Cloud Computing Load Balancing Mechanism Taking into Account Load Balancing Ant Colony Optimization Algorithm

Jing He

College of Artificial Intelligence and Big Data, Chongqing Industry Polytechnic College, Chongqing, China

Correspondence should be addressed to Jing He; hejing@cqipc.edu.cn

Received 19 January 2022; Revised 8 February 2022; Accepted 9 February 2022; Published 11 April 2022

Academic Editor: Daqing Gong

1. Introduction

Cloud computing is a relatively new technology based on software services such as electricity and computing. It does its job by providing service and feedback. Service provision and use conditions are important features that distinguish it from the previous complex. Low database efficiency will affect system performance, reduce system response time, and even more seriously cause node downtime. For the scheduling of cloud resources, the diversity of cloud components and the diversity of user needs make some nodes overloaded, affecting the use of the entire system and resources. Cloud computing can realize the sharing of resources, can process massive amounts of data, and can provide an unlimited amount of storage space. Therefore, cloud computing is applied to military, school, medicine, scientific research, and other fields. However, when cloud computing faces such a large number of users, it has to deal with a lot of data and tasks. When a large number of requests are processed at the same time, this will affect the performance of the entire system. Therefore, how to achieve source access load balancing has become one of the most important issues in the implementation of cloud computing.

Cloud computing is already a significant area of research as a computer scientist. Load balancing technology is considered to be one of the most vital research directions in cloud computing and is a popular research direction that is constantly evolving. In cloud computing, the evolution must rely on the development of load balancing technology. So far, there is no ideal research result for downloading load balancing technology used in cloud environment, so it can be studied from different aspects. Due to its positive feedback mechanism and robustness, ant colony algorithm has been extensively studied and improved since it was proposed, especially for combinatorial optimization problems in application. Ant colony algorithm shows great advantages, so this article will use ant colony algorithm as the basis to apply it to the resource allocation problem in cloud computing.
A large amount of research has been done by domestic and international experts on the application of ant colony algorithms. Zhao et al. proposed a fault classification technique for neutral and invalid grounding systems based on binary ant colony algorithm (BACA) with fuzzy neural network (FNN). Test samples and field experimental findings indicate the high accuracy and good adaptability of the proposed fault classification method [1]. Vu et al. proposed a traffic load balancing framework that seeks to strike a balance between network applications (such as total delivery delays) and green energy consumption. The statistical results show that the proposed traffic load balancing process can achieve adjustable compensations between the use of grid power and the total traffic delivery delay and saves a lot of network-connected energy at a certain cost [5]. Sarma et al. reviewed the program documentation proposed by current load balancing technologies. Studies have shown the importance of the ACO method in supporting decision-making processes that help improve organic waste management and SWS control [4].

As for load balancing methods, there are fewer related studies. Hajiesmaili et al. proposed a traffic load balancing method that combines scene filtering procedures, precise algorithms, and parallel ant colony algorithms. Experiments show that building multiple initial solutions with reduced scene sets for the ant colony algorithm can significantly reduce the total time required to find the best solution. The proposed algorithm can be used for the automated construction of reliable control systems [3]. Jiang et al. used the ACO algorithm, which uses anaerobic digestion (AnD) for combination to increase biogas production to increase the release of organic waste from various waste sources in real time. An enrichment plant has been developed based on a virtual case study of organic waste affecting urban wastewater treatment plants and agri-food operations. Studies have shown the importance of the ACO method in supporting decision-making processes that help improve organic waste management and SWS control [4].

2. Cloud Computing Load Balancing Method

2.1. Cloud Computing. Cloud computing is a supercomputing model based on the Internet. It realizes the sharing of resources, and the resources and software in the cloud are allocated to users on demand. The goal of cloud computing is based on network technology to bring together traditional scattered computers to form a cluster. This combined cluster strengthens the processing capacity of the entire system.

The basic idea of cloud computing is that, in a distributed computing environment, each user can freely find the services he/she needs in this system quickly [9]. How to improve the average service response rate of the system has become a crucial issue in cloud computing [10]. The results of the cloud computing system are shown in Figure 1.

Cloud computing can be roughly divided into three categories, called the SPI model: software as a function, platform as a service, and infrastructure as a function. The infrastructure as a service provides the basis for platform services [11]. The platform as a service uses the infrastructure as a service to provide software as a service [12]. The platform as a service is similar to the web hosting service in the web hosting market. The corresponding relationship is shown in Figure 2.

Cloud computing has the following three characteristics:

(1) Cloud computing is based on virtualization technology to realize rapid deployment of resources and access to services.

(2) Cloud computing is provided to users in the form of services that are oriented to massive amounts of information and provided through the Internet.

(3) The resources of cloud computing can be dynamically expanded and configured according to the needs of users and paid according to the actual use of
users. Users do not need to manage them themselves, which can reduce the burden of users’ processing and dependence on IT expertise.

Cloud storage uses grid technology, aggregation technology, or file sharing technology to fully organize and coordinate multiple clouds of different storage devices on each device in the cloud-covered data center [13]. Cloud storage is a cloud computing storage system. User data is stored in the cloud. For users, local storage systems and storage devices are not required [14]. More specifically, cloud storage is a service in which data is provided by a collection of multiple storage devices and servers, and users purchase services on demand.

Cloud storage deals with massive amounts of data, and it is necessary to complete the storage and management of massive amounts of data. The storage layer is composed of storage devices and storage device management systems. The storage application management system mainly completes the maintenance and upgrading of storage facilities and the monitoring of storage equipment. The structure of the cloud storage system is shown in Figure 3.

Cloud data centers usually include thousands of servers; these server nodes are heterogeneous and unevenly distributed. The storage and scheduling of resources are coordinated by many servers, and the user’s access to the server is random, diversified, and complicated. The existence of these factors can easily cause some server nodes to be idle. While other server nodes are in a state of overload saturation; that is, the load of server nodes is not balanced. The load of server nodes includes not only the load of data storage, but also the load of resource scheduling. Obviously, the existence of this imbalance will cause a waste of cloud resources, increase the burden on the network, and affect the use of users. Therefore, it is essential that the issue of load balance in relation to cloud computing is addressed. A suitable load balancing strategy allows for optimal configuration of services and increased working power.

2.2. Load Balancing Algorithm. Load balancing is realized by monitoring the usage of resources on each physical node and then analyzing the collected resources; by analyzing the algorithm of load balancing, the load on the node is dynamically migrated to the node with a small load; and finally the purpose of load balancing is achieved. Any computer or server source can be optimized, the response time can be as short as possible, and the processor performance can be as high as possible. Under normal circumstances, the user partition is finally obtained by analyzing the source of each controller (such as CPU usage, memory space, and bandwidth). Normally, the architecture of load balancing is shown in Figure 4.

A load balancing system is mainly composed of three parts: the cloud platform of the IaaS layer realizes the collection of resources, the realization of the load balancing strategy in the PaaS layer, and the web customer service terminal of the SaaS layer makes requests. At the IaaS layer, the underlying virtualization is mainly realized through the OpenStack cloud platform, and the monitoring function is mainly used to monitor and collect the resources of each node through Ganglia.

The PaaS layer is the core module to achieve load balancing, which is achieved through load balancing based on the dynamic load balancing algorithm mentioned in Section 3. In this layer, the dynamic scheduling of virtual machines on the nodes is realized through WSO2 software. The PaaS layer is mainly used to implement load balancing strategies and scheduling. The core processing module is the strategy controller and the queue manager. The strategy selection is mainly to analyze the load data collected by the server. By analyzing which physical machines and virtual machine resources are overloaded, underloaded, etc. and saving the data in the database, the corresponding data needs to be fetched from this database during scheduling. However, the queue manager mainly queues data.

Common load balancing algorithms include round-robin balancing algorithms, weighted round-robin symmetry, temporary balancing algorithms, and mechanical balancing algorithms.

The rotation balance algorithm used is to determine the rotation frequency before sending the request and then divide the request, rotation, etc. on the respective servers according to this process.

At the weighted round-robin balancing algorithm, the performance of each server is not considered in the round-
The performance of each server is inconsistent, and greater configuration and processing capacity mean greater weight; then, the load allocated to the server will be more, and the higher the load, the more efficient the operation.

The concept of load balancing is first introduced at each server node in the static payload trading algorithm, and then the corresponding function is allocated to each server, rather than the estimated resource power allocation. In the process of adjusting the concept and load balancing, the load status of each server is taken, but a value is assigned to the load, and this value will not change with the change of the load.

Consequently, the dynamic burden balancing method is a very good solution. The animated workload trading algorithm balances by monitoring the source status of each server and analyzing the load capacity adjustment of each node.

Through the above analysis and comparison of several load balancing algorithms, this article chooses a dynamic load balancing algorithm to achieve resource load balancing. The collection of PM (physical machine) and VM (virtual machine) data of each node in the cluster is the basis of load balancing, because algorithm balancing can be completed through the loading function of each node, and finally load balancing is determined. Because load balancing is implemented dynamically, real-time collection is required when collecting node resource information. Through analysis of the collected data, the causes of uneven load of virtual machine resources can be efficiently analyzed and processed. While collecting nodes, it is necessary to monitor the resource status of each node in real time, and the related scheduling work can be realized at the end through monitoring.

3. Ant Colony Algorithm
For the network, load balancing technology is the basis for the stable operation of the network and the saving of operating resources. The article will use ant colony algorithm to achieve load balancing of network links. At the same time, through the experimental analysis and selection of some parameters in the ant colony algorithm, we finally obtain a better load balancing effect.

The principle of the ant colony algorithm is that when insects carry food, they will leave a recognizable pheromone on the street, and other ant colonies can recognize this pheromone and know the concentration of the pheromone. Then, a pathway with a high pheromone concentration will be more likely to be selected, other ant colonies will pay attention to this pathway accordingly, and the pheromone concentration will increase accordingly, thus forming a positive feedback system. Therefore, the pheromone on the street will also change over time, and the pheromone concentration will decrease.
The article studies the mathematical principles, theories, and related applications of the ant colony algorithm. In particular, it proposes an improved ant colony algorithm for the shortcomings of the algorithm that is prone to stagnation, and finally verifies its effectiveness with specific TSPs.

Through simulation experiments, the influence of parameter range on the optimal solution is studied. The main parameters are as follows: $\alpha$ is the relative importance of pheromone (pheromone heuristic factor); $\beta$ is the relative importance of heuristic factor (expected heuristic factor); $\rho$ is the pheromone volatilization coefficient; $m$ is the number of ant colonies.

In the cloud computing environment, the problem of resource allocation is still a combination optimization problem. In the cloud computing environment, there are many factors affecting resource allocation, which correspond to the length of the intercity path in the traveling salesman problem. From the above application experiment of genetic ant colony algorithm (GACA) which improves ant colony algorithm through genetic algorithm, it is known that the fusion of intelligent algorithms is a more efficient and correct way to solve the problem. However, the merged algorithm can still have premature convergence in the later stage. Therefore, we need to maintain the diversity of the population in the iterative process and expand the search solution space to a certain extent, so as to prevent the occurrence of premature convergence to a certain extent. The definition of matching factor is introduced to express the matching degree of each task and each resource node.

$$\text{Match}_{ij} = \frac{1}{\sqrt{\sum_{n=1}^{N} (G_{ix} - V_{yi})^2}}$$

Supposing $X$ is a mapping sequence of tasks and resources, the degree of load balance is defined as follows:

$$\text{Load}(X) = \sum_{i=1}^{n} \frac{1}{1 - C_i^k \frac{L_{vi}}{Y_i}}.$$

Among them, $Y_i$ indicates how many tasks are allocated to resource node $i$.

Approximate nondeterministic tree search (ANTS) is an ACO algorithm that uses mathematical programming ideas. The ANTS algorithm uses a new action selection rule and pheromone update rule, as shown in the following formula:

$$\begin{align*}
\tau_{ij}^k &= \frac{\gamma \tau_{ij} + (1 - \gamma) \eta_{ij}}{\sum_{n=1}^{N_i} \gamma \tau_{ij} + (1 + \gamma) \eta_{ij}} \\
\tau_{ij} &= \tau_{ij} + \sum_{k=1}^{\Delta k_{ij}} \\
\Delta k_{ij} &= \begin{cases} 
\tau_{ij} \left( \frac{C_i^k - LB}{L_{\text{avg}} - LB} \right), & \text{if } \text{allow}\kappa \beta \tau_{ij}^k \text{ allowed} \\
0, & \text{otherwise}
\end{cases}
\end{align*}$$
Among them, $\theta$ is a parameter, and LB is an optimal solution calculated at the beginning of the algorithm.

For any given data, there is a uniquely determined membership function value $\mu A(x) \in [0, 1]$ corresponding to it, which can be expressed as follows:

$$\mu A(x): X \rightarrow [0, 1].$$  \hspace{1cm} (11)

Usually, the expression is as follows:

$$\bar{A} = \frac{\mu A_1(x_1)}{x_1} + \frac{\mu A_2(x_2)}{x_2} + \ldots + \frac{\mu A_n(x_n)}{x_n}. \hspace{1cm} (12)$$

This kind of representation can be represented by sequential even representation:

$$\bar{A} = \{(x_1, \mu A_1(x_1)), (x_2, \mu A_2(x_2)), \ldots, (x_n, \mu A_n(x_n))\},$$ \hspace{1cm} (13)

or using vector notation:

$$\bar{A} = \{\mu A_1(x_1), \mu A_2(x_2), \ldots, \mu A_n(x_n)\}. \hspace{1cm} (14)$$

In the cloud computing system $t$, there are $N$ tasks that need to be allocated to $M$ nodes for processing, and the current load status of each server is different. The load balancing strategy is to find the most reasonable task allocation plan, so that the average processing time of tasks is the shortest, and the service quality of the cloud computing system is improved. The load deviation rate is used to measure the system load distribution after assigning tasks, and the load deviation rate is calculated as follows:

$$E(x_1, x_2, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{M} e_i. \hspace{1cm} (15)$$

Among them, $M$ represents the predicted value of the resource load required on the $i$-th server under this task allocation plan, and the calculation formula is as follows:

$$e_i = f_i(x_i, L_i) - F(x_1, x_2, \ldots, x_n). \hspace{1cm} (16)$$

Among them, $f_i(x_i, L_i)$ represents the predicted value of resource load required on the $i$-th server under this task allocation plan, and the calculation formula is as follows:

$$f_i(x_i, L_i) = (x_i + 1) \times L_i. \hspace{1cm} (17)$$

In the current server running state, the calculation formula for the impact of allocating a unit of computing resource on the load is as follows:

$$F(x_1, x_2, \ldots, x_n) = \frac{1}{M} \sum_{i=1}^{M} f_i(x_i, L_i). \hspace{1cm} (18)$$

According to the above calculation formula, the load deviation rate takes a value in the range of 0 to 1. The smaller the value, the more balanced the load distribution of the cloud computing system, and the better the performance of the current system [16].

4. Experiments and Results of Cloud Computing Load Balancing

4.1. Experiment Preparation. The cloud computing simulation platform is selected to conduct experimental analysis and evaluation of the cloud computing load balancing process based on virtual machine relocation. The main performance indicators are the load balance of CPU, memory, and other sources and the average operating time. By monitoring the load situation in the network in real time, the ant colony algorithm can give the forwarding link with the smallest load, so as to realize the relative balance of dynamic link load. This can optimize the configuration of limited network resources and effectively solve the problems of low network bandwidth utilization and low throughput. In order to simplify the processing and reflect the reliability of the experimental results of different strategies, in this simulation experiment, the cloud computing data center is configured with the same type of servers, virtual machines with the same parameters, and the same network environment and performs a set of the same tasks. The specific conditions are shown in Table 1.

It is measured by the load standard deviation for the CPU and memory of the system. Their corresponding utilization standard difference will be calculated by detecting the CPU or memory utilization at a given time for each host.

The level of memory load balancing is compared between layered management-based virtual machine migration strategy (MLVM) and agent-based virtual machine migration strategy (PAVM). After that, the entire CPU utilization of the system is read at different time points, and the standard deviation of the system CPU load balance is calculated. The experimental results are shown in Table 2.

Through experiments, the task response time for different task requests is counted, and the average task response time is calculated. The average task response time of the two strategies is shown in Table 3.

4.2. Algorithm Optimal Solution. We perform statistics on the optimal solutions for different $\alpha$ values in the ant colony algorithm, and the results are shown in Figure 5.

The heuristic factor $\alpha$ characterizes the relative importance of the pheromone accumulated by the ants in the process of constructing the solution for the path selection in the ant search. The larger the value of $\alpha$, the more likely the ants will pick the previous paths they traveled and the searching paths will be less stochastic. When $\alpha$ takes different values, the corresponding optimal iterative optimal solution and number of iterations are shown in Figure 6. When $\alpha = 1$, the iterative optimal solution gets the minimum value, and then as $\alpha$ increases, the iterative optimal solution value also increases; the number of
iterations also decreases as $\alpha$ increases. This reflects the overall impact of $\alpha$ on the algorithm, that is to say, the larger the value of $\alpha$, the more likely the ants are to choose previously walked paths; there will be less randomness in the search paths, they will converge faster, and they will be easily restricted to limiting themselves to locally optimal solutions.

We carry out statistical analysis on the load quantity of each resource of MLVM and PAVM, and the results are shown in Figure 7.

It can be seen that PAVM accepts the criteria to increase the possibility of finding a better allocation method, which is subject to certain restrictions on the execution time of the algorithm, but there is a certain improvement in the scheduling time. We compare the task time spans of several
most commonly used algorithms, such as ant colony algorithm, genetic ant colony algorithm, and improved algorithm, and the results are shown in Figure 8.

In a load balancing system, validity is an important criterion for judging the quality of its algorithm, so it is particularly important to verify this criterion. Due to the instability of the experimental results, the method of seeking a tie for each set of experiments is used to obtain the final result. We judged the effectiveness of different methods by the response time when reading files of different sizes. The experimental results are shown in Figure 9.

It can be seen that the response time for reading each file varies, and this time is not completely related to the file size, which means that not all large files have a long response time, and not all small files have a short response time. However, in the ACO algorithm, the average response time is about 30% lower than that in other algorithms. This shows that the ACO algorithm can play a role.

5. Discussion

In order to avoid data loss and improve system reliability, copy technology is usually used in cloud storage systems. The copy technology is to make multiple copies of a data block, store them on multiple nodes, and use data redundancy to ensure data reliability. The use of copy technology can well solve the load balancing problem of the system. Multiple copies are distributed on different server nodes. By accessing the copy, the load of a single server node is distributed to multiple nodes, thereby reducing the load of the nodes and achieving load balancing.

The ant colony optimization algorithm has undergone more than ten years of development since it was proposed and has achieved remarkable results in algorithm theory research and application. Its application scope involves various optimization fields such as routing problems, allocation problems, scheduling problems, subset problems, machine learning problems, and emergence of ant colony algorithm bionic hardware, showing the strong vitality and broad development of ant colony optimization algorithm prospect.

With the continuous update and development of computer technology, the number of users has doubled, the architecture of traditional IP networks has become more and more complex, and the development of the network structure is relatively lagging. In this paper, the load balance is realized by using the ant colony algorithm, which is implanted in the controller, and the network load is
monitored in real time through the traffic monitoring software. Through the execution of the ant colony algorithm in the controller, the data stream will be transmitted along the link with the least load at a certain moment, which can effectively use network resources, reduce network congestion, increase network throughput, and realize network dynamic load balancing of each link in the network. We also carried out theoretical experimental research and analysis on the heuristic factor, the pheromone volatilization coefficient, and the number of ants in the ant colony algorithm. Through the selection of parameters, the performance of the ant colony algorithm was improved, and the load balancing was also able to obtain better results. We conduct load balancing experiments on the self-defined network topology. Through the analysis of experimental data, the introduction of ant colony algorithm can achieve dynamic load balancing of each link.

6. Conclusion

For the purpose of this paper, we will be looking at the allocation of resources for cloud computing. According to the existing algorithm, the intelligent algorithm is fused, and the pheromone initial allocation of the ant colony algorithm is introduced to solve the blind search problem and avoid the later local optimal solution problem. The improved algorithm was applied to the specific problem of cloud computing resource allocation, and experimental calculation and analysis were performed. It was found that the time span performance and resource load degree had relatively good results, which proved the applicability of the algorithm. The improved algorithm proposed in this paper has certain performance optimization but also has certain disadvantages. The algorithm becomes slightly more complicated and has a great test on the execution speed. Moreover, the actual situation should be fully considered in the actual application process; for example, issues such as more diverse and dynamic user needs and complexity of resources require comprehensive consideration of the needs and interests of users and providers. In addition, the ant colony algorithm is often aimed at a certain type of special problem, and its versatility is poor. In future research, if we want to use the ant colony algorithm in other fields, we need to improve its universality. In addition, the research in this paper only considers the small network with few hosts and single controller and does not consider the load balancing of the large network with multiple hosts and multiple controllers, and the researcher hopes that this part of the research can be completed through corresponding experiments in the future.

Data Availability

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

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This research study was sponsored by Chongqing Yubei District Science and technology planning project: "Big Data Development of Service PAAS Automatic Delivery and Management Platform."

References

[1] 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.
[2] T. K. Vu, M. Bennis, S. Samarakoon, M. Debbah, and M. Latva-Aho, "JOINT load balancing and interference mitigation IN 5G HetNets," IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 1–15, 2017.
[3] C. Pham, N. H. Tran, C. T. Do, E.-N. Huh, and C. S. Hong, "Joint consolidation and service-aware load balancing for datacenters," IEEE Communications Letters, vol. 20, no. 2, pp. 292–295, 2016.
[4] D. Jiang, P. Zhang, Z. Lx, and H. Song, "Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1437–1447, 2016.
[5] M. H. Hajiesmaili, L. Deng, M. Chen, and Z. Li, "Incentivizing device-to-device load balancing for cellular networks: an online auction design," IEEE Journal on Selected Areas in Communications, vol. 35, no. 2, pp. 265–279, 2017.
[6] A. Sarma, S. Chakraborty, and S. Nandi, "Deciding handover points based on context-aware load balancing in a WiFi-WiMAX heterogeneous network environment," IEEE Transactions on Vehicular Technology, vol. 65, no. 1, pp. 348–357, 2016.
[7] H. Akbari, P. Berenbrink, and T. Sauerwald, "A simple approach for adapting continuous load balancing processes to discrete settings," Distributed Computing, vol. 29, no. 2, pp. 143–161, 2016.
[8] L. Kong, K. Ma, B. Qiao, and X. Guo, "Adaptive relay chain routing with load balancing and high energy efficiency," IEEE Sensors Journal, vol. 16, no. 14, pp. 5826–5836, 2016.
[9] J. R. Aviles, M. Toril, S. Luna-Ramirez, V. Buenestado, and M. A. Regueira, "Analysis of limitations of mobility load...
balancing in a live LTE system,” *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 417–420, 2017.

[10] P. Ren, M. A. Kinsy, and N. Zheng, “Fault-aware load-balancing routing for 2D-mesh and torus on-chip network topologies,” *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 873–887, 2016.

[11] I. Cabria and I. Gondra, ”Potential- $K$- means for load balancing and cost minimization in mobile recycling network,” *IEEE Systems Journal*, vol. 11, no. 1, pp. 242–249, 2017.

[12] R. Trestian, K. Katrinis, and G. M. Muntean, “OFLoad: an OpenFlow-based dynamic load balancing strategy for data-center networks,” *IEEE Transactions on Network & Service Management*, vol. 14, no. 4, p. 1, 2017.

[13] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, ”Secure and sustainable load balancing of edge data centers in fog computing,” *IEEE Communications Magazine*, vol. 56, no. 5, pp. 60–65, 2018.

[14] O. Brun and B. Prabhu, ”Worst-case analysis of non-cooperative load balancing,” *Annals of Operations Research*, vol. 239, no. 2, pp. 471–495, 2016.

[15] W. Xu and Y. Yin, ”Functional objectives decision-making of discrete manufacturing system based on integrated ant colony optimization and particle swarm optimization approach,” *Advances in Production Engineering & Management*, vol. 13, no. 4, pp. 389–404, 2018.

[16] W. Yu, G. Hou, P. Xia, and J. Li, ”Supply chain joint inventory management and cost optimization based on ant colony algorithm and fuzzy model,” *Tehnicki vjesnik-Technical Gazette*, vol. 26, no. 6, pp. 1729–1737, 2019.