

Research of Resource Scheduling based on ACA-GA in the Cloud Computing

Xuan Chen, Wenfei Song, Zhaoguo Li

(Zhejiang Industry Polytechnic College Shaoxing Zhejiang 312000 China)
734140999@qq.com

Abstract

How to better conduct research resource scheduling has long been a research direction of cloud computing. This paper, aiming at slow convergence and easiness of falling local optimum of ant colony algorithm, has integrated genetic algorithm into the ant colony algorithm and obtained hybrid algorithm (ACA -GA); in the initial solution of the ant colony algorithm, it has adopted selection, crossover and mutation operations of genetic algorithm to obtain an effective initial solution; secondly, it has used the perception threshold of ant colony algorithm path setting to regulate individual selection optimal path; finally, it has improved volatile factor so as to significantly improve the updating efficiency of pheromone. The algorithm in the paper proved that the performance of the algorithm has been also significantly improved through classical test functions. Cloudsim platform shows that, the algorithm above mentioned reduces the time and cost spent in resource scheduling of, hence has some promotional value.

Keywords: ant colony algorithm, genetic algorithm, perception threshold, cloud computing

1. Introduction

Cloud computing is a kind of shared computing model featured with be synchronized and collaborated with the advantages of realizing sharing of software and hardware resources and avoiding producing redundant network information [1]. However, its deficiency lies in that the amount of resources in cloud computing is limited with far more tasks to be processed in each cloud server than the amount of resources. Therefore, it is the focus of current cloud computing research as how to realize rational distribution of resources [1]. As resource distribution of cloud computing is in essence a kind of NP problem, so both domestic and foreign scholars have introduced artificial intelligent algorithm into resource distribution and achieved some certain results. Literature [2] proposes a method to estimate the minimum variance of multiple resources based on heuristic cloud computing and establishes the Cloud-P2P integration model to access multiple resources. In Literature [3], this paper first introduces resource scheduling model of cloud computing, introduces optimized bat algorithm in the model and cubic mapping function into chaos algorithm in optimization for chaos optimization of individual bats while narrowing the search space and accelerating convergence speed. Literature [4] designs a kind of task and resource distribution model of cloud computing based on Q learning and two-way ACO algorithm to realize the final distribution of tasks and resources. Literature [5] proposes a kind of resource scheduling model of cloud computing (IABC) based on improved artificial bee colony algorithm to introduce the optimal value and random vector of current individuals into the bee colony's searching process, accelerating the searching speed and improving searching ability. Literature [6] proposes a kind of fair and balanced genetic scheduling algorithm (FBGSA), taking full account of fairness of resources and tasks during the process of task scheduling as well as factors like resources' load and balance. This algorithm cannot only effectively reduce the

total time to complete tasks, but also meet the fairness of resources and tasks in the system, effectively realizing load and balance of resources. Literature [7] proposes a bat algorithm based on film computing, which makes global optimization towards improved individuals by seeking the optimal individual bat in auxiliary film so as to meet the requirement of resource optimization and scheduling of cloud computing.

Aiming at the above researches, this paper proposes resource scheduling scheme of cloud computing based on Ant Colony Algorithm-Genetic Algorithm (ACA-GA in short) and adopts choosing, crossing and mutation of genetic algorithm during the process of getting the initial solution of ant colony algorithm to get the effective initial solution. Second, in choosing path of ant colony algorithm, sensory threshold is set to adjust the probability for individual to choose the optimal path. Finally, improve evaporation factor to improve the efficiency of updating pheromones. Classic test functions show that algorithm in this paper has superior performance to basic ant colony algorithm and the searching effect of genetic algorithm. Cloudsim experiment shows that algorithm in this paper has certain advantages in time and costs of cloud computing.

2. Resource Scheduling Model of Cloud Computing Based on QoS

Quality of Service is mainly used in cloud computing as a standard for reference to measure whether resource scheduling of cloud computing is rational and can make cloud users satisfied. Its essence is the main task of the cloud under certain circumstances how the allocation of resources to ensure the completion of the time and cost of network bandwidth (cost) be allocated according to different weights so that the overall minimum. This study is based on the following two points: (1) task completion time. The cloud covered the different types of users, so the time required to complete the task is not the same, because the cloud with real-time dynamic characteristics, and therefore need to complete scheduled tasks in the shortest possible time; (2) the mandate costs. As cloud computing takes a long time to occupy the broadband resources, how to complete the task in the shortest possible time, thereby saving the cost required.

In cloud computing, assign n independent sub-tasks to m resources ($m < n$). Herein, tasks is shown as $Task = \{task_1, task_2, \dots, task_n\}$, $task_i$ refers to i subtask with the amount of resources being as $Resource = \{resource_1, resource_2, \dots, resource_n\}$; $resource_i$ refers to i resource, and ETC_{ij} refers to the time needed for task i on resource j , so ETC matrix is shown as follows:

$$ETC = \begin{pmatrix} ETC_{11} & \dots & ETC_{1n} \\ \vdots & \ddots & \vdots \\ ETC_{m1} & \dots & ETC_{mn} \end{pmatrix} \quad (1)$$

Similarly, RCU_{ij} can be used to refer to the cost spent on resource j by subtask i , so RCU matrix is shown as follows:

$$RCU = \begin{pmatrix} RCU_{11} & \dots & RCU_{1n} \\ \vdots & \ddots & \vdots \\ RCU_{m1} & \dots & RCU_{mn} \end{pmatrix} \quad (2)$$

According to ETC matrix and RCU matrix, time and cost spent on implementing the assigned subtask by each resource node.

$$sumTime(i) = \max_{i=1}^{resource} \sum_{j=1}^n T(i, j) \times ETC(i, j) \quad (3)$$

$$sumCost(i, j) = \sum_{i=1}^{resource} \sum_{j=1}^n resource(i, j) \times RCU(i, j) \quad (4)$$

In the formula, $T(i, j)$ refers to the time needed for the j subtask in subtask i , $resource(i, j)$ refers to the costs needed for resource j in subtask i , and formula (3) and formula (4) refer to the total time and costs to complete tasks. The ultimate goal of resource scheduling in cloud computing is to achieve the minimum time and cost to complete tasks. In this paper, fitness function is introduced for explanation, set α as the weight value of time to complete task, and β as the weight value of cost to complete task, $\alpha + \beta = 1$. Therefore, resource scheduling of cloud computing is described as follows, and it can be found from the formula that the minimum time and cost of resource scheduling is the optimal solution of resource scheduling.

$$f = \min(\alpha \square sumtime + \beta \square sumcost) \quad (5)$$

3. Description of Basic Algorithms

3.1 Basic Ant Colony Algorithm

Ant colony algorithm (ACA) is an algorithm to solve the discrete system optimization problem during the process for ant colony to search food, and this algorithm mainly contains two aspects:

1. State Transition Mechanism

Suppose m refers to the amount of ants, d_{ij} refers to the distance between city i and city j , and $\eta_{ij} = 1/d_{ij}$ is heuristic information. τ_{ij} refers to the amount of pheromones at ij , and $p_{ij}^k(t)$ is the probability for ant $p_{ij}^k(t)$ in city i to choose city j at the moment t , then:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (6)$$

In formula (6), $allowed_k$ is the aggregate of cities that can be chosen at present. Tabu table $tabu_k (k = 1, 2, \dots, m)$ is used to record the current city of ant k , $allowed = \{C - tabu_k\}$ refers to the city to be chosen by ant k next while α and β refer to the weight of τ and η in calculation.

2. Rules to Update Pheromones

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (7)$$

In the formula, ρ is the factor showing evaporation of pheromones, and $\Delta\tau_{ij}^k(t)$ refers to the pheromone released by ant k at moment t when it passes through (i, j) .

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & k \text{ Ant in the cycle } (i, j) \text{ after the first } k \\ 0, & otherwise \end{cases} \quad (8)$$

It can be found from the formula that with the increase of path L_k , time for searching becomes longer, causing pheromones to be concentrated to the optimal path, affecting the algorithm's pause in seeking the optimal solution so as to stop further search of the space.

Thus, it causes the probability of being easy to lose the optimal solution.

3.2 Genetic Algorithm

Genetic Algorithm (GA in short) is a random search algorithm simulating genetic choosing and natural elimination. This algorithm contains four steps:

- (1) Determination of data encoding scheme, and produce an initialized individual at random;
- (2) Give the fitness value to evaluate the advantages and disadvantages of individuals; herein, evaluation of individuals is to calculate the length of path for each individual and regard this length as individual's fitness function, which is shown as follows:

$$f(x) = \sum_{i=1}^{n-1} \text{length}(i, j) + d(1, n) \quad (9)$$

In the formula, $\text{length}(i, j)$ is the distance between city i and city j . Individuals with smaller fitness shows shorter path, then the individuals are better.

- (3) Judge whether the algorithm meets the condition of convergence, it so, output the search results; otherwise, continue.
- (4) Implement crossover operation according to crossover probability and mutation probability respectively.

4. Hybrid Algorithm Based on Ant Colony Algorithm and Genetic Algorithm in Cloud Computing

It has always been the focus of resource scheduling as how to optimize time and cost targets in cloud computing. Traditionally, weight coefficient is used to determine the fitness function of genetic algorithm and heuristic function of ant colony algorithm. This algorithm is not so vigorous to some extent, mainly because the setting of weight coefficient is random.

4.1 Initialize Ant Colony Algorithm with Genetic Algorithm

Genetic algorithm is used to initialize the ant colony algorithm so as to significantly improve the efficiency of ant colony algorithm at the later stage. Set the size of ant colony algorithm as M , and divide all the ant individuals into two colonies: paternal colony and child colony. Herein, paternal colony is $Z = \{\alpha_1, \alpha_2, \dots, \alpha_M\}$, set the fitness function of each individual as $f(\alpha_i)$; child colony is $X = \{\delta_1, \delta_2, \dots, \delta_M\}$, sequence all the paternal colonies from large to small, and the colony after sequencing is $Z' = \{\alpha'_1, \alpha'_2, \dots, \alpha'_M\}$. Sequence fitness value from large to small $f(\alpha'_1) > f(\alpha'_2) > \dots > f(\alpha'_M)$. Sum up the fitness of all the individuals in paternal colony after sequencing as $\sum_{i=1}^M f(\alpha'_i)$, and

calculate the probability for each ant colony individual to be chosen as $P(i) = \frac{f(\alpha'_i)}{\sum_{i=1}^M f(\alpha'_i)}$.

The accumulated probability of each individual is $Q_i = \sum_{i=1}^M P(i)$. Produce M random numbers R within $[0,1]$ by adopting corona. If $R \leq Q_i$, choose chromosome b_1 ; otherwise, choose the i chromosome b_i . Therefore, the chosen individual M_i is $b_{\min(i,1,2,\dots,in)}$;

combine M_i and sub colony X to make $X(t) = X(t-1) \cup M_i$. Then, get the final solution of these individuals through crossover and mutation operation as the ant colony algorithm's initial solution.

4.2 Sensory Threshold Setting — Path Selection

During the process of ant colony individual's initial stage, ant routing strategy is a path up the pheromone path, which easily lead to some ants walking on the same path, and other ants that not enough time to choose to embark on the same path, which resulted in the ant individuals from the initial search exist with larger probability concentrated choice in the current local length shorter path, in order to avoid the occurrence of such a situation, set a sensory threshold γ , the setting on the path of information element quantity did not reach the sensory threshold, the ants can ignore this path pheromone, so you can continue to look for. When the sensory threshold is less than the amount of pheromone value, ants can choose path pheromone on large.

Therefore, the k ant changes from state i to state j according to the following probability.

$$j = \begin{cases} \max\{\tau_i^\alpha, \eta_i^\beta\}, s \in allowed_k, \text{if } r \leq \gamma \\ p_{i \rightarrow j}^k(t) \text{ otherwise} \end{cases} \quad (10)$$

4.3 Improvement of Pheromone Play Factor ρ

In ant colony algorithm, due to the function of evaporation factor in the pheromone updating formula, values of those pheromones on the paths that have never been searched becomes gradually fewer until this path is never visited. On the other hand, when the value of evaporation factor on some other paths is relatively larger, pheromone of the solution becomes gradually larger, and paths that have been searched previously are likely to reparticipate in the choosing of global path, reducing the global search ability and consuming time of searching. Therefore, it is particularly important to process the evaporation factor ρ . In this paper, self-fitness factor is used to change the value of ρ , suppose the initial value of ρ is 0.999 and the minimum value is ρ_{\min} . Meanwhile, when the algorithm's iteration times have not changed obviously within certain range, they gradually reduce according to certain times.

$$\rho(t+1) = \begin{cases} 0.999 - \eta \square rand() / 10 \square time & \rho(t) > \rho_{\min} \\ \rho_{\min} & \end{cases} \quad (11)$$

In the formula, $rand()$ is a random function, set η as the self-fitness factor and $time$ as the time of iteration. This self-fitness factor can ensure the global search ability of the algorithm in certain search range.

4.4 Algorithm Description

According to the above algorithm, model diagram of algorithm in this paper is as shown in Figure.1.

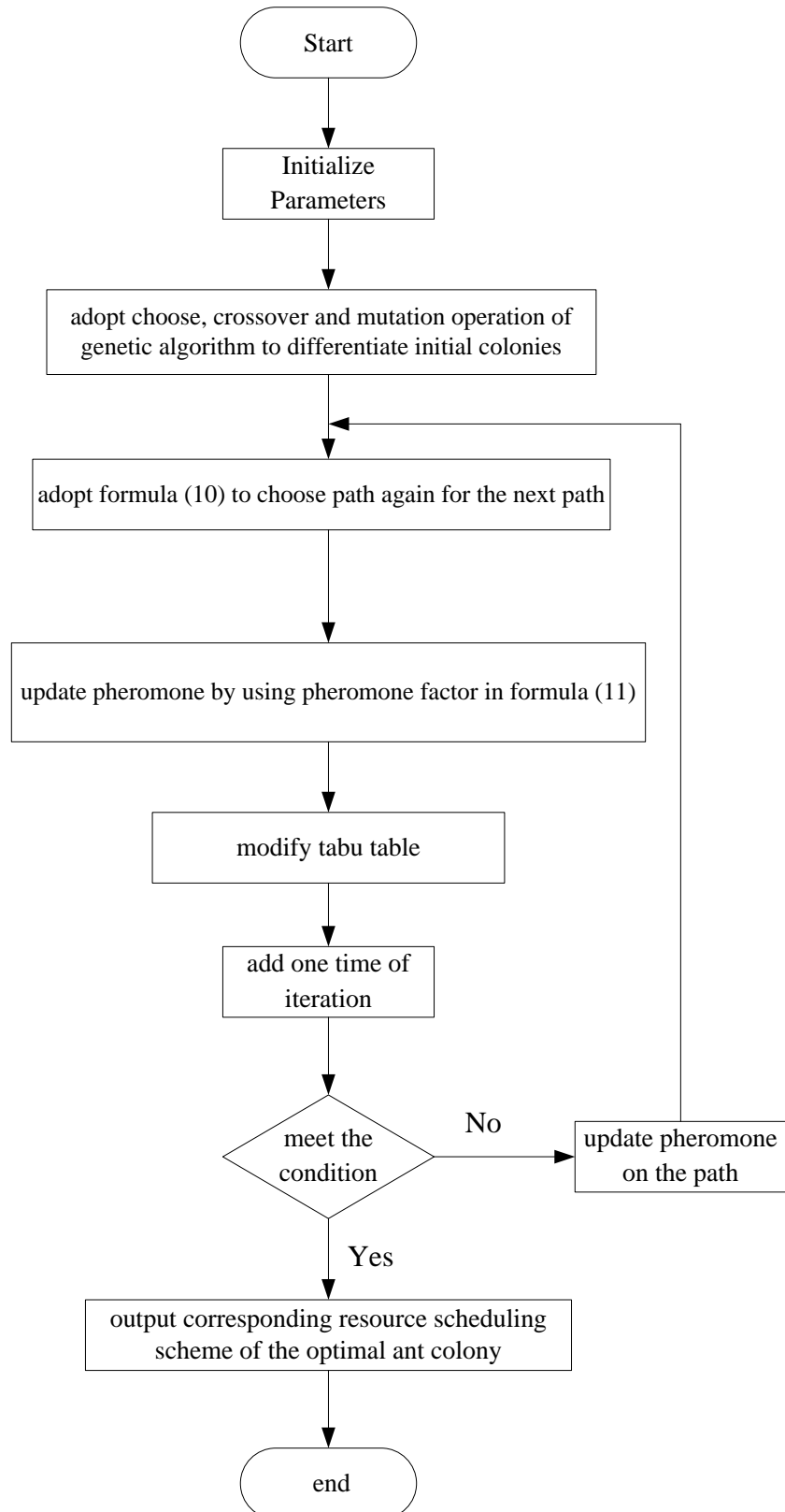


Figure 1. The Cloud Computing Resource Scheduling of this Algorithm

Step 1: Initialize basic parameters of ant colony algorithm and genetic algorithm. Set the corresponding resource scheduling scheme of cloud computing towards ant colony algorithm individuals with time weighted value as α , cost weighted value β as 0.5, and self-fitness factor η as 0.9, γ as 0.5. Suppose the minimum value of ρ is 0.001, $rand()$ is [0, 1], initialize the scale of ant colony's scale as $N=100$. Amount of resources in cloud computing is 100, that of tasks is 50 and iteration times of the algorithm are 1000.

Step 2: Make choosing, crossover and mutation operations to initialize ant colony algorithm according to the description in Chapter 3.1

Step 3: Choose path for ant colony through formula (10), and obtain the path forwards through setting of sensory threshold γ .

Step 4: Make improvement towards pheromone factors according to formula (11) so as to make pheromone on the path even.

Step 5: modify tabu table, and add one time of iteration

Step 6: Judge whether iteration time is met, if so, turn to step 7, otherwise, turn to step 3.

Step 7: Output the optimal ant colony individual and its corresponding optimal resource scheduling scheme.

5. Analysis of Simulation Experiment

Software environment of the experiment is as follows: CPU is Core i3, memory is 8GDDR3, software environment is Window 7 operation system, simulation platform is Matlab2012 and cloud computing simulation platform is Cloudsim.

5.1. Comparison of Performance with Basic Ant Colony Algorithm and Genetic Algorithm

Suppose the probability of crossover p_c in genetic algorithm is 0.75, and mutation probability is 0.05. In ant colony algorithm, Q is 100, self-fitness factor η is 0.9, and γ is 0.5. Suppose the minimum ρ value is 0.001, scale of ant colony is 100, and iteration time is 100. Table 1-3 describes the results of optimizing the three algorithms, and Figure.2-4 is the comparison of convergence effects of three algorithms. It can be found from the table and figure that algorithm in this paper is superior to ant colony algorithm and genetic algorithm in both performance and target function value, providing theoretical basis for resource scheduling in cloud computing.

$$f_1(x) = \sum_{i=1}^n x_i^4 + random(0,1), \text{ and the search space is } [-1.28, 1.28] \quad (1)$$

$$f_2(x) = 1 + \sum_{i=1}^n \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right), \text{ and the search space is } [-600, 600] \quad (2)$$

$$f_3(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n x_i, \text{ and the search space is } [-10, 10] \quad (3)$$

Table 1. Comparison of f_1 Function Optimization

Dimension	Algorithms	The Optimal Value	The Minimum Value
30	Genetic Algorithm	1.447e-10	7.173e-10
	Ant Colony Algorithm	2.873e-22	9.143e-20
	Algorithm in this Paper	3.912e-40	4.274e-30
50	Genetic Algorithm	1.217e-06	7.323e-07
	Ant Colony Algorithm	4.29e-04	3.172e-05
	Algorithm in this Paper	7.217e-03	7.292e-01
100	Genetic Algorithm	6.372	7.126
	Ant Colony Algorithm	8.416	4.162
	Algorithm in this Paper	12.174	1.381

Table 2. Comparison of f_2 Function Optimization

Dimension	Algorithms	The Optimal Value	The Minimum Value
30	Genetic Algorithm	2.148e-13	4.132e-15
	Ant Colony Algorithm	3.912e-26	7.743e-23
	Algorithm in this Paper	8.912e-36	9.174e-32
50	Genetic Algorithm	3.714e-07	7.323e-09
	Ant Colony Algorithm	5.427e-04	7.172e-02
	Algorithm in this Paper	7.251e-02	6.192e-01
100	Genetic Algorithm	2.962e+03	3.926e+03
	Ant Colony Algorithm	2.576e+03	4.172e+03
	Algorithm in this Paper	1.127e+02	1.851e+02

Table 3. Comparison of f_3 Function Optimization

Dimension	Algorithms	The Optimal Value	The Minimum Value
30	Genetic Algorithm	5.917e-03	6.432e-09
	Ant Colony Algorithm	7.413e-07	8.273e-17
	Algorithm in this Paper	5.212e-09	8.314e-24
50	Genetic Algorithm	7.234e-02	6.323e-03
	Ant Colony Algorithm	9.127e-05	5.182e-05
	Algorithm in this Paper	7.927e-06	2.182e-08
100	Genetic Algorithm	6.6812	6.1452
	Ant Colony Algorithm	8.2871	17.3036
	Algorithm in this Paper	3.1541	4.38e-04

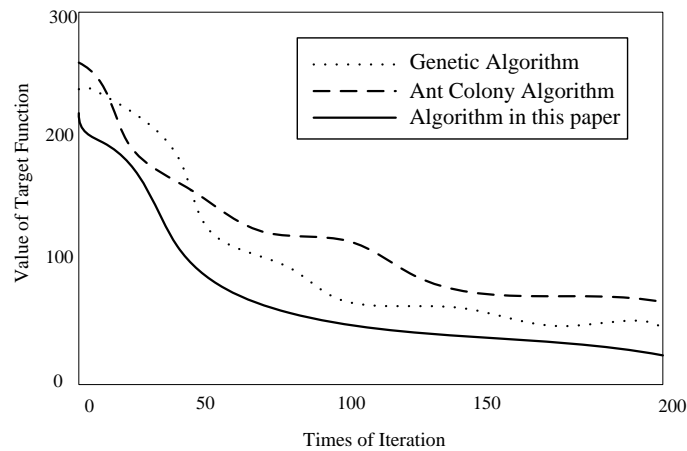


Figure 2. The Test Results of Three Algorithms in f_1

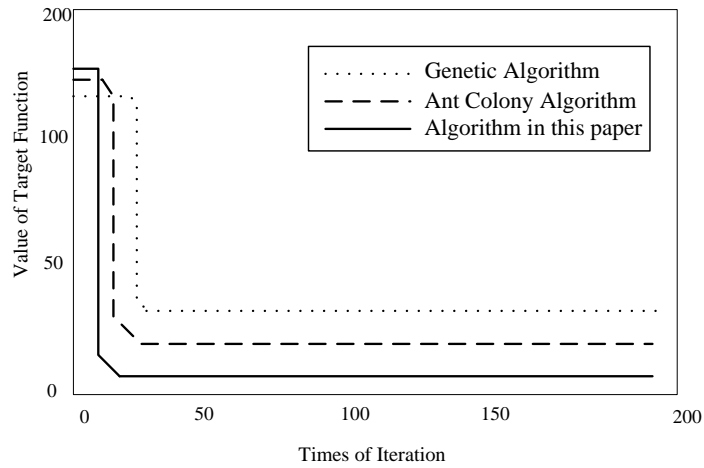


Figure 3. The Test Results of Three Algorithms in f_2

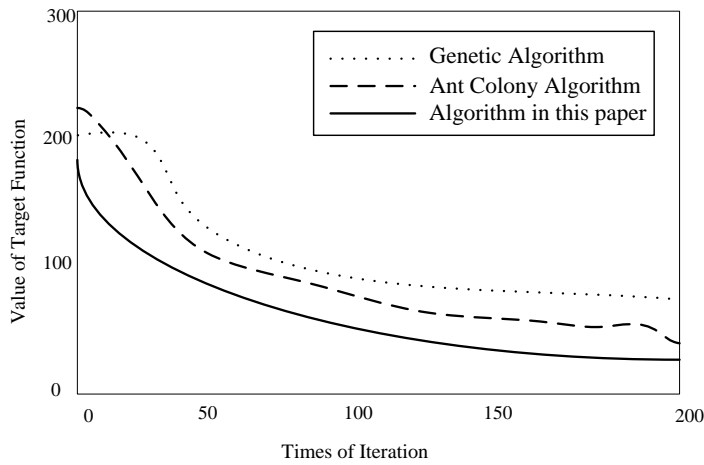


Figure 4. The Test Results of Three Algorithms in f_3

5.2 Comparison with other Intelligent Algorithms in Cloud Computing

Algorithm in this paper is compared with other algorithms of cloud computing's resource scheduling in references while supposing there are 800 virtual tasks, 10 virtual resources and 100 times of iteration. The results of comparison are as shown in Figure.5-6.

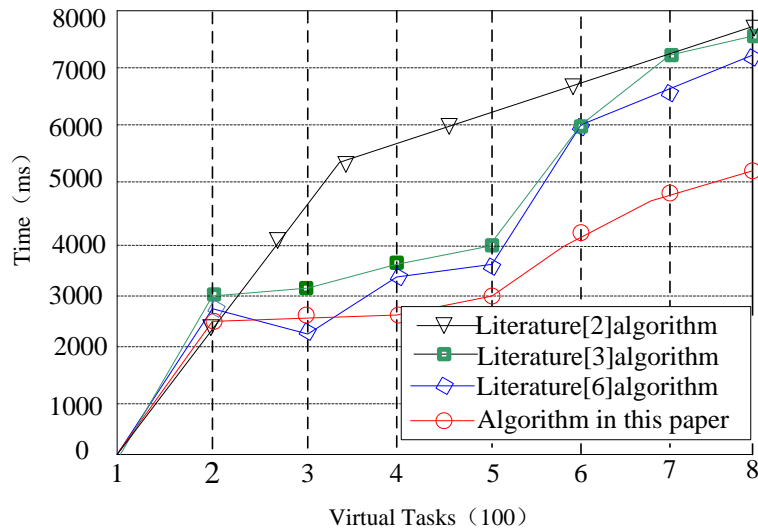


Figure 5. Task Completion Time Comparison of Four Algorithms

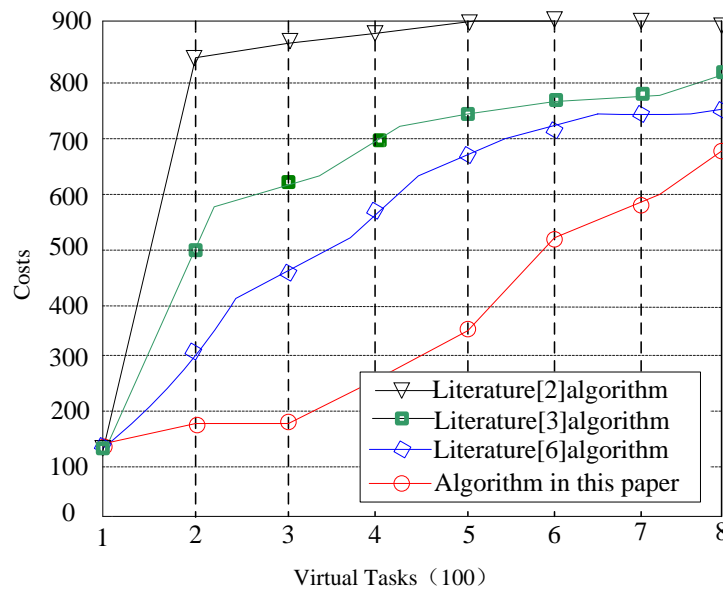


Figure 6. Cost Consumption Comparison of Four Algorithms

It can be found in Figure.5 that algorithm in this paper is superior to other three algorithms in references in terms of time to complete task. In addition, it can be found from the figure that algorithm in this paper is relatively stable with small amplitude magnitude, indicating that algorithm in this paper is stable. Figure. 6 shows that algorithm in this paper is obviously superior to algorithms in references in terms of costs spent during the process of resource scheduling, indicating that certain effects have been achieved during the initializing process of algorithm in this paper so as to avoid follow-up searches without direction and further avoid waste of costs. Therefore, it well adapts to the requirement of resource scheduling in cloud computing.

5.3. User QOS Analysis

QOS solves the problem whether the efficiency of resource scheduling in cloud computing can reach the requirements of cloud users. In this paper, algorithms in Literature [6] and Literature [8] are chosen to compare with algorithm in this paper, and

different tasks in Table 4 are chosen for comparison. It can be found from Figure 7 that algorithm in this paper is superior to algorithms in two literatures in terms of QoS, indicating that algorithm in this paper can not only meet resource scheduling, but also meet requirements of cloud users. Thus, it is certain value for reference.

Table 4. The Number of Resources Corresponding to Different Tasks

Type of Task	Amount of Task	Amount of Resources
Task 1	1000	100
Task 2	2000	200
Task 3	5000	300
Task 4	10000	400
Task 5	20000	500

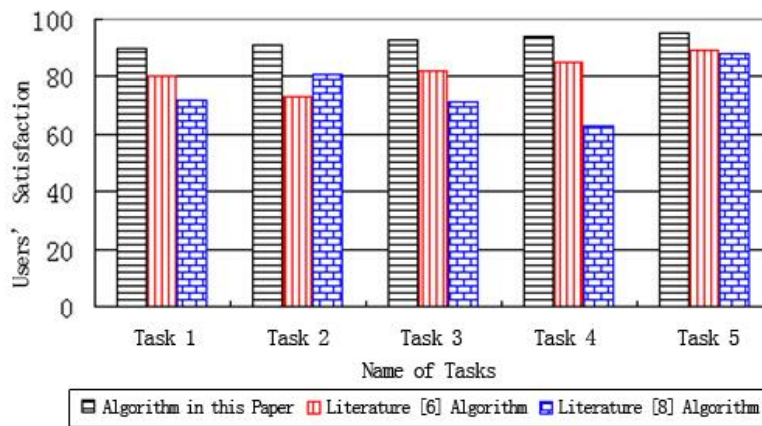


Figure 7. Effectiveness Analysis of User QoS Allocation

6. Conclusion

This paper proposes resources scheduling scheme of cloud computing based on ACA-GA, and adopts choosing, crossover and mutation of genetic algorithm during the process of getting the initial solution of ant colony algorithm so as to get the effective initial solution. Secondly, set sensory threshold during the process of choosing path in ant colony algorithm to adjust the probability for individuals to get the optimal solution. Finally, improve evaporation factors to improve the efficiency of updating pheromone. Classic test function shows that algorithm in this paper has superior performance to basic ant colony algorithm and better searching effect than genetic algorithm. Cloudsim experiment shows that algorithm in this paper has some advantages in time and cost of cloud computing. Meanwhile, it can be indicated through comparing with algorithm in reference that algorithm in this paper has certain advantage in time and costs to complete task and user satisfaction.

References

- [1] L. wei-wei, Q. de-yu, "Survey of Resource Scheduling in Cloud Computing [J] Computer Science", vol. 39, no. 10, (2012), pp. 1-5
- [2] W. shun-ping, W. jie, "Minimum Variance Estimation of Multi Resource Access Feature Based on Heuristic Cloud Computing [J]", Bulletin of Science and Technology, vol. 31, no. 4:13, (2015), pp. 3-135
- [3] L. shu-hong, "Research on the Cloud Computing Resource Scheduling Based Strategy and Bat Optimization Algorithm on Chaos [J]", Bulletin of Science and Technology, vol. 30, no. 7, (2014), pp. 144-149
- [4] S hua, Z jing-xin, "Design of Task — Resource Allocation Model Based on Q — Learning and Double Orientation ACO Algorithm for Cloud Computing [J]", Computer Measurement & Control , vol. 22, no. 10, (2014), pp. 2243-3346.
- [5] Z. tao , Z. yin, "Scheduling Model Cloud Computing Resource Based on Improved Artificial Bee Colony Algorithm [J]", Microelectronics & Computer , vol. 31, no. 7, (2014), pp. 147-150.
- [6] Z. ai-ke, X. cui-hua, "Task Scheduling Algorithm in Cloud Computing Based on Fairness a and Load Balancing [J]", Computer Applications and Software , vol. 32, no. 2, (2015), pp. 268-271.
- [7] N. bin, G. qiong,W. zhao, "Bats algorithm research in cloud computing resource scheduling based on membrane computing [J]", Application Research of Computers, vol. 32, no. 3, (2015), pp. 830-83
- [8] W. qian, S. zhen-guo, S. wan-jie, "PEPA model approach for performance evaluation of dynamic resource provision in cloud computing [J]" , Application Research of Computers , vol. 32, no. (4), (2015), pp. 1179-1183.

Author

Xuan Chen (1979.03-), male, master, lecturer, research direction: information security, wireless sensor and cloud computing

Wenfei Song (1979,09-),Female,Master,Associate Professor. Her research interests includes digital library,chinese information processing and semantic web.

Zhaoguo Li (1980.05-),male, master, lecturer, research direction: information security, wireless sensor and cloud computing.