

Task Scheduling of an Improved Cuckoo Search Algorithm in Cloud Computing

Wenli Liu^{*}, Cuiping Shi, Hongbo Yu, and Hanxiong Fang

Qiqihar University, Qiqihar, 161006, China

Abstract

In view of the low efficiency of task scheduling in cloud computing, this paper introduces the cuckoo algorithm to optimize task scheduling. Firstly, the cloud computing task scheduling model is established. Secondly, the particle swarm algorithm and quantum algorithm are introduced for the short search ability of the cuckoo algorithm and the low precision of optimization. The cuckoo is fixed as a "particle" in the search direction in three-dimensional space, so that it cannot be randomly offset. Through the binary algorithm, the particle can be made faster by having the Levy flight randomly generate the step size. The optimal solution direction moves, which speeds up the convergence speed of the algorithm and avoids the blindness in the search process. By using four classical benchmark functions, the simulation results show that the improved algorithm has better performance and improves the efficiency of task scheduling and scheduling under cloud computing.

Keywords: swarm intelligence; cuckoo; particle swarm; quantum

(Submitted on March 18, 2019; Revised on May 15, 2019; Accepted on June 15, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

Task resource scheduling in cloud computing is an important part of cloud computing. Effectively and reasonably allocating the relationship between task resources is key to the efficiency of cloud computing [1]. Resource scheduling that introduces intelligent algorithms into cloud computing has become the mainstream of current research. Domestic and foreign scholars use a single intelligent algorithm for cloud computing resource scheduling, such as the PSO algorithm [2-3], genetic algorithm [4-5], ant colony algorithm [6-7], artificial bee colony algorithm [8], and fireworks algorithm [9]. The integrated intelligent algorithm is also used for cloud computing resource scheduling. Chen [10] proposed the fusion of particle swarm optimization and the ant colony algorithm for cloud computing resource scheduling. The disadvantage of this approach was the lack of comparison with other intelligent algorithms. Zhao [11] proposed the genetic ant colony algorithm in cloud computing resource scheduling and improved the solution time of the algorithm. Wang [12] proposed a resource scheduling algorithm based on the production function to maximize the revenue of cloud service providers, and the simulation results showed that the scheduling algorithm has better performance. Chen [13] proposed a new approach for efficient skew handling in outer joins over a cloud computing environment. Experimental results showed that the approach is scalable and, at least in cases of high skew, significantly faster than the state-of-the-art methods.

Cloud computing task scheduling is essentially a nonlinear optimization problem. The meta-heuristic algorithm has attracted the attention of the academic community in recent years due to its excellent performance of solving complex problems such as in the engineering industry. It is better than traditional algorithms in dealing with some nonlinear, multi-constrained, multi-variable, and other complex problems, and it has good search ability. The cuckoo algorithm is the most representative algorithm in the meta-heuristic algorithm. It has the advantages of simplicity and ease, and it does not require a large number of parameters to solve special problems. Based on the above research, the cloud computing task scheduling scheme corresponds to the cuckoo individual. The problem of the cuckoo algorithm is analyzed. Firstly, the particle swarm-

^{*} Corresponding author.

E-mail address: Qqhr1w1982@sina.com

quantum algorithm is used for local optimization. Secondly, through the binary algorithm and Levy flight improvement algorithm, the individual can move to the optimal solution direction more quickly, avoiding the blindness of the search.

2. Cloud Computing Task Scheduling

In a cloud computing environment, the allocation of task resources is a very important component. The most critical part of the overall cloud computing resource scheduling is the resource management part. The virtual resources in the cloud computing environment are allocated according to demand and automatically grow. However, due to the limitations of the node itself, the amount of growth will not exceed the upper limit of the node itself. At the same time, when performing the same virtual resource task allocated by the parallel processing resource management, the execution time difference between each sub-task is required to reach a minimum as much as possible. Therefore, according to the characteristics of task scheduling, the content of the task load in the cloud computing environment involves the time, cost, security, and reliability of each task. This paper is represented by four functions:

(1) Task time function $T(i, j)$

$$T(i, j) = \begin{cases} \frac{t_{ij} - t(i, j)}{t_{ij}}, & t(i, j) < t_{ij} \\ 0, & t(i, j) \geq t_{ij} \end{cases} \quad (1)$$

In Equation (1), t_{ij} represents the expected execution time of task t_j on virtual machine x_i and $t(i, j)$ is the actual execution time of task t_i on virtual machine x_i .

(2) Task cost function $C(i, j)$

$$C(i, j) = \begin{cases} \frac{c_{ij} - c(i, j)}{c_{ij}}, & c(i, j) < c_{ij} \\ 0, & c(i, j) \geq c_{ij} \end{cases} \quad (2)$$

In Equation (2), c_{ij} represents the expected cost of task t_j on the virtual machine x_i and $C(i, j)$ is the actual cost of task t_i on virtual machine x_i .

(3) Task efficiency function $E(i, j)$

$$E(i, j) = \begin{cases} \frac{e_{ij} - e(i, j)}{e_{\max} - e(i, j)}, & e(i, j) < e_{ij} \\ \frac{e_{ij} - e(i, j)}{e(i, j) - e_{\min}}, & e(i, j) \geq e_{ij} \end{cases} \quad (3)$$

In Equation (3), e_{ij} represents the expected efficiency of task t_i on virtual machine x_i and $e(i, j)$ is the actual efficiency of task t_j on virtual machine x_i .

Through the analysis of the above three factors, this paper sets α_1 , α_2 , and α_3 as the weight values of task time, task cost, task efficiency, and task security for all nodes in cloud computing. To satisfy $\alpha_1 + \alpha_2 + \alpha_3 = 1$, the cloud server dynamically adjusts the weight value according to the different aspects of the task load task requirements (that is, the objective function).

$$F(i, j) = \alpha_1 \times T(i, j) + \alpha_2 \times C(i, j) + \alpha_3 \times E(i, j) \quad (4)$$

Therefore, the maximization of the objective function is an important reference to measure the effectiveness of cloud computing task scheduling.

3. Cloud Computing Task Scheduling based on Improved Cuckoo Search Algorithms

3.1. Cuckoo Search

Cuckoo search (CS for short) is a heuristic algorithm based on natural elements that was proposed by English scholars Xin-She Yang and Sua Deb in 2009. The idea of this algorithm comes from cuckoos' nest parasitic behavior and birds' unique Levy behavior. The algorithm is described as follows:

(1) N nests are randomly generated, and the best nest location is chosen: $X_0 = (x_1^0, x_2^0, \dots, x_N^0)$. Retain this location for the next generation.

(2) By updating Levy's flight position, the next generation position in each nest is obtained and compared with the previous generation's nest position. Take the better position of the two as the current position.

(3) A random number $r \in [0, 1]$ is compared with the probability P_a of finding exotic eggs by host birds. When $r > P_a$, the position of the nest is randomly changed; otherwise, it does not change. Compared with the previous generation's nest location, select a better group of nest location $X_t = (x_1^t, x_2^t, \dots, x_N^t)$.

(4) When the position of the bird's nest meets the requirement of accuracy or termination iteration condition, the bird's nest is the global optimal solution; otherwise, it will continue to rotate (2). In the cuckoo search algorithm, each egg in the nest represents a solution, and the cuckoo egg represents a new solution. The algorithm is as follows:

$$x_i^{t+1} = x_i^t + \alpha \otimes L(\lambda) \quad (5)$$

In the equation, x_i^{t+1} represents the t nest position in the i generation. \otimes is a point-to-point multiplication, α is a step, and $L(\lambda)$ is a random search path. After updating the location, the random number r is compared with P_a . If $r > P_a$, change x_i^{t+1} ; otherwise, it will remain unchanged.

3.2. Local Optimization-Particle Swarm Optimization-Quantum Algorithm

Considering the characteristics of cuckoos' movement, each cuckoo is regarded as a particle. When the position of particles is updated, Levy's flight randomness produces a large jump. This increases the running time of the algorithm and the number of iterations of the algorithm. Therefore, the robustness and convergence of the algorithm cannot be guaranteed. In this paper, a quantum algorithm is introduced so that the search direction is not random when the position of each iteration is updated. Therefore, it can increase the convergence speed of the algorithm, enhance the algorithm to find the optimal solution, improve the efficiency of local optimization, and avoid the algorithm falling into local convergence.

In quantum mechanics, the wave function $\Psi(X, t)$ is used to represent the state of particles, and $X = (x, y, z)$ is used to represent the position coordinate vector of particles in three-dimensional space. The meaning of the wave function is that when a particle appears at a point X , the probability density is the square of the modulus of the wave function.

$$|\Psi|^2 dx dy dz = Q dx dy dz \quad (6)$$

In the equation, Q represents the probability density function. Normalize it as follows:

$$\int_{-\infty}^{+\infty} |\Psi|^2 dx dy dz = \int_{-\infty}^{+\infty} Q dx dy dz = 1 \quad (7)$$

The Schrödinger equation is used to describe the transformation of particle states in quantum mechanics.

$$ih \frac{\partial}{\partial t} \psi(X, t) = \hat{H} \psi(X, t) \quad (8)$$

In the equation, \hat{H} is a Hamiltonian operator, expressed in the following form:

$$\hat{H} = -\frac{h^2}{2m} \nabla^2 + V(X) \quad (9)$$

In the equation, m denotes the mass of the particle and h is the Planck constant. From the point of view of quantum mechanics, $V(X)$ represents the potential field of a particle.

In quantum systems, every particle has quantum behavior, and the state of each particle is described by a wave function. Through the analysis of the convergence behavior of particles in particle swarm optimization, it is concluded that there is a point p and there is an attraction centered on this point. Therefore, a potential field can be established at point p as follows:

$$V(x) = -\gamma \delta(X - p) = -\gamma \delta(Y) \quad (10)$$

In the equation, $Y = X - p$ and m denotes the mass of particles, so the corresponding Hamiltonian operator is:

$$\hat{H} = -\frac{h^2}{2m} \frac{d^2}{dY^2} - \gamma \delta(Y) \quad (11)$$

The Schrödinger equation for particles is:

$$\frac{d^2 \psi}{dY^2} + \frac{2m}{h^2} [E + \gamma \delta(Y)] \psi = 0 \quad (12)$$

By solving the wave function of the equation, the following results are obtained:

$$\psi(Y) = \frac{1}{\sqrt{L}} e^{-\frac{|Y|}{L}} \quad (13)$$

The corresponding probability density function Q is:

$$Q(Y) = |\psi(Y)|^2 = \frac{1}{L} e^{-2\frac{|Y|}{L}} \quad (14)$$

The function F of its probability distribution is expressed as:

$$F(Y) = 1 - e^{-2\frac{|Y|}{L}} \quad (15)$$

The quantum state function $\psi(Y)$ can only express the probability density function $Q(Y)$ of particles at point p . Through Monte Carlo stochastic simulation, the position of particles is obtained as follows:

$$x_i = p_i \pm \frac{L}{2} \ln\left(\frac{1}{u}\right) \quad (16)$$

In the equation, u denotes the random number between (0,1) and L is the characteristic length. L is expressed as $L(t)$ so that the position of particles can be changed with time and convergence. It can be found from the equation that L represents the search space of particles. When the value of L is too large, the particle swarm will diverge, and then the

convergence speed and convergence ability will be reduced. On the contrary, the premature convergence of the whole particle swarm leads to local optimum. Therefore, the average value of the optimal position of particles is set to $mbest$.

$$mbest = \frac{1}{M} \sum_{i=1}^M pbest_i = (\frac{1}{M} \sum_{i=1}^M pbest_{i1}, \frac{1}{M} \sum_{i=1}^M pbest_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M pbest_{iD}) \quad (17)$$

Therefore, $L_i(t) = 2\beta \cdot |mbest - x_i(t)|$, the evolution equation of particles is as follows:

$$x_{ij}(t+1) = p_{ij}(t) \pm \beta \cdot |mbest - x_i(t)| \cdot \ln(\frac{1}{u_{ij}(t)}) \quad (18)$$

3.3. Global Optimization using Binary Concepts

In order to improve the efficiency of the cuckoo search algorithm, the concept of binary systems is introduced into the cuckoo search algorithm in reference [14]. Position updates are performed by Levy flight. On the premise of keeping the random step size of Levy flight, quantum search is used to control the iteration direction, so that the particles can move to the optimal solution faster. This method not only retains the original random search ability in PSO-QA, but also increases the convergence speed of the algorithm and avoids blindness in the search process.

Select a binary encoding string with length N . This coding string represents the value of the j dimension variable in the i nests of the m generation. Use $x_{ijk}^m = (0,1)$ to represent the result of the encoding. According to the transformation equation, the binary equation of Levy flight is obtained as follows:

$$sig(Step) = \frac{1}{1 + \exp(-Step)} \quad (19)$$

$$x_{ijk}^m = \begin{cases} 1, & rand() \leq Sig(Step) \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

When $Step \leq 0$, then:

$$sig(Step) = 1 - \frac{2}{1 + \exp(-Step)} \quad (21)$$

$$x_{ijk}^m = \begin{cases} 0, & rand() \leq Sig(Step) \\ x_{ijk}^m, & \text{otherwise} \end{cases} \quad (22)$$

When $Step > 0$, then:

$$sig(Step) = \frac{2}{1 + \exp(-Step)} - 1 \quad (23)$$

$$x_{ijk}^m = \begin{cases} 1, & rand() \leq Sig(Step) \\ x_{ijk}^m, & \text{otherwise} \end{cases} \quad (24)$$

According to Equation (18), suppose $\eta = l_{ij}(t+1) - x_{ij}(t)$, where η represents the updated direction vector of the particle position and $\frac{\eta}{|\eta|}$ represents the vector value in the unit direction of the vector. According to Equation (5), suppose $\varepsilon = |v_{i+1}(t) - v_i(t)|$, where ε represents the direction vector for position updating through Levy flight, and then $|\varepsilon|$

represents the step size. For the same particle in space, the particle swarm optimization-quantum algorithm and Levy flight are used to update the position simultaneously, and the position obtained by the particle swarm optimization-quantum algorithm is updated to the direction. Taking the modulus of Levy's flight step as the size, the actual position of particles is updated, and the position update equation based on the particle swarm optimization quantum binary cuckoo search algorithm is obtained.

$$x_{ij}(t+1) = x_{ij}(t) + \frac{l_{ij}(t+1) - x_{ij}(t)}{|l_{ij}(t+1) - x_{ij}(t)|} |v_{i+1}(t) - v_i(t)| \quad (25)$$

3.4. Algorithm Steps

Step 1 The cloud computing task scheduling scheme is matched with the cuckoo search algorithm's individual location, and the best individual location found is the best task scheduling scheme;

Step 2 Initialize the nest location, set the values of probability P_a , precision requirements, search space range, and quantum factor β ;

Step 3 The fitness function of the bird's nest is calculated, and the optimal fitness values $fbest$ and $bestNest$ are retained;

Step 4 The location of the nest is binary coded. Levy flight is carried out according to Equations (19)-(24) to obtain the position update step. At the same time, Equations (16)-(18) are used to search these nests by particle swarm optimization and quantum search to obtain new directions. Finally, Equation (21) is used to update the location of the nests;

Step 5 Comparing the new nest location with the previous generation nest location, the location with good fitness is selected for preservation, the current optimal fitness value is recorded as new , and the corresponding location information is stored in $best$;

Step 6 Comparing the random number with P_a , if the random number is greater than P_a , the new host nest can be found randomly; otherwise, the existing host nest location will remain unchanged;

Step 7 According to the optimal fitness of the new nest location in step 5, it is stored in new , and the corresponding location information is transferred to $best$;

Step 8 Comparing the values of new and $fbest$, if the former is better than the latter, then new covers $fbest$. At the same time, $best$ covers the value of $fbest$; otherwise, it will remain unchanged;

Step 9 Determine whether $fbest$ meets the accuracy requirement. If the accuracy is achieved, step 10 is changed; otherwise, step 3 is changed;

Step 10 Determine whether the number of iterations reaches the maximum. If so, proceed to step 11; otherwise, return to step 3;

Step 11 The best cuckoo individual is the best cloud computing scheduling scheme.

4. Simulation Experiment

4.1. Algorithm Performance Comparison

In order to better illustrate the advantages of this algorithm, the algorithm is compared with the CS algorithm and ICS algorithm. Four test functions are selected to test the three algorithms, and the simulation curves are obtained by using the simulation platform of MATLAB 2012. Among them, the three test functions are the Sphere function, Rosenbrock function, and Schwefel function. Both the CS and ICS algorithms adopt the parameters of reference [15]. The number of nests initialized by both algorithms is 30. In order to ensure the objectivity and consistency of the evaluation, the same number of

neests is selected in this paper. P_a chooses 0.25, β is 0.5, the other parameters are set in the same way, and the setting accuracy is 0.00001. Each algorithm is run 50 times. The comparison results are given in Tables 1 to 3, and the simulation curves are shown in Figures 1 to 4.

Table 1. The optimization results of the three algorithms of Sphere function

Dimension	Algorithm	Optimal value	Worst value	Average value	Iterative algebra
10	CS	1.1447e-13	5.3327e-15	3.4232e-14	480
	ICS	3.687e-28	4.7832e-30	3.7323e-29	291
	Algorithm in this paper	3.8072e-42	5.7839e-44	4.7832e-41	220
20	CS	1.2721e-8	7.3249e-10	3.3748e-9	953
	ICS	2.8732e-11	4.1013e-13	3.6712e-12	563
	Algorithm in this paper	3.4258e-14	1.7637e-16	2.3802e-15	460
50	CS	40.0871	77.9348	59.3921	-----
	ICS	0.1328	0.2573	0.01542	-----
	Algorithm in this paper	1.3018e-03	2.9834e-05	2.0128e-04	-----
100	CS	9.8276e+03	1.3873e+04	5.9823e+04	-----
	ICS	1.5283e+03	1.2432e+03	1.3553e+03	-----
	Algorithm in this paper	67.2617	1.3981e+02	1.0783e+02	-----

Table 2. The optimization results of the three algorithms of Rosenbrock function

Dimension	Algorithm	Optimal value	Worst value	Average value	Iterative algebra
10	CS	0.6123	1.3829	0.8732	-----
	ICS	0.1137	0.7236	0.0442	-----
	Algorithm in this paper	1.8072e-38	4.7849e-40	2.7832e-39	-----
20	CS	1.3823e+01	5.6249e+01	3.3648e+01	750
	ICS	2.3901e+01	5.2334e+01	3.4712e+01	-----
	Algorithm in this paper	1.1256e+01	2.2354e+01	1.5892e+01	-----
50	CS	42.0871	64.9418	53.3845	-----
	ICS	49.1328	58.2583	55.3283	-----
	Algorithm in this paper	44.9472	45.4791	45.5981	-----
100	CS	2.9276e+02	3.7573e+02	3.2123e+02	-----
	ICS	2.5483e+03	3.3132e+03	1.3553e+03	-----
	Algorithm in this paper	1.1216e+02	1.8530e+02	1.6383e+02	-----

Table 3. The optimization results of the three algorithms of Schwefel function

Dimension	Algorithm	Optimal value	Worst value	Average value	Iterative algebra
10	CS	7.9006e-08	1.8912e-10	1.4289e-9	772
	ICS	8.8761e-18	1.2796e-20	7.1287e-19	284
	Algorithm in this paper	4.2378e-25	4.7849e-27	2.7832e-26	-----
20	CS	5.2038e-02	6.6249e-04	3.3648e-03	748
	ICS	9.0238e-09	6.2334e-10	1.4712e-10	676
	Algorithm in this paper	1.1393e-12	1.2354e-10	1.1892e-11	-----
50	CS	0.2314	0.2923	0.02591	-----
	ICS	0.0114	0.0172	0.0143	563
	Algorithm in this paper	6.9538e-04	1.3507e-06	45.5981	-----
100	CS	5.9342	6.7391	3.2123e+02	-----
	ICS	7.2664	17.3602	1.3553e+03	-----
	Algorithm in this paper	0.3437	0.6009	1.6383e+02	-----

4.2. Cloud Computing Task Scheduling

In order to further illustrate the effectiveness of the proposed algorithm in resource scheduling in cloud computing, the algorithm is compared with CS and ICS in terms of load balancing, time consumption, and cost expenditure of virtual machines. The hardware environment includes Core I3 CPU, 4GDDR memory, 1000G hard disk capacity, and Linux operating system. Cloudsim is used to simulate the cloud computing environment. Three different task sets are selected, namely Task 1 [100,1000], Task 2 [1000,10000], and Task 3 [10000,100000], and the number of resources is 500.

4.2.1. Load Balancing Comparison

In this paper, {s1, s2, s3, s4, s5, s6, s7, s8, s9, s10} and {s1, 150, 200, 250, 300, 350, 400, 450 and 500} processing tasks from S1 to S10 are set in ten resource points. The comparative effect of the three algorithms in the three task set states is shown in Figures 1-3. Each cloud computing resource processing point has different capabilities, so the load is different. From Figures 1-3, it is found that when the number of tasks is small, the load balancing of the three algorithms is almost the same, and the number of tasks increases gradually. The load balancing of this algorithm is relatively stable, but the loads generated by the CS and ICS algorithms are different, mainly because the introduction of particle swarm optimization and the quantum algorithm makes the search direction fixed. Therefore, the algorithm is stable, which makes the load value more balanced. Compared with the other two algorithms, the load of the algorithm in this paper is reduced by about 10%. This shows that the algorithm in this paper has an obvious effect in cloud computing distribution and stable load.

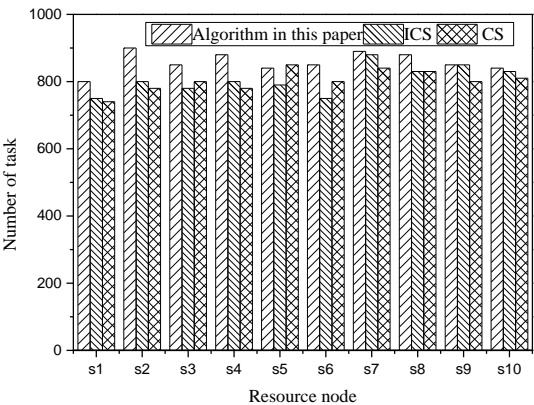


Figure 1. Load balancing comparison of three algorithms in Task1

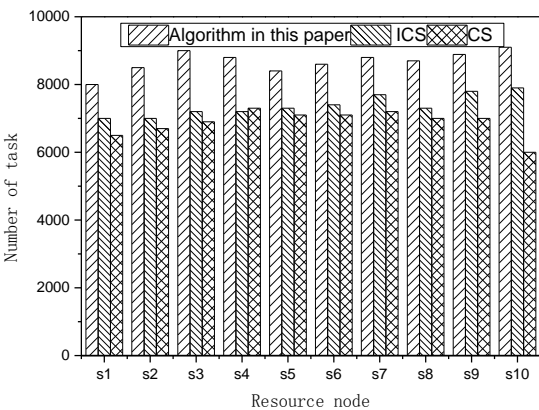


Figure 2. Load balancing comparison of three algorithms in Task2

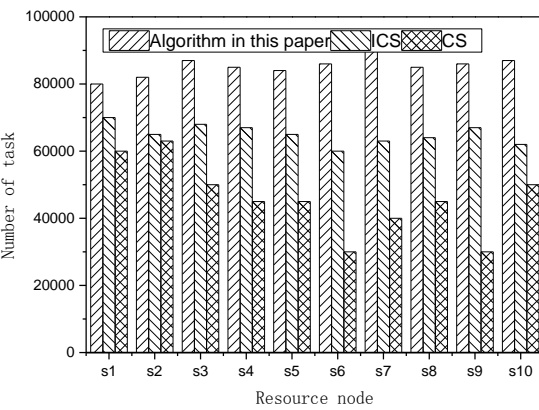


Figure 3. Load balancing comparison of three algorithms in Task3

4.2.2. Execution Time Comparison

Figures 4-6 show a comparison of execution times under three task sets. It can be found from the graph that the execution times of the three algorithms increases to some extent with an increase in the number of tasks. When the number of tasks is small, the execution times of the three algorithms do not differ much. When the number of tasks is large, the execution times of the three algorithms are obviously different. The execution time of this algorithm is obviously better than those of CS and ICS. The main reason is that the binary algorithm can make the particles move to the optimal solution faster, accelerate the convergence speed of the algorithm, and avoid blindness in the search process while keeping the random step size of Levy flight. Compared with the CS and ICS algorithms, the proposed algorithm saves about 15% of the execution time, which shows that the proposed algorithm can adapt to task resource scheduling in cloud computing.

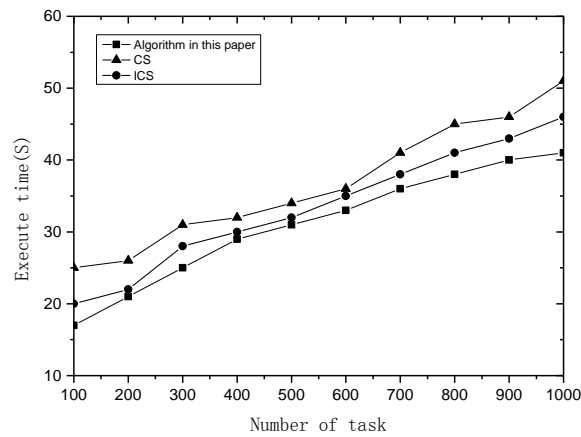


Figure 4. Execution time comparison of three algorithms in Task1

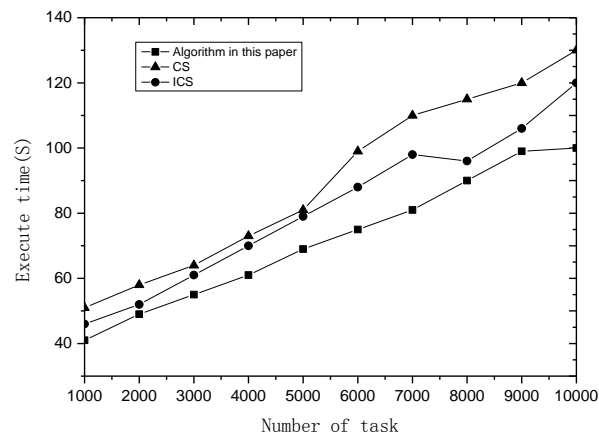


Figure 5. Execution time comparison of three algorithms in Task2

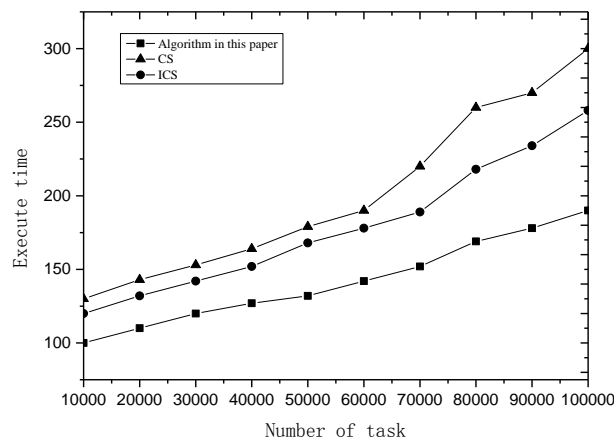


Figure 6. Execution time comparison of three algorithms in Task3

4.2.3. Consumption Cost Comparison

Figures 7-9 show the comparison of consumption costs under three task sets. It can be found from the graph that the costs of the three algorithms increase with an increase in the number of tasks. When the number of tasks is small, the cost curves of the three algorithms increase slowly. When the number of tasks increases gradually, the cost curves of the three algorithms increase in varying degrees; the curve growth of the algorithm in this paper is gentle, while those of CS and ICS algorithm are larger. This shows that this algorithm has great advantages in terms of consumption cost and can adapt to it.

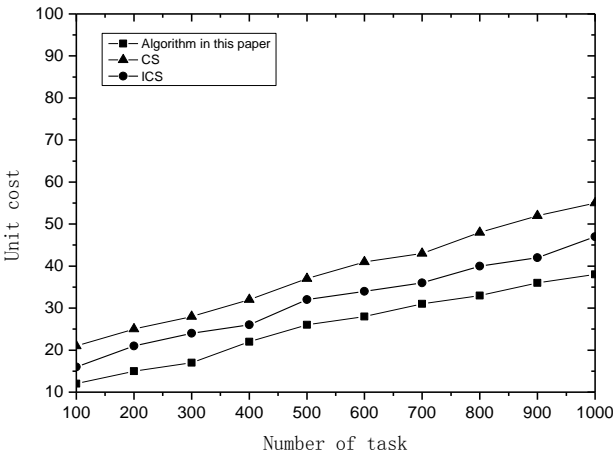


Figure 7. Consumption cost comparison of three algorithms in Task1

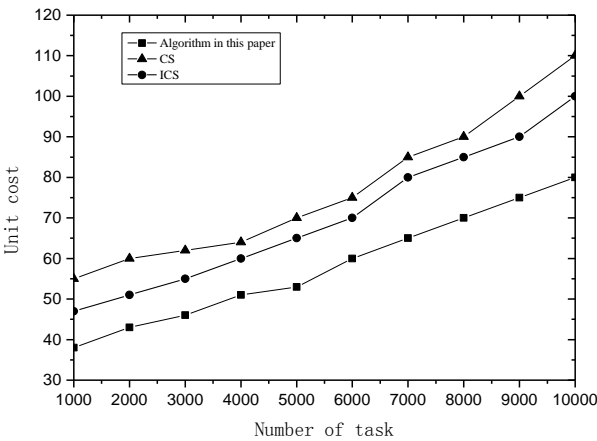


Figure 8. Consumption cost comparison of three algorithms in Task2

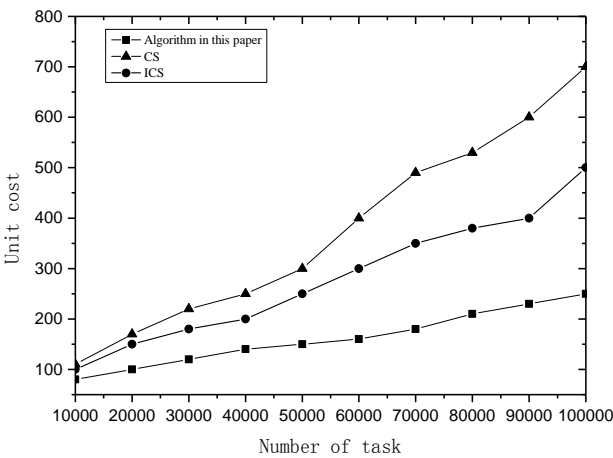


Figure 9. Consumption cost comparison of three algorithms in Task3

5. Conclusions

Task scheduling in cloud computing has always been a research hotspot. This paper introduces an enhanced cuckoo search algorithm to improve the efficiency of task scheduling in cloud computing. The simulation experiments show that the algorithm can adapt to resource scheduling in cloud computing from the aspects of load balancing, execution time, and cost consumption, and it achieves good results. However, this paper ignores energy consumption as a reference standard in cloud computing indicators, and this will be studied in future research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China Youth Fund (No. 41701479), Heilongjiang Science Foundation Project (No. QC2018045), and Qiqihar University Project (No. 2014K-M31).

References

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, et al., "A View of Cloud Computing," *Communications of the ACM*, Vol. 53, No. 4, pp. 50-58, 2010
2. M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, "A Survey of PSO-based Scheduling Algorithms in Cloud Computing," *Journal of Network and Systems Management*, Vol. 25, No. 1, pp. 122-158, 2017
3. N. Kumar and P. Patel, "Resource Management using ANN-PSO Techniques in Cloud Environment," in *Proceedings of the 2016 International Congress on Information and Communication Technology*, pp. 419-428, 2016
4. S. Shahdi-Pashaki, E. Teymourian, and R. Tavakkoli-Moghaddam, "New Approach based on Group Technology for the Consolidation Problem in Cloud Computing-Mathematical Model and Genetic Algorithm," *Computational and Applied Mathematics*, Vol. 37, No. 1, pp. 693-718, 2018
5. H. Aziza and S. Krichen, "Bi-Objective Decision Support System for Task-Scheduling based on Genetic Algorithm in Cloud Computing," *Computing*, Vol. 100, No. 2, pp. 65-91, 2018
6. A. Ragmani, A. E. Omri, N. Abghour, K. Moussaid, and M. Rida, "A Performed Load Balancing Algorithm for Public Cloud Computing using Ant Colony Optimization," in *Proceedings of 2016 2nd International Conference on Cloud Computing Technologies and Applications*, pp. 221-228, 2016
7. V. S. Kushwah and S. K. Goyal, "A Basic Simulation of ACO Algorithm under Cloud Computing for Fault Tolerant," in *Proceedings of the International Conference on Data Engineering and Communication Technology*, Vol. 468, No. 46, pp. 465-472, 2017
8. H. J. Yang, "A Job Scheduling Algorithm based on Artificial Bee Colony in Cloud Computing," *Mathematics in Practice and Theory*, Vol. 24, No. 10, pp. 115-120, 2012
9. W. J. Huang and F. Guo, "Multi-Objective Task Scheduling based on Fireworks Algorithm in Cloud Computing," *Application Research of Computers*, Vol. 34, No. 6, pp. 1718-1720, 2017
10. X. Chen and D. Long, "Task Scheduling of Cloud Computing using Integrated Particle Swarm Algorithm and Ant Colony Algorithm," *Cluster Computing*, pp.1-6, 2017
11. J. P. Zhao, J. Y. Yin, T. B. Jin, and W. N. Zeng, "Application of Genetic Ant Colony Algorithm in Cloud Computing Resource Scheduling," *Computer Engineering and Design*, Vol. 38, No. 3, pp. 693-697, 2017
12. Y. Wang, D. Qin, and J. Liu, "User Utility Optimization of Cloud Computing Resource Scheduling Algorithm based on Production Function," *Application Research of Computers*, Vol. 34, No. 2, pp. 397-400, 2017
13. L. Cheng and S. Koutolas, "Efficient Skew Handling for Outer Joins in a Cloud Computing Environment," *IEEE Transactions on Cloud Computing*, Vol. 6, No. 2, pp. 558-571, 2018
14. G. X. Zhang, N. Li, and W. D. Jin, "A Novel Quantum Genetic Algorithm and Application," *Acta Electronica Sinica*, Vol. 32, No. 3, pp. 476-479, 2004
15. X. S. Yang and S. Deb, "Cuckoo Search via Levy Flights," in *Proceedings of World Congress on Nature & Biologically inspired Computing*, pp. 210-214, 2009

Wenli Liu is a lecturer at Qiqihar University. She received her master's degree from North University of China. Her research interests include cloud computing and signal and information.

Cuiping Shi is an associate professor at Qiqihar University. She received her doctorate degree from Harbin Institute of Technology. Her research interests include cloud computing and signal and information.

Hongbo Yu is an associate professor at Qiqihar University. He received his master's degree from Xi'an Polytechnic University. His research interests include cloud computing.

Hanxiong Fang is a lecturer at Qiqihar University. She received her master's degree from Qiqihar University. Her research interests include cloud computing.