Proceedings of the 8th Faculty of Science International Conference (FOSIC 2025), Delta State University, Abraka, Nigeria. 12th – 14th November, 2025. Pp. 113 - 126

A HYBRID OPTIMIZATION ALGORITHM (ANT COLONY AND GENETIC) FOR ADAPTIVE RESOURCE ALLOCATION IN CLOUD COMPUTING

Lesson W.A.[1] and Akazue M.I.[2]

[1] Computer Science Department, Delta State University, Abraka, Nigeria.
[2] Computer Science Department, Delta State University, Abraka, Nigeria.

ABSTRACT

Cloud environments are dynamic and heterogeneous, effective resource allocation remains a significant challenge in cloud computing. For adaptive resource allocation, this paper suggests a hybrid optimization technique that combines the Genetic technique (GA) with Ant Colony Optimization (ACO). The hybrid ACO-GA model combines the positive feedback mechanism and pheromone-based learning of ACO with the crossover and mutation operations of GA to balance exploration and exploitation during the search phase. In the suggested method, initial high-quality resource allocation paths are built using ACO, and these solutions are then refined by GA to avoid premature convergence and enhance global optimization performance. In order to minimize reaction time, maximize resource usage, and lower overall energy consumption in cloud data centers, the algorithm constantly adjusts to changes in workload. According to simulation results, the hybrid ACO-GA is a promising method for intelligent and adaptive resource management in cloud computing environments because it performs better than conventional standalone metaheuristic algorithms in terms of scalability, load balancing effectiveness, and convergence speed. The created approach is tested using various cloudlets and varying numbers of virtual machines. Cost, Makespan, Reliability, and Throughput are the four parameters that are compared to the fundamental ACO Algorithm. By utilizing machine learning and real-time data analytics, researchers can investigate improved cost optimization and flexible resource allocation techniques in cloud computing. In the end, this would lower operating costs and boost performance by enabling systems to dynamically modify resource utilization in response to demand, workload patterns, and pricing fluctuations.

INTRODUCTION

Instead of depending on local infrastructure or personal devices, people and businesses now need access to computer resources including servers, storage, databases, networking, software, and analytics via the internet. (Obidike *et al*, 2025). Cloud computing has drastically changed how businesses and individuals use computer resources. Without needing an initial financial outlay, it offers on-demand supply of computer resources, including as memory, internet, apps, and services. Because of its cost, scalability, and adaptability, cloud computing is becoming more and more popular in a number of areas, including e-commerce, finance, and healthcare (Barath *et al*, 2023). However, in order to maximize resource utilization while upholding service-level agreements and improving energy efficiency, cloud computing necessitates efficient resource allocation (Bohn *et al*, 2022). The process of arranging computer resources, such as a processor, storage, and backup of several items and programs, in a way that maximizes efficiency and minimizes energy consumption is known as resource allocation in cloud computing (Ijaz et al, 2020). Because cloud computing operations are dynamic, resources are diverse, and activities and apps have different priorities, resource allocation is a complicated problem (Senthilkumar et al., 2023). To solve this issue, researchers have put up a number of methods for allocating resources in cloud computing.

In cloud computing, resource allocation is the process of allocating suitable resources to effectively complete activities requested by customers (Manzoor et al, 2020). This means allocating virtual machines with the properties that the clients have chosen. Jobs submitted by users may take a variety of lengths of time to finish. Other elements of effective cloud resource use include workload management and virtualization allocation (Pingulkar et al, 2023). When choosing when to start or complete a computer activity, it's crucial to take into account a number of factors, including the distribution of resources, the amount of time spent, the actions of successors, and the linkages with previous jobs (Nagamani et al, 2019). Additionally, pooling available resources, selecting the finest resources, providing them, planning their use, and managing resources collectively are all included in resource allocation (Edavalath et al, 2023).

In the domains of information and communication technology (ICT), a paradigm known as "cloud computing" (CC) has taken center stage in recent years. Even if cloud innovation directly or indirectly supports the daily search service through Internet activities, cloud clients may not always be aware of its benefits. Cloud computing has become a more common phrase for communication due to its significance in the computer and engineering industries.

Services for cloud computing unfettered accessibility enables developing and underprivileged nations to make rapid economic progress. It was challenging for a company to build a traditional data center prior to the era of cloud innovation due to the expensive initial equipment investment and continuing maintenance expenditures. On the other hand, cloud computing makes it simple to deploy programs by enabling customers to rent computer resources as needed. Optimizing CC innovation services is essential since so many businesses, big and small, are searching for methods to reduce expenses without compromising effectiveness. This is due to the fact that these services provide numerous advantages that are directly related to what companies require.

When cloud computing performance approaches are based on a utility-based commercial model, users can readily, reliably, and scalably access a common pool of programmable network assets (Jeyaraman et al., 2024).

Cloud service providers are attracted to CC because they can serve clients and consumers, including entities and financial institutions, by reducing or eliminating infrastructure running costs. Clients demand service providers to ensure the security of extremely sensitive cloud-based enterprise applications. A warranty between the provider and the consumer is often provided through a service-level contract. Virtualization has been more popular recently as a means of boosting the effectiveness of cloud networks (Shukur et al, 2020).

Allocating resources to the live process of computing demands is one of the most crucial responsibilities. The research proposes an optimal resource allocation paradigm as well as a cost-based resource allocation technique for the heterogeneous cloud environment.

MATERIALS AND METHODS

Adaptation of ACOGA Hybrid Algorithm

The Ant Colony Optimization (ACO) and Genetic Algorithm (GA) are two meta-heuristic algorithms that were hybridized in the study to create ACOGA. In the cloud context, the ACOGA tends to enhance resource allocation cost optimization.

There are two primary stages to the ACOGA: the first is when ACO is applied, and the second is when GA is applied. The ACO algorithm was optimized by taking inspiration from the natural world. For example, some types of ants initially travel at random before returning to their colony when they discover food, leaving pheromone trails in their wake. If more ants find this route, they will probably stop moving at random and instead follow the path, coming back and strengthening it if they eventually find food. But the pheromone trail gradually disappears, making it less appealing. The pheromones have more time to dissolve the longer an ant walks down the track and back. The pheromone density is higher on shorter paths than on longer ones because a short path is marched over more frequently. Another advantage of pheromone evaporation is that it prevents convergence to a locally optimal solution. The tracks left by the first ants would be too tempting for the subsequent ones if there was no evaporation. The study of the solution space would be constrained in that scenario. Pheromone evaporation is crucial in artificial systems, but its function in real ant systems is unclear. In general, other ants are more likely to follow suit when one finds an advantageous (i.e., quick) route from the colony to a food source, and positive feedback eventually results in a large number of ants taking the same route. By sending "simulated ants" around a graph that depicts the issue to be solved, the ant colony algorithm seeks to mimic this behavior.

GA implementation is part of the second Phase. Using biologically inspired operators like selection, crossover, and mutation, genetic algorithms are frequently used to create excellent solutions to optimization and search issues. Usually starting from a population of randomly generated individuals, the evolution process is iterative, with each iteration being called a "generation." Fitness is usually the value of the objective function in the optimization problem being addressed, and each member of the population is assessed in every generation. To create a new generation, the fit people are randomly selected from the current population, and their genomes are modified (recombined and occasionally changed at random). The subsequent iteration of the algorithm uses the newly created candidate solutions. Usually, the process ends when the population reaches a desired level of fitness or when the maximum number of generations has been produced. By improving the solutions found in the initial stage of the ACOGA application, this method was used to improve resource allocation and cost optimization in the cloud environment.

Design of the ACOGA Hybrid Algorithm

This study developed a system of a hybrid algorithm called ACOGA which combines two metaheuristic algorithms (ACO and GA) for resource allocation and optimization in the cloud environment. This study tends to develop a model that resolves the problem of slow and premature convergence, especially in complex and multi-modal environments. The cost effect is also put in consideration as the developed model reduces the cost the cloud environment and maintaining adaptability to changes. ACO and GA are two metaheuristic algorithms combined to create the hybrid algorithm. As shown in Figure 4.1, the number of tasks and VMs are set then the pheromone where also initialized. The ACO and GA are run for the number of iterations to get the best solution for the tasks and available VMs.

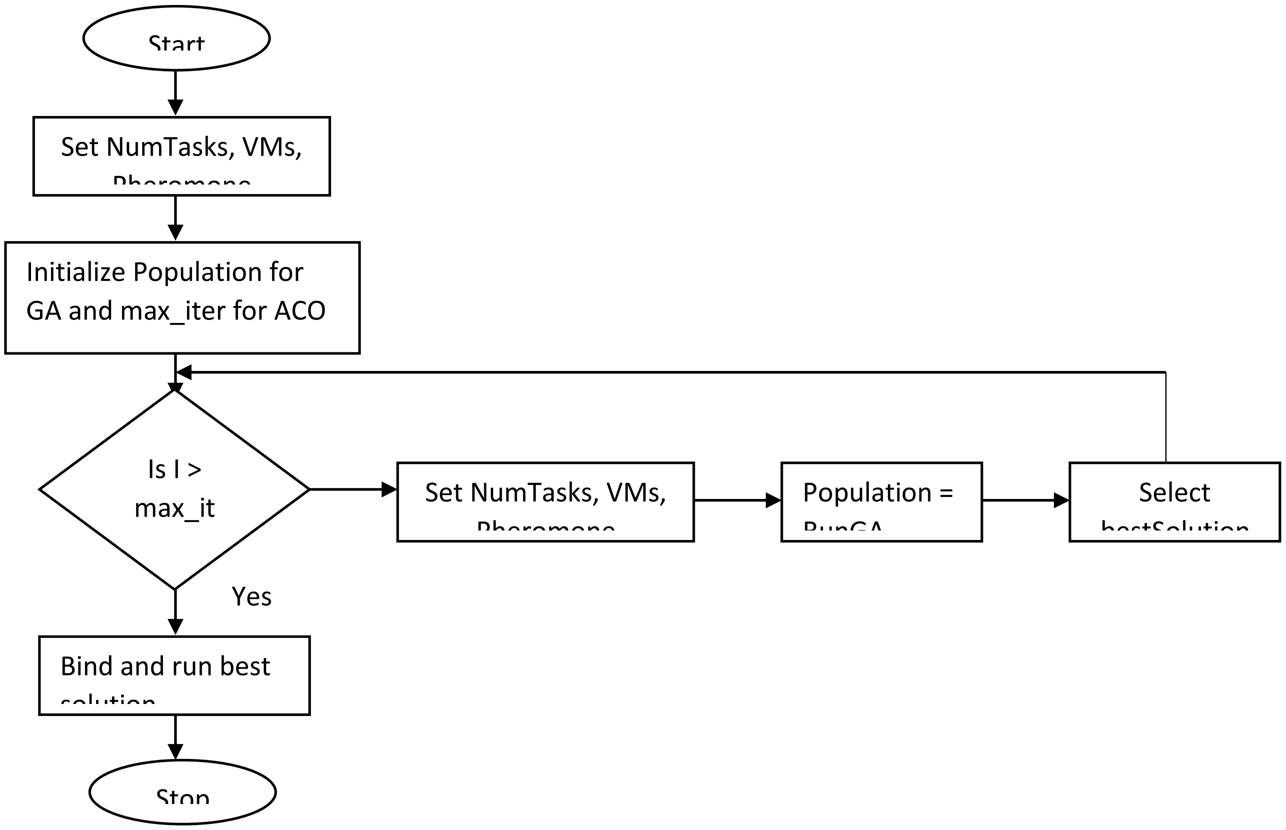

Figure 1: Flowchart of the proposed system

Algorithm 1: ACOGA

```
1.  Start
2.  Initialize Pheromone
3.  Initialize Population
4.  Define Max Iteration m
5.  For i=1 to m do
6.      Add ants to population
7.      Run Genetic Algorithm
8.      Select solution
```

```
9.    Update Pheromone
10. Bind and run best solution
11. Stop
```

Experimental Setup

For the simulation, a 2.50GHz Core i5 laptop running 64-bit Windows 11 with 8GB RAM was utilized. The cloud computing environment is simulated in this evaluation using the popular CloudSim toolkit and the Java programming language. The primary classes needed to construct the cloud, including the host, cloudlet, and data center classes, are provided by CloudSim. Ten ants were utilized in the ACO Algorithm, and the maximum iteration was set at fifty. Next, a population of 20 was used for the GA. For the stop condition, the convergence of the hybrid model was considered. The simulations were conducted in a variety of environments. Table 1 lists the cloudsim setups.

Table 1: CloudSim Configurations

Data Center	Number of Data Centers	1
Host	Number of Host	2
	PES	4
	MIPS	6000
	RAM	20 GB
	Bandwidth	10 GB
	Storage	1 TB
Virtual Machine	Number of VMs	5 – 10 - 15
	MIPS	1000 - 5000
	RAM	1 GB to 5 GB
	Bandwidth	100 MB to 500 MB
	Storage	10 GB
Cloudlets	Number of Cloudlets	50 – 100 - 200
	Length	3000 - 10000
	Type	Heterogenous
	Submission time	Poisson distribution of parameters

Performance Metrics

The finished job is evaluated using four performance metrics. These are throughput, cost, makespan, and reliability.

Cost: The total cost of executing a task (or set of tasks) is determined by multiple resource usage factors and their associated unit pricing. These factors typically include CPU, memory, storage, and bandwidth. Cost is represented mathematically as shown in Equation below.

$$\sum_{i=1}^{m}(ET_{ij} \times CPS) + (M_{ij} \times CPM) + (S_{ij} \times CPM) + (B_{ij} \times CPM)$$

Where i is the id of Virtual Machine (VM) from 1 to m and j is the number of task from 1 to n. ET is the Execution time, CPS is the Cost per second, M is the memory cost, CPM is the Cost per memory, S is the storage cost, and B is the bandwidth cost.

Makespan: It shows the interval of time between the first task's submission and the results' receipt (the point at which the last job was finished). seen as the overall amount of time needed to finish every task. The size and complexity of the jobs, the quantity of resources available, the system load, and the task scheduling algorithms are some of the variables that affect it. In order to maximize resource utilization and improve cloud data center performance, it is essential to reduce the makespan. The equation below is a mathematical representation of makespan.

$$makespan = \max(ETv_i) \quad \forall\, i \in 1,2,3, \ \ldots .m$$

The execution time of all tasks allocated to the ith VM is denoted by $ETvi$, as explained below.

$$ETv_i = \sum_{j=1}^{m} di,j \times M_{i,j}$$

Reliability: it represents the probability that a task will be executed successfully, without any resource failure. Our model for measuring reliability is based on a failure rate which is an intrinsic property of the resource. It is determined as follows.

$$Reliability = exp^{-\sum_{i=1}^{n} TE(T_i) * \gamma_j}$$

Where TE(Ti) is the task's execution time. The machine performing the work has a failure rate of Ti and λj.

Throughput: It refers to the total number of jobs completed within a specified makespan, can be computed as

$$Throughput = \frac{\sum M_{I,J}}{makespan}$$

Where $\Sigma M_{i,j}$ is the number of successfully completed task.

RESULTS

In this section, the results shown is gotten after simulations. The developed algorithm is experimented with different number of VMs and different cloudlets. The 4 metrics (Cost, Makespan, Reliability and Throughput) are used to compare with the basic ACO Algorithm.

Cost

Looking at Figure 4.2 where the both algorithms are run on 5 Virtual Machines, for 50 cloudlets the cost of ACO and ACOGA are 4,251 and 21,980 respectively. For 100 cloudlets the cost of ACO and ACOGA are 14,693 and 46,960 respectively while for 200 cloudlets the cost for ACO and ACOGA were 41,386 and 105,920 respectively.

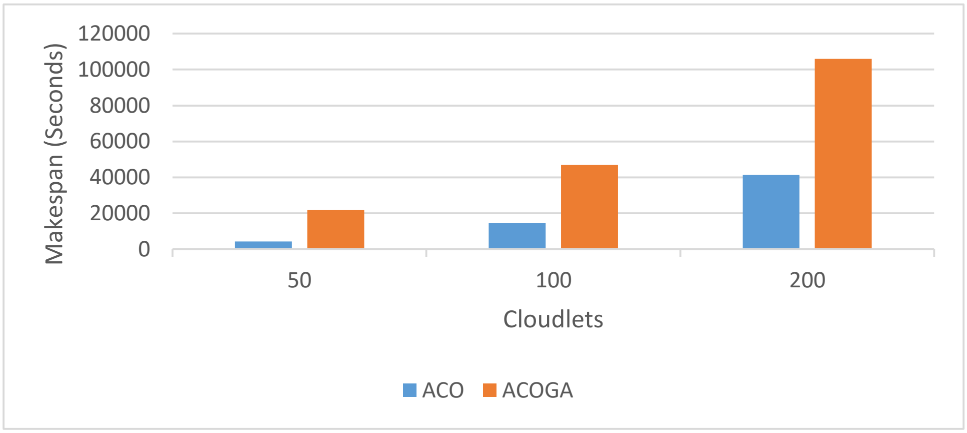

Figure 2: Comparison in terms of cost for 5 VMs

Looking at Figure 3 where the both algorithms are run on 10 Virtual Machines, for 50 cloudlets the cost of ACO and ACOGA are 3,750 and 21,230 respectively. For 100 cloudlets the cost of ACO and ACOGA are 10,879 and 43,960 respectively while for 200 cloudlets the cost for ACO and ACOGA were 21,214 and 93,920 respectively.

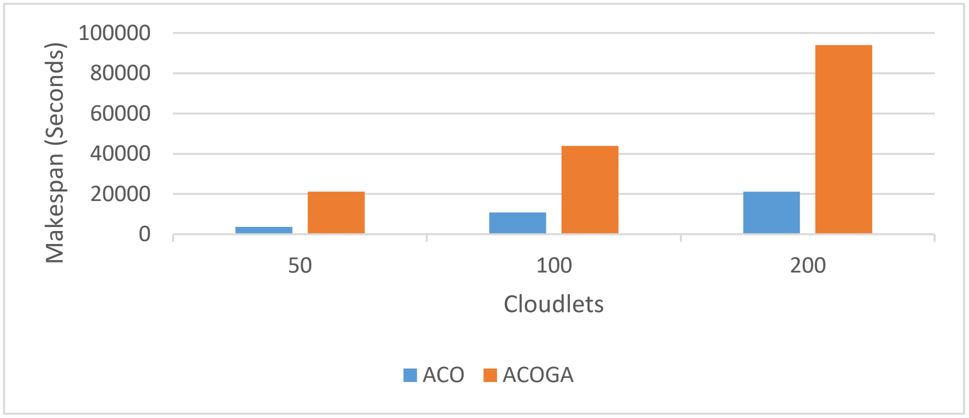

Figure 3: Comparison in terms of cost 10 VMs

Looking at Figure 4 where both algorithms are run on 10 Virtual Machines, for 50 cloudlets the cost of ACO and ACOGA are 3,369 and 20,990 respectively. For 100 cloudlets the cost of ACO and ACOGA are 7,284 and 42,970 respectively while for 200 cloudlets the cost for ACO and ACOGA were 14,640 and 89,924 respectively.

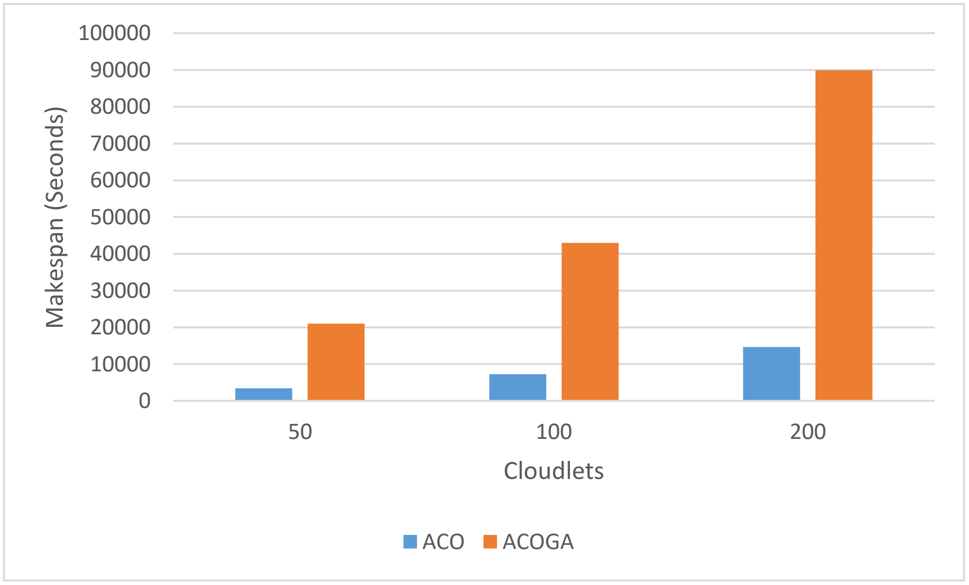

Figure 4: Comparison in terms of cost 15 VMs

Makespan

Looking at Figure 5, where the both algorithms are run on 5 Virtual Machines, for 50 cloudlets the makespan of ACO and

ACOGA are 321 and 101 respectively. For 100 cloudlets the makespan of ACO and ACOGA are 880 and 201 respectively while for 200 cloudlets the makespan for ACO and ACOGA were 1560 and 410.

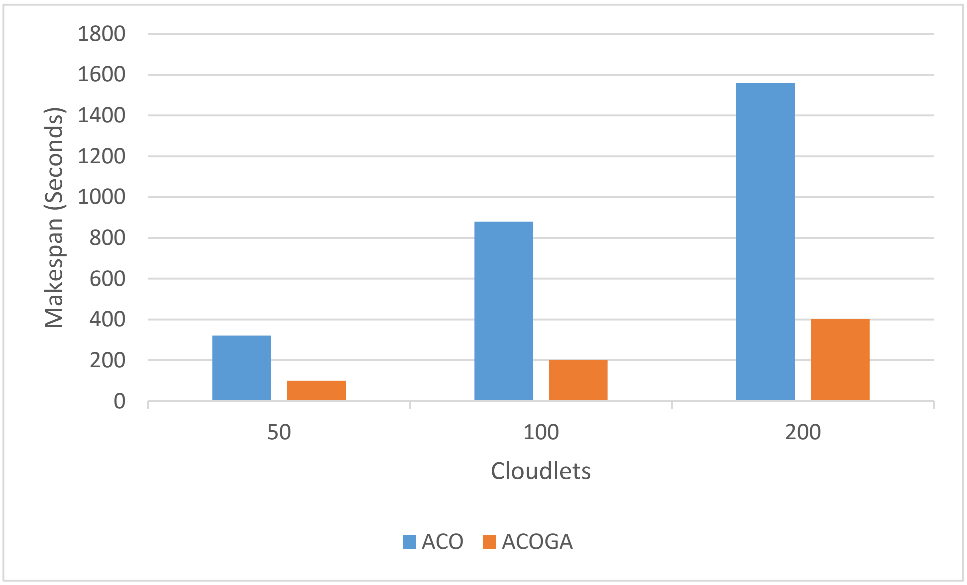

Figure 5: Comparison in terms of Makespan 5 VMs

Looking at Figure 6, where the both algorithms are run on 10 Virtual Machines, for 50 cloudlets the makespan of ACO and ACOGA are 240 and 51 respectively. For 100 cloudlets the makespan of ACO and ACOGA are 720 and 101 respectively while for 200 cloudlets the makespan for ACO and ACOGA were 880 and 201.

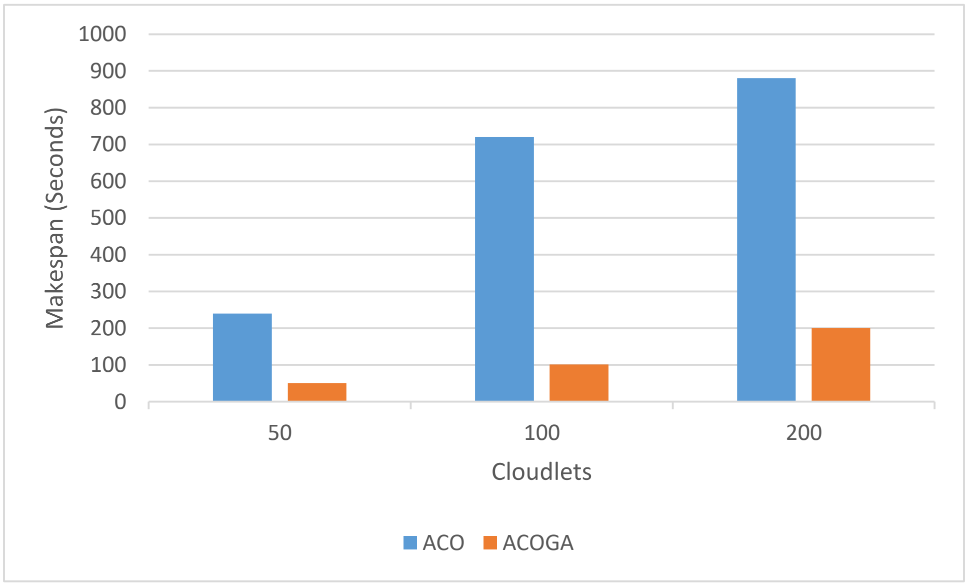

Figure 6: Comparison in terms of Makespan 10 VMs

Looking at Figure 7 where the both algorithms are run on 15 Virtual Machines, for 50 cloudlets the makespan of ACO and ACOGA are 201 and 41 respectively. For 100 cloudlets the makespan of ACO and ACOGA are 321 and 71 respectively while for 200 cloudlets the makespan for ACO and ACOGA were 480 and 141.

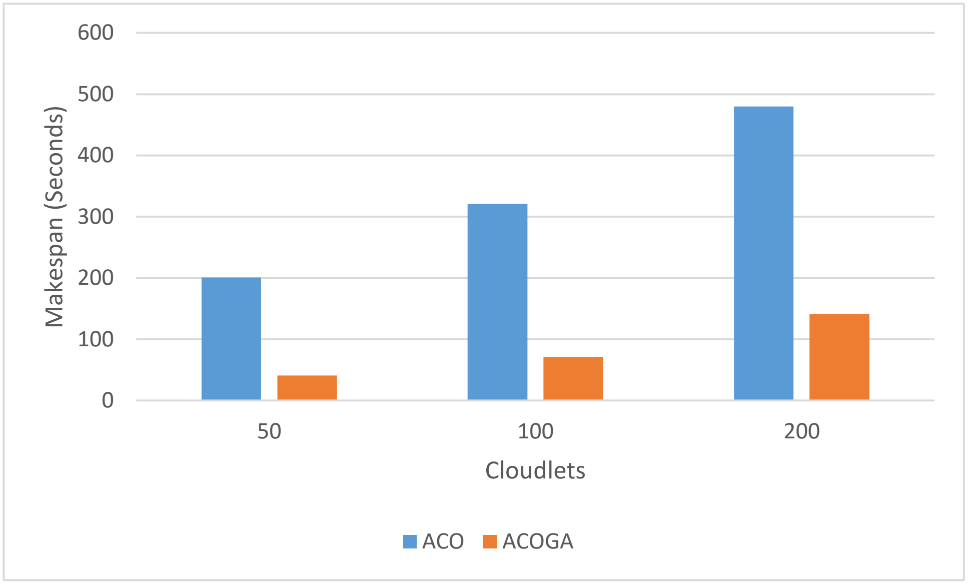

Figure 7: Comparison in terms of Makespan 15 VMs

Reliability

Looking at Figure 8 where the both algorithms are run on 5 Virtual Machines, for 50,100 and 200 cloudlets the reliability of ACO is 0.30, 0.38 and 0.35 respectively while the developed model maintained a reliability of 1 for all instances.

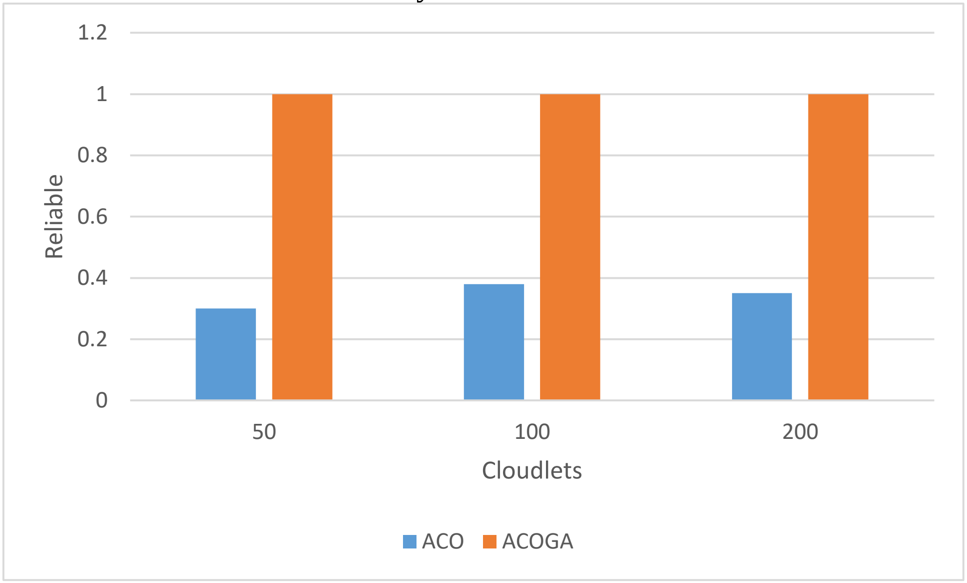

Figure 8: Comparison in terms of Reliability 5 VMs

Looking at Figure 9 where the both algorithms are run on 5 Virtual Machines, for 50,100 and 200 cloudlets the reliability of maintained a reliability of 1 for all instances. ACO is 0.24, 0.28 and 0.20 respectively while the developed model

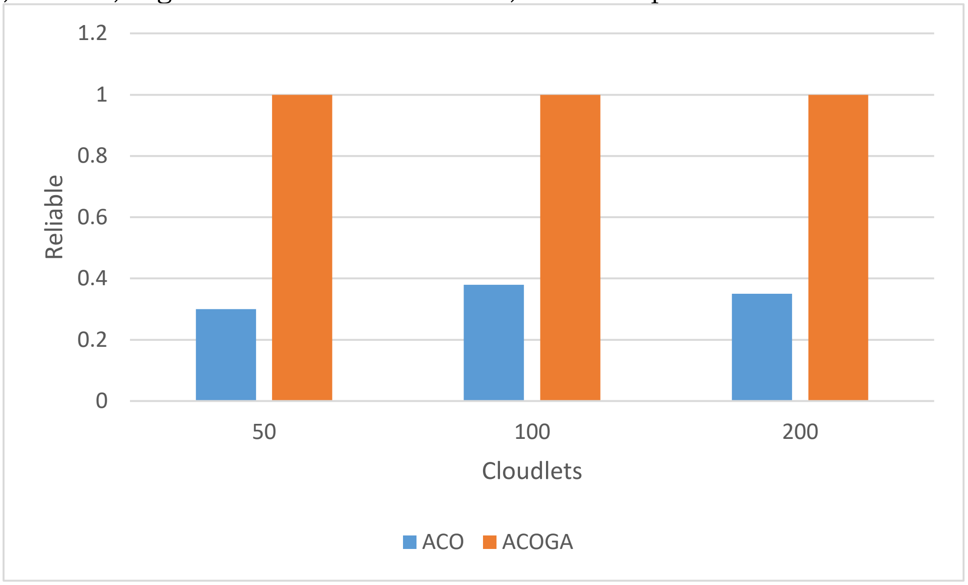

Figure 9: Comparison in terms of Reliability for 10 VMs

Looking at Figure 10 where the both algorithms are run on 5 Virtual Machines, for 50,100 and 200 cloudlets the reliability of ACO is 0.18, 0.16 and 0.12 respectively while the developed model maintained a reliability of 1 for all instances.

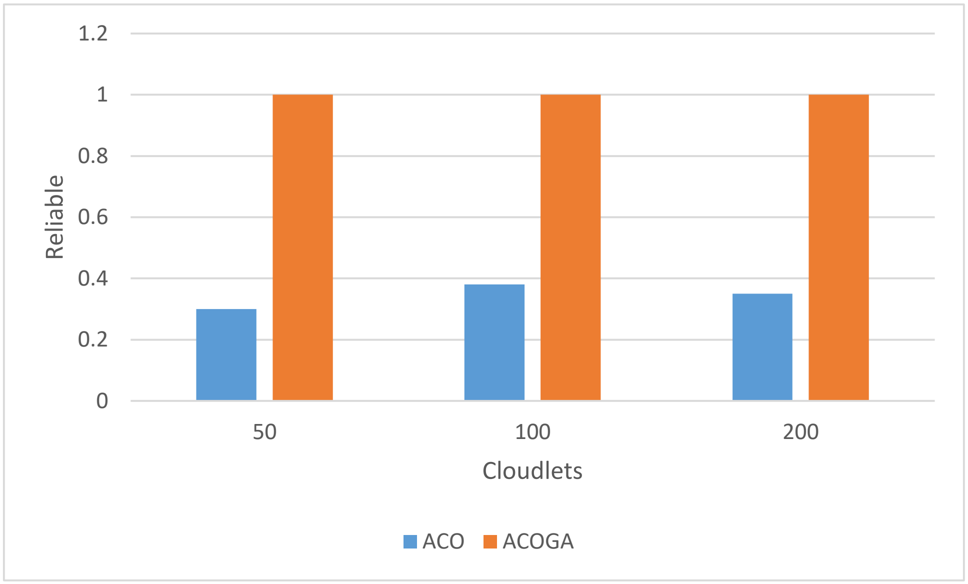

Figure 10: Comparison in terms of Reliability for 15 VMs

Throughput

Looking at Figure 11 where the both algorithms are run on 5 Virtual Machines, for 50 cloudlets the throughput of ACO and ACOGA are 0.47 and 4.95 respectively. For 100 cloudlets the throughput of ACO and ACOGA are 0.43 and 4.98 respectively while for 200 cloudlets the throughput for ACO and ACOGA were 0.45 and 4.99.

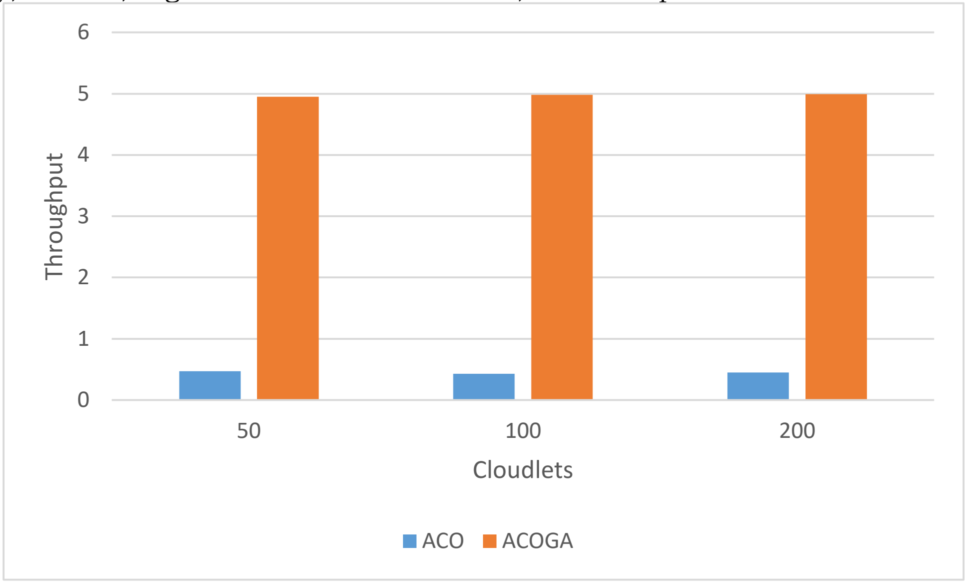

Figure 11: Comparison in terms of throughput for 5 VMs

Looking at Figure 12 where the both algorithms are run on 10 Virtual Machines, for 50 cloudlets the throughput of ACO and ACOGA are 0.50 and 9.80 respectively. For 100 cloudlets the throughput of ACO and ACOGA are 0.38 and 9.90 respectively while for 200 cloudlets the throughput for ACO and ACOGA were 0.45 and 9.95.

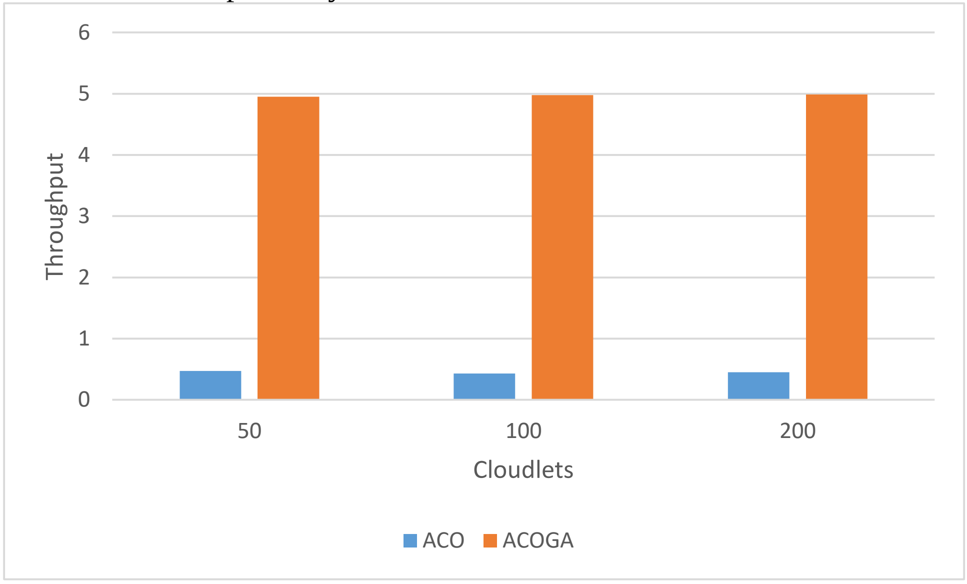

Figure 12: Comparison in terms of throughput for 10 VMs

Looking at Figure 13, where the both algorithms are run on 15 Virtual Machines, for 50 cloudlets the throughput of ACO and ACOGA are 0.45 and 12.2 respectively. For 100 cloudlets the throughput of ACO and ACOGA are 0.50 and 12.2 respectively while for 200 cloudlets the throughput for ACO and ACOGA were 0.49 and 14.18.

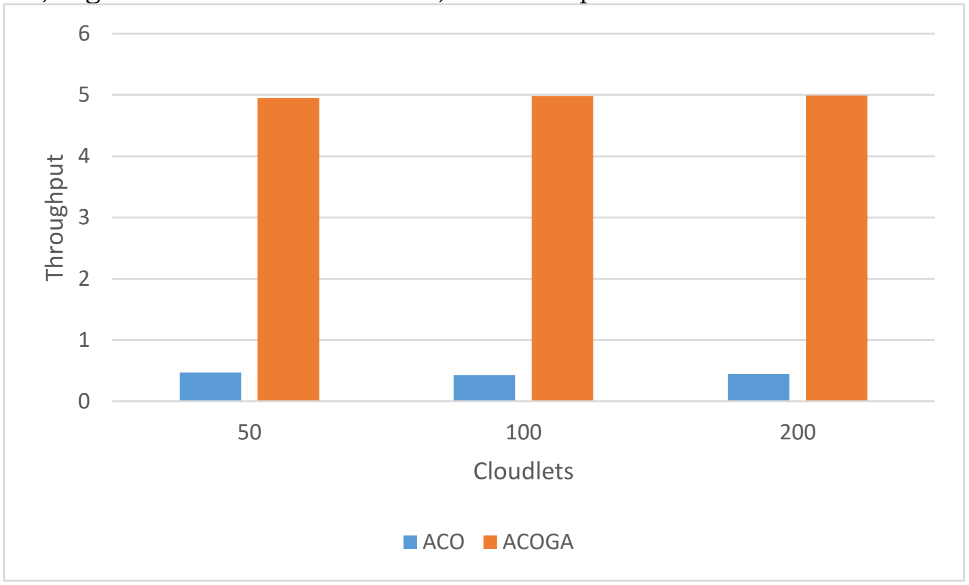

Figure 13: Comparison in terms of throughput for 15 VMs

DISCUSSION

To address the issue of resource allocation and cost optimization in the cloud environment, numerous academics have put forth various optimization algorithms. The current research has certain limitations, such as Lu, W. (2020); Kniazhyk et al. (2023); the optimization method employed in these studies was Ant Colony Optimization (ACO). Slow convergence is a common problem for this ACO, particularly in big or dynamic cloud environments.

In complex or large-scale resource landscapes, the algorithm may prematurely converge to a suboptimal solution (local minimum), particularly if pheromone trails reinforce poor paths early. This leads to poor quality scheduling decisions. Another limitation of the current systems was observed in Lu, W. (2020) and Kniazhyk et al. (2023) works because cost, reliability, and throughput were not taken into consideration. In this research, the hybridization of Ant Colony Optimzation and Genetic Algorithm was developed to address these issues.

CONCLUSION

Faster convergence is provided by the hybrid paradigm, particularly in big or dynamic cloud environments. Response time and resource utilization efficiency will both rise as a result. In addition to increasing computing efficiency, improving the ACO with GA will assist avoid premature convergence, which occurs when the algorithm becomes stuck at local optima. The hybrid model is appropriate for cloud environments due to its adaptability. The hybrid model is economical.

REFERENCES

Barath Tej. R.R.V.S.S. (2023). A Comprehensive Study of Resource Allocation in Cloud Computing Environments. International Journal of Research Publication and Reviews, 4(5): 2035-2038.

Bohn, R. B., Chaparadza, R., Elkotob M. and Choi, T. (2022). The Path to Cloud Federation through Standardization. International Conference on Information and Communication Technology Convergence, Korea: 942-946. https://doi.org/10.1109/ICTC55196.2022.9952660

Edavalath, S., and Sundaram, M. S. (2023). M.A.R.C.R.: Method of allocating resources based on the cost of the resources in a heterogeneous cloud environment. The Scientific Temper, 14(3): 576-581. https://doi.org/10.58414/ SCIENTIFICTEMPER.2023.14.3.03

Edavalath, S., and Sundaram, M. S. (2023). *Cost-based resource allocation method for efficient allocation of resources in a heterogeneous cloud environment*. *14*, 1339–1344. https://doi.org/10.58414/SCIENTIFICTEMPER.2023.14.4.41

Edje E. A., and Muhammad. L. S. (2020) "Cloud Computing Enabled Data Center Infrastructure Development and Deployment by IT Firms." International Journal of Computing and Digital Systems, 9(1), 37+

Edje E. A. (2021) "Enhanced non-parametric sequence learning shceme for internet of things sensory data in cloud infrastructure" Doctoral dissertation, Universiti Teknologi Malaysia.

Ijaz, S., Safdar, T. and Khan, A.(2020). Challenges and Limitations of Resource Allocation in Cloud Computing. Springer International Conference on Intelligent Technologies and Applications, Intelligent Technologies and Applications: 723-737. https://doi.org/10.1007/978-981-15-5232-8_62

Jeyaraman, J., Bayani, S. V., and Arasu, J. N. (2024). *Optimizing Resource Allocation in Cloud Computing Using Machine Learning Jawaharbabu Jeyaraman , Samir Vinayak Bayani and Jesu Narkarunai Optimizing Resource Allocation in Cloud Computing Using Machine Learning*.

Kniazhyk, T., and Muliarevych, O. (2023). *Cloud Computing With Resource Allocation Based On Ant Colony Optimization*. *8*(2).

Lu, W. (2020). *Resource Allocation Algorithm of Cloud Computing Infrastructure Services based on Continuous Optimization Algorithm Resource Allocation Algorithm of Cloud Computing Infrastructure Services based on Continuous Optimization*. https://doi.org/10.1088/1757-899X/750/1/012204

Manzoor, M. F., Abid, A., Farooq, M. S., Nawaz, N. A. and Farooq. U. (2020). Resource Allocation Techniques in Cloud Computing: A Review and Future Directions. Elektronika Ir Elektrotechnika, 26(6): 40-51. http://dx.doi.org/10.5755/j01.eie.26.6.25865

Nagamani, T.S., Lakshmi, V K N V S K and Bhavani, B. L. (2019). A new dynamic and enhanced resource allocation algorithm in cloud computing. International conference on computer vision and machine learning, I.O.P. Conf. Series: Journal of Physics, 1228: 1-8. https://doi.org/10.1088/1742 6596/1228/1/012033

Obidike, C. A., Edje, E. A., and Fasanmi, E. A., (2025). Multi Schemes For Dynamic Taskscheduling In Cloud Computing. June. https://doi.org/10.4314/sa.v24i2.9

Pingulkar, S., Tiwary, A. and Tyagi, S. (2023). Resource Allocation Techniques in Cloud Computing: A Comprehensive Review. International Journal of Engineering Research & Technology, 12(7): 350-357.

Senthilkumar, G., Tamilarasi, K., Velmurugan, N. and J. K. Periasamy. (2023). Resource Allocation in Cloud

Computing. Journal of Advances in Information Technology, 14(5): 1063-1073. https:// doi.org/10.12720/jait.14.5.1063-1072

Shukur, H., Zeebaree, S., Zebari, R., Zeebaree, D., Ahmed, O. and Salih, A. (2020). Cloud computing virtualization of resources allocation for distributed systems. J. Appl. Res. Technol. Tren., 1, 3, 98-105.