxiter, L, and T_t represent population size, the size of cellular space, maximum iteration times, Markov chain length, and termination temperature values which are identified. The basic parameters of GA are initialized as the population size Num = 100, generation gap GGAP = 0.9, and maximum iteration times Maxiter = 10000. The basic parameters of SA are also initialized as Markov chain length $L = 500$ and termination temperature Tend = 0.001. The basic parameters of CGA are assigned initial values as mutation probability $P_m = 0.5$ and maximum iteration times Maxiter = 10000. The other parameters that are not mentioned are the same as those of SCGA.

4.2. Comparison of Cellular Automata Rules. In order to test the optimization of the three cellular automata rules, the experiment is performed using rule 1, rule 2, and rule 3 in SCGA using the same parameters applied in the previous experiments. Figure 5 shows the optimal paths of eil51, pr107, bier127, and kroa200. Table 5 shows the best solutions, the worst solutions, the mean value, and the standard deviation of 10 repeated experiments. The boldface in the table represents the optimal value. As can be seen, in the instances of att48, berlin52, chn31, oliver30, pr107, and pr152, the best solutions obtained using rule 1, rule 2, and rule 3 are equal. In the instances of bier127, ch130, and kroa100, the best solutions obtained using rule 2 and rule 3 are also equal, and both are better than the best solutions obtained using rule 1. In the instances of eil51 and eil76, the best solutions obtained using rule 2 are better than the best solutions obtained using rule 1 and rule 3. Only in the instance of pr136, the best solutions obtained by using rule 1 and rule 3 are better than the best solutions obtained using rule 2. In the instance of kroa200, the best solution obtained using rule 3 is better than the best solution obtained using rule 2. In most instances, the worst solutions obtained using rule 2 are better than the worst solutions obtained using rule

```
Elitist strategy
Begin
  Input: Central cell S_1 and new individual S_2
  city coordinates: X = {x_1, x_2, ..., x_n}, Y = {y_1, y_2, ..., y_n}
  Calculate f(S_1) and f(S_2) by Equation (2)
  if f(S_1) > f(S_2)
  S_1 = S_2
  else
    Optimize S_2 using simulated annealing algorithm
    S_1 = S_2
  End if
  Output: S_1
End
```

Algorithm 7: The pseudocode of elitist strategy.

```
The hybrid cellular genetic algorithm (SCGA) for solving TSP
 Begin
   Input: city coordinates: X = {x_1, x_2, ..., x_n}, Y = {y_1, y_2, ..., y_n}. P_c, P_m, r, T_0, Num,
   Num. C, Maxiter, L, T_t
   Initialization: using ordinal coding to generate randomly the initial population as cellular space, generate randomly an n-th order
 square matrix N composed of 0 and 1 as state matrix, number of iteration k = 1
   while k ≤ Maxiter
     for i = 1 to n^2
       if N(i) = 1
         Genetic operations
         Elitist strategy
       End if
       Save the optimal result in the iteration
     End for
     k = k + 1
   End while
   Output: the optimal result: R = {N_1, N_2, ..., N_n}, f(R)
 End
```

Algorithm 8: The pseudocode of the hybrid cellular genetic algorithm (SCGA) for solving TSP.

Table 4: The parameters of instances.

Instance	Num	Num.C	Maxiter	L	T_t
att48	400	20×20	50	100	0.01
berlin52	400	20×20	50	100	0.01
bier127	1600	40×40	100	500	0.001
ch130	1600	40×40	100	500	0.001
chn31	400	20×20	50	100	0.01
eil51	400	20×20	50	100	0.01
eil76	900	30×30	100	500	0.01
kroa100	400	20×20	50	100	0.01
kroa200	1600	40×40	100	500	0.001
oliver30	400	20×20	50	100	0.01
pr107	1600	40×40	50	500	0.001
pr136	1600	40×40	100	500	0.001
pr152	1600	40×40	100	500	0.001

1 and rule 3. Only in the instances of kroa100 and pr136, the worst solutions obtained using rule 1 and rule 3 are better than the worst solutions obtained using rule 2. At the same time, in 10 repeated experiments, the mean value and the standard deviation computed using rule 2 are better than the mean value and the standard deviation computing using rule

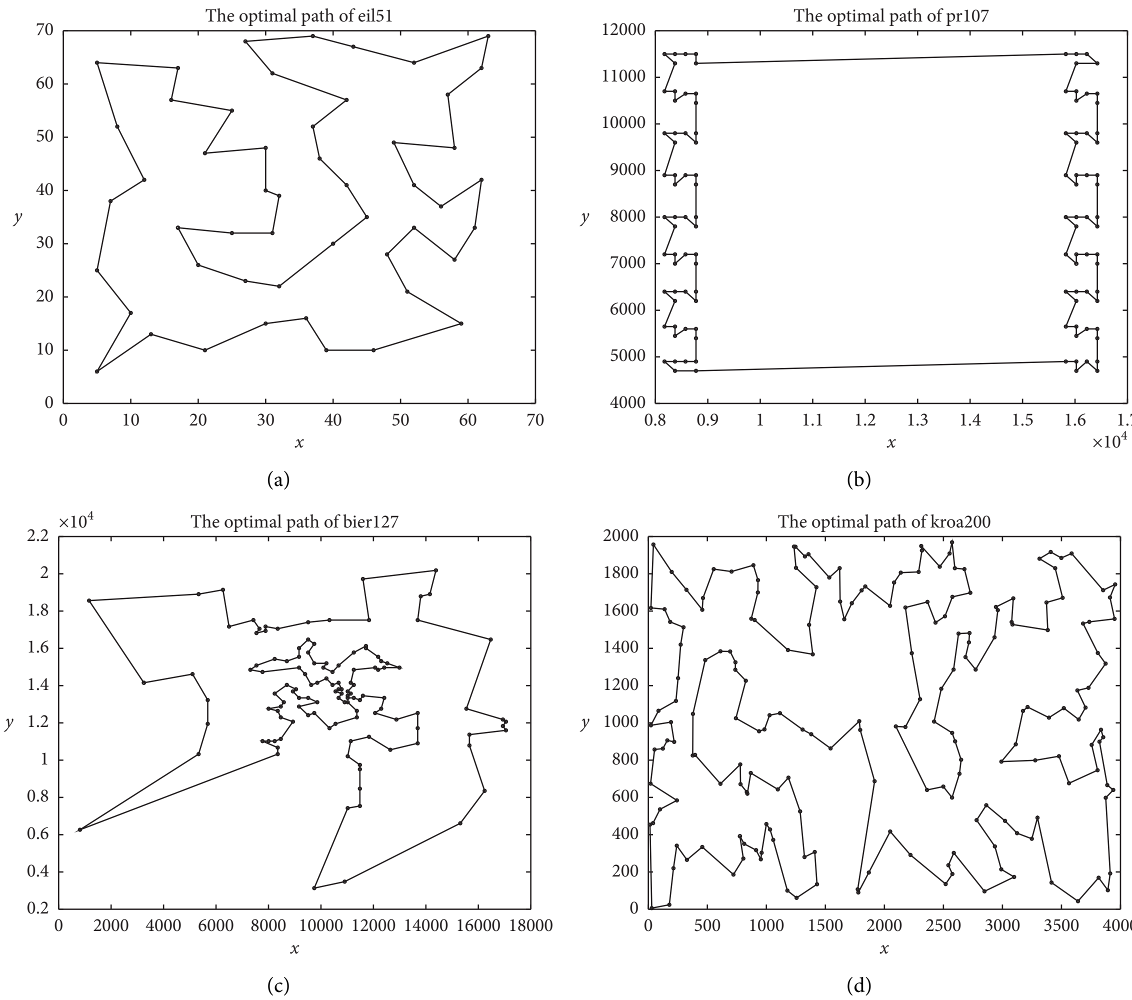


Figure 5: The optimal paths: (a) eil51; (b) pr107; (c) bier127; (d) kroa200.

Table 5: Comparison of cellular automata rules.

Instance	Opt	Rule	Best	Worst	Mean	StD
		Rule 1	**33523.7085**	33555.2765	33530.0221	12.6272
att48	33522	Rule 2	**33523.7085**	**33523.7085**	**33523.7085**	**0.0000**
		Rule 3	**33523.7085**	33555.2765	33526.8653	9.4704
		Rule 1	**7544.3659**	**7544.3659**	**7544.3659**	**0.0000**
berlin52	7452	Rule 2	**7544.3659**	**7544.3659**	**7544.3659**	**0.0000**
		Rule 3	**7544.3659**	**7544.3659**	**7544.3659**	**0.0000**
		Rule 1	118822.0753	118822.0753	118822.0753	**0.0000**
bier127	118282	Rule 2	**118293.5238**	**118293.5238**	**118293.5238**	**0.0000**
		Rule 3	**118293.5238**	**118293.5238**	**118293.5238**	**0.0000**
		Rule 1	6225.4433	6225.4433	6225.4433	**0.0000**
ch130	15377	Rule 2	**6183.0251**	**6183.9732**	**6183.4043**	0.4645
		Rule 3	**6183.0251**	6205.6532	6189.9083	10.3112
		Rule 1	**15377.7113**	**15377.7113**	**15377.7113**	**0.0000**
chn31	15377	Rule 2	**15377.7113**	**15377.7113**	**15377.7113**	**0.0000**
		Rule 3	**15377.7113**	**15377.7113**	**15377.7113**	**0.0000**
		Rule 1	430.4455	436.7443	433.2549	1.9062
eil51	426	Rule 2	**428.8718**	**432.1911**	**429.9723**	**1.1064**
		Rule 3	430.4455	434.1932	432.0974	1.1403

Table 5: Continued.

Instance	Opt	Rule	Best	Worst	Mean	StD
		Rule 1	549.4996	552.0326	550.6944	**0.6931**
eil76	538	Rule 2	**547.1706**	**551.4043**	**548.1847**	1.4609
		Rule 3	547.3471	552.0326	549.5251	1.4484
		Rule 1	21320.9570	21767.5565	21513.1620	142.7024
kroa100	21282	Rule 2	**21285.4432**	21362.9795	**21309.4379**	**27.5103**
		Rule 3	**21285.4432**	**21359.8649**	21315.2179	28.7554
		Rule 1	29834.3635	29865.9464	29840.6801	12.6332
kroa200	29368	Rule 2	29588.2340	**29599.4047**	29596.0535	**5.1191**
		Rule 3	**29533.0620**	**29599.4047**	**29577.9152**	22.7067
		Rule 1	**423.7406**	**423.7406**	**423.7406**	**0.0000**
oliver30	420	Rule 2	**423.7406**	**423.7406**	**423.7406**	**0.0000**
		Rule 3	**423.7406**	**423.7406**	**423.7406**	**0.0000**
		Rule 1	**44301.6837**	**44301.6837**	**44301.6837**	**0.0000**
pr107	44303	Rule 2	**44301.6837**	**44301.6837**	**44301.6837**	**0.0000**
		Rule 3	**44301.6837**	**44301.6837**	**44301.6837**	**0.0000**
		Rule 1	**97102.3170**	97138.1891	97105.9042	10.7616
pr136	96772	Rule 2	97160.0055	97181.1904	97164.2425	**8.4740**
		Rule 3	96795.4030	**97102.3170**	**96856.7858**	122.7656
		Rule 1	**73683.6406**	**73683.6406**	**73683.6406**	**0.0000**
pr152	73682	Rule 2	**73683.6406**	**73683.6406**	**73683.6406**	**0.0000**
		Rule 3	**73683.6406**	**73683.6406**	**73683.6406**	**0.0000**

1 and rule 3. It can be seen that, in most instances, a better and more stable optimal solution can be obtained by using rule 2 in SCGA. In other words, the optimization performance of cellular automata rules with a moderate total number of living cells is giving better results than other rules.

4.3. Comparison of Other Algorithms. In order to verify the optimization of SCGA, SCGA, CGA, GA, and SA are applied to the instances. The experimental results are shown in Tables 6–8.

As shown in Table 6, based on the instances of kroa200, bier127, pr136, and ch130, the distance of the SCGA best solutions is shorter than the distance of the GA best solutions by 9.73%, 8.64%, 8.25%, and 5.80%, respectively. Besides, based on the instances of kroa200, pr136, bier127, eil76, and ch130, the distance of the SCGA best solutions is shorter than the distance of the SA best solutions by 15.79%, 8.86%, 6.69%, 8.25%, 5.14%, and 5.02%, respectively. In the instances of kroa200 and pr136, the distance of the best solutions obtained by using SCGA is shorter than the distance of the best solutions obtained by using CGA by 7.42% and 5.43%, respectively. The SCGA best solutions are better than the GA best solutions in these 13 instances. Only in the instances of oliver30 and chn31, the distance of the SCGA best solutions is equal to the CGA best solutions' distance. Besides, in the instance of oliver30, the SCGA best solution's distance is equal to the distance of the SA best solution. Moreover, the SCGA worst solutions are better than the GA, SA, and CGA worst solutions in these 13 instances.

It is shown in Table 7, based on the instances of kroa200, pr136, bier127, kroa100, ch130, berlin52, ch130, and pr107, that the SCGA solution mean distance is shorter than the GA mean distance solutions by 26.30%, 12.16%, 11.07%, 10.15%, 8.98%, 8.90%, and 8.71%, respectively. In the instances of kroa200, pr136, pr152, bier127, ch130, and kroa100, the SCGA mean distance is shorter than the SA distance mean value by 17.92%, 10.61%, 9.12%, 8.87%, 8.86%, and 8.03%, respectively. In the instances of kroa200 and pr136, the distance of the SCGA mean value is shorter than the CGA mean distance by 9.60% and 6.88%, respectively. At the same time, the standard deviations of the SCGA are the smallest.

This paper uses SCGA to solve these instances and accounts for four decimal places. So, there will be a slight error when comparing with the known optimal solution. In Table 8, the calculation equations for the best solution error and the mean value error are as follows:

$$\text{Best.Err} = \frac{\text{Best} - \text{Opt}}{\text{Opt}},$$
$$\text{Mean.Err} = \frac{\text{Mean} - \text{Opt}}{\text{Opt}}. \quad (7)$$

It is shown in Table 8, in the instances of chn31, pr107, and pr152, that the SCGA best solution errors are about 0%. Only in the instances ch130 and eil76, the SCGA best solution errors are greater than 1%.

From Tables 6–8, we conclude the following.

The optimal solution, the worst solution, the mean value, and the standard deviation obtained by solving the TSP instances using our proposed algorithm (SCGA) are better than GA, SA, and CGA. Using SCGA to solve the TSP instances gives the closest values to the theoretical optimal value. Figure 6 shows the box plots of 10 independents runs of instances kroa100, pr152, att48, and eil76. The box plots of SCGA always locate the lowest position in four instances. Furthermore, these box plots of SCGA are close to the straight lines, which indicate that SCGA has the smallest

Table 6: The best solution and the worst solution.

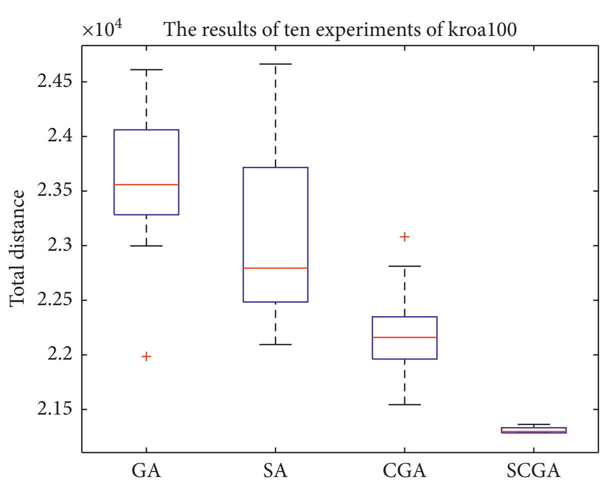
Instance	Opt	Metric	GA	SA	CGA	SCGA
att48	33522	Best	34149.3743	34013.2819	33961.1106	**33523.7085**
		Worst	36386.8892	36208.3069	34609.5047	**33555.2765**
berlin52	7542	Best	7899.0083	7684.5829	7812.5665	**7544.3659**
		Worst	8538.6311	8240.4600	8252.4864	**7544.3659**
bier127	118282	Best	128515.2927	126213.1835	123577.8803	**118293.5238**
		Worst	135685.9343	132724.0594	127274.5241	**118822.0753**
ch130	6110	Best	6541.8980	6493.4978	6350.0191	**6183.0251**
		Worst	6997.0840	6901.1848	6559.8327	**6225.4433**
chn31	15377	Best	15487.2062	15450.5079	15377.7113	**15377.7113**
		Worst	16265.9926	16097.4187	15891.3431	**15377.7113**
eil51	426	Best	437.6323	439.5980	438.5469	**428.8718**
		Worst	464.2338	458.5167	459.2966	**436.7443**
eil76	538	Best	570.9642	575.2977	562.0046	**547.1706**
		Worst	608.9891	619.4997	584.2497	**552.0326**
kroa100	21282	Best	21985.3351	22094.4038	21544.0953	**21285.4432**
		Worst	24611.4693	24663.04523	23080.9158	**21767.5565**
kroa200	29368	Best	32405.9486	34196.6537	31724.8527	**29533.0620**
		Worst	41893.2866	36233.3604	33755.3014	**29865.9464**
oliver30	420	Best	424.6918	423.7406	423.7406	**423.7406**
		Worst	476.1180	425.1044	429.5888	**423.7406**
pr107	44303	Best	45983.4339	45161.7422	44579.1881	**44301.6837**
		Worst	50060.8782	49392.8367	47261.2359	**44301.6837**
pr136	96772	Best	104784.6839	105372.1700	102047.9935	**96795.4030**
		Worst	114883.2612	110519.6460	106732.0575	**97181.1904**
pr152	73682	Best	75801.2620	76992.7749	74888.9388	**73683.6406**
		Worst	81017.3352	83021.7641	78170.3976	**73683.6406**

Table 7: The mean value and standard deviation.

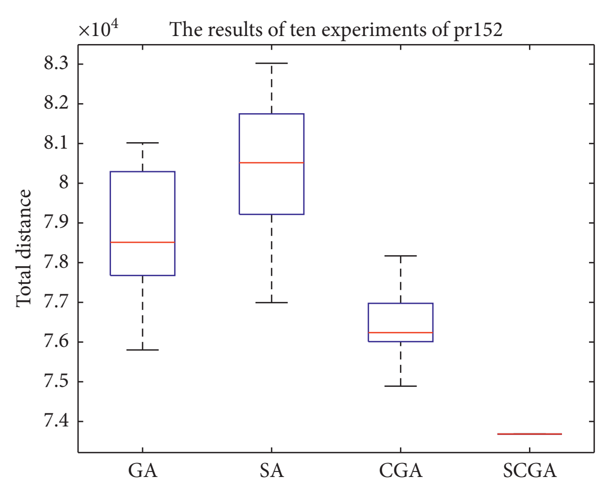
Instance	Opt	Metric	GA	SA	CGA	SCGA
att48	33522	Mean	35198.2428	34873.5741	34341.4737	**33526.8653**
		StD	775.3089	742.4642	218.0112	**9.4704**
berlin52	7542	Mean	8221.6318	7964.2032	7967.7532	**7544.3659**
		StD	175.0246	166.5235	108.1287	**0.0000**
bier127	118282	Mean	131581.2788	128983.7461	125262.8687	**118469.7076**
		StD	2448.7082	2325.3037	1235.5069	**249.1616**
ch130	6110	Mean	6751.4105	6749.0506	6467.4983	**6199.5853**
		StD	147.3786	125.4857	70.3231	**19.4134**
chn31	15377	Mean	15781.2528	15697.6860	15568.1492	**15377.7113**
		StD	256.7062	205.2499	171.7538	**0.0000**
eil51	426	Mean	454.4414	446.8963	443.9726	**431.7748**
		StD	9.1465	6.2678	5.5769	**1.9750**
eil76	538	Mean	590.0867	589.7360	570.9942	**549.4681**
		StD	13.1262	11.3760	6.3239	**1.6194**
kroa100	21282	Mean	23548.7465	23095.5248	22203.6352	**21379.2726**
		StD	674.8707	803.5809	451.3302	**127.6113**
kroa200	29368	Mean	37475.0045	34987.5192	32519.8526	**29671.5496**
		StD	2563.0760	589.8787	536.6872	**120.7940**
oliver30	420	Mean	443.1219	424.0513	425.4695	**423.7406**
		StD	19.0721	0.5317	1.4590	**0.0000**
pr107	44303	Mean	48158.4584	47656.8289	45861.5128	**44301.6837**
		StD	1505.6024	1540.8192	833.1544	**0.0000**
pr136	96772	Mean	108845.5545	107333.6988	103717.2423	**97042.3108**
		StD	2690.4693	1450.6517	1462.0469	**151.2063**
pr152	73682	Mean	78641.4797	80403.5388	76394.6319	**73683.6406**
		StD	1537.2679	1712.5745	907.1630	**0.0000**

Table 8: The best solution error and the mean value error.

Instance	Metric	GA (%)	SA (%)	CGA (%)	SCGA (%)
att48	Best.Err	1.87	8.01	1.31	**0.01**
	Mean.Err	5.00	4.03	2.44	**0.01**
berlin52	Best.Err	4.73	9.26	3.59	**0.03**
	Mean.Err	9.01	5.60	5.65	**0.03**
bier127	Best.Err	8.65	12.21	4.48	**0.01**
	Mean.Err	11.24	9.05	5.90	**0.16**
ch130	Best.Err	7.07	12.95	3.93	**1.20**
	Mean.Err	10.50	10.46	5.85	**1.47**
chn31	Best.Err	0.72	4.69	0.00	**0.00**
	Mean.Err	2.63	2.09	1.24	**0.00**
eil51	Best.Err	2.73	7.63	2.95	**0.67**
	Mean.Err	6.68	4.91	4.22	**1.36**
eil76	Best.Err	6.13	15.15	4.46	**1.70**
	Mean.Err	9.68	9.62	6.13	**2.13**
kroa100	Best.Err	3.30	15.89	1.23	**0.02**
	Mean.Err	10.65	8.52	4.33	**0.46**
kroa200	Best.Err	10.34	23.38	8.03	**0.56**
	Mean.Err	27.60	19.13	10.73	**1.03**
oliver30	Best.Err	1.12	1.22	0.89	**0.89**
	Mean.Err	5.51	0.96	1.30	**0.89**
pr107	Best.Err	3.79	11.49	0.62	**0.00**
	Mean.Err	8.70	7.57	3.52	**0.00**
pr136	Best.Err	8.28	14.21	5.45	**0.02**
	Mean.Err	12.48	10.91	7.18	**0.28**
pr152	Best.Err	2.88	12.68	1.64	**0.00**
	Mean.Err	6.73	9.12	3.68	**0.00**

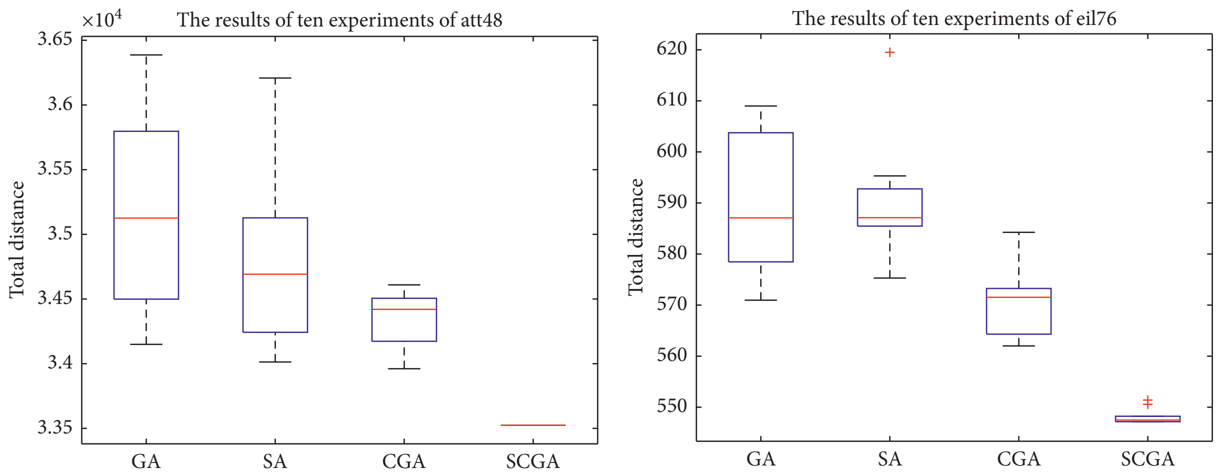


(a)



(b)

Figure 6: Continued.



(c) (d)

Figure 6: Box plots for comparing 10-runs results of GA, SA, CGA, and SCGA: (a) kroa100; (b) pr152; (c) att48; (d) eil76.

degree of the difference between the best value, the mean value, the worst value, and the median value. As mentioned before, SCGA significantly outperforms GA, SA, and CGA.

5. Conclusions

In order to solve the TSP effectively, a hybrid Cellular Genetic Algorithm with SA (SCGA) is proposed. The algorithm combines the principles of GA, CA, and SA to improve the optimization GA's performance. Experimental results show that SCGA is superior to GA, SA, and CGA in solving the TSP. It proves that SCGA has high optimization performance and robustness when solving the TSP.

We studied the optimization performance of three cellular automata rules for the TSP solution. In addition to the three cellular automata rules studied in this paper, other cellular automata rules may have higher optimization performance. We combined the simulated annealing algorithm and the cellular genetic algorithm to enhance the TSP solution's optimization performance. In addition to the simulated annealing algorithm, the cellular genetic algorithm was combined with other algorithms and effectively enhanced the TSP solution's optimization performance. Common traveling salesman problems include symmetric traveling salesman problem, asymmetric traveling salesman problem, multiperson traveling salesman problem, and multitarget traveling salesman problem. This paper studied only the symmetric traveling salesman problem based on the Euclidean distance. The optimization performance of the asymmetric traveling salesman problem and other traveling salesman problems has not been verified. In addition, it is not clear how to apply the hybrid cellular genetic algorithm to other optimal combination problems, such as knapsack problems, production scheduling problems, and clustering. These issues are worthy of our further discussion, and they will be the focus of our future work.

Data Availability

Datasets used to support the findings of this study are available from http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/XML-TSPLIB/instances/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 61772006), the Special Fund for Bagui Scholars of Guangxi (Grant no. 2017), the Science and Technology Major Project of Guangxi (Grant no. AA17204096), the General Program of Guangxi Natural Science Foundation (Grant no. 2019GXNSFAA185033), and the Talent Introduction Project of Guangxi University for Nationalities (Grant no. 2020KJQD05). And, the authors would like to express their gratitude to EditSprings (https://www.editsprings.com/) for the expert linguistic services provided.

References

[1] G. Gutin and A. P. Punnen, *The Traveling Salesman Problem and its Variations*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[2] J. E. Beasley and N. Christofides, "An algorithm for the resource constrained shortest path problem," *Networks*, vol. 19, no. 4, pp. 379–394, 1989.

[3] R. D. Franceschi, M. Fischetti, and P. Toth, "A new ILP-based refinement heuristic for vehicle routing problems," *Mathematical Programming*, vol. 105, no. 2-3, pp. 471–499, 2006.

[4] M. Widmer and A. Hertz, "A new heuristic method for the flow shop sequencing problem," *European Journal of Operational Research*, vol. 41, no. 2, pp. 186–193, 1989.

[5] K. H. Nguyen and C. Y. Ock, "Word sense disambiguation as a traveling salesman problem," *Artificial Intelligence Review*, vol. 40, no. 64, pp. 405–427, 2013.

[6] C. Papalitsas, P. Karakostas, T. Andronikos et al., "Combinatorial GVNS (general variable neighborhood search) optimization for dynamic garbage collection," *Algorithms*, vol. 11, no. 38, 2018.

[7] C. H. Papadimitriou, "The euclidean travelling salesman problem is NP-complete," *Theoretical Computer Science*, vol. 4, no. 3, pp. 237–244, 1977.

[8] G. Laporte, "The traveling salesman problem: an overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.

[9] Y. J. Song, D. Q. Wu, W. Deng et al., "Multi-population parallel co-evolutionary differential evolution for parameter optimization," *Energy Conversion and Management*, vol. 228, p. 2021.

[10] Y. J. Song, D. Q. Wu, A. W. Mohamed et al., "Enhanced success history adaptive DE for parameter optimization of photovoltaic models," *Complexity*, vol. 2021, Article ID 6660115, 14 pages, 2021.

[11] W. Deng, J. J. Xu, X. Z. Gao et al., "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[12] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, vol. 100, Article ID 106724, 2021.

[13] W. Deng, J. Xu, H. Zhao, and Y. Song, "A novel gate resource allocation method using improved PSO-based QEA," *IEEE Transactions on Intelligent Transportation Systems*, p. 1, 2020.

[14] W. Deng, J. Xu, Y. Song, and H. Zhao, "An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application," *International Journal of Bio-Inspired Computation*, vol. 16, no. 3, pp. 158–170, 2020.

[15] L. Babel, "New heuristic algorithms for the dubins traveling salesman problem," *Journal of Heuristics*, vol. 26, no. 4, pp. 503–530, 2020.

[16] K. Yang, X. You, S. Liu, and H. Pan, "A novel ant colony optimization based on game for traveling salesman problem," *Applied Intelligence*, vol. 50, no. 12, pp. 4529–4542, 2020.

[17] Y. Dong and Z. Huang, "An improved noise quantum annealing method for TSP," *International Journal of Theoretical Physics*, vol. 50, no. 12, 2020.

[18] J. W. Ohlmann and B. W. Thomas, "A compressed-annealing heuristic for the traveling salesman problem with time windows," *Informs Journal on Computing*, vol. 19, no. 1, pp. 80–90, 2017.

[19] Y. Zhou, Q. Luo, H. Chen, A. He, and J. Wu, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, pp. 1227–1236, 2015.

[20] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1743–1756, 2017.

[21] G. Pan, K. Li, A. Ouyang, and K. Li, "Hybrid immune algorithm based on greedy algorithm and delete-cross operator for solving TSP," *Soft Computing*, vol. 20, no. 2, pp. 555–566, 2016.

[22] J. H. Holland and H. John, "Genetic algorithms and the optimal allocation of trials," *Siam Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.

[23] L. Yao, W. A. Sethares, and D. C. Kammer, "Sensor placement for on-orbit modal identification via a genetic algorithm," *AIAA. Journal*, vol. 31, no. 10, pp. 1922–1928, 2012.

[24] K. Shinichi, T. Daisuke, A. Masanori et al., "Dynamic modeling of genetic networks using genetic algorithm and s-system," *Bioinformatics*, vol. 5, pp. 643–650, 2003.

[25] R. Martínez, O. Castillo, and L. T. Aguilar, "Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms," *Information Sciences*, vol. 179, no. 13, pp. 2158–2174, 2009.

[26] P. Ahmadi and I. Dincer, "Exergoenvironmental analysis and optimization of a cogeneration plant system using multimodal genetic algorithm (mga)," *Energy*, vol. 35, no. 12, pp. 5161–5172, 2010.

[27] J.-Y. Potvin, "Genetic algorithms for the traveling salesman problem," *Annals of Operations Research*, vol. 63, no. 3, pp. 337–370, 1996.

[28] H. H. Liu, C. Cui, and J. Chen, "An improved genetic algorithm for solving travel salesman problem," *Transactions of Beijing Institute of Technology*, vol. 33, no. 4, pp. 390–393, 2013.

[29] Y. Y. Yu, Y. Chen, and T. Y. Li, "Improved genetic algorithm for solving TSP," *Control and Decision*, vol. 29, no. 8, pp. 1483–1488, 2014.

[30] Y. N. Wang and H. W. Ge, "Improved simulated annealing genetic algorithm for solving TSP problem," *Computer Engineering and Applications*, vol. 46, no. 5, pp. 44–47, 2010.

[31] L. Y. Zhang, Y. Gao, and T. Fei, "Firefly genetic algorithm for traveling salesman problem," *Computer Engineering and Design*, vol. 40, no. 7, pp. 1939–1944, 2019.

[32] L. H. Tao, Z. N. Ma, P. T. Shi et al., "Dynamic ant colony genetic algorithm based on TSP," *Machinery Design & Manufacture*, vol. 12, pp. 147–149+154, 2019.

[33] X. Li, J. Wu, and X. Li, "Cellular genetic algorithms," in *Theory of Practical Cellular Automaton,* Springer, Singapore, 2018.

[34] M. Srinivas and L. M. Patnaik, "Genetic algorithms: a survey," *Computer*, vol. 27, no. 6, pp. 17–26, 2002.

[35] L. Feng and Y. Qi-wen, "Improved genetic operator for genetic algorithm," *Journal of Zhejiang University-SCIENCE A*, vol. 3, no. 4, pp. 431–434, 2002.

[36] A. S. Wu and I. Garibay, "The proportional genetic algorithm: gene expression in a genetic algorithm," *Genetic Programming and Evolvable Machines*, vol. 3, pp. 157–192, 2002.

[37] P. Karthikeyan, S. Baskar, and A. Alphones, "Improved genetic algorithm using different genetic operator combinations (GOCs) for multicast routing in ad hoc networks," *Soft Computing*, vol. 17, no. 9, pp. 1563–1572, 2013.

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4596, pp. 671–680, 1983.

[40] N. Leite, F. Melício, and A. C. Rosa, "A fast simulated annealing algorithm for the examination timetabling problem," *Expert Systems with Applications*, vol. 122, pp. 137–151, 2019.

[41] W.-H. Chen and B. Srivastava, "Simulated annealing procedures for forming machine cells in group technology," *European Journal of Operational Research*, vol. 75, no. 1, pp. 100–111, 1994.

[42] E. Aarts and J. Korst, "Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial

optimization and neural computing," *Siam Review*, vol. 12, no. 2, p. 323, 2006.

[43] P. Laarhoven, J. M. Van, and A. J. K. Lenstra, "Job shop scheduling by simulated annealing," *Operations Research*, vol. 40, pp. 113–125, 1992.

[44] V. Neumann, *Theory of Self-Reproduction Automata*, University of Illinois Press, " Urbana, IL, USA, 1966.

[45] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, no. 3, pp. 601–644, 1983.

[46] S. Wolfram, "Computation theory of cellular automata," *Communications in Mathematical Physics*, vol. 96, no. 1, pp. 15–57, 1984.

[47] E. Jen, "Global properties of cellular automata," *J Stat Phys*, vol. 43, p. 1986.

[48] B. Chopard, "Cellular automata modeling of physical systems,," in *Encyclopedia of Complexity and Systems Science*, R. Meyers, Ed., Springer, New York, NY, USA, 2009.

[49] S. W. Wei and Q. H. Hu, "An improved cellular genetic algorithm with evolutionary rules," *Computer Integrated Manufacturing Systems*, vol. 36, no. 8, pp. 247–252, 2019.

[50] Y. M. Lu, M. Li, and L. Li, "The cellular genetic algorithm with evolutionary rule," *Acta Electronica Sinica*, vol. 38, no. 7, pp. 1603–1607, 2010.