A Study into Load Balancing in Cloud Computing Based on Whale Optimization Algorithm

Xiaoxia Li
Artificial Intelligence and Big Data College, Zibo Vocational Institute, Zibo, China

Abstract

In view of the low efficiency and low utilization of load balancing under cloud computing, this paper proposes a cloud computing load balancing scheme based on whale optimization algorithm. Firstly, the balanced load under cloud computing is explained; secondly, based on the whale optimization algorithm, the population diversity is improved through population initialization, the sine and cosine algorithm is used for local optimization, and the leapfrog algorithm is used for individual screening; finally, the optimized Whale optimization algorithm is used in cloud computing load balancing. The simulation experiment shows that the algorithm in this paper has better effects in task execution time, fairness and energy consumption compared with ant colony algorithm, particle swarm algorithm, and whale optimization algorithm.

Keywords: *Cloud computing, particle swarm optimization, membrane computing*

I. Introduction

Cloud computing is the product of the integration and development of grid computing, distributed computing, and parallel computing technology. It greatly improves the efficiency of users' processing of data in the network. But the ensuing problem is that a large number of task requests and a large number of network devices need to be maintained, and the imbalance of network nodes is the problem of low utilization of cloud computing [1]. Scholars have conducted research from different directions. For example, Literature [2] proposed a load balancing algorithm based on the idea of artificial bee colony. In the load balancing decision, the overall load of the cloud system is calculated, and the following steps are initiated under the premise that the overall cloud system is not overloaded and the load is not balanced. In the virtual machine grouping, all virtual machines in the cloud system are classified into three groups: overload, low load, and load balance. The computing tasks that need to be migrated in the subsequent steps are usually on the overloaded virtual machine, and the migration destination is usually a low-load virtual machine. In computing task scheduling, tasks on overloaded virtual machines are migrated outward, and rules are designed to select appropriate low-load virtual machines as the migration destination. Experiments verify the effectiveness of the proposed algorithm; Literature [3] proposes a hierarchical workflow scheduling model, which is classified according to task priority; Secondly, based on the improvement of the hybrid frog leaping algorithm, the time greedy algorithm is used to optimize the initial population to improve the search efficiency, and the reconstruction strategy of the local optimal individual is added to jump out of the local optimal and enhance the global search ability. The experimental results show that after the same number of iterations, the load balance of ISFLA is optimal. In terms of the overall completion time of the workflow, ISFLA is also significantly lower than other algorithms; in terms of search efficiency, the search time of ISFLA is greatly reduced. Literature [4] proposed a resource load balancing scheduling method based on ant colony algorithm, which uses the pheromone update strategy in ant colony algorithm to obtain the resource load in cloud computing, and applies this scheme to resource load scheduling under cloud computing. Experiments show that the method has a high degree of balance and can effectively improve the load balance of cloud computing resources and improve the utilization of cloud computing resources; Jiang Literature [5] proposed a particle swarm optimization algorithm combined with random forest classifiers to solve the load balancing problem of virtual machines. The algorithm can not only balance the CPU utilization and memory utilization in the virtual machine, but also take the time length of the total task on the virtual machine as the optimization goal, so as to achieve the balance of the virtual machine resource utilization and effectively reduce the total waiting time of the task. The simulation results show that the algorithm can effectively solve the load balancing problem of the virtual machine; Literature [6] proposed a storage strategy based on Hopfield neural network (HNN), analyzed the resource characteristics that affect the storage efficiency, and then established the resource constraint model, designed the Hopfield energy function, and simplified the energy function. The experimental results show that this method can better realize the resource load balance in application, which will help to improve the storage capacity of Hadoop and speed up the retrieval; Literature [7] proposed an improved ant colony algorithm (LBIACO) to improve the load balance of virtual machines, which improves the defect that traditional pheromone updating is easy to fall into local optimization. The simulation results show that LBIACO algorithm had obvious advantages in task execution cost and execution time, and the resources of virtual machines can be effectively utilized, which improves the efficiency of the whole cloud computing system. Thus, the load balance of virtual machines in cloud computing is effectively maintained; Literature [8] proposed a multi-dimensional resource load balancing strategy. Firstly, by collecting the resource information of each physical node and virtual node in real time, using the newly improved weighted factor algorithm and the improved genetic algorithm, the optimal chromosome load, fitness and selection probability are calculated iteratically. The experimental results show that when the physical machine is seriously overloaded, the above methods can improve the load balance and the efficiency of each physical machine in a very short time, and can improve the performance of the cluster; Literature [9] proposed a virtual machine resource scheduling optimization method based on cat swarm optimization algorithm. Firstly, the mathematical model is constructed according to the resource scheduling optimization objective of virtual machine, and then the fitness function of cat swarm optimization algorithm is constructed by considering the shortest time and the optimal load, and the optimization of virtual machine resource scheduling optimal scheme is realized by simulating the daily behavior of cats.

On the basis of the above research, this paper uses whale optimization algorithm in cloud computing load scheduling. The whale optimization algorithm is initialized to increase the population diversity, the sine and cosine algorithm is used for local optimization, and the frog leap algorithm is used for individual screening. Finally, the optimized Whale optimization algorithm is used in cloud computing load balancing. Simulation experiments show that the improved Whale optimization algorithm has a better balance effect.

II. Brief Description of Load Balancing

Load balancing is a kind of network method which can realize the best utilization of computer cluster, network resources, CPU resources and other resources. Load balancing technology can maximize the throughput and minimize the response time. Virtualization and server cluster are the two most important parameters in cloud computing, and excellent load balancing technology can provide them with efficient service support. Load technology only distributes the user requests and task requests evenly to the server in a static form to improve the utilization of resources. With the massive growth of tasks under the cloud computing, different tasks are assigned to different nodes. Some nodes have weak computing power and need to take a long time to deal with them. The longer the waiting time is, the longer the processing delay will increase, and may result in the reduction of the overall system efficiency. As the status of nodes in the cluster changes continuously, the load balancing technology has also been rapidly developed, and the dynamic distribution has been realized. According to the state of each node in the cluster, the service application request can be allocated or migrated to the low load node. Through the virtualization technology, the overload node can be migrated to the low load node, which improves the response speed of the cloud computing center and makes reasonable use of resources. It avoids unnecessary power consumption and idle resource waste. Generally speaking, load balancing is divided into two stages, the first stage is the initial distribution, the second stage is redistribution. The first stage is to assign the task to each physical node before the task enters the node, so that the system can run efficiently. The second stage is that in the process of node operation, when the node is overloaded, the overloaded node service or virtual machine can be migrated to the node with light load to keep the load balance of the system.

Virtualization technology is a core technology in cloud computing. Through the dynamic allocation of resources, it can make full use of the storage resources and computing resources in the cloud computing. It makes software and hardware isolated from each other through the underlying hardware resources, logically expands physical resources, and makes full use of the resources of cloud computing data centers through the form of virtual resource sharing. Virtualization technology can provide the technical basis for the migration of virtual machines. The node target virtual machines with more tasks can be migrated to the local cloud data center or other nodes with less tasks to balance the load of the cloud system. In order to minimize virtual machine migration, the load resource scheduling in this article is defined as follows: Collection of data center node servers for cloud computing $H = \{H_1, H_2, \cdots H_n\}$,where the n is the number of servers, collection of deployed virtual machines in node j , $V_j = \{V_1, V_2, \cdots V_m\}$,where the m is the nodes Collection of virtual machines already deployed in the number of virtual machines of node j ,collection of resources for node servers $S_i = \{S_{imemory}, S_{icpu}, S_{ibroadband}\}$,where the $S_{imemory}$ represents the memory resources, S_{icpu} represents the CPU resources, $S_{ibroadband}$ represents broadband resources. The load of the node is shown in the formula (1-2)

$$L_i = w_1 \times C_i + w_2 \times M_i + w_3 \times N_i \tag{1}$$

$$w_1 + w_2 + w_3 = 1 \tag{2}$$

Where, L_i represents the load balancing of node i ; C_i , M_i and N_i represents the CPU, memory and broadband of the node; w_1 , w_2 and w_3 represents the weight. Set the load balance of the node to determine the balancing effect, where, m represents the number of resources, x_j represents the load value of the resource, μ is the average of the resources, σ is the variance of the resource load, which is shown as follows

$$\sigma = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(x_j - \mu)^2} \tag{3}$$

III. Whale Optimization Algorithm

Whale Optimization Algorithm [10] is an algorithm based on the population behavior of whales throwing fish in the sea. In the whale algorithm, each humpback whale is the candidate solution of the optimization problem. The algorithm gradually determines the optimal solution of the optimization problem through continuous iterative search.

3.1 Shrinkage mode

The shrinkage mode in the whale algorithm mainly means that the current best candidate solution is the target position or is close to the target position. When the best search agent position is determined, the whale individuals in other locations will run to the best search agent position and update their own position, which is expressed as follows:

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \bullet \vec{D} \tag{4}$$

$$\vec{D} = |\vec{C} \bullet \vec{X}^*(t) - \vec{X}(t)| \tag{5}$$

ISSN: 0010-8189

$$\vec{A} = 2\vec{a} \bullet \vec{r} - \vec{a} \tag{6}$$

$$\vec{C} = 2 \bullet \vec{r} \tag{7}$$

Where, t is the current number of iterations, $\vec{X}^*(t)$ is the best position in the iterations t, $\vec{X}(t+1)$ is the search agent location in the iterations $t+1$, $\vec{A}$ and $\vec{C}$ is the coefficient vector, $\vec{a}$ gradually decreasing from [2, 0], $\vec{r}$ is the random number vector between [0, 1].

3.2 Rotating bubble mode

Whale individuals often use bubbles to capture food. In order to simulate this behavior, the spiral equation is used between the whale position and the captured prey position to represent the humpback whale's spiral forward in the sea.

$$\vec{X}(t+1) = \vec{D}' \bullet e^{bl} \bullet \cos(2\pi l) + \vec{X}^*(t) \tag{8}$$

$$\vec{D}' = |\vec{X}(t) - \vec{X}^*(t)| \tag{9}$$

Where, b is the constant, l is a random number between [- 1, 1]

3.3 Search phase

When $\vec{A}$>1 or $\vec{A}$<-1,the whale algorithm will force the search agent individual to stay away from the reference humpback whale, and the random search agent individual will update the position as the best agent, so that the whale algorithm has the global search ability. Where, $\vec{X}_{rand}$ is the location of the random search agent.

$$\vec{D}'' = |\vec{C} \bullet \vec{X}_{rand} - \vec{X}| \tag{10}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \bullet \vec{D}'' \tag{11}$$

IV. Cloud Computing Resource Load Scheduling Based on Improved Whale Optimization Algorithm

Like most bionic algorithms, whale optimization algorithm also has shortcomings such as easy to fall into local optimum and slow convergence speed. Therefore, this paper optimizes the performance of the algorithm from the aspects of population initialization, local optimization, and individual location.

4.1 Population initialization

It can be found in the artificial whale optimization algorithm that the algorithm is initialized in a random manner, which affects the quality of the global optimal solution to a certain extent and reduces the efficiency of the algorithm. Therefore, this paper initializes the population of the artificial Whale optimization algorithm. The basic idea is to use the introduction of a pseudo-reverse learning strategy to generate a pseudo-reverse population, which increases the diversity of the population. In this way, the Whale group can carry out the behavior of carefully

ISSN: 0010-8189

searching for food in the neighboring cells, avoiding skipping the optimal solution, and then selecting the elite from the current population and the pseudo-reverse population, thereby effectively finding the optimal solution. The detailed process is given:

Step1: Suppose $X=(x_1,x_2,\cdots x_n)$ is a n dimensional solution, $x_1,x_2,\cdots x_n\in R$ and $x_i\in[l_i,u_i]$, $i\in\{1,2,\cdots,n\}$, and then define the reverse solution $OX=(\hat{x}_1,\hat{x}_2,\cdots\hat{x}_n)$

$$\hat{x}_i=l_i+u_i-x_i \tag{12}$$

Step2: On the basis of the reverse solution, define the pseudo reverse solution $QOX=(\hat{x}_1^q,\hat{x}_2^q,\cdots\hat{x}_n^q)$, which is expressed as:

$$\hat{x}_i^q=rand\left(\frac{l_i+u_i}{2},\hat{x}_i\right) \tag{13}$$

The pseudo reverse solution QOX is closer to the optimal solution than the reverse solution OX. If the adaptability of the pseudo-reverse solution is obviously dominant, replace the current solution X with the pseudo-reverse solution QOX; otherwise, the current solution X remains unchanged. In this way, the pseudo-reverse learning strategy is used to generate the position of the global leader before updating the position of the Whale group with the formula (13), so as to find the optimal solution more accurately, thereby improving the optimization performance of the algorithm.

4.2 Improved particle swarm optimization and membrane computing

Sine and Cosine Algoritm (Sine and Cosine Algoritm, SCA) is a new type of swarm intelligence optimization algorithm proposed by Professor Mirjalili [11]. The algorithm has unique advantages. The periodicity and volatility of sine and cosine functions make the sine algorithm have the characteristics of “global search” and “local development” within a limited range. This allows the algorithm to search in a large range during the iterative process to ensure the comprehensiveness of the iterative update results. At the same time, within a local scope or after iterative update, it can get rid of the problem that the swarm intelligence algorithm is easy to fall into the local optimum. The expression is as shown in (14):

$$x_i^{t+1}=\begin{cases}x_i^t+r_1\sin(r_2)\times|r_3x_j^t-x_k^t|,r_4<0.5\\ x_i^t+r_1\cos(r_2)\times|r_3x_j^t-x_k^t|,r_4\geq 0.5\end{cases}(k,j\neq i) \tag{14}$$

In the formula, x_i^{t+1} refers to the location of individual i at the $t+1$ cyclic update when individuals in the current population iterate after choosing an update method, x_i^t and x_k^t are the specific locations of any two individuals in the t cyclic update. At the same time, the selection of these two positions is random and different from the position of the individual i, r_1, r_2, r_3 and r_4 are two control parameters, in which r_1 controls the search direction and $r_1=a-a\times\frac{t}{t_{\max}}$ ($t_{\max}$ is the maximum iteration times, t is the iteration times, and a is a constant number), r_2 is the controlled search distance, and $r_2 \sim U[0,2\pi]$. r_3 controls two randomly selected population individuals and $r_3 \in (0,+\infty)$, r_4 controls the update switch of this sine and cosine and $r_4 \sim U[0,1]$.

The embedding of the sine-cosine optimization strategy, on the one hand, can well fill the defect of dependence on the position update formula of the whale optimization algorithm. Regardless of whether it is a sine mechanism or a cosine mechanism, individual whales can communicate with the food source to facilitate the transmission of optimal information in the population. Each individual whale can better use the positional difference information between itself and the food source to prompt the individual whale to move toward the optimal solution; on the other hand, this allows the individual whale to further perform global and local searches in different ranges of the same search space. The sine mechanism enables the global search to find the optimal solution, reduces the optimization blind spot of the cosine mechanism, and reduces the whale squatting ascidian individuals into the local optimum. The cosine mechanism allows local development to fill the shortcomings of the full global search convergence speed of sine, improve exploration capabilities, and accelerate algorithm convergence. The mutual use of sine and cosine can better balance the exploration and development capabilities of the algorithm, and jointly promote the optimization of algorithm performance.

4.3 Individual location optimization

The basic whale optimization algorithm has insufficient global search capabilities, while the Shuffled Frog Leading Algorithm (SFLA) has good global search capabilities. In this paper, the leapfrog algorithm is used to update the individual positions of the whale optimization algorithm, so that the individual corresponding to the optimal target value is the individual whale. Leapfrogging algorithm is a heuristic optimization algorithm, which executes heuristic search by executing heuristic function to obtain the global optimal solution. The idea is to first decompose the frog group into different numbers of subgroups, secondly search in the subgroups according to a certain strategy, and finally conduct a global exchange.

Suppose there are a total of M frogs, and the number of sub-groups is k, the number of candidate solutions in the group is n. $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ is the i candidate solution, in which D refers to the dimension of candidate solutions. If the candidate solution obtained in the process of subgroup search is the worst, it needs to be updated. Otherwise, you need to use the candidate solution replacement at this time to continue the search. After all the subgroups complete the internal search, all subgroups are re-divided and searched in the group until the optimal solution is found, the end condition ends.

4.3.1 Mutation operation

In the early stage of the algorithm, in order to maintain the diversity of the population and improve the global search ability, the worst individual in the subgroup was updated according to formula (12), and three individuals were randomly selected. One of the individuals is used as the target individual, and the other two individuals are used to update the moving step length, using the rand difference operator. Herein, X_{r^1} is the target individual, X_{r2} and X_{r^3} are two other individuals randomly selected, X'_w is a new individual, $F_1 \in (0,1)$, $F_2 \in (0,1)$,and $F_1 + F_2 = 1$. In the later stage of the algorithm, in order to help converge to the best point, use the best individual in the subgroup as the target individual, introduce the best mutation of the difference operator, and update the strategy as formula (15-16). In the formula, X_{r2} and X_{r^3} are two other individuals randomly selected, X'_w is a new individual, and X_b is the optimal individual $F \in (0,1)$ of the group.

$$X'_w = X_{r^1} + F_1 * (X_{r2} - X_{r^3}) \tag{15}$$

ISSN: 0010-8189

$$X_w^{'} = X_b + F_2 * (X_{r2} - X_{r^3}) \tag{16}$$

4.3.2 Selection operation

After an iteration, the fitness of the new individual $X_w^{'}$ and the worst individual X_w of the subgroup are evaluated. According to the laws of nature, individuals with better fitness values are selected to enter the next generation population, as shown in formula (17).

$$X_w^{'} = X_w \ \ if \ \ f(X_w^{'}) >= f(X_w) \tag{17}$$

4.3.3 Cross operation

In order to further provide the local search ability of the algorithm and maintain the diversity of the population, the crossover operation is introduced to improve, and the update strategy is as (18).

$$X_w^{'j} = \begin{cases} X_w^{'j} & rand_i^j \geq CR \\ rand(0,1) & otherwise \end{cases} \tag{18}$$

In the formula, $X_w^{'j}$ is the value of the current individual at the j dimension, CR is the crossover factor, $rand(0,1)$ is a random factor.

Let the number of all frog subgroups be m , and the optimal frog individuals in each subgroup are: $P_b(1), P_b(2), \cdots P_b(m)$, the global optimal solution is P_g . Choose any two individuals in $P_b(1)$ and $P_b(m)$ as the father generation, and conduct crossover at P_g , and then generate the later generations according to formula (19).

$$\begin{cases} P(i) = r_1 P_b(i) + r_2 P_b(j) + r_3 P_b g \\ P(j) = r_1 P_b(j) + r_2 P_b(i) + r_3 P_b g \end{cases} \tag{19}$$

In the formula, $r_1 + r_2 + r_3 = 1$, and the size of $r_1, r_2, r_3 \in (0,1)$, r_1, r_2, r_3 determines the size of the cross domain between the parent individuals. Use an elite retention strategy in the frog population to eliminate poor individuals. However, as P_b represents the internal optimal individuals in each subgroup, it will fall into local optimal. Therefore, crossover operations between the optimal individuals in different subgroups can avoid falling into the local optimal and achieve the global optimal.

4.4 Algorithm steps

Step1: Initialize the relevant parameter values under the resource load under cloud computing, map the individual whales to the cloud computing load scheduling plan, and set the maximum number of iterations;

Step 2: Initialize the population of Whale optimization algorithm according to section 4.1;

Step 3: Local optimization of individual whales according to section 4.2;

Step 4: The individual whales after each iteration are selected according to section 4.3;

Step 5: When the number of iterations of the algorithm reaches the maximum number of iterations, execute Step 6, otherwise execute Step 2;

ISSN: 0010-8189

Step 6: The best individual whale is the best scheduling plan.

V. Experimental Simulation

In order to better illustrate the effect of the algorithm in this paper in cloud computing load balancing, this paper compares the ant colony algorithm, particle swarm algorithm, whale optimization algorithm and the algorithm in this paper. This paper uses CloudSim platform to perform simulation, user task input vector, and specific parameters are shown in Table 1. This experiment contains 5 types of virtual machines, CPU, Memory, Bandwidth configuration parameters are shown in Table 2.

Table 1 Description table of user input task

Task	Task description
CloudletId	Task ID
C1oudletType	Task type
C1oudletLength	Task length
CloudletIn_size	Task input file size
CloudletOutput_size	The bandwidth actually required by the task
ActualBw	The bandwidth actually required by the task
Priority	User priority

Table 2 VM configuration parameters

CPU(MIPS)	Memory(MB)	BandWidth(Kbit/S)
800	2048	200000
1000	1024	300000
1500	1024	300000
500	512	150000
2000	512	250000

5.1 Task execution time comparison

Figure 1 shows a comparison diagram of the execution time of the five types of virtual machine resources that are allocated to different numbers of tasks through four algorithms. It can be seen from Figure 1 that when the tasks are 50, 80, 100, and 150, the execution time of the algorithm in this paper is less than that of WOA, PSO and ACO, achieving the goal of high task execution efficiency. Because the algorithm in this paper is in the process of optimizing, according to the classification of user tasks, tasks with preferred time are allocated to virtual machine resources with high CPU and memory configuration. In addition, the task of preferring bandwidth is allocated to the virtual machine resources with high bandwidth configuration. It will not allocate all tasks to the virtual machine resources with high CPU and memory as much as possible like the other three algorithms, resulting in virtual machines with high CPU and memory. The resource load is too heavy and the task execution time is too long. Therefore, the execution time of the algorithm in this paper will be slightly reduced compared to the other three algorithms.

ISSN: 0010-8189

www.converter-magazine.info

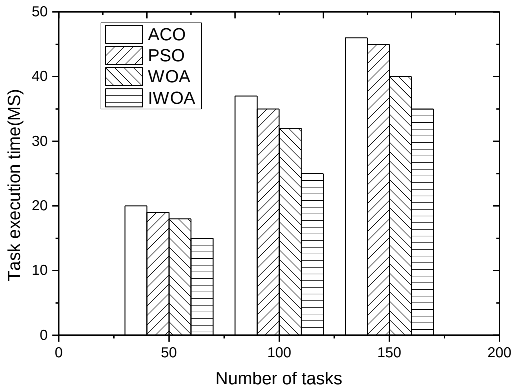

Fig 1: Comparison of execution time of four algorithms

5.2 Fairness comparison

It is generally believed that task fairness is manifested in the analysis of task bandwidth. Asymmentric Digital Subscriber Line ADSL (Asymmentric Digital Subscriber Line ADSL) usually, for downlink speed and uplink speed, when the supplier provides 1.5Mb/s and 256kb/s- 1Mb/s, the downlink bandwidth actually has an occupancy rate of 71%-84.7%. When comparing fairness in this experiment, the actual occupancy rate of the downlink bandwidth is selected as the reference for this experiment. If the bandwidth requested by the task is 10Mb/s, and the virtual machine resource provides 7Mb/s, the bandwidth occupancy rate is 70%. According to the principle of fairness, the bandwidth-preferred tasks with the number of tasks of 50, 100, and 150 are selected in the article, and the four algorithms are used to schedule the tasks respectively. The proportion of the downlink broadband occupancy rate is shown in Figure 2

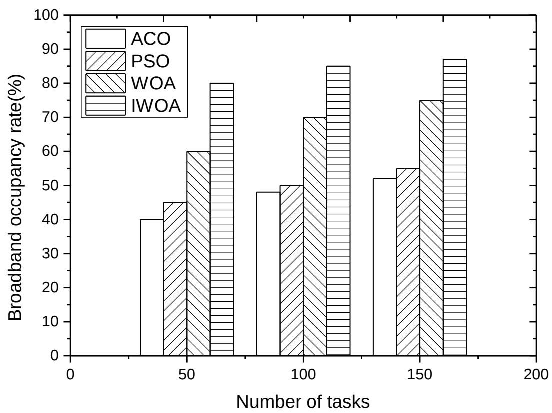

Fig 2: Comparison of broadband occupancy of four algorithms

5.3 Energy consumption comparison

The energy consumption comparison diagram of the five types of virtual machine resources configured above is allocated to different numbers of tasks through four algorithms as shown in Figure 3. As can be seen from Figure 1, when the tasks are 50, 80, 100, 150, the energy consumption of the algorithm in this paper is less than that of WOA, PSO and ACO. This shows that the optimized algorithm can reasonably allocate tasks to virtual machine resources with high CPU and memory configuration according to the classification of user tasks. Assign bandwidth-preferred tasks to virtual machine resources with high bandwidth configuration, reducing the energy consumption of the virtual machine. Compared with the other three algorithms, this algorithm reduces energy consumption compared to the other three algorithms.

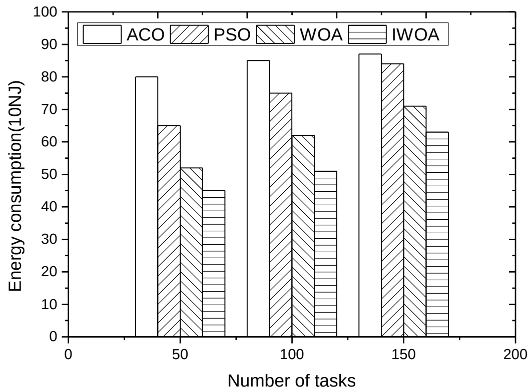

Fig 3: Comparison of energy consumption of four algorithms

VI. Conclusion

In this paper, the Whale Optimization Algorithm is used in the load balancing scheduling of cloud computing, and the performance of the Whale Optimization Algorithm is improved through methods such as population initialization, local optimization and individual selection. Simulation experiments show that the algorithm in this paper has good effects in terms of task execution time, fairness and energy consumption.

References

[1] P. Gupta, M.M. Goyal, P. Kumar. "Trust and reliability based load balancing algorithm for cloud Iass." In Advance Comuting Conference. New York: IEEE, pp. 65-69, February 2013.

[2] J. Jia and D.J. Mu. "Load Balancing in Cloud Computing by Artificial Bee Colony Algorithm." Science Technology and Engineering, vol. 20, no.16, pp. 6532-6537, June 2020.

[3] J. Xu, Q.H. Xiang, G. Xiao. "Load Balancing Scheduling Optimization of Cloud Workflow Using Improved Shuffled Frog Leaping Algorithm." Computer Science, vol. 46, no. 11, pp. 315-322, November 2019

[4] L.X. Meng and L.W. Meng. "Simulation of Resource Load Balancing Scheduling under Cloud Computing." Computer Simulation, vol. 35, no. 4, pp. 386-389, April 2018.

[5] W. Jiang, Z.H. Liu, Q.R. Qiu. "A virtual machine load balancing algorithm in cloud environment." Journal of Fuzhou University (Natural Science Edition), vol. 6, no. 4, pp. 451-457, April 2018.

[6] Q. Li and X.F. Liu. "Load balancing strategy of cloud storage based on Hopfield neural network." Journal of Computer Applications, vol. 37, no. 8, pp. 2214-2217, August 2017.

[7] Q.B. Nie. "Optimization management of task scheduling for cloud resource load balance." Computer Engineering and Design, vol. 38, no. 1, pp. 18-21, January 2017.

[8] C.H. Li and Y.B. Xie. "Optimal Scheduling of Resource Load Balancing in Cloud Computing Environment." Computer Simulation, vol. 12, pp. 420-425, December 2017

[9] Z.C. Kuang and S.Y. Xiei. "Load Balancing Scheduling of Virtual Machine Resource in Cloud Computing Based on Cat Swarm Optimization Algorithm." Journal of Jilin University (Sci Ed), vol. 54, no. 5, pp. 1117-1122, May 2016.

[10] S. Mirjalili and A. Lewis. "The whale optimizaion algorithm." Advances in Engineering Software, vol. 95, pp .51-67, May 2016.

[11] S. Mirjalili. "SCA: A sine cosine algorithm for solving optimization problems." Knowledge Based Systems, vol. 96, pp. 120-33, March 2016.