IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative Analysis Of Task Scheduling Algorithms Through CloudSim

Sikha Suhani Bhuyan[1], Ashis Kumar Mishra[2]

Odisha University of Technology and Research, Bhubaneswar, Odisha, India

Department of Computer Science and Engineering

ABSTRACT : Cloud computing is becoming increasingly popular among businesses and organisations in recent years. Cloud computing does, however, provide certain issues. Cloud services are widely employed by enterprises and individuals because to its scalability, dependability, and affordability. A primary drawback with cloud computing is how aid management is handled, which is based on a virtualization approach and a pay-as-you-go approach (e.g. CPU, Memory, Network, Storage, and so forth.). Many studies have been conducted on the management of those assets. This paper presents a variety of job scheduling techniques for allocating and executing tasks in a utility. This suggested set of rules aims to minimize as much as possible the amount of time, cost and resources used to complete the work, and to calculate how long each rule set will last. The CloudSim toolbox was used to analyse the overall performance of this proposed set of rules.

Keywords: Cloud Computing, Cloudsim, Makespan, Scheduling Technique

I. INTRODUCTION

Cloud computing is a utility-based environment with a pay-per-use paradigm accomplished by Internet-based Parallel, Distributed, and Cluster computing. On-demand self-service, scalability, and flexibility are all advantages of cloud computing. The cloud user may request, implement, configure, and pay for their own services via on-demand self-service. Virtualization is used to achieve scalability. Cloud computing provides endless computing resources due to its elasticity (CPU, Memory and Storage).

To achieve the shortest execution time, the fast expansion of cloud environment applications is run in parallel. Tasks are allocated to machines (matching), and the sequence in which the tasks are executed is referred to as scheduling. In the cloud, scheduling is one of the methods for ensuring service quality. Any scheduling strategy or approach can be implemented to improve the quality of service. In a cloud environment, numerous cloud users have varying service quality expectations.

The task scheduling categorization is NP-Complete. As a result, performance-based optimization techniques may be utilised to address the issue (i.e., completion time, cost, resource utilization, etc.). In order to reduce execution costs, maximise resource utilization, and optimize job completion time, we are developing a task scheduling algorithm based on PSO, ACO, and GA-PSO in a Cloud Computing environment.

II. RELATED WORK

The issue of task scheduling is non-polynomial complete. To overcome this difficulty, a variety of heuristic methods and meta-heuristic strategies were investigated inside the beyond. The most serious issue is the execution time, which must be kept to a minimum.

Improvements to the task scheduling algorithm R. Kaur and S. Kinger [1] recommended GA. Using mean and grand mean values, it uses an unique fitness function. They believe their method might be used to task and resource scheduling. As described in [2], GE Junwei proposes an algorithm that takes into account the overall challenge, average task completion time, and cost constraint.

Kokilavani et al. [3] devised a MinMin (LBMM) load balancing technique that complements assistance use while reducing makespan. This was done in two stages: the first ran the traditional Min-Min algorithm, while the second rescheduled commitments for efficient resource consumption. In grid computing, Xiaoshan et al. [4] established a unique set of principles for venture scheduling driven by QoS. In contrast to existing procedures based on the principle of great prediction created by obscure facts, the suggested methodology has been found to be superior.

Meta-heuristic scheduling methods include simulated annealing (SA), genetic set of rules (GA), particle swarm optimization (PSO), and ant colony optimization (ACO). ACO is a set of natural-inspired principles for resolving complex optimization issues [5]. Tawfeek et al. [6] use the ACO set of criteria to schedule issues in the cloud. The effectiveness of the set of rules is evaluated by comparing round-robin and FCFS scheduling algorithms.

Researchers have previously used weighted Directed Acyclic Graph (DAG) to schedule projects based solely on AACO, a method similar to structured challenge/workflow programmes[7]. The authors compared the proposed regulations to the NSGA-II, focusing on two key goals: makespan and reliability. Ref. [8] authors developed a set of multi-constrained grid scheduling rules that incorporated fault tolerance and load balancing. While enhancing asset utilisation, the proposed strategy reduces time schedule expenditures, makespan, and challenge failure rates.

S. Ravichandran and D. E. Naganathan [9] created a machine that solves the problem by allowing arriving responsibilities to queue while the scheduler recomputes and arranges them. Accordingly, the first venture in the queue is assigned to a resource based on GA and subsequently scheduled. This system's purpose is to minimise execution time while enhancing resource utilisation.

[10] In a Cloud computing environment, three scheduling methods were compared: round-robin, preemptive priority, and shortest final time first.

Furthermore, from the perspective of a Cloud firm, one of the most important responsibilities of challenge scheduling is to maximise profit through proper resource use. A model for scheduling work in the Cloud computing environment has been developed based on this approach, which has substantial economic benefits for the service provider.

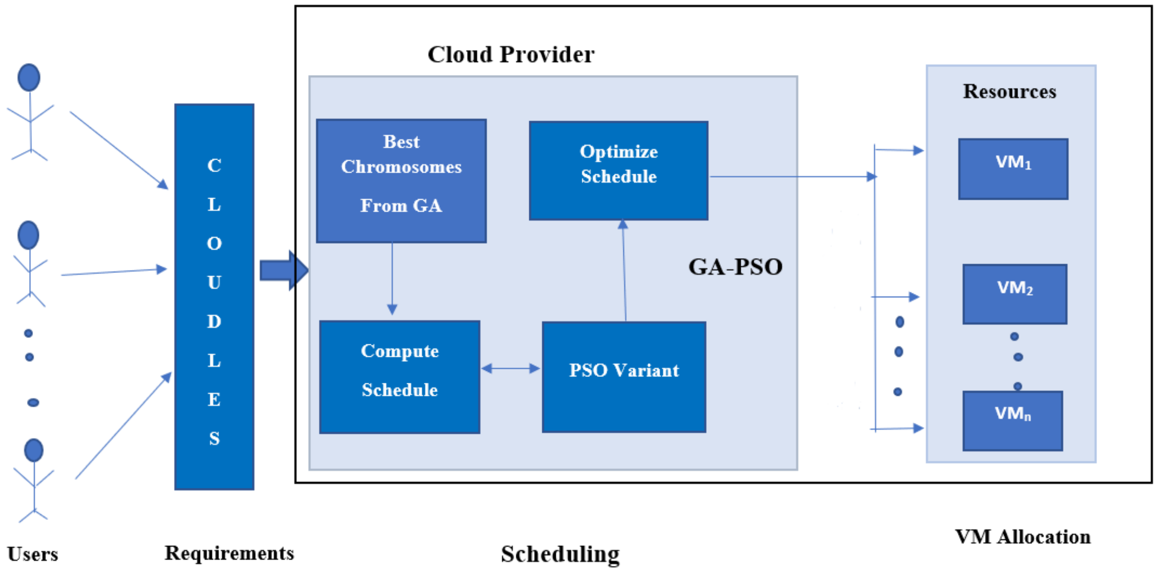
S. Singh presented numerous possibilities for labour scheduling within the Cloud computing setting in [12], providing a comprehensive understanding of GA.

He's designed a methodology for tackling work scheduling difficulties that uses the Max-Min method to build the initial population and offer the most trustworthy outcomes in terms of "makespan" by modifying GA. The authors describe the Cuckoo search-based paradigm for task performance in this work.

III. PROPOSED STATEMENT

Each tasks has deadline, along with penalty that must be paid if deadline is missed. Task Scheduling is a problem in cloudsim that reduced the makespan. So Scheduling of resources must be done in such a way that it reduces the total resource cost, time of execution, failure and improve Throughput. Meta heuristic algorithms find the best or near best solution in reasonable amount of time by making random choices to find the solution.

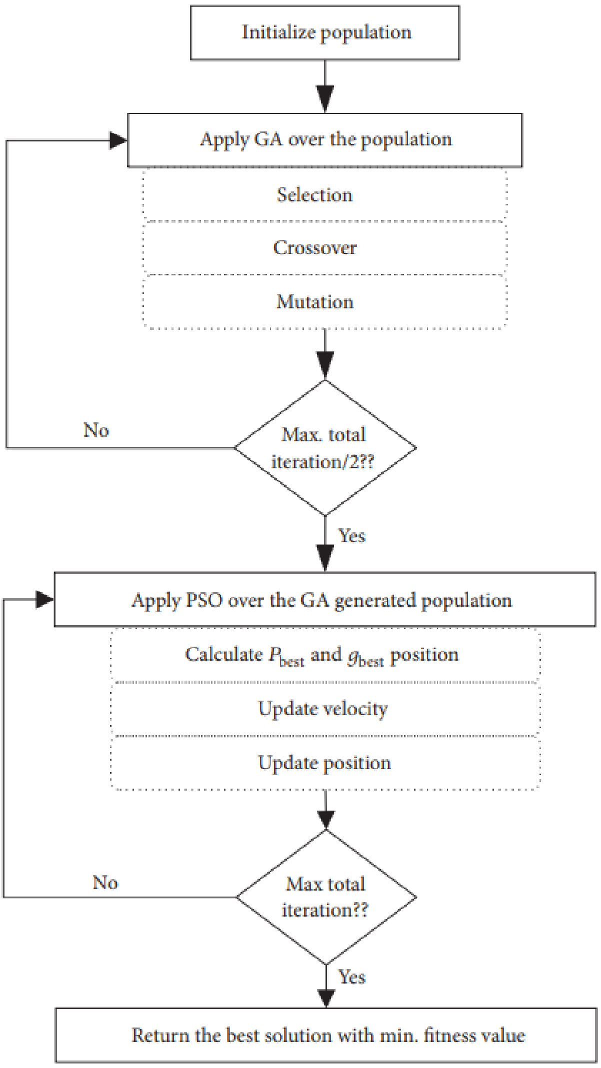
IV. PROPOSED MODEL



(Fig 1: Overall Architecture)

V. PROPOSED GA-PSO ALGORITHM

Diagram 2 illustrates the key steps in the GA-PSO algorithm. Randomly selected sample populations are generated in GA-PSO, and the number of iterations is specified. The workflow tasks in each solution are divided among VMs able to handle the workload, and the population represents alternative solutions to the workflow tasks problem. The GA algorithm is run (n/2) times with the adjusted population for the first half of an iteration if the number of iterations is (n). A simplified approach was adopted by employing an iteration of (n/2) as this has a tremendous effect on the GA technique in terms of chromosomes and particles as well as number of iterations.



(Fig 2: Workflow Of Proposed Algorithm Improved PSO)

With each iteration, GA algorithms improve the chromosomes (such as selection, crossover, and mutation) that they produce. In the second half of the prescribed rounds, the produced chromosomes are given to the PSO algorithm.

Chromosomes are referred to as particles in the PSO approach, and the particles improve with each repetition. The solution to the workflow task challenge is the particle with the lowest fitness value.

The maximum number of iterations attained is used to indicate the algorithm's termination criteria. When the conditions are completed, the workflow application's scheduling solution is given as the solution with the lowest fitness value in the population produced over the previous rounds.

Input: workflow W $\{N, E\}$ and set of resources $\{VM_1, VM_2, VM_3, \ldots, VM_j\}$
Output: g_{best} // the best solution to allocate W over VM_j
  For $i = 0$ to p
    population ← randomize() // initialize population, P is the population size
  End For
  While not Reach $n/2$ **do** // n is number of iterations
      While not Reach max P **do**
      $chromosom_j \leftarrow$ tournament(population) //selection operator
      $chromosome_i \leftarrow$ tournament(population)
      $offspring\_chromosome_j \leftarrow crossover(chromosome_j, chromosome_i)$
      $Newchromosome_j \leftarrow mutation(offspring\_chromosome_j)$
      Repeat
  Repeat
Set $Newchromosome_j$ as $particle_j$ // j is the index of the particles
Initialize particles position and velocity randomly
Calculate the (g_{best}) and (p_{best}) values
  While not Reach n **do**
  velocity matrix ← update($particle_j$ velocity)
  position matrix ← update($particle_j$ position)
  Repeat

IV. P PERFORMANCE EVALUATION

Beginning with a description of the experimental setting, this part gives an experimental assessment of the GA-PSO algorithm[14] on the PSO, and ACO algorithms.

A. The Experimental Environment

Cloudsim contains modelling and simulation tools for cloud computing infrastructures. Each cloudlet has its own set of properties, including file size and the amount of instructions that must be executed. These cloudlets will be supplied to the broker, who will organise them into VMs according to the timeline. The ability to set broker-driven rules is one of CloudSim's features. The virtual machine that may be constructed on the hosts is represented by the CloudSim class VM. By allocating each VM to its own host, the broker is in responsible of creating hosts. Brokers can adapt the number of hosts and virtual machines in a datacenter on the fly, and a datacenter can only hold a certain number of hosts.

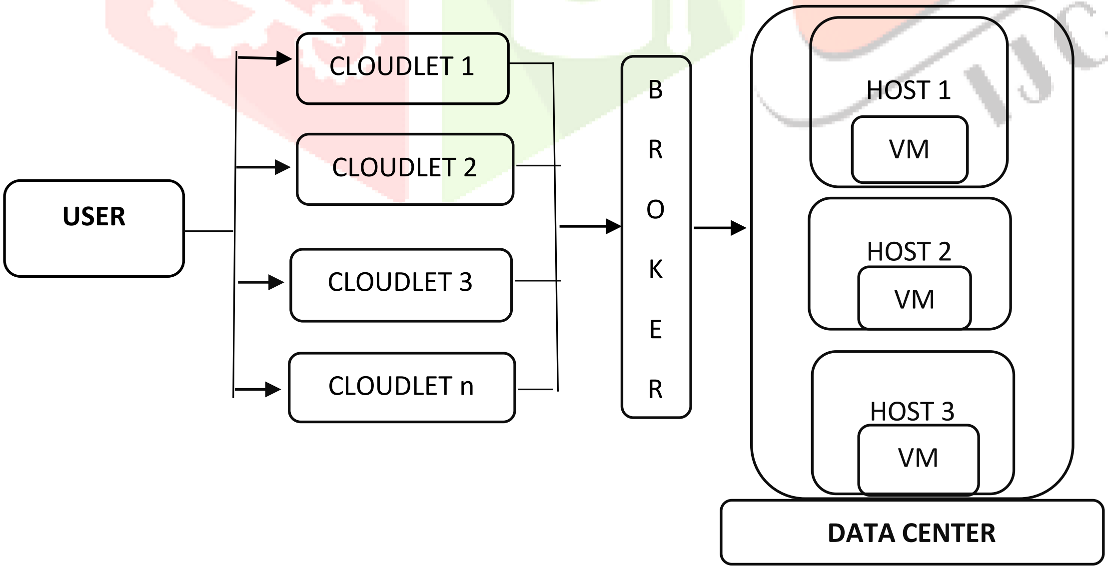


Fig. 3: CloudSim Behaviour

B. Experimental Results

The suggested GA-PSO was developed using the CloudSim toolkit, and a comparison study was conducted amongst four algorithms: PSO, ACO, and GA-PSO. To evaluate the performance, makespan , cost ,completion time and waiting time parameters are used.

Parameters	Value
Population Size	100
Mutation Rate	0.05
Crossover	single Point
No. of iterations	100
No. of Execution	500
C1	1
C2	1.1
r1, r2	[0,1]
α_1, α_2	0.4
α_3	0.2

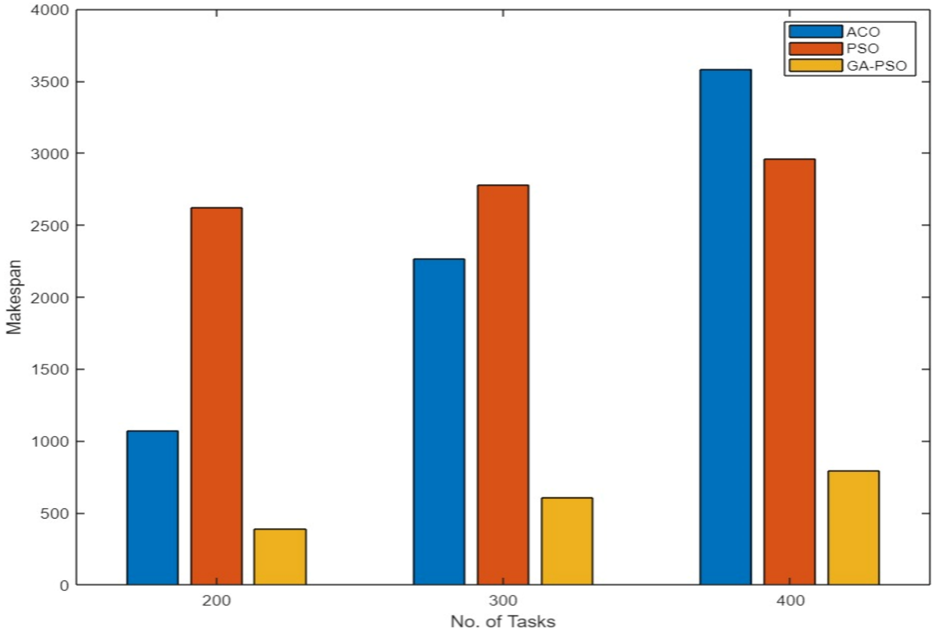
(Table1 : GA-PSO parameters)

Parameter	Value
No, of Tasks	200-500
No. of VMs	5
MIPS	250- 2500
RAM	256- 1024 (MB)
BW	250 – 2500 (mbps)
Processor Speed	10,000
No. of Processor	4
VM Policy	TIME_SHARED

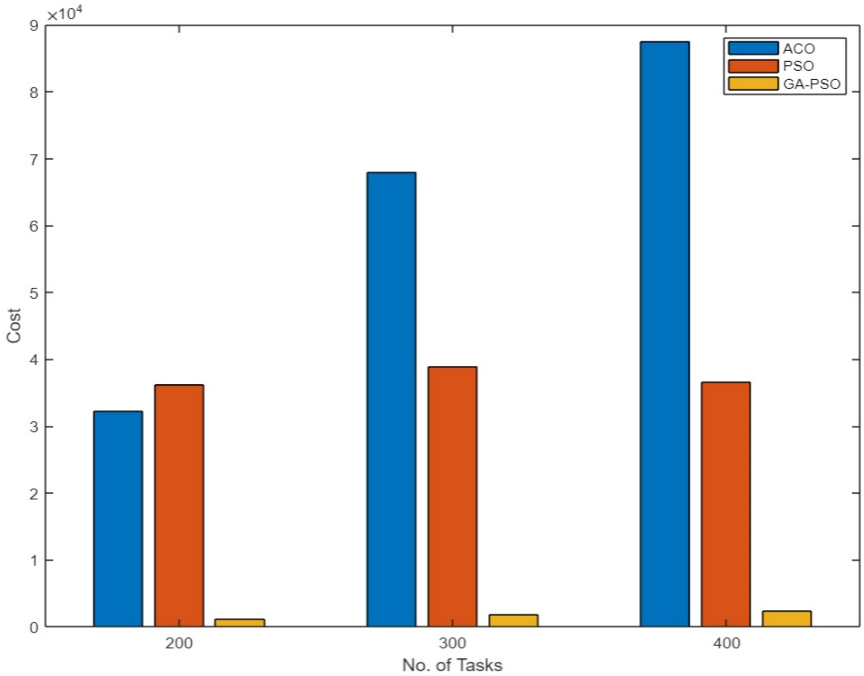
(Table 2: Parameters for simulation)

Tasks	Algorithms	Cost	Avg. Completion Time	Avg. Waiting Time	Makespan
200	ACO	32202.74	110.36	39.74	1073.41
	PSO	36256.51	2370.04	2309.61	2623.02
	GA-PSO	1163.133	193.21	191.22	387.81
300	ACO	67991.12	159.35	57.72	2266.37
	PSO	38926.00	2520.88	2477.63	2781.23
	GA-PSO	1828.03	300.95	298.91	609.44
400	ACO	87515.09	204.15	82.25	3583.83
	PSO	36601.50	2656.44	2625.02	2962.63
	GA-PSO	2387.06	398.32	396.33	795.78

(Table 3 : Parameters of different Task scheduling algorithms.)



(Fig 5: makespan of different task scheduling algorithms)



(Fig 6: Cost of different task scheduling algorithms)

V. CONCLUSION AND FUTURE WORK

Using GA-PSO, we present a task scheduling system for the cloud computing paradigm, in which customers are responsible for a varying range of computing assets and tasks. As a result, in a dynamic environment like cloud computing, making the most of the underlying property is crucial; otherwise, the cloud service provider may incur a big financial loss. For such optimization difficulties, the GA-PSO set of rules works better, and the outcomes are also tied to this fact, as GA-PSO-based project scheduling outperforms alternative project scheduling systems in terms of cost and makespan in comparison with PSO and ACO.

We'll look at the algorithm's applicability for Quality of Service (QoS) elements depending on user needs in future study.

VI. REFERENCES

1. R. Kaur and S. Kinger, "Enhanced Genetic Algorithm based Task Scheduling in Cloud Computing," International Journal of Computer Applications, vol. 101, 2014.
2. J. W. Ge and Y. S. Yuan, "Research of cloud computing task scheduling algorithm based on improved genetic algorithm," in Applied Mechanics and Materials, 2013, pp. 2426-2429.
3. Kokilavani, T., and DI George Amalarethinam. "Load balanced min-min algorithm for static meta-task scheduling in grid computing." International Journal of Computer Applications 20, no. 2 (2011):pp 43-49.
4. He, XiaoShan, XianHe Sun, and Gregor Von Laszewski. "QoS guided min-min heuristic for grid task scheduling." Journal of Computer Science and Technology 18, no. 4 (2003):pp 442-451
5. Sallim, Jamaludin, et al. "A Background Study on Ant Colony Optimization Metaheuristic and its Application Principles in Resolving Three Combinatorial Optimization Problem." (2007).
6. Tawfeek, Medhat A., Ashraf El-Sisi, Arabi E. Keshk, and Fawzy A. Torkey. "Cloud task scheduling based on ant colony optimization." In Computer Engineering & Systems (ICCES), 2013 8th International Conference on, pp. 64-69. IEEE, 2013.
7. Garg, Ritu. "Multi-Objective Ant Colony Optimization for Task Scheduling in Grid Computing." In Proceedings of the Third International Conference on Soft Computing for Problem Solving, pp. 133-141. Springer, New Delhi, 2014.
8. Keerthika, P., and P. Suresh. "A multi-constrained grid scheduling algorithm with load balancing and fault tolerance." The Scientific World Journal 2015 (2015).
9. S. Ravichandran and D. E. Naganathan, "Dynamic Scheduling of Data Using Genetic Algorithm in Cloud Computing," International Journal of Computing Algorithm, vol. 2, pp. 127-133, 2013.
10. V. Vignesh, K. Sendhil Kumar, and N. Jaisankar, "Resource management and scheduling in cloud environment," International Journal of Scientific and Research Publications, vol. 3, p. 1, 2013.
11. K. Thyagarajan, S. Vasu, and S. S. Harsha, "A Model for an Optimal Approach for Job Scheduling in Cloud Computing," in International Journal of Engineering Research and Technology, 2013.
12. S. Singh and M. Kalra, "Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014, pp. 565-569.
13. Agarwal, M., Srivastava, G.M.S.: A genetic algorithm inspired task scheduling in cloud computing. In: The Proceedings of 2nd IEEE Conference on Computing, Communication and Automation 2016 (2016).
14. Agarwal, M., Srivastava, G.M.S.: A cuckoo search algorithm-based task scheduling in cloud computing. In: Advances in Computer and Computational Sciences, pp. 293–299. Springer, Singapore (2018).
15. Abohamama, A.; Hamouda, E. A hybrid energy–Aware virtual machine placement algorithm for cloud environments. Expert Syst. Appl. 2020, 150, 113306.
16. Muteeh, A.; Sardaraz, M.; Tahir, M. MrLBA: Multi-resource load balancing algorithm for cloud computing using ant colony optimization. Cluster Comput. 2021.
17. Devaraj, A.F.S.; Elhoseny, M.; Dhanasekaran, S.; Lydia, E.L.; Shankar, K. Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments. J. Parallel Distrib. Comput. 2020.