Figure 1: Caption

Research Article

Cloud Computing Task Scheduling Model Based on Improved Whale Optimization Algorithm

LiWei Jia , Kun Li, and Xiaoming Shi

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

Henan 451191, China

Correspondence should be addressed to LiWei Jia; zzujialiwei@126.com

Received 30 May 2021; Revised 15 September 2021; Accepted 26 November 2021; Published 13 December 2021 Academic Editor: Omprakash Kaiwartya

The efficiency of task scheduling under cloud computing is related to the effectiveness of users. Aiming at the problems of long scheduling time, high cost consumption, and large virtual machine load in cloud computing task scheduling, an improved scheduling efficiency algorithm (called the improved whale optimization algorithm, referred to as IWC) is proposed. Firstly, a cloud computing task scheduling and distribution model with time, cost, and virtual machines as the main factors is constructed. Secondly, a feasible plan for each whale individual corresponding to cloud computing task scheduling is to find the best whale individual, which is the best feasible plan; in order to better find the optimal individual, we use the inertial weight strategy for the whale optimization algorithm to improve the local search ability and effectively prevent the algorithm from reaching premature convergence; we use the add operator and delete operator to screen individuals after each iteration which is completed and updated to improve the quality of understanding. In the simulation experiment, IWC was compared with the ant colony algorithm, particle swarm algorithm, and whale optimization algorithm under a different number of tasks. The results showed that the IWC algorithm has good results in terms of task scheduling time, scheduling cost, and virtual machine. The application is in

cloud computing task scheduling

1 Introduction

The allocation of resources between users and enterprises is currently a concern of people from all walks of life. Task scheduling is a key technology in cloud computing to control resources and improve system stability, which directly affects user experience. So far, the design of many task scheduling algorithms has become a hot topic. Effective combination in the existing task scheduling algorithm can save the task completion time, meet the user's service quality requirements, and improve the load balance of the system. Cloud computing follows the principle of on-demand allocation [1, 2

, by establishing a huge resource pool, and then selecting the most appropriate resource for the user according to the user's needs. It uses virtualization technology to centralize various resources and then uses specialized software to automatically manage the resources, so that users do not have to worry about other problems besides tasks. In this mode, the relationship between user task processing time and cost will be inseparable. Cloud computing often has to deal with a huge number of tasks, and resource scheduling has become a core issue. In scheduling, issues like cost, load balancing, and service quality are all unavoidable factors [3]. The whale optimization algorithm (WOA) is one of the relatively new metaheuristic algorithms. It can provide good global search capabilities and can be widely used in various engineering problems. In this article, we try to use the whale optimization algorithm to solve the task scheduling in cloud computing. The experimental results show that the algorithm has a better scheduling effect under different number of tasks. Literature [4] merges Min–Min and Max–Min into a genetic algorithm and uses this algorithm for cloud computing task scheduling. In cloud computing scheduling, a given task optimizes the task execution time, load, and cost price of cloud computing; maps the task scheduling scheme to the whale algorithm model; and obtains the optimal solution by using WOA. We propose an advanced method called IWC (improved whale optimization algorithm), which is mainly used to improve the search ability of the best solution of the WOA algorithm. The contributions of this paper are as follows: (1) in order to improve the efficiency of cloud computing task scheduling, a multiobjective optimization model for task scheduling is proposed, and WOA is used to solve the entire problem; (2) an IWC algorithm is proposed, which improves the convergence and accuracy of the WOA-based method which improves the efficiency of task scheduling; (3) describes the implementation process of the IWC algorithm and compares it with the ACO, PSO, and WOA algorithms. The experimental results show that the algorithm works under different task quantity conditions. Down has a better scheduling effect. The rest of this article is organized as follows. The second section introduces the related work. The third section describes our scheduling system model. The fourth section introduces IWC; the fifth section proposes the implementation details of the improved IWC; in the sixth section, we simulated the algorithm and explained the scheduling effect; and the seventh section is the end of this article.

2 Related Work

For task scheduling under cloud computing, many scholars first established cloud computing task scheduling models from different perspectives and then used metaheuristic algorithms to solve the scheduling models and achieved good scheduling results. Due to space limitations, this article only elaborates on commonly used metaheuristics. Literature [5] uses the Genetic Algorithm to optimize task scheduling with energy consumption as the main goal; Literature [4] merges Min–Min and Max–Min into the Genetic Algorithm for the cloud computing task scheduling. The above results show that the use of the Genetic Algorithm in cloud computing tasks can reduce task completion time, reduce energy consumption, and improve resource utilization. Literature [6] used Ant Colony Optimization to handle the node load in the cloud in order to bring users a better user experience; Literature [7] used Ant Colony Optimization in the green cloud computing. The above results show that cloud computing effectively reduces the completion time, improves the efficiency of the node load, and reduces operating costs. Literature [8] used Particle Swarm Optimization based on two different inertial weights in cloud computing task scheduling to reduce task completion time; Saleh et al. [9] used the improved Particle Swarm Optimization for cloud computing task scheduling with the goal of average task length and scheduling success rate. Literature [10] proposed a binary-based Artificial Bee Colony for grid computing; Literature [11] used the Artificial Bee Colony to handle the allocation of energy-aware resources under cloud computing. The above results show that using Artificial Bee Colony can effectively reduce energy consumption and save user costs. Literature [12] proposed the use of a hybrid Shuffled Frog Leaping Algorithm for resource and workflow scheduling in cloud computing; Literature [13] proposed a cloud computing task scheduling algorithm based on ACO and PSO. Experiments show that this algorithm can help improve the efficiency of cloud computing scheduling. Literature [14] proposed a cloud computing task scheduling algorithm based on a mixture of ACO and WOA. Simulation experiments show that this algorithm is indeed better than ACO and WOA in cloud computing scheduling. Literature [15] proposed a QoS-aware scheduling algorithm (QoSDPSO); experimental results show that QoS-DPSO can effectively improve the performance and obtain the high reliability; Literature [16] proposed a cloud computing task scheduling algorithm based on game theory. Simulation experiments show that the algorithm has better performance. Literature [17, 18] proposed the Huffman coding method and the whale optimization algorithm used in wireless sensor networks, which provide a new idea for cloud computing resource scheduling.

3 . Cloud Computing Task Scheduling Model

In cloud computing, the task scheduling strategy will directly affect the resource utilization efficiency of the underlying system. Therefore, how to allocate tasks has become a key issue in cloud computing scheduling. This article mainly focuses on the performance of different scheduling algorithms. Therefore, assuming that all tasks submitted by users are logically independent of each other, the task scheduling process in the cloud environment can be summarized as the following three steps. First, we input the detailed information of the task and the available computing resources. Secondly, the tasks and resources will be mapped according to certain strategies, and the operation will be performed according to the mapping. The task plan of the control layer will generate an optimized task execution plan to meet certain assigned requirements (that is, the optimization goal). Finally, the optimized plan is delivered to the underlying task processing layer for execution, and the output result is sent to the user. The advantage of adopting this mode is that it can reduce the calculation delay, reduce the calculation cost, and improve the user experience effect. This article takes load balancing, completion cost, and execution time as the reference basis for users to evaluate cloud computing (Quality of Service, QoS) services [19]. Let the task set be T = fT1, T2,TNg and the resource node collection is N = fN1, N2,NMg, in which N ¿ M. Task scheduling under cloud computing can be represented by the following matrix: A = a11 a12 a1M
a21 a22 a2M
aN1 aN2 aNM
2 6 6 6 6 6
4 3
7 7 7 7 7
5 : 1 In the matrix A, aij is 1, which means the task i is performed on task j, or aij is 0, i $\frac{1}{2}$1, N, j $\frac{1}{2}$1, M. This article defines the attributes of each resource node including processing capacity (that is, processing time), initial memory (that is, the load of response processing), and resource bandwidth (that is, reflecting the cost of processing). Therefore, the three vectors of virtual machine resources are the processing power vector En, load capacity vector Sn, and resource bandwidth vector Cn. At the same time, each task corresponds to three matrices, which are the processing power vector Et, load capacity vector St, and resource bandwidth vector Ct; P are the unit prices. Therefore, the time target function time, load target function load, and cost target function cost are shown as follows. time = N i=1 M j=1 aij Et,i En,j , 2 load = N i=1 M j=1 aij St,i Sn,j , 3 cost = N i=1 M j=1 aij Et,i En,j × Ct,i Cn,j × P: 4 In the formula, Et,i refers to the Et value of the i task, En,j refers to the En value of the j virtual machine, St,i refers to the St value of the i task, Sn,j refers to the Sn value of the j virtual machine, Ct,i refers to the Ct value of the i task, and Cn,j refers to the Cn value of the j virtual machine. For our presentation in the following, we use the notations as listed in Table 1. Since the results represented in the three matrices of resource nodes and task nodes have different standards, they need to be normalized, so the above three objective functions are expressed as follows: Ex-

etime = 1 N N i=1 M j=1 aij Et,i /En,j max i,j Et,i /En,j , 5 Vmload = 1 N N i=1 M j=1 aij St,i /Sn,j max i,j St,i /Sn,j , 6 Execost = 1 N N i=1 M j=1 aij PEt,i Ct,i / En,j Cn,j max i,j PEt,i Ct,i / En,j Cn,j : 7 Therefore, the task function set in this article is F i = 1 × Exetime i + 2 × Vmload i + 3 × Execost i : 8 In the formula, , , and are the weight values of Exetimei, Vmloadi, and Execosti, respectively, and 1 + 2 + 3 = 1. Therefore, min Fi is the optimal scheme for cloud computing task scheduling. 3.1. Whale Optimization Algorithm. In 2016, Mirjalil and Lewis [20] proposed a whale optimization algorithm (WOA) based on the behavior of whales' preying in the sea. In the WOA algorithm, the humpback whale in the search space is a candidate solution in the optimization Table 1: Main notes in the task scheduling model. Symbol Meaning N Number of tasks processed M Number of virtual machines aij Indicates that the i-th task corresponds to the j-th virtual machine En Et VM (task) processing power vector Sn St VM (task) load capacity vector Cn Ct Resource bandwidth vector of VM (task) P Unit price 1, 2, 3 Weighting factor

4 Task Scheduling in Cloud Computing Based on Improved Whale Optimization Algorithm

Compared with other advanced algorithms, the WOA algorithm has the advantages of simple operation and few parameters, but it has the problems of slow convergence, easily falling into the local optimum, and low convergence precision. Therefore, it was improved from two aspects of local search and global search in this dissertation. 4.1. Introduce Inertial Weight to Improve Local Search Ability. In Ref. [21], the inertial weight adopted in the PSO algorithm was beneficial to the local development of the algorithm and accelerated the convergence speed. Therefore, the inertial weight had an important influence on the convergence speed and local optimum. The inertial weight had a great influence on the WOA algorithm. Generally speaking, the WOA algorithm was to introduce the inertial weight, which decreased linearly with the increase of the number of iterations. This method satisfied that the algorithm required a large inertial weight in the early iteration but a small weight in the later period. However, if the global optimal value appeared in the early iteration of the algorithm and the inertial weight could not be efficiently and quickly reduced, then it might affect the tracking speed and accuracy of the algorithm; if it only depended on the inertial weight, of which the number of iterations was linearly decreasing, then it was difficult to effectively jump out of the local optimum for the local convergence of the algorithm. In order to make the inertial weight effectively adjusted, an adaptive inertial weight was proposed, which not only could depend on the change of the iteration number but also needs to consider the influence of the concentration of humpback whales, so as to achieve the purpose of improving the convergence speed and the optimal solution accuracy. In the adaptive WOA algorithm, an iteration number factor and a humpback whale clustering factor are introduced. indicates the relationship between the current iteration and the total number of iterations. The formula for calculating the iteration factor is as follows: = ﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬁﬀﬃﬃﬃﬃﬃﬃﬃ exp t/Q K q , 17 where t indicates the number of current iterations, Q is the maximum number of iterations, and K is a constant greater than 1. Obviously, the curve of this iterative factor is a decreasing function related to t, and a larger weight can be obtained in the early iteration of the algorithm but a smaller weight in the later iteration. However, when the iterative period reaches the optimal value of the problem to be optimized, the inertial weight cannot be effectively reduced. In this paper, is used to adjust the aggregation degree of humpback whales, and the fitness function is used to represent the value of . Although the average cost of the IWC algorithm is lower than the other three algorithms, overall, the difference between the four algorithms is very small. When is relatively larger, it means that a large inertial weight is needed. On the contrary, it means that a smaller weight is needed. Therefore, the calculation formula is as follows: f avg = N k=1 f Pk N , 18 E = 1 N 1 N k=1 f Pk f avg h i2 , 19 = sin art tan E : 20 In (18)–(20), f avg is the average value of the adaptive value, fPk indicates the adaptive value of the humpback whale k, E is the variance of the adaptive function value, and N is the total number of humpback whales. Step 1: generate a reference point R within the search range; Step 2: among the current clustering individuals P, select the point X closest to R; Step 3: in P fXg, find out the points closest to M 1 and X to form a subcluster; Step 4: delete M individuals in P; Step 5: repeat step 2–step 4 until the cluster has been divided into Np M classes. Algorithm 1: Establishment of the S1 set. Step 1: initialization: generate individual points n and the search area ½xmin, xmax; Step 2: generate initial points (r1, r2, rNd), ri = 2 cos 2i/p, 1 i Nd, and p is a minimum prime number meeting p 3/2 Nd; Step 3: generate n points: xk = fk r1g, fk r2g,fk rNd g, k = 1, , n; fg indicates the fractional part. Step 4: map the spawn point to the search domain xk = xmin + xkxmax xmin. Algorithm 2: Establishment of the S2 set.

5 Algorithm Complexity Analysis

Time complexity refers to the computational workload required in the execution of the algorithm, which mainly depends on the number of repeated executions of the problem. In the basic whale optimization algorithm, the time complexity is mainly influenced by the population size N, the number of iterations T, and the search dimension D, and the time complexity of the basic WOA is ON T D . On the basis of the WOA, the complexity of the IWOA proposed in this paper has been increased as follows. The improved inertial weight to improve local search ability the complexity of OT.The addition operators and deletion operators increase the complexity of OT D. Therefore, the total complexity of the IWOA is ON T D + OT + OT D. The overall time complexity is higher than that of the WOA.

6 Experimental Simulation

In order to further verify the task scheduling effect of the algorithm in cloud computing, the algorithm IWC is compared with ACO, PSO, and WOA algorithms. The parameters required by the algorithm are shown in Table 1. Select Table 2: Relevant parameters of comparison algorithm. Algorithm Parameter Value Description ACO 0.0005 Pheromones i 0.01 Pheromone coefficient p 0.5 Path selection probability w 0.5 Inertial weight PSO c1 0.5 Learning factor of particle swarm c2 0.5 Learning factor of particle swarm WOA F 0.5 Follower number rand 0.5 Random number 0.5 Random weight IWC wmax 0.9 Maximum weight wmin 0.2 Minimum weight K 5 Control number of clustering L 1000 IWOA parameter PSmax 10 Upper limit of clustering PSmin 1 Lower limit of clustering Table 3: Test functions. No. Function name Test function F1 Sphere f x = n i=1x2 i F2 Schwefel2.22 f x = n i=1 xi + Yn i=1xi j=1xj

Table 4: Time comparison of 4 algorithms in test functions. Algorithm Dimension F1 F2 F3 F4 F5 F6 F7 ACO 2 0.084 0.086 0.701 0.073 0.246 0.072 0.098 5 0.08 0.1 1.666 0.081 0.441 0.084 0.134 10 0.102 0.121 3.717 0.102 0.853 0.092 0.199 30 0.155 0.182 11.891 0.146 2.219 0.153 0.507 50 0.221 0.257 22.119 0.228 2.802 0.219 0.683 100 0.337 0.397 58.007 0.316 4.189 0.332 1.064 PSO 2 0.046 0.045 0.080 0.041 0.048 0.033 0.043 5 0.095 0.112 0.315 0.093 0.181 0.092 0.114 10 0.186 0.223 0.900 0.188 0.423 0.191 0.261 30 0.595 0.706 6.673 0.602 2.121 0.601 1.201 50 1.054 1.250 17.800 1.054 4.981 1.071 2.712 100 2.463 2.942 76.739 2.473 19.647 2.502 8.952 WOA 2 0.025 0.018 0.0351 0.014 0.017 0.013 0.017 5 0.034 0.041 0.133 0.031 0.072 0.032 0.042 10 0.081 0.092 0.440 0.076 0.189 0.082 0.116 30 0.433 0.478 3.510 0.428 1.159 0.431 0.783 50 1.130 1.191 9.859 1.101 3.124 1.161 2.117 100 4.238 4.461 41.151 4.590 12.967 4.481 8.720 IWC 2 0.034 0.021 0.041 0.016 0.027 0.012 0.012 5 0.033 0.038 0.143 0.031 0.074 0.028 0.029 10 0.074 0.176 0.436 0.075 0.192 0.073 0.084 30 0.451 0.502 3.713 0.434 1.183 0.432 0.481 50 1.176 1.312 10.618 1.112 3.067 1.115 1.263 100 4.211 4.516 41.094 4.263 12.172 4.236 4.741 0 20 40 60 80 100 120 140 160 180 200 0.24 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 Econmic cost Iteration times ACO WOA PSO IWC Figure 2: Comparison of the four algorithms' normalized cost. 8 Wireless Communications and Mobile Computing test functions have both high dimensions (30, 50, 100) and low dimensions (2, 5, 10) that can be compared with the ACO, PSO, WOA in all aspects at time comparison (shown in Table 4). Table 4 show a comparison of the usage times of the 4 algorithms in different dimensions under the 7 test functions. It is found that the usage time of this algorithm is longer than those of ACO, PSO, and WOA in all dimensions. This shows that the algorithm in this paper has good performance. (2) Cloud task scheduling normalization index. Figures 2–4 show the comparison of the four algorithms for normalized cost values, normalized time values, and normalized load values. Figure 2 shows the normalized cost of the four algorithms. The normalized value of the ACO algorithm is much larger than the other three algorithms, and the cost normalized

values between the PSO, WOA, and IWC algorithms are not much different. When the number of tasks is between [0, 65], the normalized result of the IWC algorithm is lower than the WOA algorithm which is higher than the PSO algorithm, and when the task exceeds 65, the normalized result of the IWC algorithm is lower than the PSO and WOA algorithms. From the overall effect, the cost normalization values of the four algorithms tend to be stable. Compared with the ACO algorithm, the PSO algorithm, and the WOA algorithm, the IWC algorithm is reduced by 31.329.570 20 40 60 80 100 120 140 160 180 200 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42 0.43 0.44 Time consumption Iteration times ACO WOA PSO IWC Figure 3: Time value of the four algorithms' normalized cost. 0 20 40 60 80 100 120 140 160 180 200 0.686 0.688 0.690 0.692 0.694 0.696 0.698 0.700 0.702 0.704 Load Iteration times ACO WOA PSO IWC Figure 4: Load value of the four algorithms' normalization. Wireless Communications and Mobile Computing 9 algorithm can effectively reduce the task scheduling cost under cloud computing. Figure 3 shows the normalized time comparison of the four algorithms. The normalized time result of the ACO algorithm is much larger than the other three algorithms. When the number of tasks is between [0, 65], the normalization result of the IWC algorithm is lower than the WOA algorithm which is higher than the PSO algorithm. When the number of tasks exceeds 65, the normalization result of the IWC algorithm is lower than that of the PSO algorithm and the WOA algorithm. From the overall effect, the normalization time of the four algorithms tends to be stable. Compared with the ACO algorithm, the PSO algorithm, and the WOA algorithm, the IWC algorithm is reduced by 18.29Figure 4 shows the normalized load values for the four algorithms. After the number of tasks is greater than 60, the ACO algorithm tends to a fixed value. After the number of tasks is greater than 18, the PSO algorithm tends to a fixed value. After the number of tasks is greater than 20, the algorithm curve tends to a fixed value. The IWC algorithm tends to a fixed value after the number of tasks is greater than 70. Overall, the load value of the IWC algorithm is lower than 15.99compared to the ACO algorithm, PSO algorithm, and WOA algorithm. (3) Comparison of the number of small-scale tasks. Figures 5–7 show the cost, time consumption, and load comparison of the four algorithms for small100 200 300 400 500 600 700 800 900 1000 0.000 0.002 0.004 0.006 0.008 0.010 0.012 0.014 0.016 Econmic cost Number of Task ACO WOA PSO IWC Figure 5: Comparison of the four algorithms' cost in small-scaled task. 100 200 300 400 500 600 700 800 900 1000 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 Time consumption(MS) Number of Task ACO WOA PSO IWC Figure 6: Time comparison of the four algorithms' cost in small-scaled task. 10 Wireless Communications and Mobile Computing scale tasks. The cost consumption of the four algorithms is illustrated in Figure 5. The curves of four algorithms are gradually increasing with the increase in the number of tasks, but the curve of the IWC algorithm is relatively flat, although the average cost is numerically lower than the other three algorithms, but the overall difference between the four algorithms is small. Figure 6 shows the completion time of the four algorithms. It is found from the figure that as the number of tasks increases, the time consumption of the four algorithms becomes larger, but the overall four algorithms are not much different. However, the IWC algorithm has certain advantages in terms of time. Figure 7 shows the comparison of the load values of the four algorithms. From the curve in the figure, the load value curves of the four algorithms are basically consistent. This shows that the four algorithms have similar effects on the load. (4) Comparison of the number of large-scaled tasks. Figures 8–10 show the cost, time consumption, and load comparison of the four algorithms for largescale tasks. Figure 8 shows the cost of the four algorithms. Along with the increasing number of tasks, the curves of the four algorithms are gradually increasing, and the cost of ACO, PSO, and WOA algorithms is not much different. The cost curve of the IWC algorithm is obviously better than that of the other three algorithms. IWC is reduced by 52.94100 200 300 400 500 600 700 800 900 1000 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 Load Number of Task ACO WOA PSO IWC Figure 7: Load comparison of the four algorithms' cost in small-scaled task. 3000 4000 5000 6000 7000 8000 9000 10000 0.02 0.04 0.06 0.08 0.10 0.12 0.14 Econmic cost Number of Task ACO WOA PSO IWC Figure 8: Cost comparison of four algorithms under large-scale tasks. Wireless Communications and Mobile Computing 11 PSO, and WOA, respectively, which shows that the IWC algorithm can effectively reduce the task cost. Figure 9 shows the completion time of the four algorithms. It is found from the figure that as the number of tasks increases, the time consumption of the four algorithms becomes larger, and IWC is better than ACO, PSO, and WOA. And the time spent was reduced by 39.29respectively. This shows that there is a certain advantage in terms of time completion. Figure 10 shows the comparison of the load values of the four algorithms. From the figure, the load values of the four algorithms will be slightly different with the increasing number of tasks, but the overall difference is not large. From the comparison of the above three experiments, the IWC algorithm proposed in this paper has obvious advantages in task cost and completion time and is more suitable for task scheduling under cloud computing. However, the effect on the memory load value is not very obvious, which shows that the IWC algorithm has more room for server memory optimization. 7. Conclusion This article introduces a WOA-based task scheduling method in cloud computing task scheduling, mainly to improve the effect of task scheduling under cloud computing. In order to further improve the scheduling performance 3000 4000 5000 6000 7000 8000 9000 10000 4 6 8 10 12 14 16 18 20 Time consumption Number of Task ACO WOA PSO IWC Figure 9: Time comparison of four algorithms under large-scale tasks. 3000 4000 5000 6000 7000 8000 9000 10000 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 Load Number of Task ACO WOA PSO IWC Figure 10: Load comparison of four algorithms under large-scale tasks. 12 Wireless Communications and Mobile Computing based on the WOA algorithm, on the basis of the WOA algorithm, two optimization strategies of the IWC algorithm are proposed. The experimental results show that compared with some commonly used metaheuristic algorithms, it can be used in system load and system resource utilization. In terms of cost, the efficiency of cloud computing systems is greatly improved. In future work, in order to obtain better convergence speed and accuracy in task scheduling, we will consider proposing more advanced strategies to further improve the balance between exploration and development in the IWC method. At the same time, to reduce the scheduling cost and time of this method in the face of a large number of virtual machine workloads, we will use some more advanced features to extend the proposed performance model and method. Our long-term goal is to develop a cloud computing efficient scheduling system suitable for various task workloads. Data Availability Regarding the data, you can directly Email to me. Conflicts of Interest The authors declare no conflict of interest. References [1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50– 58, 2010. [2] I. M. Ibrahim, "Task scheduling algorithms in cloud computing: a review," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 4, pp. 1041–1053, 2021. [3] M. Cusumano, "Cloud computing and SaaS as new computing platforms," Communications of the ACM, vol. 53, no. 4, pp. 27–29, 2010. [4] P. Kumar and A. Verma, "Scheduling using improved genetic algorithm in cloud computing for independent tasks," in Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12, pp. 137–142, ACM New York, 2012. [5] G. B. H. Bindu, K. Ramani, and C. S. Bindu, "Energy aware multi objective genetic algorithm for task scheduling in cloud computing," International Journal of Internet Protocol Technology, vol. 11, no. 4, pp. 242–249, 2018. [6] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, and R. Rastogi, "Load balancing of nodes in cloud using ant colony optimization,"in 2012 UKSim 14th international conference on computer modelling and simulation, pp. 3–8, Cambridge, UK, 2012. [7] A. A. A. Ari, I. Damakoa, C. Titouna, N. Labraoui, and A. Gueroui, "Efficient and scalable ACO-based task scheduling for green cloud computing environment," in 2017 IEEE International Conference on Smart Cloud (SmartCloud), pp. 66–71, New York, NY, USA, 2017. [8] N. Kumar and S. K. Sharma, "Inertia weight controlled PSO for task scheduling in cloud computing," in 2018 International Conference on Computing, Power and Communication Technologies (GUCON), pp. 155–160, IEEE, Greater Noida, India, 2018. [9] H. Saleh, H. Nashaat, W. Saber, and H. M. Harb, "IPSO task scheduling algorithm for large scale data in cloud computing environment," IEEE Access, vol. 7, pp. 5412–5420, 2019. [10] S. S. Kim, J. H. Byeon, H. Liu, A. Abraham, and S. McLoone, "Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization," Soft Computing, vol. 17, no. 5, pp. 867–882, 2013. [11] N. J. Kansal and I. Chana, "Artificial bee colony based energyaware resource utilization technique for cloud computing," Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1207–1225, 2015. [12] P. Kaur and M. Shikha, "Resource provisioning and work flow scheduling in clouds using augmented shuffled frog leaping algorithm," Journal of Parallel and Distributed Computing, vol. 101, pp. 41–50, 2017. [13] X. Chen and D. Long, "Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm," Cluster Computing, vol. 22, S2, pp. 2761–2769, 2019. [14] M. S. Sanaj and P. M. J. Prathap, "An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment," Materials Today: Proceedings, vol. 37, pp. 3199–3208, 2021. [15] W. Jing, C. Zhao, Q. Miao, H. Song, and G. Chen,"QoS-DPSO: QoS-aware task scheduling for cloud computing system," Journal of Network and Systems Management, vol. 29, no. 1, pp. 1–29, 2021. [16] J. Yang, B. Jiang, Z. Lv, and K. K. R. Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing," Future Generation Computer Systems, vol. 105, pp. 985–992, 2020. [17] Aanchal, S. Kumar, O. Kaiwartya, and A. H. Abdullah, "Green computing for wireless sensor networks: optimization and Huffman coding approach," Peer-to-Peer Networking and Applications, vol. 10, no. 3, pp. 592–609, 2017. [18] R. S. Rathore, S. Sangwan, S. Mazumdar et al.,"W-GUN: whale optimization for energy and delay-centric green underwater networks," Sensors, vol. 20, no. 5, pp. 1377–1399, 2020. [19] 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, no. 3, pp. 3117–3128, 2020. [20] S. Mirjalil and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51–67, 2016. [21] A. Alireza, "PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems," Acta Automatica Sinica, vol. 37, no. 5, pp. 541–549, 2011. [22] R. F. Wang, L. C. Jiao, F. Liu, and S. Y. Yang, "Nature computation with self-adaptive dynamic control strategy of population size,