International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I10P226

Volume 70 Issue 10, 271-276, October 2022

Original Article

Effective Task Scheduling in Cloud Computing using Improved BAT Algorithm

Shyam Sunder Pabboju[1], Adilakshmi Thondepu [2]

[1] *Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Telangana, India.*
[2] *Computer Science and Engineering, Vasavi College of Engineering, Telangana, India.*

[1]*Corresponding Author : pshyamsunder_cse@mgit.ac.in*

Received: 20 July 2022 Revised: 23 September 2022 Accepted: 08 October 2022 Published: 20 October 2022

Abstract - *The rapid adoption of cloud computing can be attributed to its high-performance distributed computing. Internet users can use its services and access shared resources through their own service providers. The scheduling of tasks is the primary challenge in cloud computing, which drags down the overall performance of the system. There is a requirement for an effective task-scheduling algorithm to increase the system's performance. The primary goal of the scheduling is to reduce the amount of time lost and the amount of work done while simultaneously increasing throughput. Therefore, the work of scheduling is necessary if one is to attain accuracy and correctness in the process of completing tasks in the cloud. Several different meta-heuristic task scheduling algorithms for the cloud, such as those based on Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO), have been investigated, and all of them have demonstrated great performance in a reasonable amount of time (PSO). In this research piece, we used a metaheuristic strategy known as the bat algorithm. The Bat algorithm was developed expressly to optimise difficult issues. The job scheduling method that has been suggested is evaluated alongside scheduling algorithms that are based on genetic algorithms, ant colony optimization, and particle swarm optimization, respectively. In light of the findings, it is clear that the proposed algorithm has a superior performance to that of the other algorithms.*

Keywords – *Cloud, Scheduling, Genetic Algorithm, Improved BAT, ACO, PSO.*

1. Introduction

Cloud computing[1] is one of the growing technologies everyone is talking about, and many businesses have already moved their operations to the cloud. The concept of cloud computing dates all the way back to the 19th century. The computing technique is known as IBM Mainframe. It was necessary to make work easier to perform large applications simultaneously from multiple places while retaining collaborative work among developers and making work easier. The provision of hardware, operating system software, and application software in the form of service is becoming increasingly reliant on cloud computing. The unlimited resources that make up the shared pool are distributed in a setting with distributed nodes.

Task scheduling in cloud computing determines the correct order of assigning jobs to virtual machines, allowing for the most efficient use of available resources and increasing throughput. Its primary objective is to maintain an even level of activity across all available resources and to prevent overburdening any machine or resource with an excessive number of jobs. When it comes to providing services to customers via cloud computing, various problems and obstacles must be overcome. The cloud service provider is responsible for delivering a scheduling procedure that is both effective and optimised. This procedure must consider various factors, including cost, time, and the SLA requirements that the users have chosen. Virtual machines (VMs) map heterogeneous physical resources to tasks during the task-scheduling process. It is done because a smart scheduling algorithm should consider how the distributed system can run more efficiently by considering cost, execution time, and other factors.

The scheduling technique becomes an NP-complete problem [2] due to the heterogeneity of the resources and the variable features of the activities. These problems do not have any precise approaches that may be followed to obtain answers in polynomial time. Recently, there has been a lot of interest in meta-heuristics-based techniques [3], which are designed to provide near-optimal solutions for challenging scenarios. The application of a meta-heuristic strategy discussed in this paper is known as the bat algorithm (BAT)[4]. The same is compared with other algorithms like the genetic algorithm (GA)[5], the ant colony optimization (ACO)[6], and the particle swarm optimization (PSO)[23].

2. Related Works

In 2010, Yang [8] proposed a new optimization technique called the BAT algorithm. He was inspired by how bats use echolocation to find their way around. Bats can figure out how far away their prey is by using echolocation. To find their prey, they fly around randomly and change their speed, position, frequency, volume, and pulse emission rate. When they are on the hunt, they can adjust the frequency, loudness, and pulse rate of their emission according to the distance between themselves and the prey. The BAT algorithm was made by looking at how bats act.

Jacob [9] used the BAT algorithm for resource scheduling in cloud computing to reduce makespan. He concluded that the BAT approach is superior to GA in terms of both its level of accuracy and its level of efficiency.

Kumar et al. [10] devised a way to schedule tasks in the cloud by putting together the algorithms known as BAT and Gravitational scheduling (GSA) and taking into account deadlines and the trust model. The trust value of the resources is used to choose which ones to use for the tasks.

The workflow scheduling issue that was present in the cloud was successfully resolved by Raghavan et al. [11] using the Bat method. The goal was to reduce the cost of processing for the whole workflow as much as possible. Compared to the Best Resource selection (BRS) algorithm, this one is better in terms of how much it costs to process.

A method for the scheduling of tasks was developed by Wang et al. [12], and it was based on a genetic algorithm. The goal was to reduce the time required to complete a task and spread the load evenly across virtual machines. They started the population with a greedy algorithm, and the fitness ratio is the foundation for their selection method.

Zhu et al. [13] introduced a Multi-Agent Genetic Algorithm known as MAGA to distribute the load among multiple virtual computers. The combination of GA with multi-agent approaches makes MAGA so effective in improving the quality of optimization outcomes and reducing the amount of time needed to arrive at a solution.

An approach for workflow scheduling developed by Chen et al. [14] and predicated on the Ant Colony System (ACS) algorithm was presented and had numerous new elements added to it to improve it. They wanted to keep the costs as low as possible while completing the job on time.

An Ant Colony Optimization (ACO) metaheuristic has been proposed by Tawfeek et al. [15]. The behaviour of real ants inspired this metaheuristic as they searched for the shortest path between their colonies and a food source. The Ant Colony Optimization (ACO) algorithm prioritizes the makespan reduction as the primary objective function. They have imposed the limitation that each Virtual Machine (VM) can only be visited once per ant, and the heuristic function they use is based on the predicted amount of time a job takes to execute and transfer.

Yassa et al. [16] suggested using PSO as the basis for an approach to process scheduling in the cloud. This strategy intends to cut down on the amount of time and money spent developing user applications under the Service Level Agreement (SLA) and the amount of power used by physical computers in the data centre. They have reduced their overall energy use by employing a technique known as Dynamic Voltage and Frequency Scaling (DVFS).

In their paper [17], Liu and Wang introduced an algorithm for load balancing among cloud-based virtual computers based on PSO. The programme endeavors to shorten the makespan of virtual machines while increasing the utilization of their resources.

In [18], George S. and colleagues discussed an algorithm that is a combination of PSO and the Multi-Objective Bat Algorithm for the purpose of maximization of profits in cloud environments. PSO is utilized for local search in the work he has proposed, and the Bat algorithm is utilized for the global update because the Bat method has a high level of global convergence.

Currently, the specific implementation of cloud computing uses distributed parallel processing technology. It means that the computing resources in the cloud are mapped into virtual computing nodes using a technology called virtualization. A complete computing task the user submits to the cloud is divided into several subtasks. The scheduling algorithm then assigns the subtasks to run on different computing nodes, and once all of the subtasks are completed, the calculation is completed.

3. Task Scheduling Based on Bat Algorithm

3.1. Improved BAT Algorithm

A swarm intelligence programme, the Bat Algorithm, was developed by merging multi-agent systems and evolutionary principles. It was inspired by how bats use echolocation to detect target behaviour [19]. Bats conduct their searches by emitting ultrasonic pulses and listening for reflections of those pulses in the surrounding area. The bat locates the target based on the strength of the echoes it receives, as well as the time delay and the amount of time that elapses before the echoes reach its ears. The concept behind the bat algorithm is to regard the individual bat to utilise the fitness function of the optimization problem to evaluate the advantages and disadvantages of each bat's position in the solution space of the optimization problem,, and to compare the bat detection target and flight movement as the best use in the iterative process. These are the three main components of the bat algorithm. A workable solution has taken the place of the ineffective one that was previously used.

Assuming that bats fly across space that has D dimensions, the following equation can be used to explain the speed and position of the bat I at time t+1:

$$v_{id}^{t+1} = v_{id}^{t} + \left(x_{id}^{t} - x_{d}^{best}\right) \text{x } f_i \tag{1}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{2}$$

In the formula, v_{id}^{t} and v_{id}^{t+1} represent the speed of the i-th bat at time t and t+1, in formula (2), x_{id}^{t} and x_{id}^{t+1} represent the position of the i-th bat at time t and t+1, and x_{d}^{best} represents the maximum obtained in the search process. The bat position corresponds to the optimal solution, fi is the frequency of the pulse sent out during the bat search, and fi is located in the interval [f_{min}, f_{max}], and f_{min} and f_{max} are the minimum and maximum values of the pulse frequency.

In the process of flying to detect the target, the bat constantly adjusts the modulation of the pulse's strength as well as frequency in order to enhance the detection efficiency. At the beginning of the search, the intensity of the pulses sent out is larger, and the frequency is lower, so a long distance can be searched.

In the process of flying to the target, the pulse intensity gradually becomes smaller and the frequency gradually becomes larger, which is beneficial to accurately grasp the constantly changing position of the target accurately. The intensity and frequency of the pulses emitted by bats can be expressed as follows:

$$A_i^{t+1} = \alpha A_i^{t} \tag{3}$$

$$R_i^{t+1} = R_i^{0} \text{ x } (1 - e^{-\gamma}) \tag{4}$$

In formula (3), A_i^t and A_i^{t+1}represent the i-th bat at time t and t+1. The pulse intensity of α is a constant in [0, 1], called the attenuation factor. In formula (4), R_i^{t+1} is the pulse frequency of the i-th bat at time t+1, R_i^0 is the initial pulse frequency and γ is a constant greater than 0, called the pulse frequency increase coefficient.

3.2. Coding of subtasks

There are T tasks and M computing nodes, and the number of subtasks that the ith task is divided into is subTask(i)(1iT), then the total number of subtasks is

$$SubTaskNum = \sum_{i=1}^{T} subTask(i) \tag{5}$$

Then number all the subtasks, and the number of the jth subtask in the ith task is

$$num(i.j) = \sum_{k=1}^{i-1} subTask(k) + j \tag{6}$$

The key to using a swarm intelligence algorithm to solve the problem lies in the coding of bionic individuals, that is, how to establish the relationship between the position of the individual in the search space and the solution of the optimization problem. The individual coding scheme of the bat algorithm is given as follows: For a scheduling problem with n subtasks, a 2n-dimensional space X is defined to represent the distribution of subtasks on computing nodes. It is stipulated that the 1 to n dimension components of X represent the nodes corresponding to the subtasks, denoted as Xr; the n+1 to 2n dimension components represent the execution order of the subtasks on the nodes, denoted as Xs. For example, if the number of subtasks is 7. The number of nodes is 3, and the value of the vector X is (1, 2, 2, 2, 2, 3, 3, 1, 4, 3, 1, 2, 2, 1), then the subtasks represented are in the order of execution on the nodes are:

–Node 1: Subtask 1;
–Node 2: Subtask 4 - Subtask 5 - Subtask 3 - Subtask 2;
–Node 3: Subtask 7 - Subtask 6.

3.3. Application of Bat Algorithm for Subtask Scheduling

Using the coding scheme given in the previous section, $f(T) = \sum_{i=1}^{n} \sum_{j=1}^{m} makespan$ij * value ij is used as the fitness value function, and the optimization goal is to make the fitness value function obtain the minimum value. Let the number of subtasks is n and the number of computing nodes is m. The steps of applying the bat algorithm for cloud computing task scheduling are given below

Initialization of the bat population. Set the following parameters: population size Q, maximum pulse intensity A0, maximum pulse frequency of R0, and pulse frequency range of [fmin, fmax], the sound intensity attenuation coefficient α, the frequency rise coefficient γ, and the maximum iteration times Tmax.

The position X of each bat is randomly generated. Each component of Xr takes a random integer between 1 and m, and each component of Xs takes a random integer between 1 and n and then corrects as follows: Xs corresponds to the same computing node. The values of the components must not be repeated and must be continuous. The bat in the best position is calculated by the formula $f(T) = \sum_{i=1}^{n}\sum_{j=1}^{m} makespan_{ij} * value_{ij}$.

1. Initialize the search pulse frequency f_i of the bat. Formula (1) is used to determine the speed of the bat, and formula (2) is used to determine the current position of the bat.
2. Generate a random number R1; if R_1r_i, randomly perturb the bat at the global extreme point, and use the perturbed position to replace the position of the i-th bat; otherwise, update the position of the bat i according to formula (2). When updating the position, if the value of the X_s component is a decimal, the decimal is rounded and normalized, and the component corresponding to the same node is subtracted from the minimum value and added 1 as the corrected value.
3. If the position of the bat i is better than the best bat position in the population after the update is completed, then the pulse intensity and frequency are updated by equations (3) and (4).
4. Determine the value of the fitness function and the relative position of the best bat in the population.
5. Repeat steps (3) to (5) until the search accuracy requirements are met, or the specified number of iterations is reached.

After the iteration, the bat in the optimal position represents an optimal subtask scheduling scheme.

4. Results and Discussions

To evaluate how well the algorithm for scheduling proposed in this study works while keeping the set of subtasks and computing nodes unchanged, CloudSim software is used as the simulation platform, and the genetic scheduling algorithm in [24] and the method are applied, respectively.

The ant colony scheduling algorithm[21], the particle swarm scheduling algorithm[22] and the bat scheduling algorithm proposed in this paper simulates the task assignment. It is assumed that there are 50 computing nodes in the cloud computing environment, and they are scheduled according to the number of sub-tasks being 200 and 500. The execution matrix ETC is randomly generated.

The population size of the four algorithms is all taken as 200, and the algorithm's parameters are listed in Table 1.

Table 1. Parameters of four scheduling algorithms

Algorithm	Parameter	Value
Genetic Algorithm (GA)	Cross Rate	0.6
	Mutation Rate	0.05
Ant Colony Algorithm (ACO)	Pheromone Weight	1
	Heuristic information weight	1
	Pheromone volatilization coefficient	0.2
Particle Swarm Optimization (PSO)	The maximum value of inertia weight	0.9
	Inertia weight minimum	0.3
	Individual learning factor	0.91
	Population learning factor	0.78
BAT Algorithm (BAT)	Search pulse frequency range	[-1,1]
	Maximum pulse frequency	0.5
	Maximum pulse sound intensity	0.25
	Pulse sound intensity attenuation coefficient	0.95
	Pulse frequency increase factor	0.05

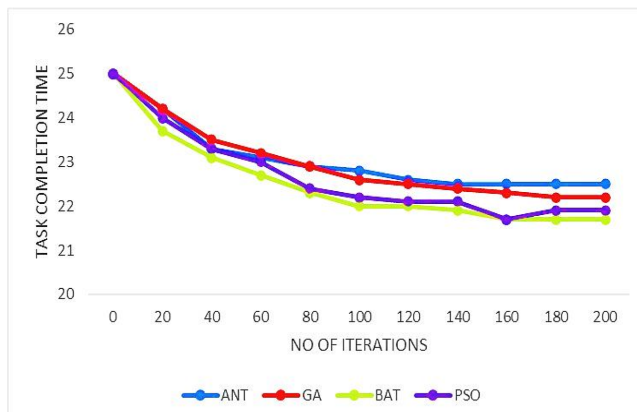

Fig. 1 Total Task completion time with 200 subtasks

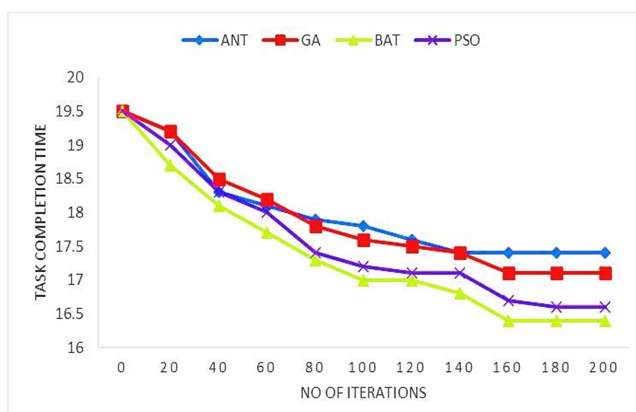

Fig. 2 Total Task completion time with 300 subtasks

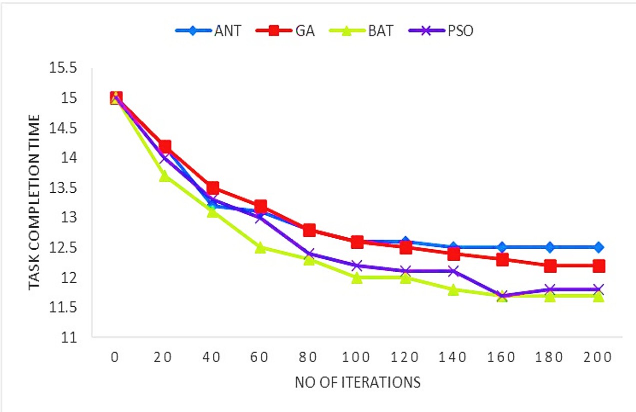

Fig. 3 Total Task completion time with 400 subtasks

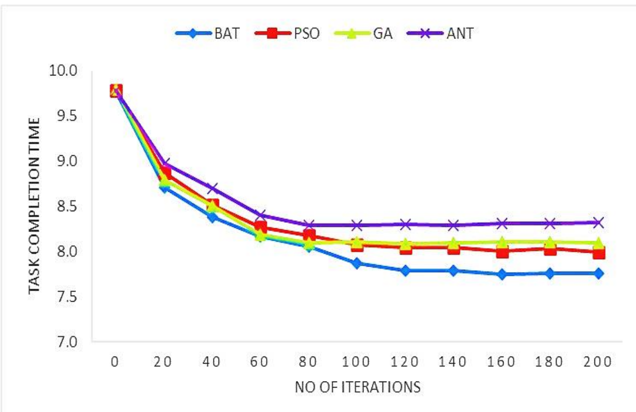

Fig. 4 Total Task completion time with 500 subtasks

Each of the four scheduling algorithms is executed 200 times, and each algorithm takes the average of the 200 runs as the final scheduling result. The simulation results when the number of subtasks is 200,300,400 and 500 are shown in Fig.1, 2, Fig.3 and 4.

It can be seen from Figure 1 to Figure 4 that the algorithm in this paper is faster than the other three algorithms in the total task completion time.

5. Conclusion

Based on the analysis of task segmentation and scheduling mechanism in the cloud computing environment, this paper designs a new subtask coding scheme according to the usage relationship and running order between each subtask and computing nodes: Two n-dimensional vectors are used to represent the nodes to which subtasks are assigned and the running order of each task on the nodes, thereby encoding a combinatorial optimization problem in continuous space, which lays the foundation for using the bat algorithm. Then, a bat population is initialized by using the characteristics of the bat algorithm, which has fast convergence speed and is not easy to fall into the local optimum. Finally, the bat algorithm solves the optimal scheduling scheme of cloud computing sub-tasks. Simulation experiments show that the algorithm proposed in this paper has obvious advantages over other swarm intelligent scheduling algorithms in the total task completion time, and its performance is better than other scheduling algorithms.

References

[1] P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," *In IEEE Transactions on Automation Science and Engineering*, vol 15, no. 2, pp. 772-783, 2018, Doi: 10.1109/Tase.2017.2693688.

[2] Gareymr, Johnson Ds , "Computers and Intractability," *A Guide to the Theory of Np-Completeness*, W.H. Freeman & Co., New York.

[3] Mala Kalra, Sarbjeet Singh, "A Review of Metaheuristic Scheduling Techniques in Cloud Computing," *Egyptian Informatics Journal*, 2015.

[4] A. K. Jayswal, "Efficient Task Allocation for Cloud Using Bat Algorithm," *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 186-190, 2020. Doi: 10.1109/Pdgc50313.2020.9315845.

[5] Y. Song, F. Wang and X. Chen, "An Improved Genetic Algorithm For Numerical Function Optimization", *Applied Intelligence*, vol. 49, no. 5, pp. 1880–1902, 2019.

[6] Zhao Junpu, Ban Jinyong, Jin Tongbiao, "Genetic Ant Colony Algorithm in Cloud Computing Resources Application in Scheduling", *Computer Engineering and Design*, vol. 38, no. 3, pp. 693-697, 2017.

[7] S. Sakthivel Padaiyatchi, S. Jaya, "Hybrid Bat Optimization Algorithm Applied to Optimal Reactive Power Dispatch Problems," *SSRG International Journal of Electrical and Electronics Engineering*, vol 9, no. 1, pp. 1-5, 2022. Crossref, https://doi.org/10.14445/23488379/IJEEE-V9I1P101.

[8] Yang Xs, "A New Metaheuristic Bat-Inspired Algorithm," *Nat Inspired Coop Strateg Optim Comput Intell* , vol. 284, pp. 65–74, 2010. http://Dx.Doi.Org/10.1007/978-3-642-12538-6_6.

[9] Jacob L, "Bat Algorithm For Resource Scheduling in Cloud Computing," *International Journal for Research in Applied Science and Engineering Technology*, vol. 2, pp. 53–7, 2014.

[10] Suresh Kumar V, Aramudhan, "Hybrid Optimized List Scheduling and Trust Based Resource Selection In Cloud Computing," *Journal of Theoretical and Applied Information Technology*, vol. 69, pp. 434–42, 2014.

[11] Raghavan S, Marimuthu, C, Sarwesh, P, & Chandrasekaran K, “Bat Algorithm For Scheduling Workflow Applications in Cloud,” *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV),IEEE*, pp. 139–44, 2015.

[12] Wang T, Liu Z, Chen Y, Xu Y, Dai X, “Load Balancing Task Scheduling Based on Genetic Algorithm in Cloud Computing,” In: *IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, pp. 146–52, 2014. http://Dx.Doi.Org/10.1109/Dasc.2014.35.

[13] Zhu K, Song H, Liu L, Gao J, Cheng G, “Hybrid Genetic Algorithm For Cloud Computing Applications,” *2011 IEEE Asia-Pacific Services Computing Conference*, pp. 182–7, 2011. http://Dx.Doi.Org/10. 1109/Apscc.2011.66.

[14] Chen W-N, Zhang J, Yu Y, “Workflow Scheduling In Grids: An Ant Colony Optimization Approach,” *IEEE Congress on Evolutionary Computation*, pp. 3308–15, 2007.

[15] Tawfeek Ma, El-Sisi A, Keshk Ae, Torkey Fa, “Cloud Task Scheduling Based on Ant Colony Optimization,” *In: 8th International Conference on Computer Engneering Syste*m, pp. 64–9, 2013. http://Dx.Doi.Org/10.1109/ ICCEs.2013.6707172.

[16] Yassa S, Chelouah R, Kadima H, Granado B, “Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments,” *Sci World Journal*, 2013. http://Dx.Doi.Org/ 10.1155/2013/350934.

[17] Liu Z, Wang X, “A Pso-Based Algorithm for Load Balancing In Virtual Machines of Cloud Computing Environment,” *Lect notes Comput Sci*ence (*Including Subser Lect notes Artif Intell Lect notes Bioinformatics*) 7331 Lncs:142–7, 2012. Http://Dx.Doi.Org/ 10.1007/978-3-642-30976-2_17.

[18] George S, “Hybrid Pso-Moba for Profit Maximization in Cloud Computing,” *International Journal of Advanced Computer Science and Applications* , vol. 6, pp. 159–63, 2015.

[19] Min Cao, Sean J. Bennett, Quanfei Shen, Ruqi Xu, "A Bat-Inspired Approach to Define Transition Rules For A Cellular Automaton Model Used to Simulate Urban Expansion", *International Journal of Geographical Information Science*, 2016.

[20] Prashant B. Jawade, D Sai Kumar, S. Ramachandram , “A Compact Analytical Survey on Task Scheduling in Cloud Computing Environment,” *International Journal of Engineering Trends and Technology* , vol. 69, no. 2, pp. 178-187, 2021.

[21] Wei, Chen Yuanyuan, "Cloud Computing Task Scheduling Model Based on Improved Ant Colony Algorithm", *Computer Engineering*, vol. 41, no. 2, pp.12-16, 2015.

[22] Agrawal, A.; Tripathi, S., “Particle Swarm Optimization With Adaptive Inertia Weight Based on Cumulative Binomial Probability”, Evolutionary Intelligence, vol. 14, pp.305–313, 2021.

[23] Yuan Zhengwu, Li Junqi, "Cloud Resource Scheduling Based on Improved Particle Swarm Algorithm", J*ournal of Computer Engineering and Design*, vol. 37, no. 2, pp.401-412, 2016.

[24] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya, R., “Cloudsim: A Toolkit For Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.