A Cloud Computing Oriented Neural Network for Resource Demands and Management Scheduling

Gaoxiang Lou and Zongyan Cai
(Corresponding author: Gaoxiang Lou)

School of Construction Machinery, Chang'an University
Xi'an, Shaanxi 710064, China
(Email: agoxzl@126.com)

(Received May 31, 2018; Revised and Accepted Dec. 17, 2018; First Online Jan. 31, 2019)

Abstract

Cloud computing, a new kind of resource sharing service system, can provide virtual resource services such as infrastructure and platform for users who access it through the Internet. Its service quality is related to resource management and scheduling. In this study, CloudSim3.0 simulation platform was used as a simulation platform for cloud computing resource scheduling to test the performance of radial base function (RBF) neural network based on particle swarm optimization (PSO) and RBF neural network based on Improved Particle Swarm Optimization (IPSO) in cloud resource scheduling and configuration. The results showed that the CPU and memory utilization rate and processing time of the two algorithms increased with the increase of processing tasks. It was found that compared to PSO-RBF, IPSO-RBF had higher CPU and memory utilization rate and shorter processing time and converged faster and found the best position of particles after only 30 iterations with small fluctuation amplitude. In addition, IPSO-RBF had better performance in balancing the load of different kinds of physical resources compared to PSO-RBF.

Keywords: Cloud Computing; Particle Swarm Algorithm; Radial Base Function Neural Network; Resource Scheduling

1 Introduction

With the development of society, the role of computer in various fields of society is becoming more and more extensive. At the same time, the demand for computing power in industries that need to use computer power is also increasing day by day [6]. Single improvement of computer hardware performance to obtain more computing power not only has limited computing power improvement, but also has low cost performance for a single user compared with the cost of hardware improvement [9].

The emergence of cloud computing solves the problem of limited performance improvement of a single computer. Relying on the Internet, cloud computing uploaded and distributed the tasks that users needed to process to the "master station" composed of a large number of servers, and applied the physical resources in the "master station" to process the tasks of users [1, 3]. Its "master station" is called the resource pool [15], also known as "cloud". Cloud computing combines virtualization, parallel computing, distributed computing and other concepts, and has the following characteristics [5].

Cloud computing has a larger computing scale than a single user; Cloud computing can complete the interaction of information through the Internet and terminals, and the resources it uses are not physical objects [10, 13]. Resource pools in cloud computing are Shared. Relevant researches are as follows. Chen et al. [4] proposed an Improved Ant Colony System (IACS) method and conducted extensive experiments based on workflows of different scales and different cloud resources. Experimental results showed that IACS was able to find a better solution with lower cost than basic particle swarm optimization (PSO) and dynamic target genetic algorithm under different scheduling scales and deadlines. Abdullahi Mohammed et al. [11] proposed a symbiotic organisms search optimization algorithm (SOS) based on simulated annealing (SA) to improve the convergence speed and quality of SOS solutions.

CloudSim simulation results showed that SASOS was superior to SOS in terms of convergence speed, response time, unbalance and MAK. Zhang et al. [18] proposed an improved Centrino Hardware Control (CHC) algorithm, which inherited the advantages of standard genetic algorithm (SGA) and CHC algorithm. The experimental results showed that the improved CHC algorithm had better efficiency and convergence, and the average time and completion time of task scheduling were relatively shorter. In this paper, CloudSim3.0 simulation platform was adopted as the simulation platform for cloud computing resource scheduling to test the scheduling configuration performance of RBF neural network based on PSO and RBF neural network based on improved particle swarm optimization (IPSO).

2 Cloud Computing Resource Scheduling

The resource scheduling model in cloud computing consists of three layers: service request, virtual resource pool and physical resource pool. Cloud computing resource scheduling was generally divided into two steps. First, the task in the service request was allocated to the virtual machine in the virtual resource pool, and then the allocated virtual machine was deployed to the physical machine in the physical resource pool, as showed in Figure 1.

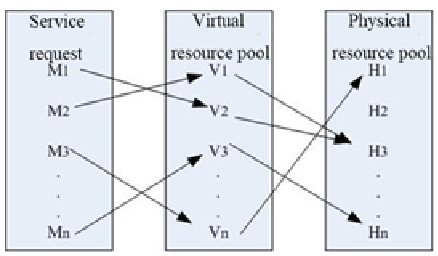

Figure 1: Cloud computing resource scheduling

Its mathematical model [2] was: If the task in the service request was divided into n mutually independent subtasks, the set of sub-tasks was $M = \{m_1, m_2, \cdots, m_n\}$, where the i-th sub-task was m_i, and a sub-task could only run in one virtual node. If there were b virtual nodes ($b < n$), the set of virtual nodes was $V = \{v_1, v_2, \cdots, v_b\}$, where v_j is the j-th virtual node. Then the distribution could be expressed by the matrix A:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{b1} & a_{b2} & \cdots & a_{bn} \end{bmatrix} \quad (1)$$

Any element in the matrix represents the relationship between the sub-task and the virtual section. When the i-th task run on the JTH node, it was 1, otherwise it was 0. For the convenience of later calculation, it was assumed that a node could only run one task at a time, and a task could only run on one node at the same time. Hence the time spent in completing task M was:

$$time = \max(\sum_{i=1}^{n} a_{ji}t_{ji}, \quad (1 \leq i \leq n, 1 \leq j \leq b). \quad (2)$$

A good resource scheduling algorithm should have the least time to complete the task, *i.e.*,

$$Cost = \min(time). \quad (3)$$

Equations (1), (2) and (3) were the mathematical model of cloud computing resource scheduling.

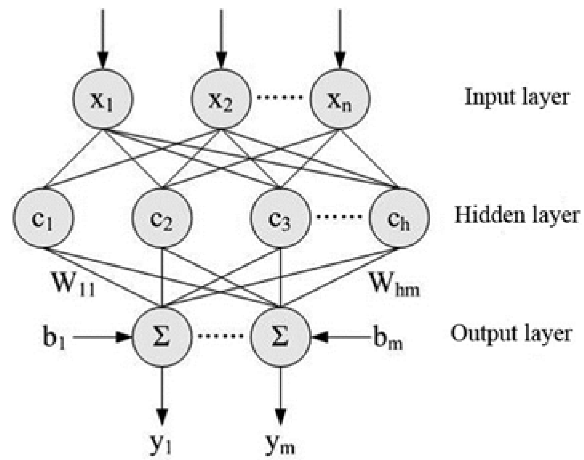

Figure 2: Basic structure of RBF network

3 Radial Basis Function (RBF) Neural Network

Radial basis function neural network [?, 19] has a simple structure. It converges quickly and could simulate any nonlinear function. Its structure is shown in Figure 2. RBF neural network belongs to feed-forward neural network, and its basic structure was divided into three layers, including input layer, hidden layer and output layer. The activation function of hidden layer is radial basis function, so each node of hidden layer has a data center c_i.

Each data node in the input layer was denoted as $x = (x_1, x_2, \cdots, x_i, \cdots, x_n)^T$. The output data in the output layer was denoted as $y = (y_1, y_2, \cdots, y_i, \cdots, y_n)^T$. The mapping between the hidden layer and the output layer was a linear mapping, and the weight matrix of the mapping between the nodes of the two layers was denoted as $w = (w_{11}, w_{12}, \cdots, w_{ij}, \cdots, w_{hm})^T$. The mapping from an input layer to a hidden layer in RBF network is a nonlinear mapping, and its mapping formula [7] was:

$$h_i = \exp(\frac{||x - c_i||^2}{2b_i^2}), \quad 1 \leq i \leq h, \quad (4)$$

where h_i refers to the output of the i-th node in the hidden layer, $|| \cdot ||$ is the euclidean distance, and b_i refers to the width of the radial basis function of the node of the i-th hidden layer. The mapping formula between the hidden layer and the output layer was:

$$y_j = \sum_{i=1}^{h} w_{ij}h_i, \quad 1 \leq j \leq m, \quad (5)$$

where y_j is the output data of the j-th node in the output layer and w_{ij} is the mapping weight between the i-th hidden layer node and the j-th output layer node.

4 RBF Network Based on PSO

PSO [17] is also known as "flock foraging algorithm" because it is derived from the research on the foraging and migration behavior of birds. Its principle is to obtain the global optimal solution by tracing the current optimal solution. PSO is one of the evolutionary algorithms. Similar to other evolutionary algorithms, PSO uses population to find the solution set in space, and iterates randomly on population initialization and end conditions to get the optimal solution. In the process of operation, the direction of the optimal solution was determined by the fitness value, and the fitness value was used to judge whether the solution was good or not. The global optimal solution was found by comparing the self-optimal solution with the currently explored optimal solution.In particle swarm optimization, individual updates were influenced by historical particles rather than random ones. The number of particles in the particle swarm optimization algorithm was n. In m-dimensional space, P_i was used as the vector position of the i-th particle, i.e., $\overrightarrow{P_i} = (p_{i1}, p_{i2}, \cdots, p_{iM})$, S_i as the velocity of the particle $\overrightarrow{V_i} = (v_{i1}, v_{i2}, \cdots, v_{iM})$.

Let the current optimal position of the i-th particle be $pbest_i$, the optimal position of the particle swarm was $gbest_i$. Then the formula [16] for the velocity change of the particle was:

$$V_{i+1} = \mu V_i + a_1 x_1 (pbest_i - P_i) + a_2 x_2 (gbest_i - P_i), \quad (6)$$

where μ refers to the value of the particle affected by inertia, a_1 and a_2 are the learning factor, x_1 and x_2 are a random number, they were evenly distributed between 0 and 1, $a_1 x_1 (pbest_i - P_i)$ is cognitive term, and $a_2 x_2 (gbest_i - P_i)$ refers to social term.

The expression of the position change of the particle was:

$$P_{i+1} = P_i + V_i. \quad (7)$$

By repeatedly updating the calculation based on Equations (6) and (7), and analyzing the fitness of particles, the optimal solution with the largest fitness could be found. This study adopted PSO algorithm to optimize RBF network. First, the mapping parameters in the RBF network were converted into the dimensional vectors of each particle in the PSO. In other words, mapping parameters such as data center c_i, width coefficient b_i, and weight w_{ij} in RBF network were taken as dimensions in corresponding particles. Then, the fitness function in PSO was taken as the mean square error in RBF network, and the optimal weight with the minimum mean square error could be obtained after PSO operation.

After the mapping parameters in RBF were converted into the dimensions of particles in the particle swarm, the optimal parameters of a set of particles were calculated according to the calculation flow in Figure 3, and then the dimensions of the particles were converted into the mapping parameters in the RBF network to participate in the scheduling and configuration of cloud resources by the RBF neural network.

5 RBF Network Based on Improved Particle Swarm Optimization

Up to now, there are various ways to improve the traditional particle swarm optimization algorithm, but the ultimate purpose is to make up for the two shortcomings of the traditional particle swarm optimization: First, when the test object is a complex function, with the increase of the number of iterations, the algorithm is likely to fall into a local extreme value, which is difficult to obtain the real optimal solution. The second is the selection of algorithm parameters, among which the inertia factor and learning factor have the greatest influence on the change of algorithm capability.

In the traditional PSO algorithm, cognitive coefficient a1 and social coefficient a2 remain unchanged, they are learning factors; as a result, when the number of iterations is small, the influence of individual cognition is large; when the number of iterations is large, the social influence is large, failing to reflect the aforementioned changes at the same time, which makes the algorithm obtain the local optimal solution and induces the phenomenon of "premature" [14]. In order to solve the above problems, the traditional particle swarm optimization algorithm was improved, and the improved formula [12] was:

$$S_{i+1} = \mu V_i + b(\frac{1}{n}) x_1 (pbest_i - P_i) + cn^2 x_2 (gbest_i - P_i). \quad (8)$$

By comparing Equations (6) and (8), it was found that the improvement made is to change the cognitive term coefficient a_1 into $b(\frac{1}{n})$, so that the individual cognitive proportion decreased with the increase of the number of iterations. The social item coefficient a_2 changes to cn^2, so that the social item proportion of the group increases with the number of iterations. Then, when calculating, b took a large value, and c took a small value, making it conform to the fact that individual cognition was the dominant factor in the initial stage. As the number of iterations increases, the proportion of social terms increases, and it started to become the dominant factor.

After the improvement, the algorithm was more consistent with the objective law of the optimal solution, and the practicability was greatly improved. The optimization steps of the IPSO for RBF network were not much different from those of PSO for RBF above. Similarly, the mapping parameters in RBF were converted into the dimensions of particles in the particle swarm. Then, the optimal parameters of the particle swarm were calculated according to the steps shown in Figure 3, and the difference was that the formula for updating the particle velocity is replaced by Equations (6) with (8). Then, the dimensions of the optimal particle swarm were converted into various mapping parameters in the RBF network, and participate in the scheduling and configuration of cloud resources by the RBF neural network.

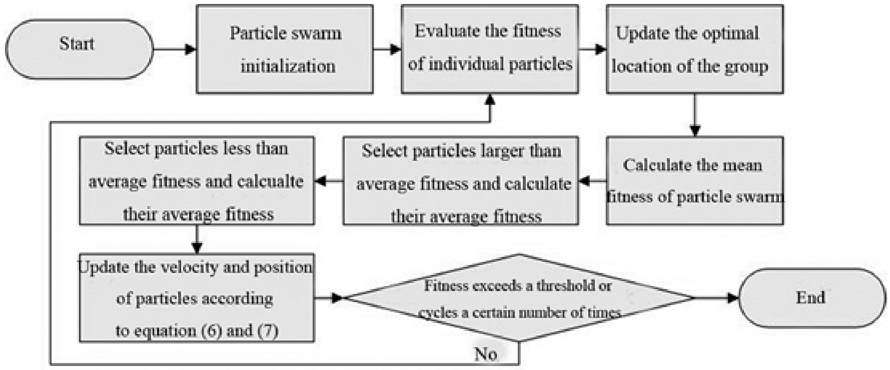

Figure 3: Calculation flow of RBF network mapping parameters based on PSO algorithm

6 Experimental Analysis

6.1 Experimental Environment

In this study, CloudSim3.0 simulation platform [8] was selected as the simulation platform for cloud computing resource scheduling to test the scheduling configuration performance of PSO-based RBF neural network and IPSO-based RBF neural network for cloud resources.The experimental environment in this paper was Windows10 operating system on hardware with 16G of memory and 1000G of hard disk storage. On the software, CloudSim3.0 cloud platform simulation simulator, compilation environment JDK1.7 and development tool MyEclipse were adopted.

6.2 Experiment Settings

Thirty physical resources were set up in the cloud computing laboratory center and converted into virtual machine resources. The configuration of each virtual machine was 2GB memory capacity and 3.0 GHz CPU processing frequency. Virtual physical resources were divided into three parts: responsible for processing document classes, for image processing, and for dealing with video class, and the maximum number of iterations was set as 100. The learning factor was set as 1.33. The number of submitted tasks was set as 100, including three types: 10 types of documents, 30 types of pictures, and 60 types of video. Each algorithm was iterated for 100 times, and the experiment was repeated for 40 times. The average value of each test result was obtained.

6.3 Experimental Results

As showed in Figure 4, under the premise that the total number of cloud resources was constant, the utilization rate of CPU and memory of the two algorithms increased accordingly with the increase of the number of tasks executed; when the number of tasks exceeded a certain number, the utilization rate was relatively stable; the CPU utilization of PSO-RBF increased with the number of tasks before executing 50 tasks, and fluctuated around 0.4 after more than 50 tasks; the CPU utilization rate of IPSO-RBF increased with the number of tasks before 60 tasks were executed, and fluctuated around 0.6 after more than 60 tasks. For memory utilization, PSO-RBF fluctuated around 0.4 after 80 tasks and IPSO-RBF fluctuated around 0.5 after 60 tasks. On the whole, both CPU utilization and memory utilization rates were higher compared with IPSO-RBF, because PSO-RBF algorithm used more physical resources, tasks were evenly distributed to each virtual machine node, and the utilization of CPU and memory was evenly distributed.

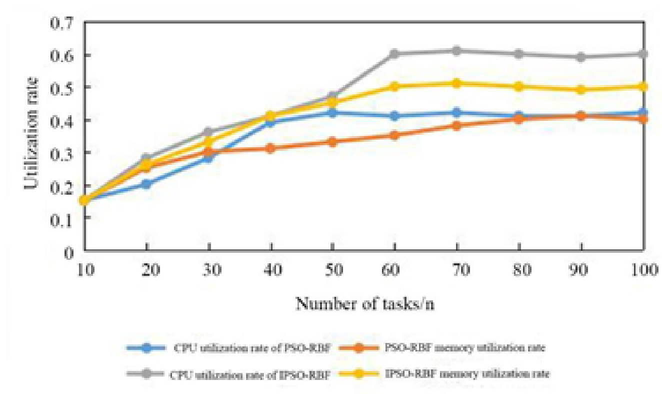

Figure 4: CPU and memory utilization rates of the two algorithms under different task number

As showed in Figure 5, with the increased of the number of processing tasks, the time required by the two algorithms for processing tasks also showed an upward trend of fluctuation. The fluctuation range of PSO-RBF was larger, and the time required to process the same number of tasks was greater than those of IPSO-RBF in most cases. It could be seen from the figure that it took the most time to process 70 tasks in this experiment, which took 910 ms. The time required for IPSO-RBF task processing fluctuated with the increase of the task, and the fluctuation range was small. It was found that it took the most time, 80 ms, to process 100 tasks in this experiment.

On the whole, IPSO-RBF was more efficient and took less time to process tasks.

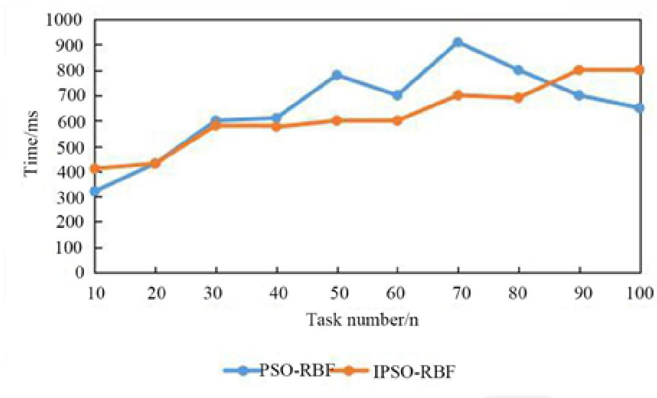

Figure 5: Time consumed by the two algorithms in processing different number of tasks

As showed in Figure 6, with the increase of iteration times, the convergence of the two algorithms became stable. In the process of convergence, the particle position of PSO-RBF increased with the number of iterations and fluctuated greatly. It did not stabilize until 90 iterations. The particle position of IPSO-RBF increased with the number of iterations, with a small fluctuation range, and the position of the optimal solution was found stably after 30 iterations. IPSO-RBF had a fast convergence speed and a better effect of finding the optimal solution.

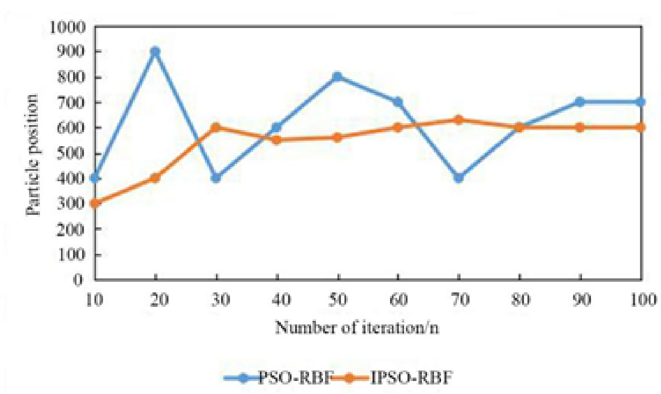

Figure 6: Convergence effect of the two algorithms

As showed in Figure 7, due to the difference in processing capacity between nodes, the number of tasks assigned on different types of physical resource nodes was different. In the PSO-RBF algorithm, the number of tasks processed on the three types of nodes was similar, and the number of actual task types was compared. It could be found that the PSO-RBF algorithm basically distributes tasks to all nodes on an equal basis without considering the difference in processing capability of different types of tasks between different nodes. However, the number of tasks on different types of nodes allocated by IPSO-RBF algorithm was obviously different. By comparing the number of actual types of tasks, it was found that this algorithm considers whether the type of tasks was consistent with the type of resource nodes in the allocation of tasks, and balanced the load between different nodes.

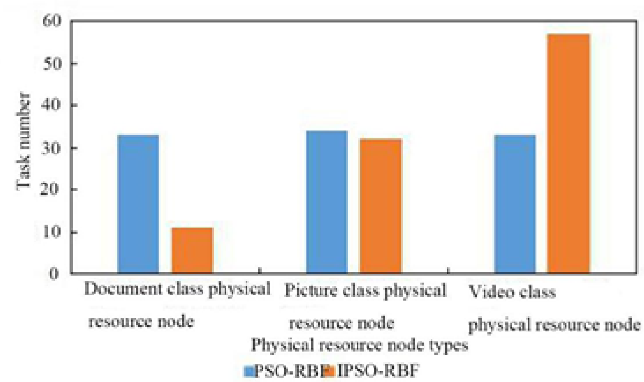

Figure 7: Load distribution on nodes of different physical resource types

7 Conclusion

Firstly, this paper briefly introduced the cloud computing resource scheduling model and the construction of RBF neural network. In cloud computing resource scheduling, tasks were allocated to virtual machine according to requests, and then virtual machine was allocated with physical resources. Then, the PSO-based RBF neural network algorithm and IPSO-based RBF neural network algorithm were proposed. Finally, simulation experiments were carried out on the CloudSim3.0 simulation platform for the two algorithms. The results showed that the CPU and memory utilization rate and running time of the virtual machine increased with the increase of the number of tasks. Moreover, IPSO-RBF algorithm had a higher utilization rate of virtual machine resources and shorter running time than PSO-RBF algorithm. In terms of the convergence effect, IPSO-RBF algorithm could achieve stable convergence faster with only 30 iterations, while PSO-RBF algorithm needed 90 iterations to achieve stable convergence. IPSO-RBF could balance the load of different kinds of physical resources better than PSO-RBF.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] A. Alexandridis, E. Chondrodima, H. Sarimveis, "Cooperative learning for radial basis function networks using particle swarm optimization," *Applied Soft Computing*, vol. 49, pp. 485–497, 2016.

[3] M. H. R. Al-Shaikhly, H. M. El-Bakry, and A. A. Saleh, "Cloud security using Markov chain and genetic algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 96–106, 2018.

[4] Z. G. Chen, Z. H. Zhan, H. H. Li, *et al.*, "Deadline constrained cloud computing resources scheduling through an ant colony system approach," in *International Conference on Cloud Computing Research and Innovation*, IEEE, pp. 112–119, 2015.

[5] M. C. S. Filho, R. L. Oliveira, C. C. Monteiro, *et al.*, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," in *Integrated Network and Service Management*, IEEE, pp. 400–406, 2017.

[6] M. B. Gawali, S. K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 4, 2018.

[7] S. Guo, J. Liu, Y. Yang, *et al.*, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. pp: 319–333, 2018.

[8] H. Han, X. Wu, L. Zhang, *et al.*, "Self-organizing RBF neural network using an adaptive gradient multiobjective particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 69–82, 2017.

[9] N. Kumar, N. Chilamkurti, S. Zeadally, *et al.*, "Achieving quality of service (QoS) using resource allocation and adaptive scheduling in cloud computing with grid support," *Computer Journal*, vol. 57, no. 2, pp. 281–290, 2018.

[10] L. Liu, Z. Cao, C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29-35, 2018.

[11] A. Mohammed, N. Md Asri, "Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment," *PLoS One*, vol. 11, no. 6, pp. e0158229, 2016.

[12] A. D. Niros, G. E. Tsekouras, D. Tsolakis, *et al.*, "Hierarchical fuzzy clustering in conjunction with particle swarm optimization to efficiently design RBF neural networks," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 1, pp. 105–125, 2015.

[13] S. Rezaei, M. Ali Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115-131, 2018.

[14] K. Shanmukhi, P. R. Vundavilli, B. Surekha, "Modeling of ECDM micro-drilling process using GA- and PSO-trained radial basis function neural network," *Soft Computing*, vol. 19, no. 8, pp. 2193–2202, 2015.

[15] W. Wei, X. Fan, H. Song, *et al.*, "Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 78–89, 2018.

[16] J. Wu, J. Long, M. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *Neurocomputing*, vol. 148, no. 2, pp. 136–142, 2015.

[17] L. Xu, F. Qian, Y. Li, *et al.*, "Resource allocation based on quantum particle swarm optimization and RBF neural network for overlay cognitive OFDM System," *Neurocomputing*, vol. 173, no. 3, pp. 1250–1256, 2016.

[18] L. Zhang, W. Tong, S. Lu, "Task scheduling of cloud computing based on Improved CHC algorithm," in *International Conference on Audio, Language and Image Processing*, IEEE, pp. 574–577, 2015.

[19] W. Zhang, D. Wei, "Prediction for network traffic of radial basis function neural network model based on improved particle swarm optimization algorithm," *Neural Computing & Applications*, vol. 29, no. 1, pp. 1–10, 2016.

Biography

Gaoxiang Lou (1989-). Ph.D. candidate from Chang'an University. His research interests include Mixed flow assembly scheduling, Manufacturing system integration and automation, etc. Funding institutions: National Natural Science Foundation, China(51305042),Construction project of Central University Educational Reform Special Foundation China(jgy16049,jgy170501). Recently published papers: Research on the Hybrid Algorithm based on Differential Evolution and Genetic Algorithm for Mixed Model Assembly Scheduling, Improved hybrid immune clonal selection genetic algorithm and its application in hybrid shop scheduling, etc.

Zongyan Cai (1964-). Doctor of Education , Professor. Worked in Chang'an University. His research interests include Manufacturing system integration and automation, Intelligent robot technology, etc. Funding institutions: National Natural Science Foundation, China(51705030),Construction project of Central University Educational Reform Special Foundation China(jgy16049,jgy170501). Recently published papers: An enhanced bearing fault diagnosis method based on TVF-EMD and a high-order energy operator, An alternative demodulation method using envelope-derivative operator for bearing fault diagnosis of the vibrating screen, etc.