
Research on Cloud Computing Task Scheduling Based on PSOMC

Kun Li*, Liwei Jia and Xiaoming Shi

*Computer Teaching and Research Section Department of Public Infrastructure,
Henan Medical College, Zhengzhou, Henan, 451191, China*

E-mail: sunlik_1982@126.com; zzujiawei@126.com; hnsxm1983@yeah.net

**Corresponding Author*

Received 17 June 2022; Accepted 15 July 2022;
Publication 09 November 2022

Abstract

How to better reduce the task scheduling time and consumption cost in cloud computing has always been a hot topic of current research. In this paper, we propose a cloud computing task scheduling strategy based on the fusion of Particle Swarm Optimization and Membrane Computing. Firstly, a task scheduling model with time function and cost function as the target is proposed, secondly, on the basis of particle swarm algorithm, chaos operation is used in population initialization to improve the diversity of rich understanding, adaptive weight factor based on sinusoidal function is used to avoid the algorithm falling into local optimum, Membrane Computing is used in individual screening to improve the quality of individual solutions, and finally, in The performance of the PSOMC algorithm is illustrated by comparing six benchmark test functions in simulation experiments, and it is also verified that the completion time and consumption cost are significantly better than those of the ACO, PSO and MC algorithms for different number of tasks.

Keywords: Cloud computing, task scheduling, chaos, adaptive weights.

Journal of Web Engineering, Vol. 21.6, 1749–1766.

doi: 10.13052/jwe1540-9589.2161

© 2022 River Publishers

1 Introduction

With the development of network technology, cloud computing, as the most widely used business service model, has a profound impact on social development and economic development [1]. One of the core technologies of cloud computing system is task scheduling, and the rationality of the scheduling strategy is directly related to the scheduling efficiency and operational performance of the whole cloud system. A reasonable scheduling strategy can reduce the task completion time and completion cost, improve the system utilization, reliability and user satisfaction, and bring a good experience to users. However, the complexity of the cloud system itself and the diversity of users make the research on task scheduling in the cloud environment extremely difficult. At present, there is still a large research space about multi-objective scheduling, and since the established multi-objective task scheduling is an NP problem [2], using meta-heuristic algorithms to solve it is one of the most used strategies by current scholars [3]. The particle swarm algorithm [4] is a bionic algorithm proposed by Eberhart and Kennedy in 1995 based on the behavior and characteristics of birds in biology during foraging, which lies in the structural nodes and the algorithm is simple and easy to implement. We firstly construct a task scheduling model based on time function and cost function in this paper, and secondly improve on the basis of particle swarm algorithm in 3 aspects: (1) in the population of individuals, we use chaos idea for optimization, which makes the individuals of the population keep diversity and enrich the format of understanding; (2) use sine function in the adaptive weight of individuals, which can avoid the individuals to fall into (3) in individual screening we use Membrane Computing to remove redundant individuals, which ensures the quality of individuals in the next iteration and effectively reduces the algorithm running time. Simulation experiments show that the algorithm has good performance in benchmark test functions and achieves better results in cloud computing task scheduling.

The subsections of this paper are organized as follows: in Section 2, the related research progress about cloud computing task scheduling is described, in Section 3, the task scheduling model based on time function and cost function is described, in Section 4, the task scheduling process based on PSOMC algorithm fusion is described, in Section 5, the algorithm of this paper is verified by simulation experiments, and in Section 6, the algorithm of this paper is summarized.

2 Related Research

Currently, there are two main types of research using metaheuristic algorithms for cloud computing task scheduling; one strategy is to use a single metaheuristic algorithm for research. For example, Literature [5] proposed an improved bat algorithm for cloud computing task scheduling, and simulation experiments illustrated that the algorithm has good performance in terms of virtual machine load and task completion time; Literature [6] proposed the use of whale algorithm for cloud computing task scheduling, which has good results in terms of virtual machine, completion time and task scheduling; Literature [7] proposed an adaptive PSO algorithm for cloud computing task scheduling strategy, and simulation experiments illustrated that the algorithm has good results in terms of time and cost indicators; Literature [8] proposed a cloud computing task scheduling algorithm based on MVOGA, and simulation experiments verified that it has good results in the number of tasks between 1000 and 2000; Literature [9] used the cat swarm algorithm for cloud computing task scheduling. Literature [10] proposed a whale algorithm with reverse integrated learning for cloud computing task scheduling strategy, by using reverse learning in the population of the whale algorithm, the performance of the whale algorithm is improved, and the results of simulation experiments show that the algorithm has a good scheduling effect in reducing the running time and running cost of cloud computing; Literature [11] proposed an electronic search strategy based on the genetic algorithm, and simulation experiments show that this algorithm outperforms traditional metaheuristics such as GA, PSO in terms of time completion and cost consumption in cloud computing; Literature [12] proposes a hybrid antlion optimization algorithm based on elite differential evolution for cloud computing task scheduling strategy, and simulation experiments illustrate that it has better results in terms of maximizing resource utilization and minimizing time.

Another strategy is the fusion of multiple meta-heuristic algorithms for scheduling processing such as Literature [13] proposed a fusion algorithm based on artificial bee colony and whale optimization algorithms; Literature [14] proposed the fusion of ACO algorithm and PSO algorithm for cloud computing task scheduling; Literature [15] proposed a fusion of GA algorithm and PSO algorithm for cloud computing task scheduling strategy; Literature [16] proposed ABC and FOA algorithm fusion for cloud computing task scheduling strategy; Literature [18] used the fusion of particle swarm

algorithm and ant colony algorithm for cloud computing task scheduling, and simulation experiments illustrated better results in terms of completion time and resource utilization.

From the above study, it is found that using a single metaheuristic algorithm is effective in improving the effect of task scheduling in cloud computing, but there is not much room for improving the performance of the algorithm because it only optimizes the algorithm itself, which directly affects the effect of the algorithm in task scheduling; while the fusion algorithm using two metaheuristics performs well in the effect of task scheduling in cloud computing, it leads to a possible increase in the complexity of the algorithm because of the existence of certain algorithmic complexity of the algorithm itself.

3 Task Scheduling Model

The problem that cloud computing task scheduling actually solves is how to map the tasks submitted by users to the virtual machine resource nodes in a reasonable way. In the process of task scheduling, the scheduling objectives and scheduling strategies are combined to optimize the allocation of resources and the operational efficiency of the cloud computing system. In this paper, we mainly consider two aspects, task completion time and task completion cost, and finally make the task completion time optimal, cost reduction and load balancing at the same time; the size of data transmission capacity has a great relationship with the size of bandwidth. Cloud computing task scheduling is to assign a certain number of tasks to different resource nodes. In order to establish an effective scheduling model, we specify that: the tasks submitted by users are split into subtasks, which are independent from each other and do not have dependencies; the number of tasks is much larger than the number of virtual machines; and the unit cost on the virtual resource nodes is known. Task definition $T = \{t_1, t_2, \dots, t_m\}$ (m denotes the number of tasks) and virtual machine definition $R = \{r_1, r_2, \dots, r_n\}$ (n denotes the number of virtual machines). Therefore, the mapping relationship between tasks and virtual resources in multi-objective task scheduling is expressed using matrix S as follows.

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \quad (1)$$

3.1 Task Completion Time

The task completion time function is the sum of the execution time of the task on the virtual machine and the transfer time of the task to the virtual machine. It is represented using the following matrix.

$$E_{m \times n} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix} \quad (2)$$

$$e_{ij} = Exe_time_{ij} + Trans_time_{ij} \quad (3)$$

In the equation, e_{ij} denotes the time for VM r_j to complete task t_i , Exe_time_{ij} denotes the time for VM r_j to perform task t_i , and $Trans_time_{ij}$ denotes the time for VM r_j to transfer task. $Exe_time_{ij} = t_{i_length}/rv_{j_comp}$, $Trans_time_{ij} = t_{i_inputsize}/r_{j_bw} + t_{i_outputsize}/r_{j_bw}$. t_{i_length} denotes the instruction length of task t_i , rv_{j_comp} denotes the computing power of virtual machine r_j , which is the product of the number of CPUs contained in the virtual machine and the computing power of a single CPU, $t_{i_inputsize}$ denotes the file size submitted by task t_i , $t_{i_outputsize}$ denotes the file size output by task t_i when execution is completed, and r_{j_bw} denotes the bandwidth.

Therefore multiple tasks are being processed serially on one VM, so the task completion time for each VM is done according to Equation (4), while VMs in cloud computing are executed in parallel, so the time used for the last VM to finish processing the task is the entire task completion time, which is shown in Equation (5).

$$r_{j_comTime} = \sum_{i=1}^k e_{ij} \quad (4)$$

$$TotalTime = \max(r_{j_comTime}) \quad (5)$$

3.2 Task Completion Costs

The task cost function is defined to represent the sum of the computation cost and transmission cost of all virtual machines. That is, Equation (6) is as follows.

$$TotalCost = \sum_{j=1}^n \sum_{i=1}^m Exe_cost_{ij} + \sum_{j=1}^n \sum_{i=1}^m Trans_cost_{ij} \quad (6)$$

where Exe_cost_{ij} denotes the execution cost of VM r_j to perform task t_i and $Trans_cost_{ij}$ denotes the transmission cost of VM r_j to perform task t_i . Exe_cost_{ij} is primarily the product of the time of VM r_j to perform task t_i , the number of CPUs in VM r_j , the processing power of a single CPU in VM r_j , and the unit cost of a single CPU in VM r_j . $Trans_cost_{ij}$ is the product of the time of VM r_j to transmit task t_i and the unit broadband cost of VM r_j .

3.3 Cloud Computing Objective Function

In this paper, we establish a task scheduling model from task completion time and task completion cost to achieve optimal task completion time, reduce the bottom cost and take into account the purpose of load balancing. In order to facilitate the calculation, firstly, the task completion time and task completion cost are normalized by using the linear function method with different scales. Therefore, the normalized functions of task completion time and task completion cost are as follows.

$$ConTime = \frac{TotalTime - TotalTime_{\min}}{TotalTime_{\max} - TotalTime_{\min}} \quad (7)$$

$$ConCost = \frac{TotalCost - TotalCost_{\min}}{TotalCost_{\max} - TotalCost_{\min}} \quad (8)$$

where $TotalTime_{\min}$ and $TotalTime_{\max}$ denote the time completed on the best performing virtual machine and the time completed on the worst performing virtual machine, respectively. $TotalCost_{\min}$ and $TotalCost_{\max}$ denote the cost completed on the best performing virtual machine and the cost completed on the worst performing virtual machine, respectively.

4 PSOMC-based Cloud Computing Task Scheduling

4.1 Particle Swarm Algorithm

The particle swarm algorithm is a meta-heuristic algorithm with good performance. Its idea comes from the relationship between birds in nature in the process of hunting in a group, which has a good performance in the global optimal solution, and at the same time, it has the advantages of simple algorithm process, less parameters involved in the algorithm, and better algorithm performance. Each particle has two characteristics of position and velocity, which are expressed as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ and

$v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$, respectively, and the velocity and position of each particle are updated a

$$v_{id}^{k+1} = \omega \times v_{id}^k + c_1 \times r_1 \times (pbest_{id}^k - x_{id}^k) + c_2 \times r_2 \times (gbest_{id}^k - x_{id}^k) \quad (9)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (10)$$

In the above equation, $pbest_{id}^k$ denotes the optimal value of individual id in dimension k , $gbest_{id}^k$ denotes the global optimal value of individual id in dimension k , c_1 is the learning factor under the individual optimal value, c_2 is the learning factor under the global optimal value, r_1 is the random number of the individual optimal value, r_2 is the random number of the global optimal value, and ω is the weight.

4.2 Particle Swarm Algorithm Based on Membrane Computing

Although the particle swarm algorithm has the advantages of few parameters and easy implementation simplicity, the algorithm will gradually fall into stagnation at a later stage as the number of iterations gradually increases, in order to avoid this phenomenon, we propose a solution to optimize the performance of the algorithm in three aspects. The chaotic operation of each individual is used to select individuals with better fitness values, which improves the diversity of the algorithm; second, for the phenomenon that the weight factor is a fixed value, we try to propose the design idea of adaptive weights, which links the number of iterations to the weight change in the process of adaptive weights, which can help the individuals to have a better speed in the later stage; in the individual screening, we use membrane calculation to divide the individual selection, we use membrane computing to divide the particles into different regions, and improve the robustness of the individual solution by selecting particles in different regions and then processing them using the global optimization capability of membrane computing.

(1) Population initialization

Chaos operation is the method used by most metaheuristics for population initialization, and it is widely used because it can improve the population diversity at a later stage. In this paper, the chaos strategy shown in Equation (11) is used in the population initialization of the particle swarm

algorithm.

$$x_i^{k'} = \begin{cases} (1 + x_i^k)/\delta & 0 < \delta \leq 0.5 \\ (1 - x_i^k)/\delta & 0.5 < \delta \leq 1 \end{cases} \quad (11)$$

$$x_i^t = \begin{cases} x_i^k & \text{fitness}(x_i^k) > \text{fitness}(x_i^{k'}) \\ x_i^{k'} & \text{fitness}(x_i^{k'}) > \text{fitness}(x_i^k) \end{cases} \quad (12)$$

In Equation (11), where δ denotes the chaos parameter, x_i^k and $x_i^{k'}$ denote the individual before and after the chaos at the k th iteration, respectively. In Equation (12), the new individuals are obtained by comparing the fitness values of the individuals before and after chaos.

(2) Adaptive weighting factor based on sine function

In the particle swarm algorithm, the weight factor has an important role for the individual in the algorithm, but in the basic particle swarm algorithm, the weight factor is a fixed array, which does not transform with the number of iterations, so it cannot guarantee that the particles still have a good speed in the later stage, therefore, we set the weight factor with adaptive nature, and the setting result is shown in the formula (13). In this formula, we set the sinusoidal function, which can ensure that the algorithm with the increasing number of iterations, the ω value can be within a reasonable transformation range, and by setting the maximum and minimum values, we can effectively ensure that ω changes in a reasonable range and avoid the possibility of the algorithm falling into the local optimal solution.

$$\omega = (\omega_{\max} - \omega_{\min}) \times \sin\left(\frac{t}{t_{\max}}\right) \quad (13)$$

where ω_{\min} and ω_{\max} denote the maximum and minimum values of adaptive weights, t and t_{\max} denote the current and maximum number of iterations, respectively.

(3) Individual screening

In the particle swarm algorithm, there is no relevant operation for the individuals after each iteration, resulting in many redundant particle individuals entering the next iteration of the algorithm, which reduces the operational efficiency of the algorithm. We use membrane computation for the screening of individuals, and the study of MC [17] comes from a computational model

proposed by the scholar G. Paun, which is a computational model abstracted from the biological mechanism of the cell from the perspective of the study of the cell, taking the constituent structure of the cell as the object of study. The model is mainly divided into a main membrane and an auxiliary membrane. The main membrane is used to assist the local optimization in the particle swarm algorithm, and the auxiliary membrane is used to assist the global search in the particle swarm algorithm. The specific process is to send the particles into different membranes, and the global optimization results obtained from the auxiliary membranes are fed into the main membrane by relying on the evolutionary rules, and then iterate continuously until the maximum number of iterations is reached, and finally the optimal particles of the whole membrane system are obtained. The evolution rule for the operation from the main membrane to the auxiliary membrane uses the structure of Equation (14) as follows:

$$X'_{1L} \dots X'_{nL} \dots X'_{dL} \rightarrow X'_{1L} \dots X'_{nL} \dots X'_{dL} (X'_{1L} \dots X'_{nL} \dots X'_{dL}) \quad (14)$$

In the individual screening, the particles are assigned to the membrane structure to ensure that the primary and secondary membranes contain at least one particle, where $q_i (i = 1, 2, 3 \dots n)$ denotes the individual particle in the secondary membrane, and let the dimension of the search space be d , the position $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$ of the particle will be regarded as the object processed by the membrane system, and the velocity of the particle denotes the current state of the individual, so the set of solutions corresponding to all particles in the population is the set of objects of the membrane system, and $x_{1i}^t, x_{2i}^t, \dots, x_{di}^t$ represents the solution corresponding to each particle in the t th generation of the iteration of the algorithm within the i th membrane.

(4) Fitness function

In the particle swarm algorithm, the fitness value is used as the condition for selecting individuals, and the larger the fitness value is, the better the individuals are, and the worse the individuals are. Let x_i^t be the i th particle of the t th iteration, $Contime(x_i^t)$ denote the execution time function corresponding to individual i , and $Concost(x_i^t)$ denote the execution cost function corresponding to individual i . In this paper, the fitness functions of execution time and consumption cost corresponding to each particle individual are

expressed as follows.

$$F_1(x_i^t) = \frac{1}{\log_2 \text{Contime}(x_i^t)} \quad (15)$$

$$F_2(x_i^t) = \frac{1}{\log_2 \text{Concost}(x_i^t)} \quad (16)$$

Since the objective function of the cloud computing task sought in this paper is to obtain the minimum value, the larger the fitness function of the individual is required (i.e., the superior individual), the value of the fitness function of the objective in this paper is as follows:

$$\text{Fitness}(x_i^t) = F_1(x_i^t) + F_2(x_i^t) \quad (17)$$

The objective function emphasizes that the cost and time indicators have equal roles in cloud computing tasks, so it can reflect the user's evaluation of cloud computing task scheduling services more objectively, and obtain the optimal cloud computing task scheduling solution by solving for the optimal individual particle position.

4.3 Algorithm steps

Step 1: Set the values of various parameters required for the task in cloud computing. Initialize all the parameters in the PSOMC algorithm, set the population size of the algorithm, and the maximum number of iterations required by the algorithm.

Step 2: According to the task scheduling model in cloud computing, we use the expression of the task scheduling function (Equation (17)) as the fitness function of the particle swarm algorithm, i.e., finding the optimal particle swarm individual is finding the optimal task scheduling solution.

Step 3: Perform population initialization in the population of the particle swarm optimization algorithm according to Equations (11)–(12).

Step 4: Performing sinusoidal optimization of the factors of adaptive weights using Equation (13).

Step 5: Performing membrane calculations using Equation (14) for individual deletion after each iteration.

Step 6: Calculating the objective function values of the tasks corresponding to the individuals of the particle swarm algorithm.

Step 7: Compare the task objective function value with the objective function value in the previous iteration, if it is less than the result of the previous iteration then replace the original particle swarm individual position, otherwise keep the same.

Step 8: Compare the relationship between the current iteration number and the maximum limited iteration number, if the former is smaller than the latter, add 1 to the number of iterations, turn to step 4, otherwise, turn to step 9.

Step 9: Output the optimal particle swarm algorithm individual position, which is the best mobile cloud computing task solution.

5 Simulation Experiment

In order to better verify the advantages of the algorithm in this paper, we choose Ant Colony Algorithm (ACO), PSO and MC as the comparison algorithms, the hardware environment is Core I7, the memory is 8G, the hard disk capacity is 1T, and the software environment is Win10, c_1, c_2 is 0.5, ω is set to 0.5, 3 auxiliary membranes in MC algorithm, ω_{\min} and ω_{\max} in PSOMC algorithm are 0.1 and 0.9, respectively, and δ is set to 0.3. The number of iterations is 200. To illustrate the effect of PSOMC algorithm in time and cost comparison in cloud computing task scheduling, we divide the number of tasks into small task sets according to [100, 1000] and [3000, 10000] and large task sets. In the number of small tasks we set between 100 and 1000, because we consider the number of tasks scheduled in some conditions of cloud computing environment is not much, so we think between 100 and 1000 is more appropriate. In the case of large tasks, we believe that the range of 3000–10000 can reasonably reflect the actual situation of scheduling some current cloud computing tasks.

5.1 Comparison of Fitness Values

The results of the fitness function of this paper are compared with those of the three algorithms ACO, PSO and PSOMC for different number of iterations, and the results are shown in Figure 1. It is found from Figure 1 that the fitness function value curves of all the four algorithms decrease to different degrees as the number of iterations gradually increases, but the value of the fitness value curve of this paper is always the lowest. This also shows that the performance of the algorithm in this paper is better than the other three algorithms, which indicates that the performance of the algorithm can indeed

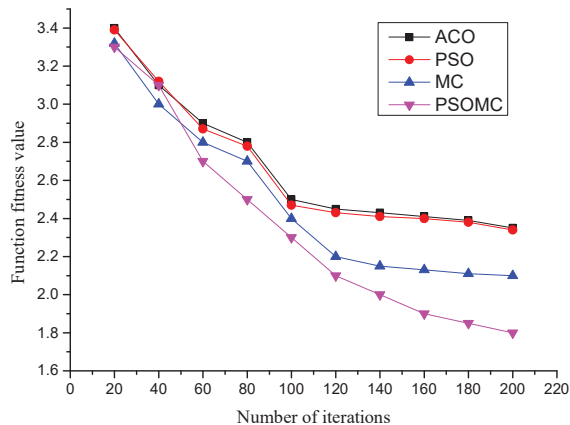


Figure 1 Comparison of the fitness values of the four algorithms.

be effectively improved by population initialization, adaptive weighting and membrane calculation, which can be better used for the scheduling of tasks under cloud computing.

5.2 Effectiveness of Cloud Computing Task Scheduling

(1) Comparison of four algorithms under small tasks

Figures 2 and 3 show the comparison results of the completion time and consumption cost of each of the four algorithms for small number of tasks. From Figure 2, it is found that the completion time of all the four algorithms increases as the number of tasks increases, and it is found that the difference between the four algorithms is not significant when the number of tasks is between 100 and 400, and between 400 and 1000, MC and PSOMC clearly outperform ACO and PSO, but PSOMC still has an advantage over MC, although In terms of the average time to complete the task, PSOMC is 8.2%, 10.8%, and 11.9% lower than MC, PSO, and ACO algorithms, respectively. From Figure 3, it is found that the consumption cost of all the four algorithms increases as the number of tasks increases, and from the results in the Figure 3, it is found that there is no significant difference between the four algorithms when the number of tasks is 100–300, while the difference between the four algorithms gradually becomes larger when the number of tasks is 300–1000. The average cost savings of PSOMC compared with MC, PSO and ACO algorithms are 9.8%, 11.3% and 12.7%, respectively. However, in general, the PSOMC algorithm has a certain cost advantage.

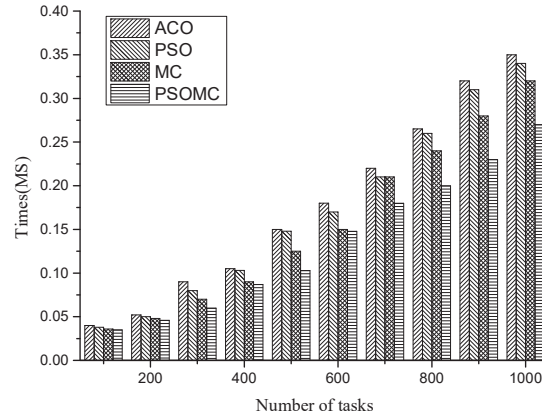


Figure 2 Comparison of the completion time under four algorithms for small tasks.

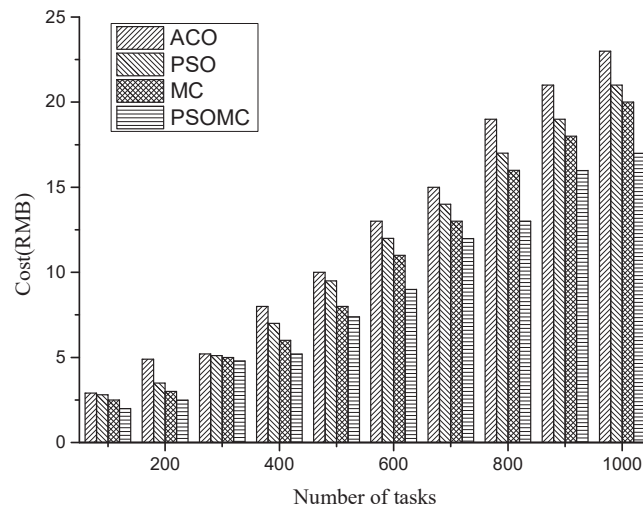


Figure 3 Comparison of consumption costs under four algorithms for small tasks.

The above comparison results show that the PSOMC algorithm has some advantages in the case of small number of tasks, but the overall advantage is not very obvious.

(2) Comparison of the four algorithms under large tasks

Figures 4 and 5 show the comparison results of completion time and consumption cost for each of the four algorithms for a large number of tasks. From Figure 4, it is found that the completion time of all four algorithms

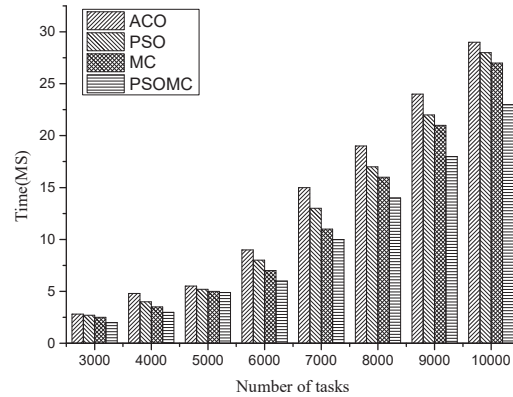


Figure 4 Comparison of the completion time under four algorithms for large tasks.

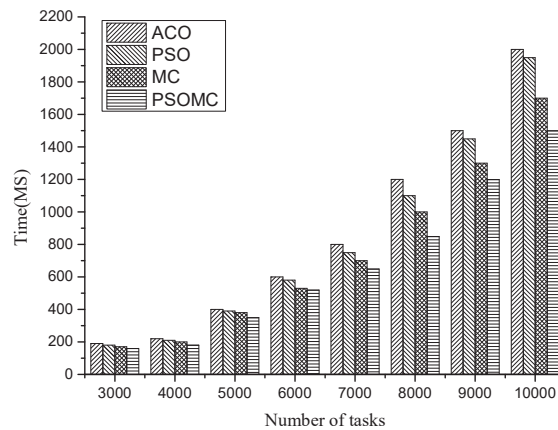


Figure 5 Comparison of consumption costs under four algorithms for large tasks.

increases with the increasing number of tasks, and the results from the Figure 4 show that there is not much difference between the four algorithms when the number of tasks is between 3000 and 4000, and from 4000 to 10000, MC and PSOMC obviously outperform ACO and PSO, but PSOMC still has certain advantages over MC. The average completion time of the whole task is 10.2%, 17.8%, and 19.9% for PSOMC compared to MC, PSO, and ACO algorithms, respectively. From Figure 5, it is found that the consumption cost of all four algorithms increases as the number of tasks increases, and the results show that there is no significant difference between the four algorithms when the number of tasks is 3000–5000, while the difference between the four algorithms gradually becomes larger when the number of

tasks is 5000–10000. The average consumption cost of PSOMC is nearly 11.8%, 19.3% and 22.7% respectively, which shows that PSOMC algorithm has a better cost advantage. The above comparison results show that the PSOMC algorithm is able to perform better in the case of large number of tasks, and the overall advantage is very obvious.

Through the comparison in terms of completion time and cost consumption under different number of tasks, we found that PSOMC algorithm has better performance effect, especially the reasonable scheduling under large number of tasks can effectively reduce the scheduling time and cost. This is because the PSOMC algorithm has undergone operations such as population initialization, adaptive weighting and individual screening, which improve the performance of the algorithm. Especially, the comparative results of the fitness function values illustrate the good performance of the PSOMC algorithm. The comparison of the simulation experiments shows that PSOMC does have good results in terms of time and cost compared with MC, ACO, and PSO.

6 Conclusion

In this paper, a cloud computing task model based on time function and cost function is constructed, and the PSOMC algorithm generated by fusion of two algorithms, PSO and MC, is used for task scheduling. PSOMC is obtained from chaos-based initialization, adaptive weighting of sinusoidal functions and individual screening of membrane computation based on particle swarm algorithm. The simulation experiments illustrate that the algorithm, compared with ACO, PSO and MC can effectively reduce the completion time and consumption cost under different number of tasks, and in the next step, the impact of virtual machine in task scheduling needs to be considered.

References

- [1] A. Razaque, N.R. Vennapusa, N. Soni, et al., ‘Task scheduling in cloud computing’, 2016 IEEE long island systems, applications and technology conference (LISAT). IEEE, Farmingdale, NY, USA, pp. 1–5. June 2016.
- [2] I.M. Ibrahim, ‘Task scheduling algorithms in cloud computing: A review’, Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 12, No. 4, pp. 1041–1053. April 2014.

- [3] A.R. Arunarani, D. Manjula, V. Sugumaran, 'Task scheduling techniques in cloud computing: A Literature survey', *Future Generation Computer Systems*, Vol. 91, pp. 407–415. February 2019.
- [4] R. Poli, J. Kennedy, T. Blackwell, 'Particle swarm optimization', *Swarm intelligence*, Vol. 1, No. 1, pp. 33–57. August 2007.
- [5] T. Bezdan, M. Zivkovic, N. Bacanin N, et al., 'Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm', *Journal of Intelligent & Fuzzy Systems*, Vol. 42, No. 1, pp. 411–423, January 2022.
- [6] X. Chen, L. Cheng, C. Liu, et al., 'A WOA-Based Optimization Approach for Task Scheduling in Cloud Computing Systems', *IEEE Systems Journal*. Vol. 14, Issue 3, pp. 3117–3128, January 2020.
- [7] S. Nabi, M. Ahmad, M. Ibrahim, et al., 'AdPSO: adaptive PSO-based task scheduling approach for cloud computing', *Sensors*, Vol. 22, No. 3, pp. 920–941, January 2022.
- [8] L. Abualigah, M. Alkhrabsheh, 'Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing', *The Journal of Supercomputing*, Vol. 78, No. 1, pp. 740–765, January 2022.
- [9] S. Mangalampalli, S.K. Swain, V.K. Mangalampalli, 'Multi Objective Task Scheduling in Cloud Computing Using Cat Swarm Optimization Algorithm', *Arabian Journal for Science and Engineering*, Vol. 47, No. 2, pp. 1821–1830, September 2022.
- [10] M. Abd Elaziz, I. Attiya, 'An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing', *Artificial Intelligence Review*, Vol. 54, No. 5, pp. 3599–3637, June 2021.
- [11] S. Velliangiri, P. Karthikeyan, V.M.A. Xavier, et al., 'Hybrid electro search with genetic algorithm for task scheduling in cloud computing', *Ain Shams Engineering Journal*, Vol. 12, No. 1, pp. 631–639, March 2021.
- [12] L. Abualigah, A. Diabat, 'A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments', *Cluster Computing*, Vol. 24, No. 1, pp. 205–223, March 2021.
- [13] N. Manikandan, N. Gobalakrishnan, K. Pradeep, 'Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment', *Computer Communications*, Vol. 187, pp. 35–44, April 2022.

- [14] J.H. Ju, W.Z. Bao, Z.Y. Wang, et al., 'Research for the task scheduling algorithm optimization based on hybrid PSO and ACO for cloud computing', *International Journal of Grid and Distributed Computing*, Vol. 7, No. 5, pp. 87–96, May 2014.
- [15] M. Agarwal, G.M.S. Srivastava, 'Genetic algorithm-enabled particle swarm optimization (PSOGA)-based task scheduling in cloud computing environment', *International Journal of Information Technology & Decision Making*, Vol. 17, No. 4, pp. 1237–1267, May 2018.
- [16] F. Hemasian-Etefagh, F. Safi-Esfahani, 'Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing', *The Journal of Supercomputing*, Vol. 75, No. 10, pp. 6386-6450, October 2019.
- [17] G. Păun, 'A quick introduction to membrane computing', *The Journal of Logic and Algebraic Programming*, Vol. 79, No. 6, pp. 291–294, August 2010.

Biographies



Kun Li received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. He is currently a Lecturer Henan Medical College, Zhengzhou, China. His research interests include cloud computing and algorithm design.



Liwei Jia received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. He is currently a Lecturer Henan Medical College, Zhengzhou, China. His research interests include cloud computing and algorithm design.



Xiaoming Shi received the B.S. degree in computer science and technology from Zhengzhou University Of Light Industry, Zhengzhou, China, in 2005 and received the M.S. degree in computer software and theory from Zhengzhou University, Zhengzhou, China, in 2008. His research interests include algorithm design and multi-agent system.