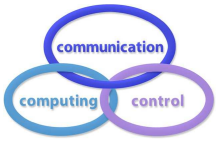
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 4, Month: August, Year: 2025
Article Number: 6814, https://doi.org/10.15837/ijccc.2025.4.6814



CCC Publications

A Genetic Particle swarm optimization based Hybrid Scheduling Algorithm for Cloud Computing Resources

Y. Wang

YiYong Wang
Zhejiang Technical Institute of Economics
Hangzhou, Zhejiang 310018, China
wyyddd@126.com

Abstract

As the quick advancement of information technology, cloud computing technology has risen rapidly, but the energy consumption and resource waste generated by data centers are also increasing. Therefore, the study analyzed the hybrid scheduling of cloud computing resources. Firstly, a improved particle swarm optimization algorithm-based resource scheduling model was raised to address the initial placement problem of virtual machines. Secondly, considering that the resource requirements of applications in cloud computing environments are dynamically changing, attention mechanisms and whale optimization algorithms were introduced to optimize the bidirectional long short-term memory network and build a resource demand prediction model. The results showed that when the amount of virtual machines was 200, the energy consumption of the improved particle swarm algorithm was 6.19 kW/h. The completion time of the algorithm always did not exceed 2000ms under different numbers of virtual machines and tasks. When the problem size was 500, the proposed resource demand forecasting model tended to converge at around 100 epochs. The prediction accuracy and recall rate of the proposed resource demand forecasting model were 94.35% and 93.62%, respectively. The experiment outcomes indicate the resource scheduling and resource demand prediction effectiveness of the raised model. The outcomes contribute to improving the service quality and effectiveness of the entire cloud platform, and promoting the development of cloud computing technology.

Keywords: cloud computing, resource scheduling, genetic algorithm, particle swarm optimization algorithm, WOA, Bi-LSTM.

1 Introduction

As the popularity of cloud computing (CP), the resource scheduling problem of cloud data centers is also receiving increasing attention. CP provides fast and secure CP services and data storage on websites [1]. By providing ultra large scale computing power and flexible resource scheduling, CP can automatically adjust resources according to user needs and provide elastic scalability capabilities [2]. However, cloud centers generally suffer from low utilization of software and hardware resources and high energy consumption. Therefore, to fully utilize the resources on the cloud platform and improve

the service quality and performance of the entire cloud platform, it is crucial to adopt effective resource scheduling strategies. CP resource scheduling refers to the process of effectively managing and allocating computing resources through intelligent algorithms and policies in a hybrid cloud environment. The main purpose is to ensure the reasonable allocation of computing resources among different tasks and users, to achieve efficient operation of the entire system [3, 4]. However, resources in cloud environments are usually dynamic, including changes in availability, delays in resource availability, and fluctuations in resource usage, all of which pose serious challenges to resource scheduling [5]. In this context, research proposes resource scheduling models with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), as well as resource demand prediction models grounded on improved Bidirectional Long Short Term Memory (Bi-LSTM) network, to solve the scheduling resource issue in complex cloud environments and raise the resource usage of cloud platforms. The innovation of the research includes the utilization of an attention mechanism, which assigns different weights to the hidden states of Bi-LSTM, and using Whale Optimization Algorithm (WOA) to raise the hyperparameters of Bi-LSTM.

2 Literature review

CP is a mode of adding, using and delivering Internet based related services, allowing users to access configurable computing resource pools through the network. Katal A et al. stated that CP data centers require a large amount of electricity to provide services, leading to an increase in carbon dioxide emissions. Therefore, they engaged in a deliberation regarding the software technologies that can be leveraged to develop environmentally sustainable data centers and the strategies for curtailing data center power consumption. This helped to reduce environmental pollution and promote the green development of CP technology [6]. Mark J et al. discussed strategies to reduce carbon emissions from data transmission in CP environments, addressing the significant challenges posed by the rapid development of CP to environmental sustainability. They also provided practical recommendations for reducing carbon emissions from hardware, algorithms, and renewable energy utilization aspects. This helped to reduce the impact on the environment while maintaining high performance and reliability standards for CP [7]. Islam R et al. stated that CP provides a wide range of architectural configurations, allowing organizations to swiftly and effectively expand or contract their computer resources in response to the dynamic demands of the business environment and market conditions. Besides, to better understand CP, the future advantages and challenges it will face were discussed, pointing out that CP may promote further innovation in artificial intelligence and machine learning [8]. Kunduru AR et al. stated that although CP has the advantages of flexibility and cost-effectiveness, applying CP in existing business models may pose significant security risks. Therefore, the merits and demerits of CP, as well as its application in information risk management, were discussed, and it was pointed out that enterprises need to actively anticipate and consider possible risks and response strategies [9]. Cinar B et al. stated that cloud services in CP have not yet developed primary forensic tools to assist in investigating criminal behavior, making it difficult to effectively prevent cloud vulnerabilities and criminal targets. Therefore, an analysis of digital forensics investigation was conducted, and the current and future trends of cloud forensics methods and tools were examined, which will help improve the security of CP technology [10].

PSO algorithm denotes a population-based stochastic optimization technology that verifies the fitness of each point by regularly moving particles multiple times in the solution space. It has been widely used in the optimization of virtual machine scheduling strategies. Nabi et al. asserted that metaheuristic algorithms based on swarm intelligence were highly suitable for cloud scheduling. However, they noted that existing PSO algorithms still require further optimization to realize optimal scheduling outcomes. Accordingly, a linear descent and adaptive inertia-weight balancing approach was utilized with the objective of achieving a balanced relationship between local and global search. The outcomes showed that the raised method improved completion time, throughput, and average resource utilization by 10%, 12%, and 60%, respectively [11]. Malik M et al. designed a hybrid algorithm that integrates PSO and grey wolf optimization algorithm to perform parallel task scheduling and find optimized virtual machines (VMs) for the task scheduling problem in CP, which helped to

reduce response time to the greatest extent possible. The outcomes indicated that the raised method was more effective than existing systems [12]. Syed D et al. stated that the widespread utilization of CP has led to a significant surge in user requests, which may result in issues such as resource surplus or low resource utilization. Therefore, the utilization of PSO algorithm and its variants was explored to evenly distribute incoming traffic with efficient resource utilization, which could help raise the effectiveness of CP systems [13]. Nabi S et al. developed a resource dynamic load balancing algorithm grounded on an improved PSO algorithm to improve user satisfaction and cloud resource utilization, to achieve task scheduling for CP. The outcomes indicated that the raised algorithm increased completion time, average resource utilization, and penalty cost by 66%, 162%, and 98%, respectively, contrast to existing advanced task scheduling heuristic methods [14]. Srivastava A et al. solved the issue of energy-efficient resource allocation in CP by training a dataset using the PSO algorithm. They proposed solving the configuration problem by scheduling tasks to VM, which helped to lessen system energy consumption and improve the effectiveness of scheduling algorithms. The outcomes indicated that the raised method could improve energy efficiency by 12% and increase average start-up time by over 50%, demonstrating certain effectiveness [15]. Mangalampalli S et al. proposed using an optimized POS algorithm for task scheduling, as most existing CP scheduling algorithms overlook the issues of energy consumption, time, and total electricity costs. The results showed that the proposed algorithm reduced energy consumption by 22% and 12% compared to traditional PSO and Client/Server algorithms, respectively, and had certain feasibility and effectiveness [16].

In summary, although previous researchers have conducted extensive research on CP and affirmed the role of PSO algorithm in CP scheduling, the current CP resource scheduling technology still faces issues of poor stability and performance. Therefore, the research on CP resource hybrid scheduling algorithm based on GA and PSO has certain practical application value and prospects.

3 Research methodology

To fully utilize resources on cloud platforms, a GA-PSO algorithm-based resource scheduling model is designed to address the initial placement issue of CP VM. Secondly, a resource demand prediction model with improved Bi-LSTM is developed to address the issue of dynamic placement of VM.

3.1 Construction of resource scheduling model based on GA-PSO algorithm

In CP, resources are scheduled in the form of VM, and the main task of virtual machine resource scheduling is to allocate and manage virtual machine resources reasonably to raise the efficiency and performance of CP systems. Therefore, a GA-PSO algorithm is proposed to schedule CP resources. The process of CP resource scheduling is shown in Figure 1.

Assuming the virtual machine is $V = \{v_1, v_2, ..., v_m\}$ and the server queue is $S = \{s_1, s_2, ..., s_n\}$, the mapping relationship between the virtual machine and the server is shown in formula (1).

$$A = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{bmatrix} \tag{1}$$

In equation (1), if virtual machine v_i is placed on server s_j, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. The study uses the resource waste rate to reflect the resource usage of servers, and the calculation of the resource waste rate w_j for the jth server is shown in formula (2).

$$w_j = \frac{\left| R_j^c - R_j^r \right| + \alpha}{U_j^c + U_j^r} \tag{2}$$

In equation (2), R_j^c and R_j^r represent the proportions of remaining CPU and memory resources to total resources. w_j represents the balance parameter, set to 0.0001 in the study. U_j^c and U_j^r represent

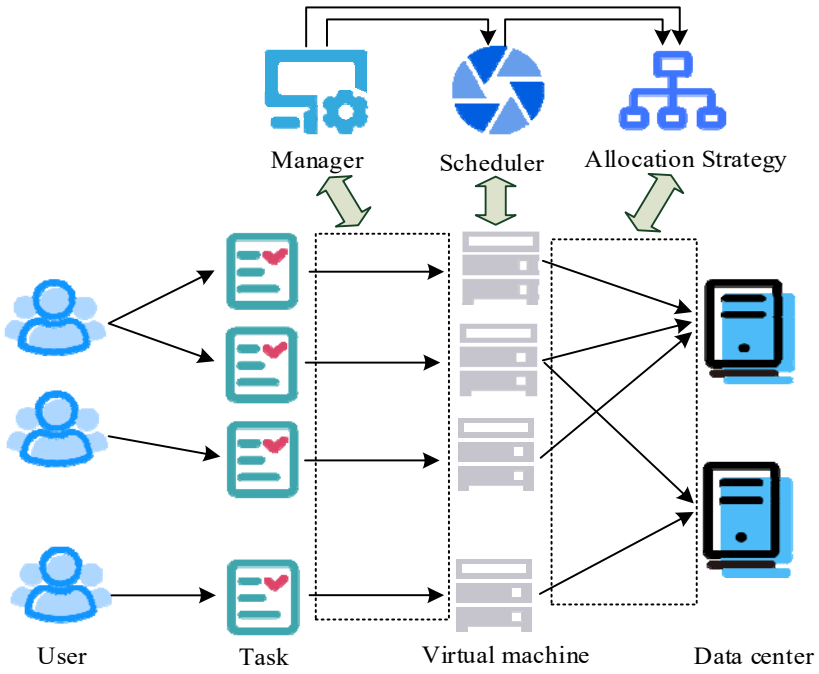


Figure 1: Cloud computing resource scheduling flowchart

the proportions of used CPU and memory resources to total resources, respectively. The calculation of the load L_j of the jth server is shown in formula (3).

$$L_j = \sum_{i=1}^{m} (r_i^c a_{ij}) + \sum_{i=1}^{m} (r_i^r a_{ij}) + \sum_{i=1}^{m} (r_i^d a_{ij}) \tag{3}$$

In equation (3), r_i^c, r_i^r, and r_i^d respectively represent the amount of CPU requests, memory requirements, and disk requirements of virtual machine i. To balance the resource load of all servers on the cloud platform, formula (4) is proposed in the study.

$$\begin{cases} \bar{L} = \frac{\sum_{j=1}^{n} (L_j)}{n} \\ S = \sqrt{\frac{\sum_{j=1}^{n} (L_j - \bar{L})^2}{n}} \end{cases} \tag{4}$$

In equation (4), $\bar{L}$ means the average load and S means the standard deviation. The calculation of server energy consumption w_j^p is shown in formula (5).

$$\begin{cases} p_j^t = p_j^{idle} + \left(p_j^{\max} - p_j^{idle}\right) U_j^c(t) \\ w_j^p = \frac{p_j^t}{p_j^{\max}} \end{cases} \tag{5}$$

In equation (5), p_j^t represents the power of the jth server at time t, $p_j^{\max}$ represents the maximum power, and p_j^{idle} represents the power when idle. Due to the involvement of multiple constraints in resource scheduling in CP, the complexity of the problem is high. Therefore, heuristic algorithms capable of handling large-scale and complex issues are adopted in the research to solve it. Heuristic algorithm is an algorithm based on intuitive or empirical construction, which combines random algorithm and local search algorithm, and can provide feasible solutions for the combinatorial optimization problem to be solved at an acceptable cost [17]. The concrete calculation method of the heuristic algorithm is illustrated in Figure 2.

The GA is a meta-heuristic algorithm that emulates natural selection processes. It employs a set of operators, including mutation, crossover, and selection, which are inspired by biological mechanisms, to develop effective solutions to optimization and search problems of the highest quality [18]. The search process of GA relies on the internal fitness function to calculate the fitness values of each

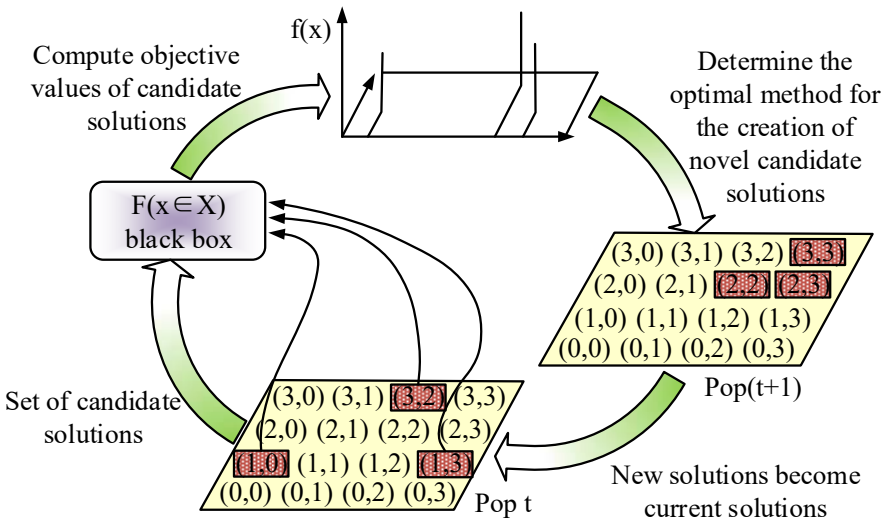


Figure 2: Calculation flowchart of heuristic algorithm

chromosome and select excellent individuals, as shown in formula (6).

$$\begin{cases} Fit(i) = \frac{1}{(F(x))^k} \\ k = b + (a(Iter\max - Iter))^2 \end{cases} \tag{6}$$

In formula (6), $Fit(i)$ means the fitness function value of individual i, a and b represent adjustable parameters, $Iter\max$ means the max amount of iterations, and $Iter$ means the current amount of iterations. Based on the individual fitness function value, the GA determines how to pass on the parent gene to the offspring through the selection operator. The probability of individual i being chosen is denoted in formula (7).

$$P(i) = \frac{Fit(i)}{\sum_{i=1}^{M} Fit(i)}, i = 1, 2, ..., M \tag{7}$$

In formula (7), M represents the population size. Although GA has certain advantages in solving optimal problems, it is more sensitive to the initial population and has lower search efficiency in the later stage. Therefore, the study combines PSO algorithm with faster convergence speed to compensate for the shortcomings of GA. The PSO algorithm is a group cooperative random search algorithms. Assuming there exists a D dimensional space, the velocity and position of particle i are updated as shown in formula (8).

$$\begin{cases} Vij(t+1) = \omega vij(t) + c1r1\left(pbestij(t) - xij(t)\right) + c2r2\left(gbestij(t) - xij(t)\right) \\ xij(t+1) = xij(t) + vij(t+1) \end{cases} \tag{8}$$

In equation (8), ω represents the inertia factor; $c1$ represents the self-learning factor, whose value affects the global search ability of particles; $c2$ represents the global learning factor, whose value affects the ability of particles in local search; $r1$ and $r2$ represent random numbers in range of (0,1); $vij(t)$ and $xij(t)$ respectively denote the velocity and position components of particle i in the jth dimension when it evolves to the tth generation; $pbestij(t)$ and $gbestij(t)$ respectively denote the individual optimum and global optimum of particle i in the jth dimension when it evolves to the tth generation; $vij(t+1)$ and $xij(t+1)$ respectively denote the velocity and position components of particle i in the jth dimension when it evolves to the $t+1$th generation. The calculation of the optimal state $Ggood(t)$ for all particles in the group is shown in formula (9).

$$Ggood(t) = \min\left\{Pgood1(t), Pgood2(t), ..., Pgoodn(t)\right\} \tag{9}$$

In equation (9), $Pgoodi(t)$ represents the optimal solution. Although the inertia factor in PSO algorithm increases its flexibility, its non-adjustable nature during iteration also makes it difficult for

PSO algorithm to maintain a balanced relationship between global and local search. Therefore, the study uses dynamic inertia weights to replace the original inertia weights, as shown in formula (10).

$$\omega_{n+1} = \omega_n + r\frac{f_n - f_{n-1}}{|f_n - f_{n-1}|} \tag{10}$$

In equation (10), ω_n and ω_{n+1} represent the current inertia factor and the inertia factor for the next iteration, r represents the gain coefficient, and f_n and f_{n-1} represent the optimal fitness values for iterations n and $n-1$, respectively. In summary, the specific calculation of the GA-PSO algorithm proposed in the study is shown in Figure 3.

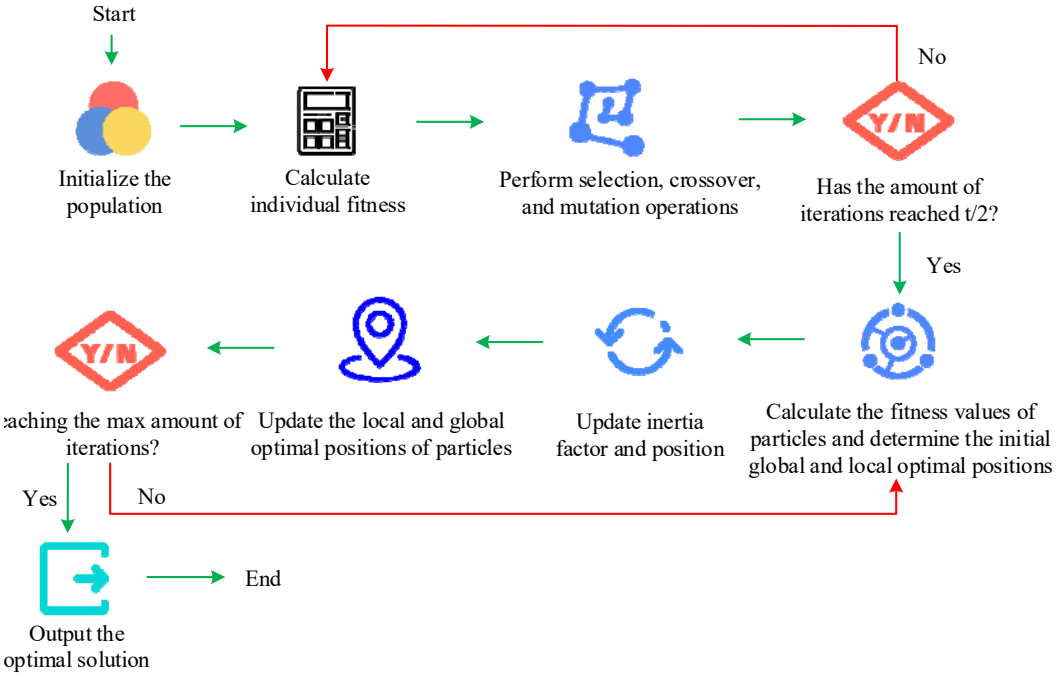


Figure 3: Flowchart of GA-PSO algorithm

3.2 Building a resource demand prediction model based on improved Bi-LSTM

A resource scheduling model with GA-PSO algorithm is raised for the initial placement of VM. However, due to the heterogeneity of CP cluster resources, VM must dynamically adapt to the CP environment. Therefore, research will further predict the resource requirements of VM to solve the issue of dynamic placement of VM. The load of VM fluctuates nonlinearly over time, and LSTM, which is suitable for handling long-term dependencies and nonlinear relationships, is studied for resource demand prediction. LSTM is built to effectively capture long-term dependencies in sequential data. The core idea of LSTM is the introduction of gating mechanisms, including forgetting gates, inputting gates, and outputting gates. Among them, the forgetting gate mainly controls the retention and discarding of old information, and the calculation of the output ft of the forgetting gate is shown in formula (11).

$$ft = \sigma\left(Wfht - 1 + Vfxt + bf\right) \tag{11}$$

In formula (11), σ represents the Sigmoid activation function, xt denotes the inputting at the current time, $ht-1$ denotes the input at the previous time, and Wf, Vf, and bf are learnable parameters. The inputting gate controls the flow of new information, and its calculation process is shown in formula (12).

$$\begin{cases} it = \sigma(Wiht - 1 + Vixi + bi) \\ ct = ft \odot ct - 1 + it \odot \bar{c}t \\ \bar{c}t = \tanh(Wcht - 1 + Vcxt + bc) \end{cases} \tag{12}$$

In formula (12), it represents the state of the input gate, Wi, Vi, and bi are learnable parameters, ct, $ct-1$, and $\bar{c}t$ denote the current, the previous, and the candidate cell states, respectively. tanh represents the activation function. The outputting gate controls the output of the new state, and its calculation process is shown in formula (13).

$$\begin{cases} ot = \sigma(Woht-1 + Voxt + bo) \\ ht = ot \odot \tanh(ct) \end{cases} \tag{13}$$

In formula (13), ot denotes the state of the outputting gate, ht denotes the outputting at the current time, and Wo, Vo, and bo are learnable parameters. However, traditional LSTM is a unidirectional recurrent neural network that cannot simultaneously consider the information before and after the sequence. Therefore, to better capture long-term dependencies in the sequence and raise the effectiveness and accuracy of the model, Bi-LSTM is studied for predicting resource demand. Bi-LSTM combines forward and backward LSTM, which can simultaneously process backward and forward information of sequence data. Its structural schematic is denoted in Figure 4.

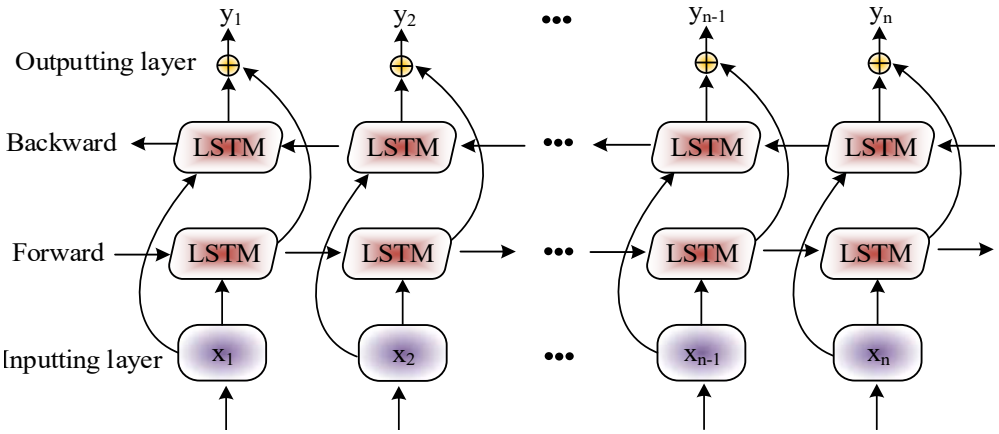


Figure 4: Structure diagram of Bi-LSTM

The calculation of the output O_t of Bi-LSTM at time t is shown in formula (14).

$$\begin{cases} \vec{h}_t = f(w_1 x_t + w_2 h_{t-1}) \\ \overleftarrow{h}_t = f(w_3 x_t + w_4 h_{t+1}) \\ O_t = g(w_5 \vec{h}_t + w_6 \overleftarrow{h}_t) \end{cases} \tag{14}$$

In formula (14), $\vec{h}_t$ means the forward layer output at time t, w means the weight, x_t means the inputting, h_{t-1} means the outputting at the previous time, h_{t-1} means the reverse layer outputting at time t, and h_{t+1} means the outputting at the next time. To better fit the CP scenario and raise the prediction accuracy and efficiency of Bi-LSTM, attention mechanism and WOA are further introduced to improve Bi-LSTM, and a WOA-Attention-Bi-LSTM algorithm is proposed. Firstly, the study utilizes attention mechanisms to assign different weights to the hidden states of Bi-LSTM, to enhance the influence of key data and reduce the loss of past information. The attention mechanism facilitates the model's capacity to selectively concentrate on salient aspects of the input data, while disregarding superfluous information. This enhances the whole effectiveness and resilience of the system [19, 20]. The calculation of attention mechanism is denoted in formula (15).

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{dk}}\right)V \tag{15}$$

In formula (15), Q indicates query, K refers to key, and V expresses value. Secondly, to address the issue of Bi-LSTM requiring a significant amount of time to customize hyperparameters, WOA is brought to automatically optimize the hyperparameters of the network model. WOA represents

an innovative heuristic optimization algorithm inspired by the hunting behavior patterns observed in natural humpback whale populations. The core idea is to explore the best solution to the issue by simulating the self-organization and adaptability of whale populations [21]. The encirclement strategy of WOA is shown in formula (16).

$$\begin{cases} D = |CX^*(t) - X(t)| \\ X(t+1) = X^*(t) - AD \end{cases} \tag{16}$$

In equation (16), D means the distance between the individual's position and the optimal position at t iterations, $X^*(t)$ means the position of the prey, $X(t)$ represents the current position of the whale, $X(t+1)$represents the position of the whale at the next moment, and C and A are two coefficient vectors. WOA's bubble net attack simulates the hunting behavior of humpback whales, including spiral updates and contraction encirclement, as shown in formula (17).

$$\begin{cases} X(t+1) = D' e^{bl} \cos(2\pi) + X^*(t) \\ D' = |X^*(t) - X(t)| \end{cases} \tag{17}$$

In equation (17), b controls the shape of the spiral, and l represents a random number. The random search phase of WOA is shown in formula (18).

$$\begin{cases} D = |CX_{rand}(t) - X(t)| \\ X(t+1) = X_{rand}(t) - AD \end{cases} \tag{18}$$

In equation (18), $X_{rand}(t)$ indicates the random position of the prey. The target of the study is to reduce the mean square error between the actual and expected outputs of the Attention-Bi-LSTM model, which serves as the fitness function, and solve it using WOA. In summary, the calculation method of the resource demand forecasting model based on the WOA-Attention-Bi-LSTM algorithm is shown in Figure 5.

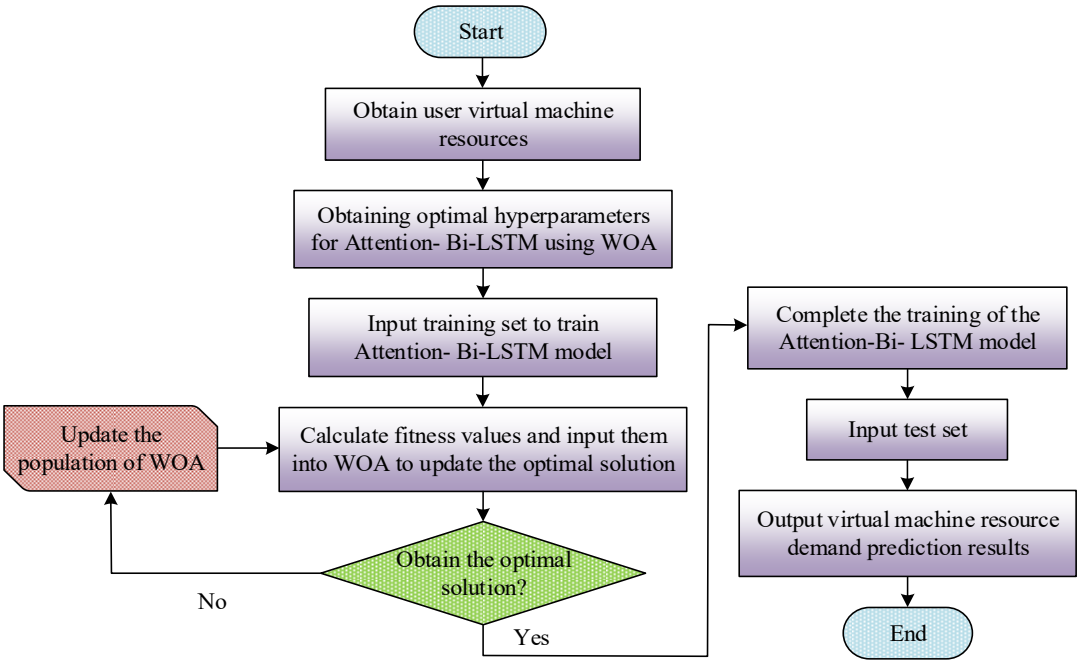


Figure 5: Flowchart of resource demand prediction model based on WOA-Attention-Bi-LSTM algorithm

4 Results and discussion

A resource scheduling model based on GA-PSO algorithm and a resource demand prediction model based on improved Bi-LSTM were proposed, but their performance still needs further validation.

The research mainly analyzed from two aspects. Firstly, it analyzed the feasibility of the resource scheduling model based on GA-PSO algorithm. Then, it verified the predictive performance of the resource requirement forecast model based on the improved Bi-LSTM.

4.1 Feasibility analysis of resource scheduling model

To verify the feasibility of the resource scheduling model based on GA-PSO algorithm, simulation experiments were conducted on CloudSim 4.0 using Windows 10 operating system and Intel (R) Core (TM) i5-1035G1 processor. The experiment used 50 physical nodes, with a host memory of 24GB, a bandwidth of 1200Mbps, specifications of CPU 1000MIPS, 2000 MIPS, and 3000 MIPS, 200 VM, 2GB of memory, 200Mbps of bandwidth, with specifications of CPU250 MIPS, 500 MIPS, and 750 MIPS. The study set the amount of iterations to 200, the initial population to 20, the inertia weight to 0.5, the learning factor to 1, the crossover probability to 0.8, and the mutation probability to 0.05. Using resource waste rate and energy consumption as evaluation indicators, the GA-PSO algorithm was compared with traditional GA, PSO algorithms, and greedy algorithms. The outcomes are denoted in Figure 6. From Figure 6 (a), among the four algorithms, the GA-PSO algorithm had the lowest resource waste rate of 16.34%. Next was the greedy algorithm, with the GA having the highest resource waste rate, reaching 20.07%. From Figure 6 (b), as the number of VM increased, energy consumption also gradually increased. Among them, the energy consumption of GA-PSO algorithm was always lower than the other three algorithms. When the amount of VM was 200, the energy consumption was 6.19kW/h. The PSO algorithm performed the worst in terms of energy consumption. The outcomes indicated that the resource scheduling model based on GA-PSO algorithm proposed by the research could effectively schedule CP resources and had certain feasibility.

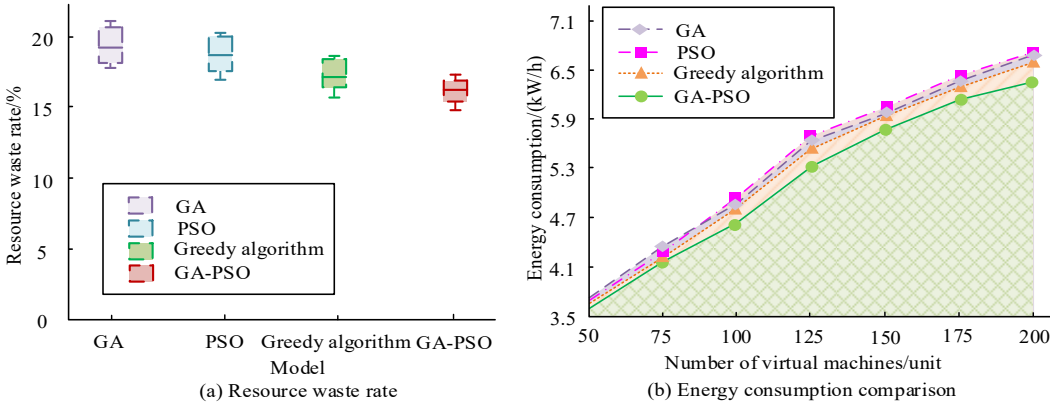


Figure 6: Comparison of resource waste rate and energy consumption among four algorithms

To investigate the stability of the GA-PSO algorithm, the amount of VM was set to 10, 20, and 30, and the completion times of the four algorithms were compared. The results are shown in Figure 7. Comparing Figures 7 (a), (b), and (c), under different numbers of VM and tasks, the completion time of the GA-PSO algorithm was always the shortest, not exceeding 2000ms, demonstrating good stability. The GA had the longest completion time and experienced significant performance fluctuations when handling large-scale tasks. The results indicated that even in situations with large task volumes, the GA-PSO algorithm could still maintain good optimization performance and had good stability.

To investigate the efficiency of the GA-PSO algorithm, experiments were conducted using the Azure VM packing trace dataset to extract virtual machine creation requests. The average running time of GA-PSO algorithm for solving problems of different dimensions was compared with GA, PSO algorithm, Differential Evolution (DE) algorithm, and Artificial Bee Colony (ABC) algorithm, with each algorithm running independently 20 times. The outcomes are indicated in Table 1. From Table 1, among the five algorithms, the GA-PSO algorithm proposed by the research had the shortest average running time. When the problem dimensions were 10, 20, 30, 40, and 50, the average running time was 0.65s, 0.99s, 1.42s, 1.84s, and 2.36s, respectively, which was significantly lower than the other four algorithms. Next was the ABC algorithm, and the GA had the longest average running time. The

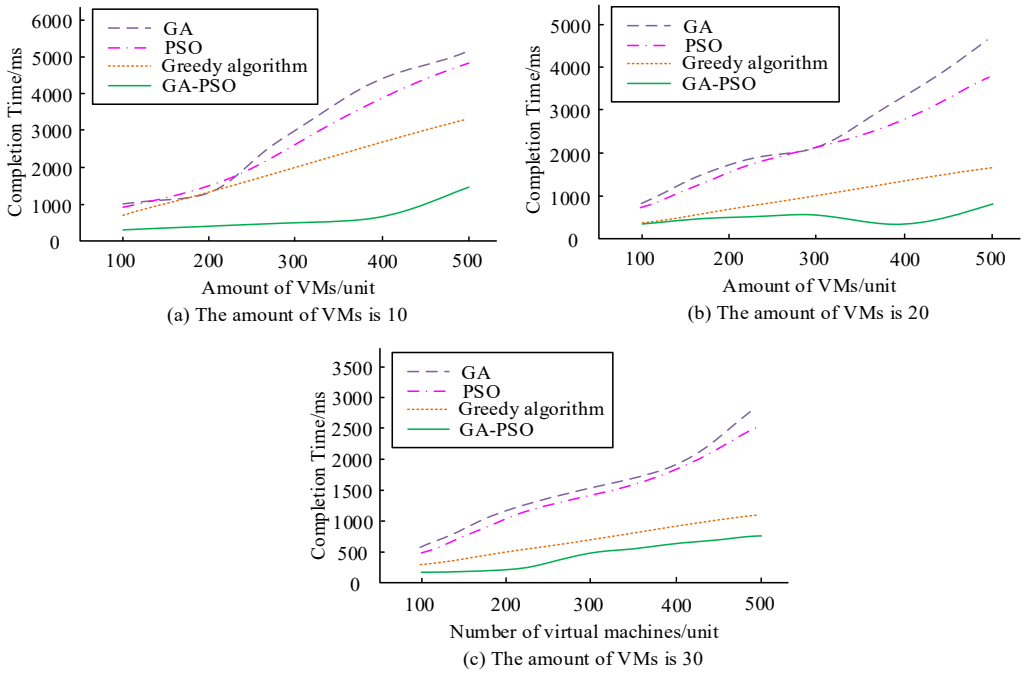


Figure 7: Comparison of completion times for four algorithms

results indicated that the GA-PSO algorithm had high computational efficiency.

Table 1: Comparison of average running time of five algorithms

Problem dimension	10	20	30	40	50
GA	3.91s	5.52s	7.44s	9.62s	12.17s
PSO	3.65s	5.28s	7.17s	9.40s	11.88s
DE	3.32s	4.83s	6.54s	8.67s	10.98s
ABC	2.18s	3.54s	5.22s	7.16s	9.56s
GA-PSO	0.65s	0.99s	1.42s	1.84s	2.36s

4.2 Performance analysis of resource demand forecasting model

To verify the predictive performance of the WOA-Attention-Bi-LSTM algorithm proposed by the research, experiments were conducted using the Azure VM packing trace dataset. The convergence curve of the WOA-Attention-Bi-LSTM algorithm was compared with the Moth Swarm Algorithm (MSA), WOA algorithm, GA, and PSO algorithm. The convergence curves of the five algorithms are shown in Figure 8. From Figure 8 (a), when the problem size was 500, the WOA-Attention-Bi-LSTM algorithm had the fastest convergence speed and tended to converge at around 100 epochs. From Figure 8 (b), when the problem size was 1500, the convergence speed of the WOA-Attention-Bi-LSTM algorithm was still better than the other four algorithms, and tended to converge at around 150 iterations. The results indicated that the WOA-Attention-Bi-LSTM algorithm proposed by the research had good convergence performance, fast convergence speed, and better fitness.

To verify the predictive efficiency of the WOA-Attention-Bi-LSTM algorithm proposed by the research, the running time of the above algorithms was compared, and the outcomes are denoted in Figure 9. From Figure 9, compared to the other four algorithms, the WOA-Attention-Bi-LSTM algorithm always had the longest running time at different problem scales. When the problem scale was 3500, the running time was 26760s, but it was still within an acceptable range. This may be because the improvement strategy proposed by the research not only enhanced the predictive effectiveness of the algorithm, but also increased the complexity of the model. The MSA had the shortest running time.

To assess the performance of the raised improvement strategy, ablation experiments were conducted. The prediction accuracy and recall of Bi-LSTM, Attention Bi-LSTM, WOA Bi-LSTM, and WOA-Attention-Bi-LSTM were compared. The outcomes of the ablation experiment are denoted in

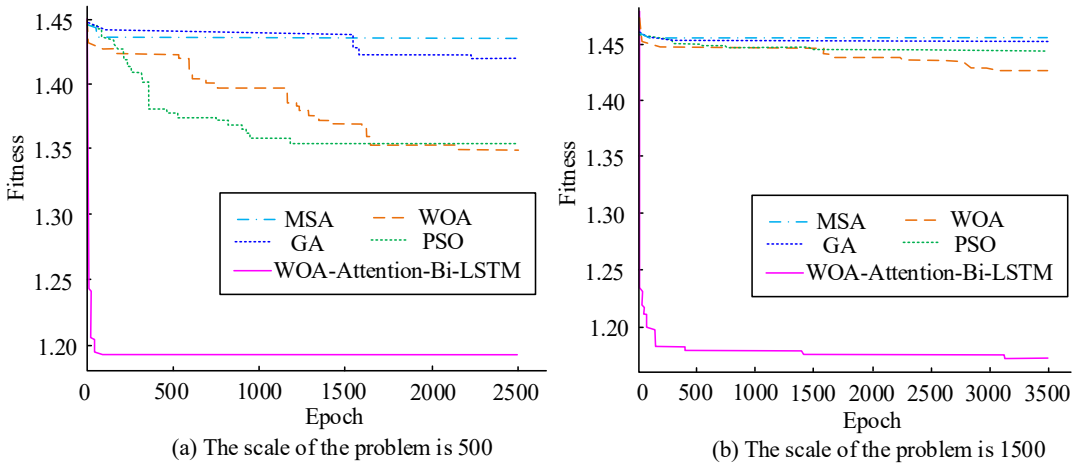


Figure 8: Convergence curves of five algorithms

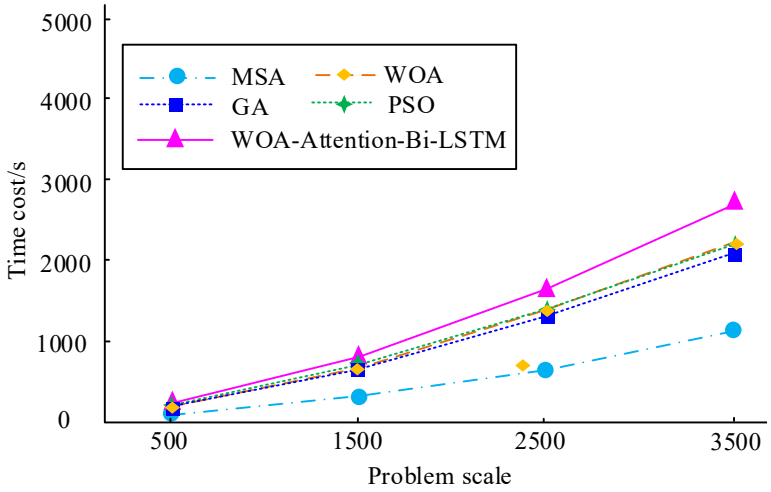


Figure 9: Comparison of running time of five algorithms

Figure 10. From Figure 10 (a), among the four models, the WOA-Attention-Bi-LSTM model had the highest forecast accuracy of 94.35%. Next was the WOA-Bi-LSTM model, with a forecast accuracy of 90.41%. The Bi-LSTM model had the lowest prediction accuracy. From Figure 10 (b), the WOA-Attention-Bi-LSTM model also performed well in terms of recall rate, with the rate of 93.62%. The results indicated that both the attention mechanism and WOA algorithm proposed by the research could effectively improve the predictive performance of Bi-LSTM, and had certain feasibility and effectiveness.

To investigate the practical application effect of a resource demand prediction model based on improved Bi-LSTM, the memory usage prediction curve of the WOA-Attention-Bi-LSTM model was compared with that of traditional LSTM. The results are shown in Figure 11. Comparing Figures 11 (a) and (b), the overall prediction performance of traditional LSTM was good, but there was still a problem of overestimation, and the prediction performance was not stable enough. The prediction curve of the WOA-Attention-Bi-LSTM model had a high degree of coincidence with the true value, and there was almost no estimation problem, demonstrating good predictive performance. The results indicated that the resource demand prediction model based on improved Bi-LSTM could effectively predict CP resource demand with high accuracy and certain effectiveness.

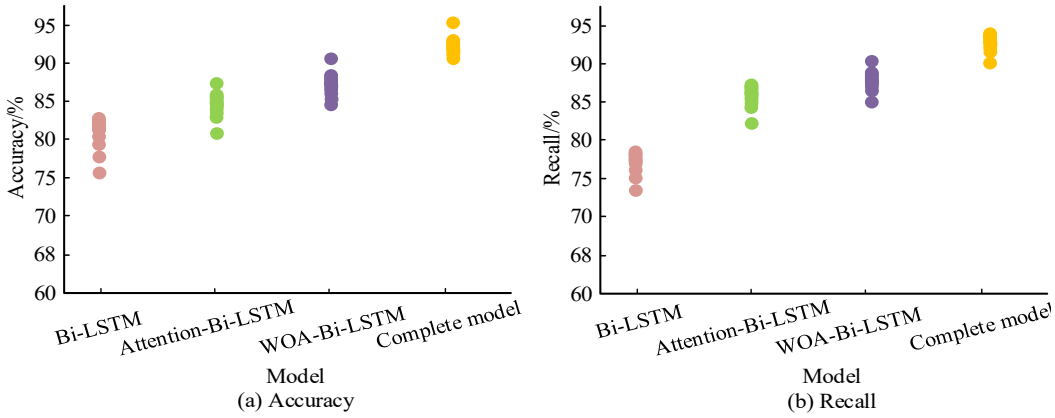


Figure 10: Results of ablation experiment

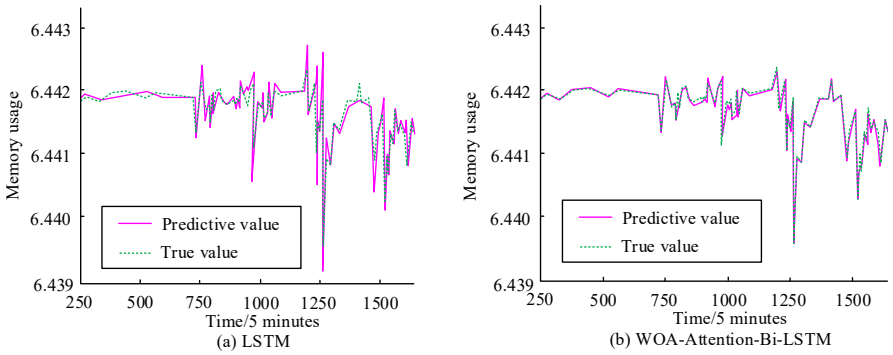


Figure 11: Comparison of memory usage prediction results

5 Conclusion

To fully utilize resources on cloud platforms, a resource scheduling model based on GA-PSO algorithm and a resource demand prediction model based on improved Bi-LSTM were proposed. The results showed that the resource waste rate of GA-PSO algorithm was 16.34%, which was lower than traditional GA, PSO algorithm and greedy algorithm. As the amount of VM increased, energy consumption also gradually increased. When the amount of VM was 200, the energy consumption of the GA-PSO algorithm was 6.19 kW/h. Under different numbers of VM and tasks, the GA-PSO algorithm consistently had the shortest completion time, not exceeding 2000ms, demonstrating good stability. The average running time of GA-PSO algorithm was the shortest. When the problem dimensions were 10, 20, 30, 40, and 50, the average running time was 0.65s, 0.99s, 1.42s, 1.84s, and 2.36s, respectively. When the problem size was 500, the WOA-Attention-Bi-LSTM algorithm had the fastest convergence speed and tended to converge at around 100 epochs. When the problem size was 1500, the convergence speed of the WOA-Attention-Bi-LSTM algorithm was still better than the other four algorithms, and tended to converge at around 150 iterations. The WOA-Attention-Bi-LSTM algorithm always had the longest running time at different problem scales. When the problem scale was 3500, the running time was 26760s, but it was still within an acceptable range. The WOA-Attention-Bi-LSTM model had the highest prediction accuracy and recall rate, at 94.35% and 93.62%, respectively. In summary, the proposed model has good resource scheduling performance and resource demand prediction performance. However, the efficacy of the WOA-Attention-Bi-LSTM algorithm proposed by the research still needs to be raised in terms of operational efficiency. Accordingly, future research should focus on further developing methodologies to enhance model efficiency while maintaining predictive performance, thus enabling more accurate forecasting of CP resource requirements.

Acknowledgement

General Research Projects of Zhejiang Provincial Department of Education in 2022 (Y202250351). Teaching Reform Project for Higher Vocational Education in Zhejiang Province during the 14th Five-Year Plan (jg20240308).

References

[1] Peñalvo FJ, Sharma A, Chhabra A, Singh SK, Kumar S, Arya V, Gaurav A. (2022). Mobile cloud computing and sustainable development: Opportunities, challenges, and future directions. International Journal of Cloud Applications and Computing (IJCAC). 12(1):1-20.

[2] Oke AE, Kineber AF, Al-Bukhari I, Famakin I, Kingsley C. (2023). Exploring the benefits of cloud computing for sustainable construction in Nigeria. Journal of Engineering, Design and Technology. 21(4):973-990.

[3] Khallouli W, Huang J. (2022). Cluster resource scheduling in cloud computing: literature review and research challenges. The Journal of supercomputing. 78(5):6898-6943.

[4] Mansour RF, Alhumyani H, Khalek SA, Saeed RA, Gupta D. (2023). Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment. Cluster Computing. 26(1):575-586.

[5] Huang X, Lin Y, Zhang Z, Guo X, Su S. (2022). A gradient-based optimization approach for task scheduling problem in cloud computing. Cluster Computing. 25(5):3481-3497.

[6] Katal A, Dahiya S, Choudhury T. (2023). Energy efficiency in cloud computing data centers: a survey on software technologies. Cluster Computing. 26(3):1845-1875.

[7] Mark J, Bommu R. (2024). Tackling Environmental Concerns: Mitigating the Carbon Footprint of Data Transmission in Cloud Computing. Unique Endeavor in Business & Social Sciences. 3(1):99-112.

[8] Islam R, Patamsetti V, Gadhi A, Gondu RM, Bandaru CM, Kesani SC, Abiona O. (2023). The future of cloud computing: benefits and challenges. International Journal of Communications, Network and System Sciences. 16(4):53-65.

[9] Kunduru AR. (2023). Security concerns and solutions for enterprise cloud computing applications. Asian Journal of Research in Computer Science. 15(4):24-33.

[10] Cinar B, Bharadiya JP. (2023). Cloud computing forensics; challenges and future perspectives: A review. Asian Journal of Research in Computer Science. 16(1):1-4.

[11] Nabi S, Ahmad M, Ibrahim M, Hamam H. (2022). AdPSO: adaptive PSO-based task scheduling approach for cloud computing. Sensors. 22(3):920-920.

[12] Malik M, Suman. (2022). Lateral Wolf Based Particle Swarm Optimization (LW-PSO) for Load Balancing on Cloud Computing. Wireless Personal Communications. 125(2):1125-1144.

[13] Syed D, Shaikh GM, Rizvi S. (2024). Systematic Review: Particle Swarm Optimization (PSO) based Load Balancing for Cloud Computing. Sir Syed University Research Journal of Engineering & Technology. 14(1):86-94.

[14] Nabi S, Ahmed M. (2022). PSO-RDAL: Particle swarm optimization-based resource-and deadline-aware dynamic load balancer for deadline constrained cloud tasks. The Journal of Supercomputing. 78(4):4624-4654.

[15] Srivastava A, Kumar N. (2023). An energy efficient robust resource provisioning based on improved PSO-ANN. International Journal of Information Technology. 15(1):107-117.

[16] Mangalampalli S, Swain SK, Mangalampalli VK. (2022). Multi objective task scheduling in cloud computing using cat swarm optimization algorithm. Arabian journal for science and engineering. 47(2):1821-1830.

[17] Kaveh M, Mesgari MS. (2023). Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. Neural Processing Letters. 55(4):4519-4622.

[18] Alhijawi B, Awajan A. (2024). Genetic algorithms: Theory, genetic operators, solutions, and applications. Evolutionary Intelligence. 17(3):1245-1256.

[19] Chen C, Zhang Q, Kashani MH, Jun C, Bateni SM, Band SS, Dash SS, Chau KW. (2022). Forecast of rainfall distribution based on fixed sliding window long short-term memory. Engineering Applications of Computational Fluid Mechanics. 16(1):248-261.

[20] Tang J, Li Y, Ding M, Liu H, Yang D, Wu X. (2022). An ionospheric TEC forecasting model based on a CNN-LSTM-attention mechanism neural network. Remote Sensing. 14(10):2433-2433.

[21] Nadimi-Shahraki MH, Zamani H, Asghari Varzaneh Z, Mirjalili S. (2023). A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. Archives of Computational Methods in Engineering. 30(7):4113-4159.

Copyright ©2025 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Wang, Y. (2025). A Genetic Particle swarm optimization based Hybrid Scheduling Algorithm for Cloud Computing Resources, *International Journal of Computers Communications & Control*, 20(4), 6814, 2025.
https://doi.org/10.15837/ijccc.2025.4.6814