ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 10, Issue 2, pp: (11-19), Month: October 2023 – March 2024, Available at: www.paperpublications.org

A HYBRID GENETIC ALGORITHM-BASED MODEL FOR JOBS SCHEDULING IN CLOUD COMPUTING

Anthony E. Umoru[1], Samuel A. Oluwadare[2], Folasade M. Dahunsi[3]

[1, 2] Department of Computer Science, School of Computing, Federal University of Technology Akure, Nigeria.

[3] Department of Computer Engineering, School of Engineering and Engineering Technology, Federal University of Technology Akure, Nigeria.

DOI: https://doi.org/10.5281/zenodo.10300111

Published Date: 08-December-2023

Abstract: **Cloud computing is a platform that provide users with computing resources via the Internet on a pay-as-you-use basis. In cloud computing paradigm, resources are shared; hence, there is a need to utilize the resources appropriately to achieve the quality of service stipulated by the user and yield maximum profit for the service provider. This requirement of resources sharing informs the need for efficient jobs scheduling. Jobs Scheduling means the set of policies used to control the order of work to be done by a computer system. The jobs scheduling problem in cloud computing focuses mainly on efficiently using computing resources to benefit both the service provider and the user. In this study a new algorithm is proposed that minimize the completion time and cost of cloudlets on resources and improve the overall experience of both service providers and customers. The hybrid genetic algorithm uses the output of max-min and first come first serve methods. Two physical machines, twenty five virtual machines and two thousand cloudlets were used to test run the algorithm. The results showed that the proposed algorithm achieved the optimal solutions for the two metrics of completion time and cost.**

Keywords: **Cloud computing, Completion time, Hybrid Genetic algorithm, Job scheduling, Max-min, FCFS, Optimization.**

1. INTRODUCTION

The term cloud computing became visible in innovation in the early 1990s. Since then, it has been promoted and gained the attention of many researchers in computing, though it is still developing. Kumar and Verma [1] defined cloud computing as a computing environment, where many computers are linked together in private or public networks, architecture for file storage, data as well as application. Such innovation significantly decreases computational cost, application hosting, content storage and delivery cost. Cloud providers make services available to their customers and charge them according to their use. This goal is achieved via virtualization and web technologies.

One of the issues that often arise from the usage of cloud server computers as the quantity of data generated grows, is completion time. The time it takes to complete a set of jobs is referred to as its completion time. Users' needs for computing resources grow frequently, and these needs vary with users, because resources need to be shared in the cloud, there is a need for proper scheduling of the available resources to avoid wastage [2].

Aladwani [3], defined scheduling as the strategies used to manage the sequence of work done in the computing environment. An efficient job scheduling method must aim to yield less completion time so that job processing can occur within an expected time and more jobs can be processed. The job scheduling problem in cloud computing focuses on the efficient use of computing resources in the cloud to benefit both the service providers and the consumers [1]. The problem

falls into the Optimization Problem (OP) or Decision Problem (DP) and is NP-hard. An optimization problem requires finding the best solution among all the feasible solutions in a set *S*. The Job Scheduling Problem in cloud computing is solved by satisfying various scheduling goals like computational cost, minimum completion time, load balancing, energy consumption, and service level agreement [4].

Genetic Algorithms are among the several optimization algorithms that randomly search for good solutions to optimization problems. Genetic algorithm was proposed by John Holland in 1975. Other solution methods have also been deployed, including tabu search, ant colony optimization, simulated annealing, Particle Swarm Optimization [5].

The motivation for this research is derived from the fact that cloud computing has become a technology used by lots of people all over the world; as a result of a large number of people using it, some issues that arise, such as completion time and cost, need to be solved. Many studies have been carried out to develop scheduling models for cloud computing, they include the work of Thiyeb and Alhomdy [6] as well as Fahd and Taghreed [7] which developed scheduling models for cloud computing using CloudSim for simulations. Thiyeb and Alhomdy [6] used hybridize min-min and max-min algorithms, which outperformed the max-min and min-min approaches.

In Ivana et al. [8], a new scheduling method was designed. The work used the hybridized Monarch Butterfly Optimization and Artificial Bee Colony, which performed better than the existing approach. Despite the encouraging results, further research is required to improve the completion time of jobs as well as cost.

This study aims to propose a genetic algorithm based model that achieves further reduction in job completion time and minimizes cost in cloud computing.

Since Jobs Scheduling problem is a vital issue in cloud computing; many researchers have employed various methods to solve problem, focusing on different metrics. For instance, Hamed and Alkinani [9] modeled a genetic algorithm that aims to reduce both the completion time and cost of tasks and improve resource utilization. In the model, genetic algorithm device as a solution to the jobs scheduling problem in cloud computing, this proved capable of solving the problem.

Kaleeswaran et al. [10], used a genetic algorithm to minimize the utilization of resources and reduce the execution time in a private cloud environment. Amine et al. [11] use an improved deep Q-network algorithm to obtain minimal makespan in cloud computing. A novel reward function was used for the DQN model convergence. Attiya et al. [4], conducted their research using a hybrid version of Simulated Annealing to search for optimum job solutions to the scheduling problem. The main objective was to improve the rate of convergence of the local search of Harris Hawk Optimization (HHO).

Abdel-Basset et al. [12] presented the job scheduling problem in cloud computing as one of the classes of Jobs Scheduling Problem. The proposed model solves the problem using heuristic algorithm based on differential evolution. The algorithm achieved a minima makespan. While Fahd and Taghreed [7], proposed a solution to the Job Scheduling Problem in Cloud Computing based on the Round-Robin method. The study modified the round-robin algorithm by tuning its time quantum based on the mean of the time quantum and applying the burst time of the task to decide the continued execution of the task during the current round. Abdelhafiz [13], suggested a new algorithm based on the tuple algorithm. The proposed model satisfied the condition of reducing completion time and maximizing usage of resource within cloud computing.

The various algorithms have their strengths and weaknesses. This work proposes a scheduling technique based on genetic algorithm aimed at yielding minimum completion time and cost.

2. MATERIALS AND METHODS

2.1 System Architecture

Job scheduling is when different jobs get executed at predetermined time or when the right event happens [14]. The model's architecture has two separate but interrelated phases: the acceptance phase, which allows users to submit their jobs, and the running phase, which assigns resources to the jobs. The two phases are further divided into five components namely: user interface, cloudlets, scheduling, genetic algorithm, and data center component. The user interface component is a Java i/o file that sends input into the system and returns the output through the console. The cloudlets component is where the user requests (jobs) are created with their requirements. The scheduling component is where schedules are designed for the cloudlets. This component formalizes cloudlet details from the cloudlet component which then serve as input for the genetic algorithm component. The Genetic Algorithm component is hybridized with two heuristic

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 10, Issue 2, pp: (11-19), Month: October 2023 – March 2024, Available at: www.paperpublications.org

algorithms;- max-min and first come first serve algorithms. This hybridized genetic algorithm determines the best way to schedule the cloudlets on the available resource to yield minimum completion time. The data center (DC) component is considered the home of the cloud computing resources. The data center consists of Virtual Machine (VM) and physical Machines (Hosts). The entire system runs on the CloudSim simulation tool. The architecture of the system is depicted in Fig 1.

2.2 Modelling

Three scheduling techniques are being deployed in this model to reduce the completion time of jobs in cloud computing. The model can be described mathematically by equations (1) to (9).

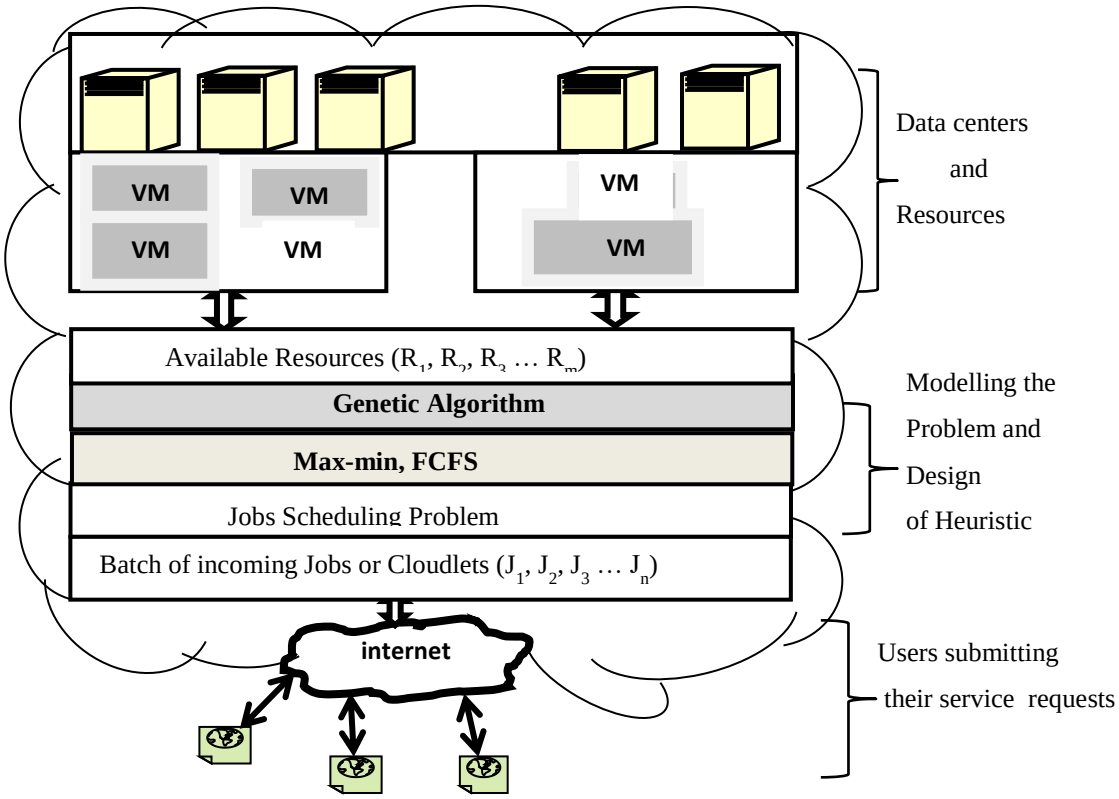

Figure 1: Components of the scheduling algorithm.

2.2.1 Acceptance Phase

The acceptance phase is stated as follows, let J represent the set of jobs to be processed in the cloud, V represent the set of virtual machines for processing, such that;

$$J = \{j_1, j_2, j_3, \dots, j_n\} \quad (1)$$

$$V = \{v_1, v_2, v_3, \dots, v_m\} \quad (2)$$

n represents the number of jobs needed to be processed at a certain time and m represents the virtual machine's number. M_{ij} is the maximum number of virtual machines that can be assigned on physical machine j at a time t, C_i Represents each physical machine available in the cloud, where i represents the physical machine number, and i is $1 <= i <= J$. C_i, V_i denote the set of virtual machines assigned on a physical machine C_i such that

$$C_i, V_i = \{V_{i1}, V_{i2}, V_{im}\} \quad (3)$$

where m is the number of VMs assigned on the physical machine. If we assign VMs v on each physical machine, then a solution set represented as $S = \{S_1, S_2, S_{3i} \dots S_n\}$ (4)

will emerge. S_i represents the mapping structure after virtual machine V_i have been assigned to physical machine C_i.

The additions of all the VMs assigned on a host machine can be referred to as the load of the host machine.

The average load on each physical machine C_i at time *t*, can be expressed as

$$AveLoadV_i\ (i,t) = \frac{1}{t}\sum_{k=1}^{n} V(i,k)\ \times (t_k - t_{k-1}) \qquad (5)$$

2.2.2 Running Phase

The running phase involves the use of hybrid genetic algorithm (HGA). The genetic algorithm is used to optimize the schedule. It finds the best schedule for all the jobs that minimizes the completion time. The following steps were taken:

a. Initial Population: An initial population is selected. In this case the initial population is the several possible permutations that exist in which cloudlets can be mapped to virtual machines and virtual machine to resources. The chromosomes represent the solutions. The initial population chromosomes were created with the outcome of two heuristic methods i.e Max-min and FCFS (Kumar and Buyya [15], Aladwani [3], Li *et al*. [16], Santhosh & Manjaiah [17]) as follows.

First the completion time for all the cloudlets was calculated to determine the cloudlet with the longest processing time and assign it to the corresponding resource with the fastest processor. The remaining cloudlets are mapped according to their arrival time while the following databases are created:

i. For all submitted jobs in Meta-jobs: J_i, calculate the minimum processing time of each job

ii. Assign job J_i that has the longest completion time to resource R_j That gives the lowest completion time.

iii. Order the job in descending order of arrival time.

iv. Order the available processors in descending order of processing power.

v. Order job from the ordered list to the ordered list of processors on a one-to-one mapping basis.

The population size is 25 chromosomes. The Selection is done according to the steps below.

i. Create chromosome S as indicated in Fig 2

ii. The first part of S is created via the outcome of max-min and FCFS from 1 to M.

iii. The other part is created via the outcome of max-min and FCFS from 1 to N while keeping the records of jobs mapped on every virtual machine

iv. Repeat step 1 to 3 to create the chromosome population.

b. Crossover takes place when new offspring emerge from the combination of several solutions available to form a new solution S_i.

2.3 Population Encoding

For most computational problems, the encoding scheme plays an important role [18]. The Selection of coding techniques for genetic algorithms depends mainly on the characteristics of the problem and the design of genetic operators. The data model in this research consists of a one-to-many scheduling relationship between cloudlets and VMs. It therefore, uses the permutation encoding method, which is useful for task scheduling or ordering problems [19].

In jobs scheduling, there are *n* jobs $J = \{J_1, J_2, \ldots, J_n\}$ and M processors $P = \{P_1, P_2, \ldots, P_m\}$. The chromosomes represent the mapping order of the virtual machines to the physical machines that will achieve the goal of finding an optimal schedule S that returns the shortest completion time and minimum cost. The chromosomes are coded with permutation encoding while the permutation derived is compared with the fitness function. Assuming there is a similarity, then Selection will be carried out. For the hybrid genetic algorithm, the chromosome is broken down into the acceptance phase and running phases. The acceptance phase represents the virtual machine's token, while the running phase depicts the jobs to be executed by each virtual machine as shown in Fig 2

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 10, Issue 2, pp: (11-19), Month: October 2023 – March 2024, Available at: www.paperpublications.org

v_0	v_1	v_2	v_3	v_4	…	v_m	j_4	j_2	j_3	j_6	j_0	j_8	…	j_n

Figure 2: virtual machine to job representation.

As indicated in Fig 3, VM v_0 executes job j_4, v_1 executes j_2. Jobs j_0 and j_8 will be executed on VM v_4 etc.

2.3.1 Fitness Function

Jianhua et al. [20], described fitness function as the criterion for the quality of the individual chromosome in the population. It directly matches the performance of the individuals. Individuals are reserved or eliminated in the next generation base on the fitness function value. The fitness function value F_i^n for this algorithm is based on completion time. The fitness function value F_i^n is obtained from the optimal schedule that gives the shortest completion time and the set of possible mapping solutions of n jobs $j_i(j_1, j_2, \ldots, j_n)$ on m resources $r_j(r_1, r_2, \ldots, r_m)$ S is represented as

$$S = \{s_1, s_2, s_3, \ldots, S_n\} \quad (6)$$

$$F_i^n = \theta(s) \quad (7)$$

while θ represents the function that returns the shortest completion time from the set is defined as

$$\theta(s) = \sum_{t=0}^{n} T_{it} \quad (8)$$

T is the total completion time (makespan) of virtual machines, *n* stands for the number of virtual machines deployed.

The fraction of total schedules whose completion time is closest to F_i^n is selected and crossover is performed on them to produce new schedules. The most outstanding schedule with the shortest completion time is then chosen from the new set of generated schedules. This is then compared with the earlier shortest completion time schedule F_i^n. If the new completion time is shorter than the earlier one;- it then becomes the new fitness function value F_i^n thereby replacing the previous one.

The Processing cost is define as $P_{cost} = \sum (pr_c + mem_c + ram_c) \times C_T$ (9)

where pr_c depicts the cost of using a processor, mem_c is the cost of storage, ram_c is the cost of using RAM and C_T is the completion time of the jobs.

2.3.2 Selection

The Selection method is used to pick a near-optimal solution for the next generation according to the law of survival of the fittest. The tournament selection method was used in this research to choose the chromosomes for the succeeding generation. Every chromosome is selected according to its fitness values from the stochastic roulette wheel in pairs. The method is computationally more efficient and tractable to multiprocessing.

2.3.3 Crossover

A Crossover operator is employed to evolve current individuals (successors) by blending the genetic features of two or more parent individuals (ancestors). The proposed model used the one-point crossover to combine the genes from the parents to yield offspring representing the new generation. One arbitrary position n is chosen for crossover, using crossover probability Pc, one successor is created by crossing over two ancestors.

2.3.4 Mutation

A Mutation operator ensures diversity in the population from one generation to the next. It saves the algorithm from premature convergence [21]. This work utilized the Order changing mutation (OCM) operator, where two numbers representing feasible schedules are selected and exchanged. The operator selects any two of the gene of a chromosome and swap their position using the mutation probability value P_m, to cause a change within a specified singular solution.

3. RESULTS AND DISCUSSION

The algorithm is implemented in Java Programming Language using Intel core i3 System 2.4 GHz processor with 320 GB HDD and 8 GB RAM on Windows 7 OS, Eclipse IDE, with CloudSim Toolkit. CloudSim has several modules. The particular module used for this work is the GA Module. The module is a collection of classes that work together to perform a genetic algorithm task.

3.1 Creation of Data Centre (DC)

The data center is the service provider's resources, referred to as the home of the resources for computing, hardware, software, network and storage [11]. The class called "Datacenter characteristics"is used to provide information about cloud resources in the datacenter. One data_center was created, which consists of two hosts and twenty five VMs.

3.1.1 Virtual Machines (VMs)

VMs are file, typically referred to as an image that behaves like an actual computer. VMs accommodate or receive the user's tasks in the cloud environment. They allow each user to have a similar experience to the actual machine. In this research work twenty VMs were created and used for the experiments.

3.1.2 Datacenter Broker

The Datacenter broker will appropriately receive the mapping schedule and use it to assign jobs to VMs appropriately. Datacenter broker performs important function between the user requests and cloud resources. The broker receives the jobs, breaks them down into tasks, and map them to the VMs according to the genetic algorithm. When the scheduling is completed, the broker returns the result to the users.

3.1.3 Cloudlets

Cloudlets are the user's tasks or jobs processed in the datacenter by the datacenter broker according to the scheduling algorithm. In this research, the number of cloudlets was varied between 100 and 2000 to test-run the system.

TABLE 1: Host machines, virtual machine and cloudlets data

Host machines parameter	**values**
number of hosts	2
Hosts CPU capacity	100000 MIPS
RAM	2GB
Hosts storage capacity	10GB
Bandwidth	10GB
Host machine policy	Space shared
TABLE1 Conti.	
Virtual Machines parameter	**values**
VMs size	10000
RAM	512 MB
VMs CPU ability	1000 mips
Storage capacity	1000
VMs bandwidth	10000 Mbps
Virtual machine policy	Time shared
VMs os	Xen
Cloudlets Data parameter	**values**
Cloudlet Numbers	100 – 3000
Length of cloudlets	60000
Size of File	400 MB

3.2 Algorithm Evaluation

CloudSim provides basic classes for describing data centers, virtual machines, applications, users, computational resources and policies for management of different parts of the system. Host and virtual machines are considered as resources, then cloudlets as jobs. Instead of random Selection, the HGA uses the output of Max-Min and FCFS to generate the initial population, and this helped to increase the fitness of the individual solution and minimize the

completion time. The number of virtual machines was fixed at 25 while cloudlets were varied at 100, 500, 1000 and 2000. The simulation was run, and the best results were recorded in each case.

3.2.1 Result Evaluation

The most common form of evaluating scheduling algorithms is by comparing them with other algorithms. The proposed Hybrid Genetic Algorithm has been compared with an existing state-of-the-art algorithm, the Monarch Butterfly Algorithm (MOBA), in terms of the completion time and cost. The result obtained is shown in Table 2 and Fig 3.

3.2.2 Bar Chart

A bar chart displays data using a number of bars, and each bar represents the value output of one of the algorithms, as shown in Fig 3. From the chart, HGA has the lower bars representing the lower completion times.

TABLE 2: Completion time comparison for HGA and MOBA

VMs fixed:25					
Number of cloudlets		100	500	1000	2000
Method Used	HGA	558.01	1431.45	1503.35	5164.93
	MOBA	1124.56	1488.74	1938.96	5632.02

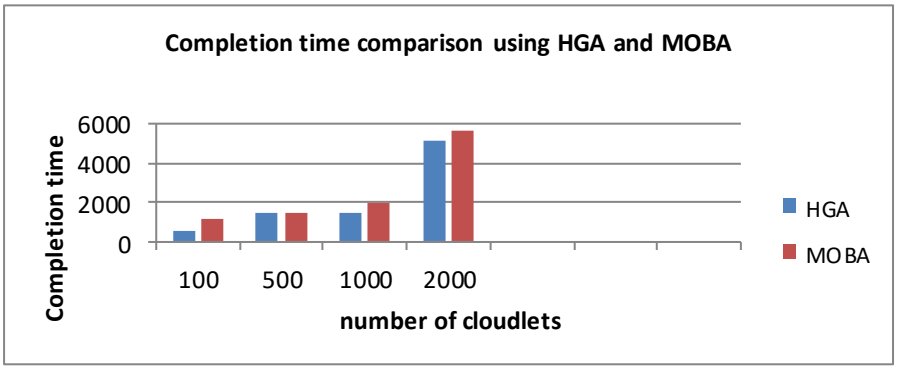

Figure 3: Bar chart

3.2.3 Result Description

In the first case, the number of virtual machines were fixed at 25 and 100 cloudlets were ran. In the other cases, the number of virtual machines were fixed at 25 and cloudlets varied at 500, 1000, and 2000. The results of the experiments showed that the proposed HGA performed significantly better in terms of Completion time. The x-axis shows the number of cloudlets used, while the y-axis shows the completion time. From the graph, it can be observed that the HGA minimizes the completion time better than MOBA. Hence, it is established that the new Hybrid Genetic Algorithm is efficient for solving the cloud computing jobs scheduling problem.

3.2.4 Cost Analysis

As done for completion time, cost metric evaluation was carried out using a set of 100, 500, 1000, and 2000 cloudlets. The outcomes of the comparison are shown in Table 3 and Fig 4.

TABLE 3: Cost comparison for HGA and MOBA

VMs fixed: 25					
Number of cloudlets		100	500	1000	2000
Method used	HGA	29.57	32.92	52.62	92.97
	MOBA	33.87	51.32	59.54	92.75

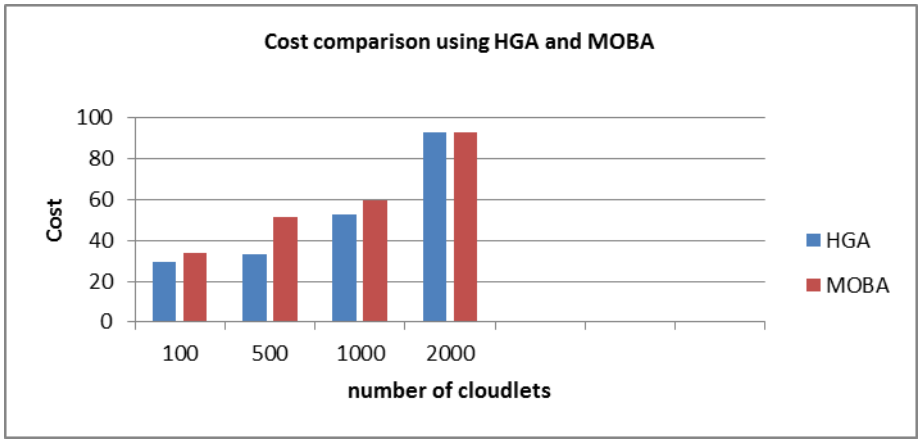

Figure 4: Bar chart

The cost result obtained in Table 3 and Fig 4 shows that HGA achieved the lower cost for 100, 500 and 1000 cloudlets while MOBA only achieved a slightly lower cost than HGA for 2000 cloudlets.

4. CONCLUSION

Jobs Scheduling has a significant role in cloud computing to ensure that user jobs are completed within a stipulated time and benefit the service provider. Since Job scheduling is a basic necessity in cloud computing, more effort should be made to ensure that this problem is well dealt with for the overall benefit of both cloud users and cloud providers.

In this paper, a hybrid genetic algorithm was proposed. The algorithm combines Max-Min, FCFS and genetic algorithms for job scheduling in cloud computing. From the result analysis as obtained for the completion time and cost metrics, in our experiments with 100, 500, 1000 and 2000 cloudlets, the proposed HGA achieved better results and proved to be efficient and robust for optimization problem-like the jobs scheduling problem in cloud computing that belongs to the class of NP-hard problems.

REFERENCES

[1] P. Kumar and A. Verma, “Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm,” International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 5, ISSN: 2277 128X, May 2012.

[2] M. Gawali and K. Subhash, “Task scheduling and resource allocation in cloud Computing using a heuristic approach, ” Journal of Cloud computing Advances, Systems and Applications, vol. 7, article 4, Feb. 2018.

[3] T. Aladwani, “Types of Task Scheduling Algorithms in Cloud Computing Environment,” International Journal of Resent Technology and Engineering, vol. 9, issue 4, pp. 209-218, Nov. 2020.

[4] I. Attiya, M. Abd Elaziz and S. Xiong, “Job Scheduling in Cloud Computing using a Modified Harris Hawk Optimization and Simulated Annealing Algorithm”, Hindawi Computational Intelligence and Neuroscience, 2020, ID 3504642, https://doi.org/10.1155/2020/3504642

[5] J. Delaram and O. Valilai, “Mathematical Model for Task Scheduling in Cloud Manufacturing Systems focusing on Global Logistics”, 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018) , Columbus, OH, USA, 2018.

[6] I. Thiyeb and S. Alhomdy, “HAMM: A Hybrid Algorithm of Min-Min and Max-Min Task Scheduling Algorithms in Cloud Computing,” International Journal of Resent Technology and Engineering, vol. 9, issue-4, pp. 209-218, Nov. 2020.

[7] A. Fahd and Z. Taghreed, ‘Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling,” 2021. https//doi.org/10.3390/computers1005006, accessed: Sep 6, 2023.

[8] S. Ivana, N. Bacanin, M. Tuba, and E. Tuba, "Resource Scheduling in Cloud Computing Based on a Hybridized Whale Optimization Algorithm," Applied Sciences. 2019, vol. 9, no. 22, pp. 4893. https:doi.org/10.3390/app 9224893

[9] A.Y. Hamed and M.H. Alkinani, "Task Scheduling Optimization in Cloud Computing Based on Genetic Algorithm," Computers, Materials & Continua, vol. 69, no. 3, pp. 3289-3301, Apr. 2021.

[10] A. Kaleeswaran, V. Ramasamy and P. Vivekanandan, "Dynamic Scheduling of Data Using Genetic Algorithm in Cloud Computing," International Journal of Advances in Engineering & Technology, IJAET ISSN: 2231-1963, vol. 5, no. 2, Jan. 2013.

[11] C. Amine, S. Ben Alla and A. Ezzati, "Makespan Optimization in Cloudlet Scheduling with Improved DQN Algorithm in Cloud Computing," 2021, https://hindawi.com/journals/sp/2021/7216795. accessed: May 5, 2023.

[12] M. Abdel-Basset, R. Mohamed, W. Elkhalik, M. Sharawi and M. Sallam, "Task Scheduling Approach in Cloud Computing Environment Using Hybrid Differential Evolution," Mathematics 2022, 10, 4049. https://doi.org/10.3 390/math10214049

[13] A. Abdelhafiz, (2022), "A new algorithm for minimizing makespan within cloud Computing," Al-Azhar Bulletin of Science, vol. 33, no 1, pp. 57-63, 2022.

[14] V. Anis and R. Nourmandi, "Resource-provision scheduling in cloud datacentre", Cumhuriyet Universitesi Fen Edebiyat Fakultesi Fen Bilimleri Dersigi, vol. 36, no 3, pp. 2603-2627. May 2015.

[15] G. Kumar and R. Buyya, "Cloud Computing and Environmental Sustainability, Australia Cloud Computing and Distributed Systems (CLOUDS)," Laboratory department of Computer Science and Software Engineering, The University of Melbourne, 2012.

[16] X. Li, Y. Mao, X. Xiao and Y. Zhuang, "An improved max-min task scheduling algorithm for the elastic Cloud, Computer, Consumer and Control (IS3C)," International Symposium on IEEE, 2014.

[17] B. Santhosh and D. Manjaiah, "An Improved Task Scheduling Algorithm based on Max-min for Cloud Computing," International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no.2, 2014.

[18] K. Sourabh, S. Chauhan and V. Kumar, "A review on genetic algorithm: past, present, and future", Multimed Tools Appl 2021, vol. 80, pp. 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

[19] J. Jianhua and G. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," 3rd International symposium in parallel architecture, algorithms and programming, 2003.

[20] L. D. Chambers, "The Practical Handbook of Genetic Algorithms Application", 2nd ed., Chapman & Hall//CRC, 2000, pp. 177-188.

[21] S. Hamad and F. Omara F, "Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment," International Journal of Advanced Computer Science and Applications, vol. 7, no. 4, 2016.

[22] J. Zheng, and Y. Wang. "A Hybrid Multi-Objective Bat Algorithm for Solving Cloud Computing Resource Scheduling Problems" *Sustainability* 13, no. 14: 7933. https://doi.org/10.3390/su13147933, accessed: Jul. 12, 2021