Journal of Theoretical and Applied Information Technology
30th September 2022. Vol.100. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN IMPROVED APPROACH FOR SCHEDULING IN CLOUD USING GA AND PSO

SHYAM SUNDER PABBOJU[1], DR. T. ADILAKSHMI [2]

[1]Assistant Professor, Department of CSE, Mahatma Gandhi Institute of Technology (A), Telangana, India

[2] Professor, Department of CSE, Vasavi College of Engineering (A), Telangana, India

E-mail: [1] pshyamsunder_cse@mgit.ac.in, [2] t_adilakshmi@staff.vce.ac.in

ABSTRACT

In order to improve the efficiency of cloud com- puting task scheduling, the Improved Genetic Algorithm (IGA) and the Improved Particle Swarm Optimization (PSO) are integrated into the IGA-IPSO algorithm for cloud computing task scheduling. The fitness function is constructed by integrating the three objectives of task completion time, task execution cost and virtual machine load balancing to find the optimal solution of task scheduling. The particle swarm algorithm is improved and the dynamic inertia weight strategy is used to improve the adaptive search of the algorithm. In the early stage of task scheduling, the IGA algorithm is used to reduce the solution space, and the Improved PSO is used in the later stage of task scheduling to quickly converge to the optimal solution. Simulation experiments show that compared with the other algorithms, this algorithm has faster convergence speed and stronger optimization ability. In cloud computing task scheduling, it can not only reduce task completion time and execution cost, but also optimize virtual machines load.

Keywords*: Cloud, Scheduling, Genetic Algorithm, PSO, Completion Time, Makespan*.

1. INTRODUCTION

The scale of cloud computing infrastructure is much larger than that of a single physical device, but the virtual resources it provides are still limited. In the face of huge cloud tasks, how to efficiently allocate subtasks to virtual resources and perform reasonable task scheduling has become an issue. To address this many attempts took place [1-4]. Various previous efforts to solve the task scheduling problem in cloud computing have mainly focused on the completion time [5-6]. With the popularity of the pay-as-you-go model, users will also focus on execution costs. For cloud service providers, load balancing of virtual machines also needs to be considered. Although several classical scheduling algorithms are simple to implement, they have obvious shortcomings, such as the Min-Min algorithm [7], the Max-Min algorithm [8] and the First In First Out (FIFO) algorithm [9].The first two algorithms fail to utilize resources efficiently, which easily leads to the problem of load imbalance.

The FIFO algorithm arranges resources in the order of task submission. If the tasks submitted earlier take up a lot of computing resources, the smaller tasks later must wait for a long time. Many researchers use heuristic algorithms to solve the problem of cloud computing task scheduling. Guol et at., [10] proposed a cloud task scheduling scheme based on particle swarm algorithm (PSO) to reduce the total execution time and task transmission time, and proved that PSO runs faster than Genetic Algorithm, effectively reducing the task However, the PSO may fall into a local optimal solution, which may easily lead to a decrease in the solution accuracy, thereby increasing the completion time and execution cost of the task. Santos E et al., [11] uses an improved ant colony algorithm for cloud computing resource scheduling, and finds a more accurate solution through the accumulation and update of pheromone. However, due to the lack of pheromone in the initial solution, the solution rate is slow, which may occupy a large number of users. time, can not well meet the user's service quality requirements.

Ding S et al., [12] proposed an adaptive parameter Genetic Algorithm based on the earliest completion time and PSO. The parameters of the crossover probability are adapted according to the current evolution state to promote evolution and find a better solution. The disadvantage is that it only considers Minimize the completion time, the optimization goal is too single.

Zhou et al., [13] integrates Genetic Algorithm and ant colony algorithm for cloud computing task scheduling, which improves the efficiency of cloud computing resource schedul- ing, but there are many parameters, and the programming implementation is more complicated, and the load balancing of the cloud system is not considered, and the task is too large. Or too much, it is easy to cause abnormal operation of the cloud system. In order to make up for the shortcomings of traditional algorithms for computing task scheduling, this paper proposes a hybrid Improved Genetic Algorithm – Improved PSO (IGA- IPSO) task scheduling algorithms for cloud computing, consid- ering three objectives of task completion time, execution cost, and cloud system load balancing. The algorithm combines the advantages of Improved Genetic Algorithm (IGA) and IPSO, and uses it in the early stage of task scheduling. The IGA is used to reduce the solution space to improve the optimization speed. In the later stage of task scheduling, the PSO with dynamic inertia weight is used to further search and quickly converge to the optimal solution. The experimental results show that the IGA-IPSO algorithm can effectively reduce the completion time, reduce the execution cost and balance the load of virtual machines in the cloud data center. The main objective of this work is to improve completion time, reduce execution cost and achieve load balancing.

2. MULTI OBJECTIVE TASK SCHEDULING IN CLOUD COMPUTING

Cloud computing uses parallelization technology to process a large number of tasks, and uses virtualization technology to establish a certain number of virtual machines in the data center. According to user needs, subtasks are allocated to appropriate virtual machines for execution. Usually, the Map/Reduce programming model is used. Through the two stages of Map and Reduce, a large task is divided into many smaller sub-tasks, and then assigned to several virtual machines for parallel execution, and finally returned operation result. This model has good scalability and fault tolerance. When a machine goes down, it can quickly transfer tasks to another node for running. This paper only considers the case where subtasks are independent of each other, and the whole task scheduling framework can be simplified as Figure 1.

Suppose there are t mutually independent tasks, such as: A ={α1, α2, …,αp, } where αi (0 ⩽ i ⩽ t) represents the ith task, and the size of each task is SIZEi. It consists of p high-speed interconnected virtual machines, such as: B = {β1, β 2, …, βp,} where βj(0 ⩽ ⩽ p) represents

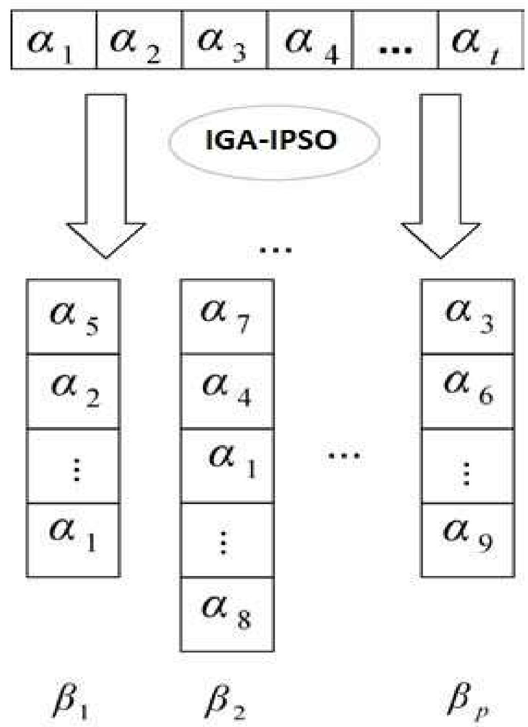

Fig. 1. Task scheduling framework

the jth virtual machine, and each virtual machine has a certain processing capacity MIPSj , using the IGA-IPSO algorithm to schedule tasks to p virtual machines for non-preemptive execution (t p).

The calculation formula of the execution time of task i on virtual machine j is as follows:

$$T_{ij} = SIZE_i/MIPS_j \quad (1)$$

The release time RT_j of the virtual machine j is initialized to 0, and its update formula is as follows:

$$RT_j = RT_j + T_{ij} \quad (2)$$

When task i is the first task on virtual machine j, its

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

start execution time is defined as 0, otherwise it is defined as the release time RT_j of virtual machine j.

$$ST_{ij} = RT_j \tag{3}$$

The completion time of task i is the sum of the start execution time and execution time of task i on virtual machine j, and its formula is as follows:

$$FT_j = ST_{ij} + T_{ij} \tag{4}$$

The maximum completion time of a task is the maximum release time of all virtual machines, and its update formula is as follows:

$$\text{Makespan} = \max \{RT_j \mid \forall \beta_j \in B\} \tag{5}$$

The load balancing of the cloud environment is defined as the standard deviation of the release time of all virtual machines, and its formula is as follows:

$$LB = \sqrt{\frac{1}{P}\sum_{j=1}^{P}\left(u - RT_j\right)^2} \tag{6}$$

In the formula $u = \frac{1}{p}\sum_{k=1}^{p} RT_k$i is the average value of the release time of all virtual machines. The more balanced the use of virtual machines in the cloud system, the smaller the standard deviation of the virtual machine release time, andthe smaller the value of cloud system load balance.

Set a certain unit price for the memory, bandwidth, proces- sor and storage space of the cloud data center, and calculate theexecution cost by counting the usage of each resource during the task scheduling process.

The scheduling process of cloud tasks is a multi-objective optimization problem, so this paper defines the fitness functionas the weighted sum of task completion time, execution cost and cloud environment load balancing, which is used to measure the performance of cloud computing task scheduling.The formula is as follows:

$$\text{fitness} = \lambda_1 \times \text{Makespan} + \lambda_2 \times LB + \lambda_3 \times \text{Cost} \tag{7}$$

In the formula: $\lambda_1 + \lambda_2 + \lambda_3 = 1$, the value of each weight can be flexibly adjusted according to the task requirements.The ultimate goal of task scheduling is to minimize the valueof the fitness function.

3. HYBRID IGA – IPSO ALGORITHM

3.1 Improved Genetic Algorithm

Professor John Holland proposed the Genetic Algorithm (GA) based on the population evolution mechanism in 1975. The algorithm proposes a process of natural selection, which will produce better solutions with the passage of time, and has the characteristics of parallel search and group optimization , which is suitable for solving problems in large spaces.

GA has the disadvantage of not being able to retain ”good parents”. No matter how good the parent’s chromosomes are, it will not be retained. Only crossover mutation operations can be performed to generate new individuals. But the new individuals are not necessarily better than the parents, which may easily lead to insufficient solution accuracy. In order to overcome this shortcoming, referring to the idea of Genetic Algorithm [14], in each iteration process of the population, two chromosomes are randomly selected as parent chromosomes and their fitness is compared, and the parent chromosome with good fitness does not do any processing, it only per- form improved crossover and mutation operations on parent chromosomes with poor fitness. If the fitness of the mutated chromosome is better than that of its parent chromosome, it and the parent chromosome with good fitness are placed on the new population as the next generation, otherwise, the original population is retained. In this way, the parent chromosomes with good fitness in the population can be effectively retained, which greatly improves the accuracy of the solution.

The GA process is shown in Figure 2. First, the population is initialized by randomly creating chromosomes. Secondly, a selection operation is performed according to the fitness value, and some chromosomes are selected to create a new popula- tion. Then, two parent chromosomes are randomly selected for improved crossover operation. Randomly intercept a gene crossing point i on the parent chromosome, and replace the 1st to i gene segments with the corresponding positions of the parent chromosome with poor fitness. The new gene replaces the original gene. Finally, the fitness of the chromosome is calculated. If the fitness of the mutated child chromosome is better, it will replace the parent chromosome with poor fitness and put it

Journal of Theoretical and Applied Information Technology
30th September 2022. Vol.100. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

on the new population, otherwise, the two parent chromosomes still placed on the new species. The whole process continues until the termination condition is met, and the chromosome with the best fitness is selected as the optimal solution.

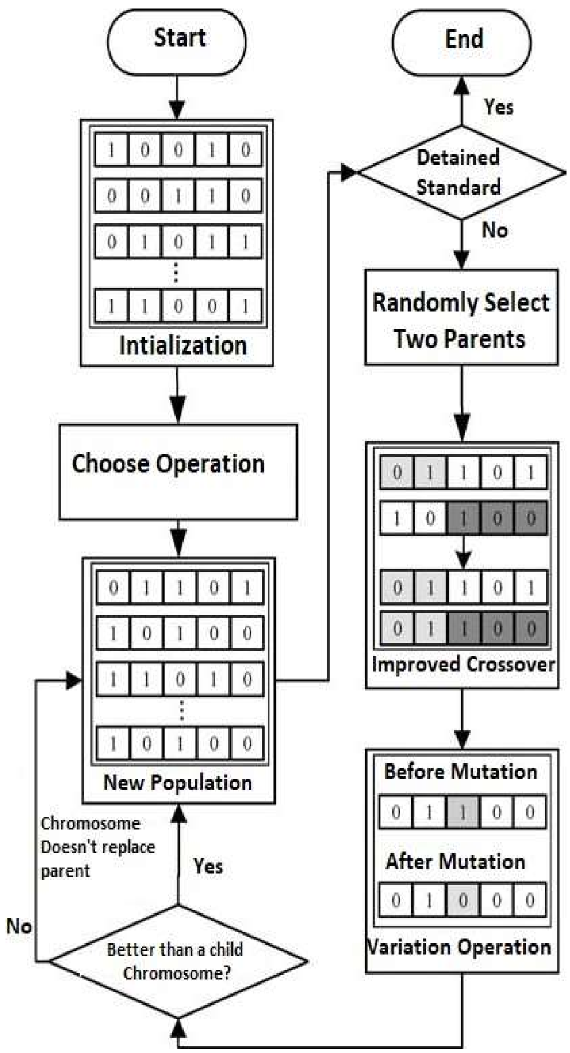

Fig. 2. Improved GA Process

3.2 Improved Particle Swarm Optimization

3.2.1 Standard Particle Swarm Optimization: Particle swarm optimization (PSO) [15] is an optimization algorithm based on swarm intelligence. The particles in the algorithm are similar to the flocks of birds flying in the process of searching for food. The information of each flock is shared, and they always search for the bird closest to the food. In the surrounding area, search for food based on flying experience. The design idea of PSO is based on this information sharing mechanism, and the position of each particle at any time is affected by the individual best position and the global best position in the problem space. The performance of the particles in PSO can be measured by the fitness function, and the particles are optimized in each iteration according to the fitness value to judge the quality of the particles. In order to describe the particle swarm algorithm, the relevant parameters are defined in Table 1. The update formulas of particle velocity and position in the algorithm are as follows:

Table 1: Definition Of Related Parameters Of Particle Swarm Optimization

Parameter	Definition
v_i^k	the velocity of particle i at k iterations
v_i^{k+1}	The velocity of particle i at iteration k+1
ω	inertia weight
c_j	acceleration coefficient; j=1,2
$rand_i$	random number between 0 and 1; i=1,2
x_i^k	The current position of particle i at iteration k times
x_i^{k+1}	The position of particle i at iteration k+1
$pbest_i$	The best position of particle i
gbest	The global best position of the particle in the population

$$v_i^{k+1} = \omega \times v_i^k + c_1 \times rand1 \times \left(P^{best} i - x_i^k\right) + c_2 \times \text{rand2 x} \left(g^{best} i - x_i^k\right) \quad (8)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (9)$$

The right side of Equation (8) can be divided into three parts: 1) "Inertia", that is, the particle's speed experience in the previous iteration; 2) "Self-awareness", which represents the distance between the particle's current position and its own optimal position; 3.) "swarm experience", which represents the distance between the particle's current position and the overall best position of the population. PSO does not have the crossover and mutation operations of GA, and needs to set fewer parameters. The programming is simple and the convergence speed is fast, which can be used to speed up the solution speed.

3.2.2 Particle Swarm Optimization with Dynamic Inertial Weights:

The value of inertia weight ω affects the performance of PSO. A larger value of ω is beneficial to the global search and can speed up the solution; a smaller value of ω is beneficial to the local search and can improve the accuracy of the solution. The fixed value of the inertia weight in the standard PSO is not conducive to the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

balance between the global search and the local search during the operation of the algorithm.

In the early stage of PSO operation, the particles are relatively scattered and have good diversity. At this time, a large inertia weight value should be maintained to enhance the global search ability of the algorithm; in the later stage of the algorithm operation, the particles are more and more concentrated. The inertia weight value should be kept small to improve the local search ability of the algorithm. Many algorithms set the inertia weight to decrease linearly with the increase of the number of iterations, which improves the performance of the algorithm to a certain extent, but the effect of the linear decrease strategy is not ideal for dynamic systems. Therefore, a dynamic inertia weight strategy that decreases nonlinearly with the number of iterations is proposed, and its mathematical formula is described as:

$$\omega = 1 - \left(\alpha_1 - \omega \max \times e^{\frac{1-n}{N}} + \alpha_2 \times \omega_{min} \times \lg(n) - \alpha_3 \times \frac{N-n}{N}\right) \tag{10}$$

In the formula: ω_{max} represents the maximum value of the inertia weight; ω_{min} represents the minimum value of the inertia weight. n is the current number of iterations. N is the maximum number of iterations. With a larger weight value, the global search is strengthened to speed up the solution. In the later stage of the algorithm, the weight value is rapidly reduced, and the local search is strengthened, thereby improving the accuracy of the solution.

3.3 Hybrid IGA-IPSO Algorithm

In this paper, Improved GA and Improved PSO are com- bined to form a hybrid IGA-IPSO for cloud computing task scheduling. In the early stage of task scheduling, IGA is used to process the initial population. Because GA needs to train for the long period proposes that only a small number of iterations are needed to narrow the solution range, and there is no need to find a more accurate solution. In the later stage of task scheduling, using the advantages of PSO to quickly converge to the optimal solution, the improved PSO is used to further optimize the solution generated in the previous stage to find the optimal solution.

GA has the advantages of strong global search ability, which can greatly reduce the solution range, improve the accuracy of the solution, and avoid the PSO falling into the local optimal solution when further optimizing the solution. The advantage of PSO to quickly converge to the optimal solution reduces the time of cloud computing task scheduling and makes up for the shortcoming that GA takes a long time to find an accurate solution.

Before GA-PSO algorithm is used for cloud computing task scheduling, it is necessary to establish the corresponding relationship between the solution of cloud computing task scheduling and the chromosomes and particles in the algo- rithm. Suppose the cloud system has 8 tasks and 3 virtual machines. In the GA stage, each chromosome consists of some genes representing virtual machines. The length of the chromosome is equal to the number of cloud tasks. In this stage, the length of the chromosome can be defined as 8, and the gene type is 3. In the stage of using the improved PSO, the particle is the task to be assigned, the dimension of the particle is the number of tasks to be assigned, and the value of each dimension assigned to the particle is the virtual machine number. In this stage, the dimension of the particle can be defined as 8. The value of the dimension can only take 0, 1 or 2.

Figure 3 shows this correspondence, which can represent both a chromosome and a particle. When representing a chromosome, the value of the 0th gene is 1, which means that task 0 is assigned to virtual machine 1; the value of the third gene is 2, which means that task 3 is assigned to virtual machine 2. When represented as a particle, the value of the first dimension is 0. which means that task 1 is assigned to virtual machine 0; the value of the seventh dimension is 2,which means that task 7 is assigned to virtual machine 2.

1	0	2	2	1	0	1	2

Fig. 3. Chromosome

The process of IGA-IPSO algorithm for cloud computing task scheduling is shown in Figure 4.

The execution steps are summarized as follows.

1) Generate a random population and specify the number of iterations. The population represents a series of solutions for task scheduling, and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

each solution is the distribution of cloud tasks on available virtual machines.

2) Use IGA to narrow the solution range. The solution of cloud computing task scheduling is called chromosomes here. Through IGA operators (ie selection, crossover, mutation), chromosomes with better fitness are retained in each iteration, and the obtained Chromosomes are passed to the modified PSO.
3) The solution is further optimized using the modified PSO, the chromosomes from the IGA are called par- ticles, and the particles are gradually enhanced by Eqs. (8)-(10) in each iteration
4) According to formula (7), select the particle with the best fitness as the solution of cloud computing task scheduling.

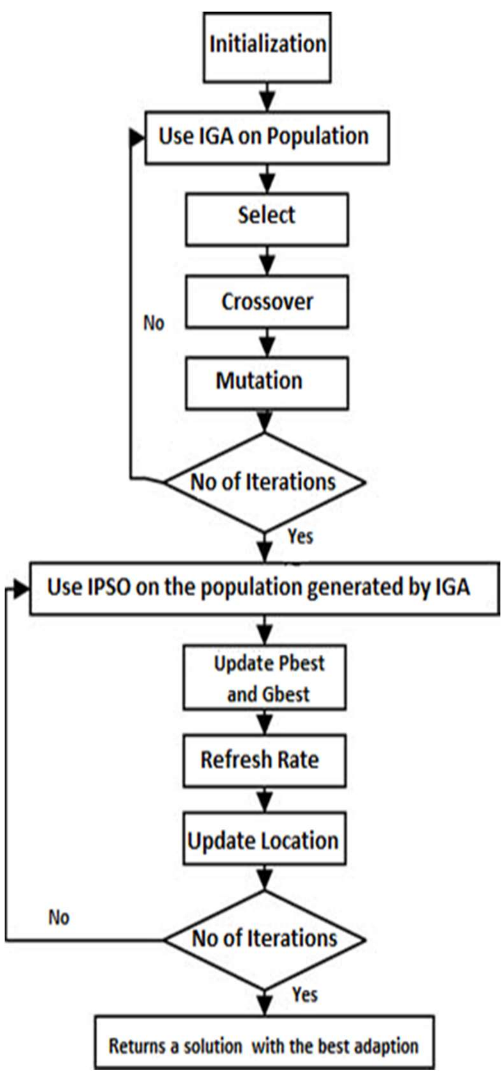

Fig. 4. IGA-IPSO Algorithm flow

Table 2: Parameters of the algorithm

Parameter	Value	Description
P	1000	Population size
M	50	GA iterations
N	100	Improved PSO iterations
crossover	1	single point crossover
mutationRate	0.015	Variation rate
ωmax	0.95	Maximum value of inertia weight
ωmin	0.3	Minimum value of inertia weight
α_1	0.13	Control factor
α_2	0.1	Control factor
α_3	0.2	Control factor
c_1	0.8	acceleration coefficient
c_2	0.8	acceleration coefficient
rand	(0,1)	random number from 0 to 1
α_1, α_2	0.2	Makeup time and load balancing weights
$\alpha {3}	0.6	Weight of execution cost

4. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

4.1 Experimental environment and parameter settings

In order to evaluate the performance of the IGA-IPSO algorithm, the CloudSim [16] cloud simulation platform de- veloped by the University of Melbourne was used. CloudSim supports the research and development of cloud computing, and algorithms written by users can be tested and run on this platform, reducing the time and cost of building a cloud platform. The simulation experiment in this paper uses the IGA-IPSO algorithm for cloud computing task scheduling and compares it with the existing cloud task scheduling algorithms, which are IGA and IPSO.

In the experiment, the three algorithms were run 22 times respectively, the maximum and minimum values of the experi- mental results were removed, and the average of the remaining 20 experimental results was taken as the effective comparison data. The relevant parameter settings in the algorithm are shown in Table 2.

Journal of Theoretical and Applied Information Technology
30th September 2022. Vol.100. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2 Performance Analysis

Figure 5 shows the optimization effect of the three algo- rithms for cloud computing task scheduling, and the number of tasks is set to 500. It can be seen that the optimization effect of IGA and IPSO for cloud computing task scheduling is general, and they fall into the local optimum prematurely. The fitness of the IGA-IPSO algorithm has not changed for a period of time, and may fall into the local optimum, but with the increase of the number of iterations, they can quickly jump out of the local optimum, and can quickly find the optimal solution. The final fitness of the IGA-IPSO algorithm in this paper is the lowest, the optimization effect is the best, and it has a strong global search ability.

Figures 6 and 7 show the comparison effect of the completion time and execution cost of the three algorithms under small-scale tasks, and the number of tasks is set to 20, 40, 60, 80, and 100.

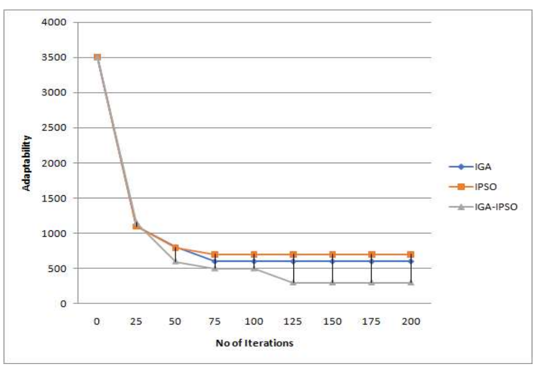

Fig. 5. Optimization curves of algorithms for Task Scheduling in Cloud.

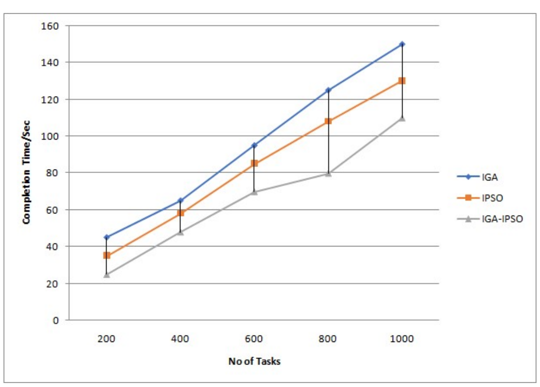

Fig. 6. Completion time of algorithms for small-scale tasks

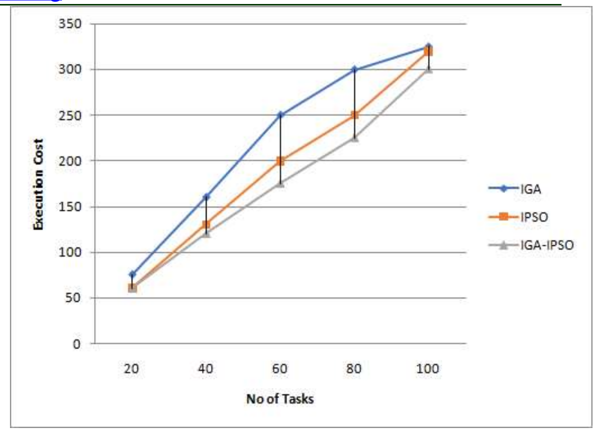

Fig. 7. Execution costs of algorithms for small-scale tasks

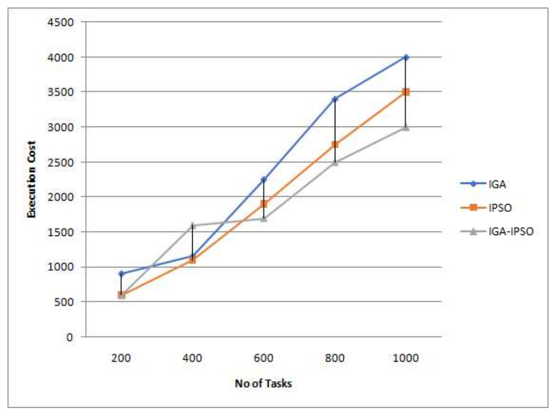

Fig. 8. Completion time of algorithms for large-scale tasks

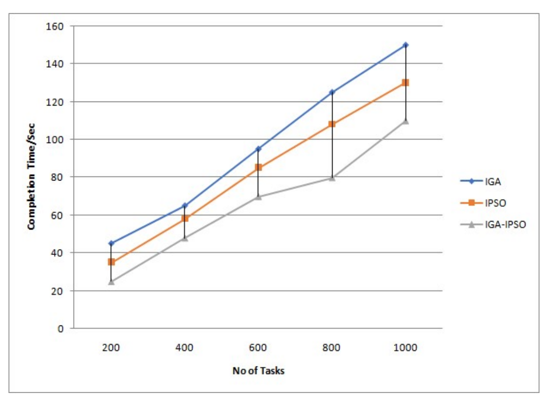

Fig. 9. Execution costs of algorithms for large-scale tasks

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

It can be seen from Figure 6 that IGA and IPSO will spend more time for cloud computing task scheduling. When the number of tasks increases, the completion time of IGA increases the fastest. Overall, the completion time of the IGA-IPSO algorithm is lower than that of other algorithms. This result is because the proposed IGA is iterative. In the process, the parent chromosome with good fitness is retained, the accuracy of understanding is improved, and the solution can be converged in a better way. It can be seen from Figure 7 that the execution cost of IGA is the highest, which may be due to the ”premature” convergence; the execution cost of IGA- IPSO algorithm for small-scale task scheduling is less. The difference is because the proposed hybrid algorithm mainly relies on PSO to converge the solution to the optimal solution, and the smaller the task scale, the smaller the difference.

Figures 8 and 9 show the comparison effect of the completion time and execution cost of the three algorithms under large-scale cloud tasks, and the number of tasks is set to 200, 400, 600, 800, and 1000. It can be seen from Figure 8 that when scheduling large-scale cloud tasks, the IGA- IPSO algorithm have great advantages in terms of completion time. As the task scale increases, the completion time of the IGA-IPSO algorithm is significantly lower than that of others. This is because the scale of the task becomes larger, and the initial population generated is more random. In the early stage of the IGA-IPSO algorithm, more excellent parent chromosomes can be retained, which greatly improves the accuracy of understanding. It can be seen from Figure 9 that in terms of execution cost, when the task is scheduled, the larger the task scale, the more obvious the advantage of the IGA- IPSO algorithm. This is because the improvement of PSO with dynamic inertia weight effectively balances the global search ability and local search ability of the algorithm, and can find a more suitable solution.

Figure 10 shows the comparison effect of virtual machine load balancing when the three algorithms are used for cloud computing task scheduling, and the number of tasks is set to 20, 80, 200, and 1000. It can be seen that the load balancing effect of the IGA-IPSO algorithm is much better than that of the IGA and IPSO.

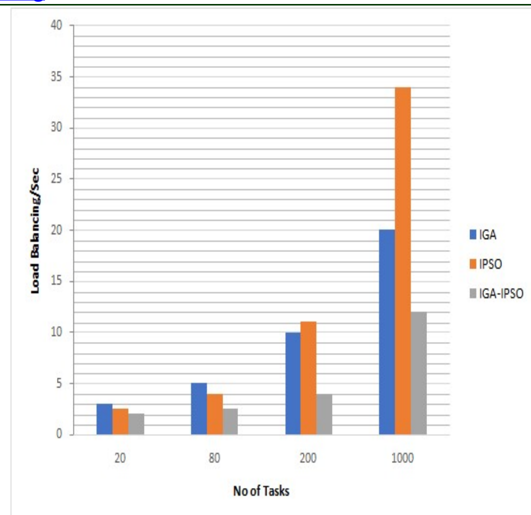

Figure 10 Shows The Comparison Effect Of Virtual Machine

. This is because the IGA-IPSO algorithm uses the three goals of task completion time, execution cost, and cloud system load balancing as the basis for the selection of virtual machines during task execution. Always choose the most suitable virtual machine to perform the task, rather than just focusing on the processing power of the virtual machine, this may select a virtual machine with less processing power to handle the cloud task and slow down the overall execution speed of the cloud task (i.e. increase the completion time), but there will be improvements in other areas, such as a more balanced virtual machine load and lower execution costs. The value of each weight in the fitness function can be adjusted flexibly to meet the needs of different aspects.

The scheduling performance of the IGA-IPSO algorithm is better than the GA-PSO algorithm of the same idea in terms of completion time, execution cost and virtual machine load balancing effect. It can be seen that the improvement of GA and PSO has improved the performance of cloud computing task scheduling. The proposed algorithm is tested for only small scale and large scale tasks and could not consider medium scale tasks. Further this work is limited to only task completion and cost.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION

In view of the advantages and disadvantages of IGA and IPSO, this paper combines IGA and Improved PSO into IGA- IPSO algorithm for cloud computing task scheduling. The algorithm fully considers the three goals of completion time, execution cost and cloud system load balancing, which reflect the user's service quality, and establishes a fitness function based on these goals as the basis for the selection of virtual machines when the algorithm runs. The simulation results show that the IGA-IPSO algorithm is superior to the IGA and IPSO algorithms in terms of completion time, execution cost and load balancing, which can meet the service quality requirements of users and effectively improve the efficiency of cloud computing task scheduling.

REFERENCES

[1] P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," in IEEE Transactions on Automation Sci- ence and Engineering, vol. 15, no. 2, pp. 772-783, April 2018, doi: 10.1109/TASE.2017.2693688.

[2] Zhi-Hui Zhan, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu- Hung Chung, Yun Li, "Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches", ACM Computing Surveys Volume 47, Issue 4, Article No.:63, pp 1–33, 2015.

[3] Yang Zhaofeng, Wang Qiming, Lu Hailian, "Cloud computing resource scheduling based on task delay degree algorithm", Computer Measure- ment and Control, 22(2),pp. 488-502, 2014

[4] Zhou Lijuan, Wang Chunying, "Cloud computing resource scheduling based on particle swarm optimization algorithm", Journal of Computer Science, 42(6), pp.279-281, 2015.

[5] Shi Feng,"Research and simulation of resource scheduling efficiency optimization in cloud environment", Machine Simulation,33(5), pp.363- 366, 2016

[6] Yuan Zhengwu, Li Junqi, "Cloud resource scheduling based on improved particle swarm algorithm", Journal of Computer Engineering and De- sign, 37(2), pp.401-412, 2016

[7] S. K. Panda and P. K. Jana, "An efficient task scheduling algorithm for heterogeneous multi-cloud environment," 2014 International Con- ference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1204-1209, doi: 10.1109/ICACCI.2014.6968253, 2014.

[8] S. Devipriya and C. Ramesh, "Improved Max-min heuristic model for task scheduling in cloud," 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 883-888, doi: 10.1109/ICGCE.2013.6823559, 2013.

[9] Santos-Neto, E., Cirne, W., Brasileiro, F., Lima, A., "Exploiting Replica- tion and Data Reuse to Efficiently Schedule Data-Intensive Applications on Grids", Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science, vol 3277. Springer, Berlin, Heidelberg, 2005 https://doi.org/10.1007/11407522 12

[10] Guo, L., Zhao, S., Shen, S., Jiang, C., "Task scheduling optimization in cloud computing based on heuristic algorithm", Journal of networks, 7(3), pp.547, 2012.

[11] Wei, Chen Yuanyuan, "Cloud computing task scheduling model based on improved ant colony algorithm", Computer Engineering, 41(2), pp.12- 16, 2015.

[12] S. Ding, J. Wu, G. Xie and G. Zeng, "A Hybrid Heuristic-Genetic Al- gorithm with Adaptive Parameters for Static Task Scheduling in Hetero- geneous Computing System," 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 761-766, 2017.

[13] Zhao Junpu, Ban Jinyong, Jin Tongbiao, "Genetic Ant Colony Algorithm in Cloud Computing Resources Application in scheduling", Computer Engineering and Design, 38(3), pp. 693-697, 2017

[14] Inman Harvey, "The microbial genetic algorithm", Proceedings of the 10th European conference on Advances in artificial life: Darwin meets von Neumann - Volume Part II, Pages 126–133, 2009.

[15] Kennedy, J., Eberhart, R., "Particle swarm optimization", In Proceedings of ICNN'95-international conference on neural networks , Vol. 4, pp. 1942-1948, 1995

[16] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya,R. , "CloudSim: a toolkit for modeling and simulation of cloud com- puting environments and evaluation of resource provisioning algorithms.Software: Practice and experience, 41(1), 23-50, 2011

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[17] Ertenlice, O.; Kalayci, C.B., "A survey of swarm intelligence for port- folio optimization: Algorithms and applications", Swarm Evol. Comput, 39, pp.36–52, 2018.

[18] Y. Song, F. Wang and X. Chen, "An improved genetic algorithm for numerical function optimization", Applied Intelligence, 49(5), pp. 1880–1902, 2019.

[19] Ibrahim, M.; Nabi, S.; Baz, A.; Naveed, N.; Alhakami, H, "Towards a task and resource aware task scheduling in cloud computing: An exper- imental comparative evaluation", Int. J. Networked Distrib. Comput. 8, pp.131–138, 2020.

[20] Ibrahim, M.; Nabi, S.; Baz, A.; Alhakami, H.; Raza, M.S.; Hussain, A., "Djemame, K. An in-depth empirical investigation of state-of- the-art scheduling approaches for cloud computing", IEEE Access, 8, pp.128282–128294, 2020.

[21] Mubeen, A.; Ibrahim, M.; Bibi, N.; Baz, M.; Hamam, H.; Cheikhrouhou,O. Alts, "An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing" Processes 2021, 9, 1514, 2021.

[22] Agrawal, A.; Tripathi, S. , "Particle swarm optimization with adaptive inertia weight based on cumulative binomial probability", Evol. Intell. 2021, 14, pp.305–313, 2021.