Cluster Computing
https://doi.org/10.1007/s10586-020-03221-z

Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm

Xueliang Fu[1] · Yang Sun[1] · Haifang Wang[1] · Honghui Li[1]

Received: 8 April 2020 / Revised: 26 November 2020 / Accepted: 8 December 2020

Abstract
Task scheduling in cloud environment is a hot topic in current research. Effective scheduling of massive tasks submitted by users in cloud environment is of great practical significance for increasing the core competitiveness of companies and enterprises and improving their economic benefits. Faced with the urgent need for an efficient scheduling strategy in the real world, this paper analyzed the process of cloud task scheduling, and proposed a particle swarm optimization genetic hybrid algorithm based on phagocytosis PSO_PGA. Firstly, each generation of particle swarm is divided, and the position of the particles in the sub population is changed by using phagocytosis mechanism and crossover mutation of genetic algorithm, so as to expand the search range of the solution space. Then the sub populations are merged, which ensures the diversity of particles in the population and reduces the probability of the algorithm falling into the local optimal solution. Finally, the feedback mechanism is used to feed back the flight experience of the particle itself and the flight experience of the companion to the next generation particle population, so as to ensure that the particle population can always move towards the direction of excellent solution. Through simulation experiments, the proposed algorithm is compared with several other existing algorithms, and the results show that the proposed algorithm significantly improves the overall completion time of cloud tasks, and has higher convergence accuracy. It shows the effectiveness of the algorithm in cloud task scheduling.

Keywords Cloud computing · Task scheduling · Phagocytosis · PSO · GA

1 Introduction

The advent of cloud computing [1] has redefined the patterns of computing and services. It is a commercialized computing model with paid resources as services. Its essence is a large aggregation, which integrates large-scale storage resources, data resources and computing resources. Both companies and individuals can obtain the required resources according to their actual needs, and these resources can be quickly provided and delivered by service providers [2, 3]. According to the actual application perspective, the delivery model of cloud computing can be divided into three levels: providing the entire business application as a service to users, namely SaaS, allowing users to rapidly develop and deploy applications in the cloud platform, namely PaaS, and users to obtain the use of infrastructure through the network, namely IaaS [4–6]. Cloud computing, as a new technology, has been widely used by companies and enterprises because of its great commercial value. With the wide application of cloud computing, the number of cloud users and cloud tasks are growing faster and faster, which makes the cloud environment face severe challenges. How to properly utilize and allocate various resources provided by the cloud environment, how to effectively schedule massive tasks submitted by users, and ensure load balancing of cloud systems have become hot issues in the field of cloud computing research. In complex cloud computing environment, task scheduling [7] process is a difficult problem to change or optimize. It is a typical NP-complete problem [8]. In order to satisfy the customer's requirement of service quality and improve the core competitiveness and

✉ Xueliang Fu
fuxl@imau.edu.cn

1 College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, China

economic benefits of service providers, it is particularly urgent to find an efficient and excellent scheduling strategy.

In recent years, natural heuristic algorithms have been applied to task scheduling, and their performance is better than traditional algorithms. For example, particle swarm algorithm and genetic algorithm are a kind of common meta heuristic algorithms. Then various heuristic intelligent scheduling algorithms based on improved optimization emerged. For example, Improved Genetic Algorithm (IGA) [9], On the basis of genetic algorithm (GA), the introduction of "three-stage selection method" and cross-region similarity reduces the possibility of genetic algorithm falling into local optimal solution. Huang et al. [10] proposed an improved particle swarm optimization algorithm combined with time-varying inertia weight strategy and verified its effectiveness by comparing with three classic heuristic algorithms. Arul et al. [11] proposed an improved chaotic social spider algorithm based on the social spider algorithm to solve the task scheduling problem in various heterogeneous virtual machines in the cloud environment. Sun et al. [12] proposed an improved genetic algorithm (PGA) based on phagocytosis, and based on this, proposed a multi-group hybrid co-evolution genetic algorithm (MPHC_GA), which used the Min-Min algorithm to generate initial multiple subpopulations, and these subpopulations are respectively evolved by standard genetic algorithm (GA) and phagocytosis based improved genetic algorithm (PGA).

According to the principle of "No Free Lunch", it is shown that no algorithm can fully satisfy all application scenarios, and only "under certain conditions, the effect of an algorithm is the best" [13]. Therefore, when solving task scheduling problems, only using a single algorithm can not solve the problem well, so it is necessary to try to fuse different algorithms. Muthulakshmi and Somasundaram [14] have integrated the functions of the simulated annealing algorithm into the artificial bee colony algorithm, and proposed a hybrid optimization algorithm ABC-SA for optimal allocation of resources in the cloud environment. Madni et al. [15] have proposed a hybrid gradient descent algorithm (HGDCS) based on gradient descent (GD) and cuckoo search (CS) to optimize and solve the resource scheduling problem in infrastructure as a service (IaaS) cloud computing. Valarmathi and Sheela [16] have proposed an improved particle swarm optimization algorithm (RTPSO), which uses the distance function and adjustment function of data locality to solve the problem of inertial weight distribution when the PSO algorithm was applied to task scheduling scenarios. Valarmathi and Sheela also combined RTPSO and bat algorithm for cloud task scheduling, which reduced the task completion time, lowered costs, and improved resource utilization. Chen et al. [17] have suggested a particle swarm optimization algorithm combined with ant colony algorithm and verified its effectiveness in cloud task scheduling through experiments. Senthil et al. [18] proposed a hybrid genetic particle swarm optimization algorithm (HGPSO), which combines genetic algorithm and particle swarm optimization algorithm to evaluate suitable resources for user tasks in on-demand queues.

At present, the cloud computing task scheduling mechanism has not formed a unified standard and specification. According to the characteristics of cloud computing task scheduling, this paper studies how to minimize the maximum time span of the task. In this paper, a particle swarm optimization genetic hybrid algorithm (PSO_PGA) based on phagocytosis is proposed for cloud task scheduling. This algorithm changes the updating method of position and speed in standard particle swarm optimization. Fitness function and standard deviation of load balancing [35] are used to divide the individual of each generation of Particle Swarm for the first time and the second time respectively. The simulation results show that the particle swarm optimization genetic hybrid algorithm (PSO_PGA) based on phagocytosis improves the completion time of cloud tasks compared with particle swarm optimization (PSO), genetic algorithm (GA), improved particle swarm optimization (PSO_CM) [19], enhanced Improved Genetic Algorithm (EIGA) [9]. Moreover, the proposed algorithm has better optimization results and higher optimization accuracy.

2 Methodology

2.1 Problem description

Cloud computing task scheduling refers to mapping or assigning tasks to specific virtual machines. The purpose of scheduling is to allocate resources reasonably for cloud tasks submitted by cloud users so as to improve resource utilization. The scheduling result is generally reflected in the mapping association between the task and the virtual machine, that is, determining which task is running on which virtual machine node [20]. That is, for a given cloud task, we should find an appropriate virtual machine to ensure that the task execution time (excluding data transfer time) is minimized [21].

In order to facilitate simulation research, the following assumptions are made:

- Ignore the influence of bandwidth, data transmission, communication time between virtual machines, etc. The execution time of a task is equal to the ratio of the length of the task to the execution speed of the virtual machine where the task is located.

- Abstract the task as the number of machine instructions, the unit is MI (million instructions) and the length is randomly selected within a certain range.
- All cloud computing resources are mapped to virtual machines whose performance unit is MIPS (millions of instructions per second) and whose performance is selected within a certain range.
- There is no interdependence between tasks. When multiple tasks are assigned to the same virtual machine, the tasks are executed according to the principle of sequential scheduling.

Therefore, this paper mainly studies how to allocate resources for tasks submitted by users, in other words, how to allocate N cloud tasks of different lengths to M virtual machines with different performance, so that the total completion time of these N cloud tasks is the shortest.

2.2 Introduction of standard particle swarm optimization (PSO)

In 1995, Kennedy et al. proposed particle swarm optimization (PSO) [22]. It originates from the simulation of migration and aggregation of birds and fish in the process of foraging [23]. In particle swarm optimization (PSO), the population is called a group and the individual is called a particle. Particles in the particle swarm algorithm look for food like a bird or fish through the search space. Each particle in a particle swarm has an attribute of velocity and location. In motion, each particle gradually approaches the optimal solution by adjusting its flight direction and speed according to its current position, speed, p_{besti} and g_{best}, just as a flock of birds or fish finds food through cooperation and information sharing among individuals in the flock. The performance of each particle is measured by fitness function. There are different fitness functions corresponding to different practical problems. Any one of the particles in the group corresponds to one of the solutions to the actual problem. Its dimensions are determined by the question itself [24]. The position and velocity of particles are randomly initialized. The velocity and position of particles in each generation of particle swarm are updated as shown in formulas (1) and (2).

$$V_i^{k+1} = wv_i^k + c_1 r_1 (p_{best_i} - x_i^k) + c_2 r_2 (g_{best} - x_i^k), \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^k, \tag{2}$$

Among them, v_i^k represents the velocity of particle i at the kth iteration, x_i^k represents the position of particle i at the kth iteration, v_i^{k+1} represents the velocity of particle i at the k + 1th iteration, x_i^{k+1} represents the position of particle i at the k + 1th iteration, w represents inertia weight, c_1, c_2 represents the acceleration factor, r_1, r_2 represents random numbers between [0,1], p_{besti} represents the optimal position that particle i is currently looking for, g_{best} represents the optimal position searched by particle swarm optimization.

Particle Swarm Optimization (PSO) has the characteristics of "memory function", easy implementation, fast convergence, etc., and has been widely used in various fields [25]. However, particle swarm optimization (PSO) needs to be adjusted dynamically according to the actual problems in parameter control. And it is difficult to adjust to the best so that the particle speed can not be effectively controlled. It is easy to fall into local optimal solution, which results in low convergence accuracy of the algorithm.

2.3 Introduction of standard genetic algorithms (GA)

In 1975, Professor Holland first proposed a genetic algorithm in his book "Adaptation in Natural and Artificial System" [26]. The algorithm is based on the mechanism of natural genetics, namely genetics, selection, crossover and mutation, etc. [27]. Before searching, the potential solution of the problem is transformed from its solution space to the search space that genetic algorithm can deal with, that is, coding [28]. The coding maps the solution space to the chromosome coding space, and different chromosome individuals form a population. For task scheduling problems, the algorithm begins with a population that represents a potential solution set of problems, each of which corresponds to a scheduling scheme. In each generation, individuals are selected according to the fitness of individuals in the problem domain [29], which represents the individual's competitive ability. Through genetic operations such as selection, crossover and mutation, the population evolves to form a new generation of population. Evolutionary new generation population again calculates the fitness of individual populations. It is judged whether the termination condition is satisfied, and if it is satisfied, the individual with the greatest fitness among the latest generation population is output as the optimal solution. Otherwise, the population will continue to evolve through genetic operations such as selection, crossover and mutation. The flow chart of the standard genetic algorithm is shown in Fig. 1.

2.4 Specific description of particle swarm genetic hybrid algorithm based on phagocytosis

Evolutionary algorithm is a kind of swarm intelligence algorithm, such as genetic algorithm (GA), ant colony optimization algorithm (ASO), particle swarm optimization algorithm (PSO), differential evolution algorithm (DE) and

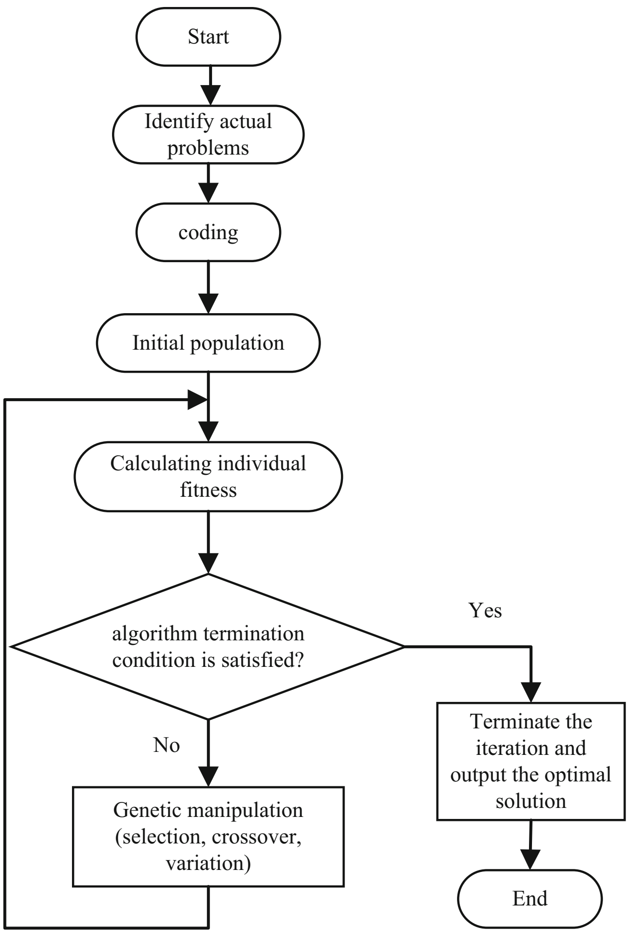

Fig.1 Operation process of genetic algorithm.

so on. These algorithms have been applied to solve various optimization problems in different fields, but they have one disadvantage: most of them are only suitable for solving numerical optimization problems in continuous domain, but can not be directly used to solve combinatorial optimization problems on discrete domains except GA and ACO [30]. As one of the classical evolutionary algorithms, particle swarm optimization (PSO) has this problem. Task scheduling under cloud computing can be regarded as solving combinatorial optimization problems in the discrete domain. Therefore, standard particle swarm optimization algorithms cannot be used directly to solve them. The existing algorithms need to be improved. At present, there are two feasible methods: one is to change the evolution operator of the original algorithm and redefine it to meet the calculation requirements in the discrete domain [31, 32]; the other is to keep the evolution operator of the original algorithm unchanged, and map the real vector corresponding to an individual into a 0–1 vector [33, 34]. This paper uses the first method to improve the standard particle swarm optimization algorithm, so that it can be better applied to task scheduling.

2.4.1 Particle coding

Using real number direct coding, each particle in particle swarm optimization represents a scheduling scheme, and the dimension of the particle is equal to the number of tasks. We assume that the number of tasks is N, the number of virtual machine resources is M, and the number of each dimension of the individual particle represents the number of tasks. The value range is 0–N–1. The positive integer value of each dimension represents the number of virtual machine resources, and the value range is 0–M–1. For example, N = 9, M = 4, and the particle coding sequence is (2, 0, 0, 3, 1, 2, 1, 2, 3). It represents the fourth and the ninth task assigned to the fourth virtual machine. The first, sixth, eighth tasks were assigned to the third virtual machine, the fifth, the seventh tasks were assigned to the second virtual machine, the second and the third tasks were assigned to the first virtual machine.

2.4.2 Fitness function

In this paper, the goal is to minimize the total completion time of cloud tasks, so the total completion time of cloud tasks is taken as the fitness function. Because the shorter the total completion time of the cloud task, the better the individual is, the calculation formula is set as shown in (3):

$$fitness = 1/\max(time_i), \quad i \in (0, 1, 2, 3, \ldots, m-1) \tag{3}$$

In formula (3), $time_i$ represents the time taken for all cloud tasks to complete on virtual machine i, and m represents the number of virtual machines. The fitness of individuals in each generation depends on the virtual machines that spend the most time to complete all cloud tasks on them, and the larger the fitness of individuals, the smaller the total completion time of cloud tasks.

2.4.3 Feedback mechanism

In particle swarm optimization, the updating of particle velocity and position is based on the optimal position of individual particle and global particle in each iteration process. Moreover, the optimal position of the individual particle corresponds to the individual's self-perception, which is the flight experience of the individual particle. The optimum location of global particles corresponds to the recognition of population, i.e. social cognition, which is the flight experience of particle companions. For each iteration, the particle swarm algorithm takes these two experiences into account and forms a feedback.

In order to better apply the algorithm to task scheduling in cloud computing, we propose a particle swarm optimization hybrid algorithm based on phagocytosis, which directly discards the influence of velocity on particle position and avoids the performance degradation caused by inertia weight, acceleration factor and other parameters can not determine the optimal value. However, the feedback mechanism of the standard particle swarm optimization algorithm is retained, and the self-recognition and social recognition of the particles are taken into account.

We set the dimension of particles, i.e. the number of cloud tasks, to N. Drawing on the basic idea of cross-strategy in Ref. [19], the implementation process is described as follows:

- A positive integer p is randomly generated. The value of p ranges from [0, N − 1], and p is used as the amount of empirical information of the best flight position p_{besti} that particle i has acquired in each iteration.
- In the range of p, the positive integer y is generated randomly, and the range of y is [0, N − 1]. In each iteration, y is used as the subscript of the specific coding sequence of the best flight position p_{besti} that particle i as experienced.
- Randomly generate a positive integer g, the value range of g is [0, N − 1], and g is used as the g_{best} information amount of the particle companion flight experience in the particle group in each iteration process.
- In the range of g, a positive integer z is generated randomly, and the range of z is [0, N − 1]. In each iteration, z is used as the specific coding sequence subscript of particle i to obtain the optimal position g_{best} currently searched by particle swarm.

2.4.4 Particle subpopulation

For each generation of particle population, we use fitness function and standard deviation of load balancing to divide it twice. The standard deviation formula of load balancing [35] is shown in (4) (5) (6).

$$BL = \sqrt{\sum\nolimits_{i=0}^{m-1} (F(i) - AVL)^2 / m}, \tag{4}$$

$$F(i) = \sum\nolimits_{j=1}^{n} VL_j^i, \tag{5}$$

$$AVL = \sum\nolimits_{i=0}^{m-1} F(i)/m, \tag{6}$$

In formula (4) (5) (6), BL denotes the standard deviation of load balancing, F(i) represents the time taken by the ith virtual machine to complete the tasks assigned to it, m represents the number of virtual machines, $VL^i{}_j$ represents the time taken by the i-th virtual machine to complete the j-th task assigned on it,n represents the total number of tasks assigned to the ith virtual machine, AVL represents the average of the sum of the time that all virtual machines complete the tasks assigned to them, that is, the average load of the virtual machines.

The partitioning process of particle swarm is described as follows:

- Formula (3) is used to calculate the fitness of each individual particle, and the average fitness of the particle swarm is calculated according to the fitness of the individual particle.
- Particles whose fitness is greater than the average fitness are divided into the initial phagocytic particle sub-population, and those whose fitness is less than the average fitness are divided into the initial pathogenic particle sub-population.
- Formula (4) (5) (6) is used to calculate the standard deviation of load balancing of particles in two sub-populations after the first partition, and the average standard deviation of load balancing of two sub-populations is calculated respectively.
- The standard deviations of load balancing of individual particles in two sub-populations are compared with the standard deviations of average load balancing of their corresponding sub-populations respectively.
- In the initial phagocytic particle sub-population, individuals whose load balance difference is greater than the average load balance difference of the initial phagocytic particle sub-population are divided into ordinary particle swarm.
- In the initial pathogen particle sub-population, the individual whose load balance difference is less than the average load balance difference of the initial pathogen particle sub-population is divided into ordinary particle swarm.

Finally, three particle subpopulations are formed, which are phagocytic particle sub-population, common particle sub-population, and pathogen particle sub-population.

2.4.5 Phagocytosis

Phagocytosis is one of the oldest and most basic defense mechanisms of organisms. Since the late nineteenth century, Eli Metchnikov first proposed that moving phagocytic cells can detect foreign particles in tissues and fight fiercely with potential pathogens, people began to realize this. Phagocytes can ingest microbial pathogens, but importantly, they can also ingest apoptotic cells [36]. In most organisms, there are some phagocytic cells, which phagocytize invading bacteria, viruses, parasites and other pathogens and some necrotic tissue debris to protect the organism itself.

Phagocytosis operation was carried out on the subpopulations of phagocytic particles and pathogenic particles formed after partition.

The process of phagocytosis [37] is described as follows:

- Particle individuals were separately extracted from the phagocytic and pathogenic sub-populations, and the coding sequence of the individual particles was segmented.
- Based on one of the individual particles, we compare the coding sequence fragments of the corresponding position of the other particle, and use the excellent particle sequence fragments with high fitness to phagocytize the bad particle sequence fragments with low fitness. Finally, we get a new particle individual.

2.4.6 Crossover operation

We perform crossover operations in genetic algorithm for the partitioned general particle sub-population. The crossover operation uses the two-point intersection method, and the left and right intersection positions are randomly selected.

2.4.7 New particle population

We merge particles that have undergone phagocytosis and crossover operations. Assuming that the number of individuals in the population of particle swarm is num, the number of individuals in the phagocytic subpopulation is phnum, the number of individuals in a common subpopulation is cnum, and the number of individuals in the pathogen subpopulation is panum. The relationship between the four is shown in formula (7).

$$num = phnum + cnum + panum, \tag{7}$$

The merging process is described as follows:

- If phnum > = panum, the new particle group is formed by the phagocytosis of the individual particles, the number is panum, the number of individuals who are not involved in phagocytosis in the phagocytic subpopulation, the number is phnum-panum, and the individual particles formed after the cross operation and the individual composition of the particles randomly generated if the population is less than num.
- If phnum < panum, the new particle group is formed by the phagocytosis of the individual particles, the number is phnum, the number of individuals in the pathogen particle subpopulation that are not involved in phagocytosis, the number is panum-phnum, the individual particles formed after the cross operation and the individual composition of the particles randomly generated if the population is less than num.

2.4.8 Mutation operation

The mutation operation was performed on the particle swarm which was merged again after phagocytosis and cross operation. Mutation operation uses basic bit mutation to change a bit or several bits in the particle coding sequence with a predetermined probability. Tables 1 and 2 respectively show the coding of particles before and after mutation. Appropriate mutation is conducive to breaking through the local optimal solution, which is similar to random search element, and avoids the loss of genetic diversity in the evolution process.

From Tables 1 and 2, we can see that the individual particles (1, 5, 0, 4, 2, 3, 0, 3, 5, 4) are changed to (1, 5, 3, 0, 4, 3, 0, 3, 5, 4) That is, a new individual is generated after the mutation operation. Compared with the particle individual before the mutation, the difference of the individual after the mutation is that Task2 is scheduled to execute on VM3, and Task3 is scheduled to execute on VM0, Task4 is scheduled to execute on VM4.

3 Results and discussion

3.1 Experimental environment and parameter setting

In this paper, eclipse is used for development, and cloudsim [38–40] is set up in it. Cloudsim is a kind of cloud computing simulation software written by Java and built in a very specific way. It is announced and launched by grid lab and gridbus project of University of Melbourne, Australia. In the experiment, the MIPs range of virtual machine is 30–200, and the task length range is 5000–20,000. In order to focus on the problems studied, the following experiments were carried out under the conditions described in Sect. 2.1. To verify the effectiveness of the PSO_PGA algorithm proposed in this article in cloud task scheduling, the following experiments were carried out:

Experiment 1: maintain the number of virtual machines in the data center, change the cloud tasks submitted by cloud users, and verify the performance of the PSO_PGA algorithm in terms of task completion time. In order to ensure the comparability of experiments, when increasing the number of cloud tasks, the cloud task data set of the next experiment always contains the cloud task data set of the previous experiment. For example, suppose that if the number of cloud tasks in experiment A is 200 and the number of cloud tasks in experiment B is 100, then A contains the cloud tasks in B.

Experiment 2: keep the cloud tasks submitted by cloud users, change the number of virtual machines in the data center, and verify the performance of the PSO_PGA

Table 1 Individual particle before mutation operation

Sequence number	1	2	3	4	5	6	7	8	9	10
Cloud task	Task0	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9
Virtual machine	VM1	VM5	VM0	VM4	VM2	VM3	VM0	VM3	VM5	VM4
Coding sequence	1	5	0	4	2	3	0	3	5	4

Table 2 Individual particle after mutation operation

Sequence number	1	2	3	4	5	6	7	8	9	10
Cloud task	Task0	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9
Virtual machine	VM1	VM5	VM3	VM0	VM4	VM3	VM0	VM3	VM5	VM4
Coding sequence	1	5	3	0	4	3	0	3	5	4

algorithm in terms of task completion time. Similarly, in order to ensure the comparability of the experiments, when increasing the number of virtual machines, the virtual machine resources used in the next experiment always include the virtual machines used in the previous experiment. For example, suppose the experiment with the number of virtual machines of 13 is A, and the experiment with the number of virtual machines of 11 is B, then the virtual machine in A contains B.

Experiment 3, 4: maintain the number of virtual machines in the data center and cloud tasks submitted by cloud users, change the iteration number of each algorithm, and verify whether the convergence accuracy of PSO_PGA is better under the same iteration number.

The algorithm parameters are set as shown in Table 3.

3.2 Analysis of experimental results

Experiment 1: the 10 virtual machines in the cloud data center remain unchanged, and the number of cloud tasks submitted by users is in the range of 100–1000. In the case of the same number of iterations of each algorithm, the experiment was repeated multiple times and the average value was taken to verify the performance of the PSO_PGA algorithm in terms of task completion time. The experimental result graph shows the cloud task completion time according to 1:1000. The experimental results are shown in Fig. 2.

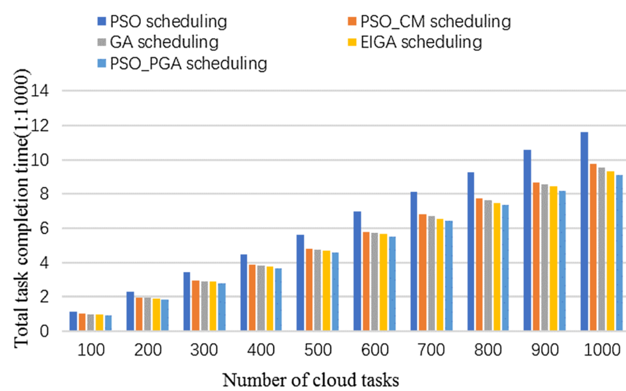

Fig. 2 Performance analysis of algorithms with different number of cloud tasks.

Discussion of experimental results: from Fig. 2 we can see that when the number of cloud tasks is 100, although the execution results of each algorithm are different, the gap is not big. As the number of tasks submitted by cloud users increases, the effect begins to appear. The PSO_PGA scheduling strategy can find a better scheduling scheme than other compared algorithms, which is more excellent than the PSO algorithm directly applied to task scheduling.

Experiment 2: the 300 cloud tasks submitted by cloud users remain unchanged, and the number of virtual machines in the cloud data center ranges from 9 to 23. In the case of the same number of iterations of each algorithm, the experiment was repeated multiple times and the

Table 3 PSO_PGA scheduling

Algorithm name	Parameter	Parameter values
PSO_PGA	Particle swarm size	300
	Number of virtual machines	9–23
	Maximum number of iterations	100
	Mutation probability	0.05
	Crossover probability	1
	Number of tasks	100–1000
	Number of individual fragments	10

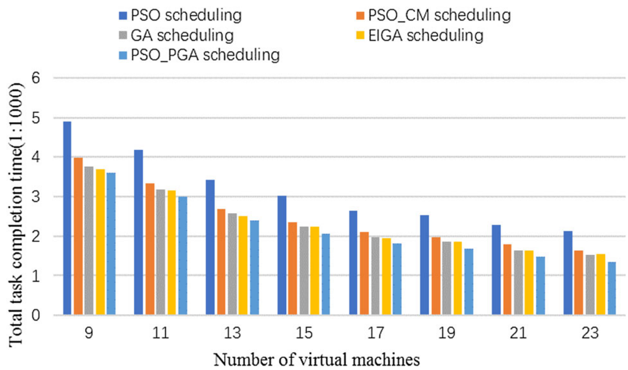

Fig. 3 Performance analysis of algorithms with different number of virtual machines

average value was taken to verify the performance of the PSO_PGA algorithm in terms of task completion time. The experimental result graph shows the cloud task completion time according to 1:1000. The experimental results are shown in Fig. 3.

Discussion of experimental results: with the increase of virtual machines, the scheduling complexity of the algorithm increases rapidly. In this case, we can see from Fig. 3 that the PSO_PGA scheduling strategy proposed in this paper can still find an excellent scheduling scheme compared with other algorithms in this paper. And it is significantly better than the particle swarm optimization algorithm directly applied to task scheduling.

Experiment 3: the number of cloud tasks submitted by cloud users 300 remains unchanged, the 10 virtual machines in the cloud data center remain unchanged, and the number of iterations is changed from 20 to 100. The experiment was repeated several times and the average was taken to verify the performance of the PSO_PGA algorithm in terms of convergence accuracy. The experimental result graph shows the cloud task completion time according to 1:1000. The experimental results are shown in Fig. 4.

Experiment 4: the number of cloud tasks submitted by cloud users 1000 remains unchanged, the 10 virtual machines in the cloud data center remain unchanged, and the number of iterations is changed from 10 to 100. The experiment was repeated several times and the average was taken to verify the performance of the PSO_PGA algorithm in terms of convergence accuracy. The experimental result graph shows the cloud task completion time according to 1:1000. The experimental results are shown in Fig. 5.

Discussion of experimental results: from Figs. 4 and 5, we can see that under the same number of iterations, the PSO_PGA scheduling strategy proposed in this paper can converge to a better solution. At the same time, we can also see from the side that the standard PSO algorithm is suitable for solving numerical optimization problems in the continuous domain, but for such problems as task scheduling, it cannot be directly solved, and it is easy to fall into a local optimal solution.

Experiments 1, 2, 3 and 4 show the effectiveness of the proposed algorithm in task scheduling, which shows that this research has practical significance.

4 Conclusion

For the cloud task scheduling problem, this paper studies how to minimize the maximum time span of the task. The standard PSO algorithm is suitable for solving numerical optimization problems in continuous domain, but for cloud computing task scheduling problem, it can not be directly solved, and it is easy to fall into local optimal solution. A particle swarm optimization genetic hybrid algorithm PSO_PGA based on phagocytosis is proposed. The algorithm uses the fitness function and the load balancing standard deviation to perform the first division and the second division for each generation of particle groups and introduces phagocytosis mechanism. The phagocytosis

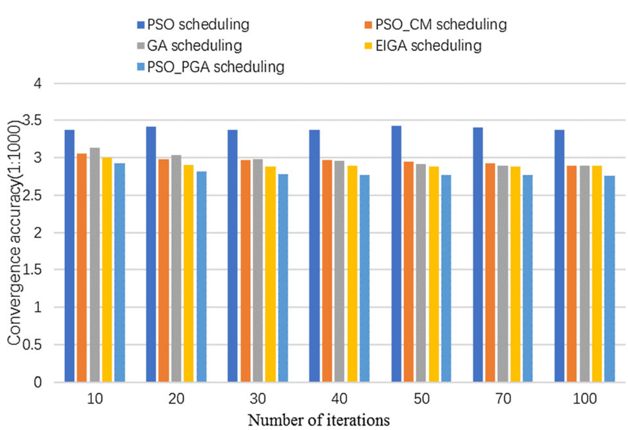

Fig. 4 Comparison of convergence accuracy of algorithms with different iterations under 300 cloud tasks.

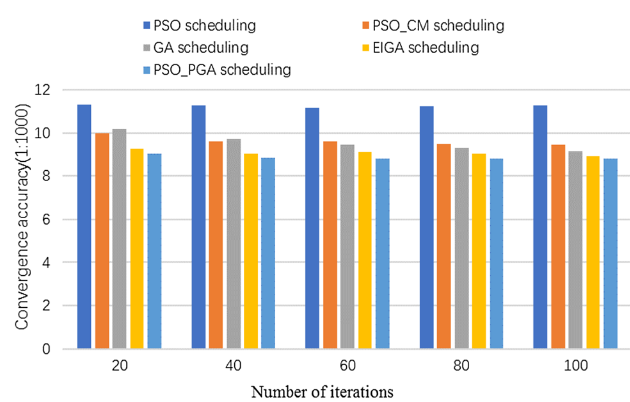

Fig. 5 Comparison of convergence accuracy of algorithms with different iterations under 1000 cloud tasks

mechanism and the crossover mutation of genetic algorithm are used to operate different sub-populations of particles, various ways to change the position of the particles, expand the search scope of the algorithm in the solution space. The merging operation of sub-populations ensures the diversity of the individual particles, and improves the possibility of the algorithm jumping out of the local optimal solution. Finally, through four different experiments designed, it is shown that the new algorithm formed by fusing the particle swarm algorithm and the genetic algorithm in the manner in this paper can improve the quality and convergence of the solution obtained by the meta-heuristic algorithm itself, which shows the effectiveness of the proposed algorithm in cloud task scheduling. It shows that this study has practical significance.

Acknowledgements This research was financially supported by National key research and development plan: integrated water ecological management and water resource smart regulation technology demonstration of typical lakes in Inner Mongolia (2019YFC0409205), and National Natural Science Foundation of China (61962047), and Natural Science Foundation of Inner Mongolia Autonomous Region of China (No. 2019MS06015).

References

1. Hayes, B.: Cloud computing. Commun. ACM **51**(7), 9–11 (2008)
2. Djebbar, E.I., Belalem, G.: Benadda M (2016) Task scheduling strategy based on data replication in scientific Cloud workflows. Multiagent Grid Syst **12**(1), 55–67 (2016)
3. Sujana, J., Jennifa, A., Revathi, T., Priya, T., Siva, S., Muneeswaran, K.: Smart PSO-based secured scheduling approaches for scientific workflows in cloud computing. Soft Comput. **23**(5), 1745–1765 (2019)
4. Somasundaram, T.S., Govindarajan, K.: CLOUDRB: a framework for scheduling and managing high-performance computing (HPC) applications in science cloud. Fut. Gener. Comput. Syst. **34**, 47–65 (2014)
5. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: issues and challenges. J. Grid Comput. **14**(2), 217–264 (2016)
6. Abdullah, M., Al-Muta'a, E.A., Al-Sanabani, M.: Integrated MOPSO algorithms for task scheduling in cloud computing. J. Intell. Fuzzy Syst. **36**(2), 1823–1836 (2019)
7. Mathew T, Sekaran K.C., Jose, J.: Study and analysis of various task scheduling algorithms in the cloud computing environment. In: International conference on advances in computing, communications and informatics (ICACCI, 2014). IEEE, 2014: 658–664.
8. Liao, Q., Jiang, S., Hei, Q., et al.: Scheduling stochastic tasks with precedence constrain on cluster systems with heterogenous communication architecture. Algorithm Arch. Parallel Process. **9532**, 85–99 (2015)
9. Yao, H., Fu, X., Li, H., Dong, G., Li, J.: Cloud task scheduling algorithm based on improved genetic algorithm. Intl. J. Perform. Eng. **13**(7), 1070–1076 (2017)
10. Huang, X., Li, C., Chen, H., An, D.: Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies. Clust. Comput. **23**(2), 1137–1147 (2020)
11. Arul Xavier, V.M., Annadurai, S.: Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. Clust. Comput. **22**(1), 287–297 (2019)
12. Sun, Y., Li, J., Fu, X., Wang, H., Li, H.: Application research based on improved genetic algorithm in cloud task scheduling. Intell. Fuzzy Syst. **38**, 239–246 (2020)
13. Zhou, J., Dong, S.-B., Tang, D.-Y.: Task scheduling algorithm in cloud computing based on invasive tumor growth optimization. Chin. J. Comput. **41**(6), 1140–1155 (2018)
14. Muthulakshmi, B., Somasundaram, K.: A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment. Clust. Comput. **22**, 10769–10777 (2019)
15. Madni, S.H.H., Abd Latiff, M.S., Abdulhamid, S.M., Ali, J.: Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Clust. Comput. **22**(1), 301–334 (2019)
16. Valarmathi, R., Sheela, T.: Ranging and tuning based particle swarm optimization with bat algorithm for task scheduling in cloud computing. Clust. Comput. **22**, 11975–11988 (2019)
17. Xuan, C., Dan, L.: Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. Clust. Comput. **22**, 2761–2769 (2019)
18. Senthil Kumar, A.M., Venkatesan, M.: Task scheduling in a cloud computing environment using HGPSO algorithm. Clust. Comput. **22**(1), 2179–2185 (2019)
19. Li, H., Yu, H.: Task scheduling strategy based on evolutionary algorithms in cloud computing. J. East China Univ. Sci. Technol **4**, 556–562 (2015)
20. Li, T., Zhang, F., Wang, M.: Improved two period cloud task scheduling algorithm with genetic algorithm. J. Chin. Comput. Syst. **38**(06), 1305–1310 (2017)
21. Fu, X., Cang, Y.: Task scheduling and virtual machine allocation policy in cloud computing environment. J. Syst. Eng. Electron. **26**(4), 847–856 (2015)
22. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks (ICNN 95), pp. 1942–1948 (1995)
23. Li, Y., Lin, Y.: Cloud task scheduling based on hybrid particle swarm optimization algorithm. Comput. Technol. Autom. **1**, 73–77 (2014)
24. Guo, L.Z., Wang, Y.J., Zhao, S.G., et al.: Particle swarm optimization embedded in variable neighbourhood search for task scheduling in cloud computing. J. Donghua Univ. **30**(2), 145–152 (2013)
25. Zhao, S., Fu, X., Li, H., Dong, G., Li, J.: Research on cloud computing task scheduling based on improved particle swarm optimization. Intl. J. Perform. Eng. **13**(7), 1063–1069 (2017)
26. Levenick, J.: Showing the way: a review of the second edition of Holland's adaptation in natural and artificial systems. Artif. Intell. **100**, 331–338 (1998). https://doi.org/10.1016/s0004-3702(98)00017-4
27. Agarwal, M., Srivastava, G.M.S.: Genetic algorithm-enabled particle swarm optimization (PSOGA)-based task scheduling in cloud computing environment. Intl. J. Inf. Technol. Decis. Making **17**(04), 1237–1267 (2018). https://doi.org/10.1142/s0219622018500244
28. Ma, Y., Yun, W.: Research progress of genetic algorithm. Appl. Res. Comput. **4**, 1201–1206 (2012)
29. Xia, G., Zhou, C., Jin, S., Huang, C., Xing, J., Liu, Z.: Sensitivity enhancement of two-dimensional materials based on genetic optimization in surface plasmon resonance. Sensors **19**(5), 1198 (2019). https://doi.org/10.3390/s19051198
30. He, Y., Wang, X., Zhao, S., Zhang, X.: Design and applications of discrete evolutionary algorithm based on encoding transformation. J. Softw. **29**(9), 2580–2594 (2018)

31. Tasgetiren, M.F., Pan, Q.K., Suganthan, P.N.: A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. Inf. Sci. **181**(16), 3459–3475 (2011)
32. Jia, D.L., Duan, X.T., Khan, M.K.: Binary Artificial Bee Colony optimization using bitwise operation. Comput. Ind. Eng. **76**, 360–365 (2014)
33. Kiran, M.S.: The continuous artificial bee colony algorithm for binary optimization. Appl. Soft Comput. **33**, 15–23 (2015)
34. He, Y.C., Wang, X.Z., Kou, Y.Z.: A binary differential evolution algorithm with hybrid encoding. J. Comput. Res. Dev. **44**(9), 1476–1484 (2007). ((**in Chinese with English abstract**))
35. Feng, X., Pan, Y.: DPSO resource load balancing in cloud computing. Comput. Eng. Appl. **49**(06), 105–108 (2013)
36. Rosales, C., Uribe-Querol, E.: Phagocytosis: a fundamental process in immunity. Biomed. Res. Int. (2017). https://doi.org/10.1155/2017/9042851
37. Yao, H.: Research on Task Scheduling Strategy Based on Improved Genetic Algorithm in Cloud Computing Environment. Inner Mongolia Agricultural University. (2018) ((**in Chinese with English abstract**))
38. Goyal, T., Singh, A., Agrawal, A.: Cloudsim: simulator for cloud computing infrastructure and modeling. In: International conference on modelling optimization and computing, pp. 3566–3572 (2012)
39. Mehmi, S., Verma, H.K., Sangal, A.L.: Simulation modeling of cloud computing for smart grid using CloudSim. J. Electr. Syst. Inf. Technol. **4**(1), 159–172 (2017)
40. Rani, E., Kaur, H.: Study on fundamental usage of CloudSim simulator and algorithms of resource allocation in cloud computing. (2017). https://doi.org/10.1109/ICCCNT.2017.8203998

Xueliang Fu graduated from Xi'an University of Technology, majoring in Computer Applications in china in 1992, and completed his M.S. and PhD at Dalian University of Technology in china in July 2005 and 2008 respectively. He is a professor and doctoral supervisor of College of Computer and Information Engineering, Inner Mongolia Agricultural University, China. Currently the director of the library of Inner Mongolia Agricultural University, China. His current research is focused on the Intelligent Computing, Data Mining, Information Safety, Network Information Optimization, Dominance of Graphs.

Yang Sun graduated from College of Computer and Information Engineering, Inner Mongolia Agricultural University in china in 2017 and recommended to school of computer and information engineering of Inner Mongolia Agricultural University for master's degree after graduation. His current research is focused on the Intelligent Computing, Data Mining, Cloud Computing, Swarm Intelligence Optimization Algorithm, Big Data.

Haifang Wang graduated from College of Computer and Information Engineering, Inner Mongolia Agricultural University in china in 2017. In September 2018, she was admitted to the school of computer and information engineering of Inner Mongolia Agricultural University in china for master's degree. Her current research is focused on the Intelligent Computing, Data Mining, Cloud Computing, Block chain Research, Big Data.

Honghui Li graduated from University of Electronic Science and Technology of China, majoring in computer software in china in 1992, completed her M.S. at Inner Mongolia University in china in July 2002 and completed her PhD at Concordia University, Canada. She is a professor and master's supervisor of College of Computer and Information Engineering, Inner Mongolia Agricultural University, China. Her current research is focused on the Computer network, Internet of things applications, Intelligent Computing, Big data technology.