Comparative Analysis of Dynamic Cost-Time Allocation Techniques in Cloud Environments

Monika Bishnoi[1*], *Neha* Dutta[2]

[1] Department of CSE Chandigarh University Mohali, India
[2] Department of CSE Chandigarh University Mohali, India

Abstract. The cloud computing environment requires effective cloud resource optimization to manage the cost of execution and the time of completion and deal with workloads that may vary. Dynamic scheduling is necessary to maximize system performance due to the fact that traditional approaches tend to be static and are not dynamic enough to address dynamic situations. This paper introduces comparative research on dynamic cost-time allocation methods in the cloud setups with reference to three well known optimization strategies which include Teaching Learning Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). The evaluation framework that has been proposed evaluates these algorithms under varying workloads based on the efficiency of task allocation, scalability and convergence behaviour. The outcomes of the experiments indicate the strong and weak aspects of each method and show that TLBO is more effective in converging much faster with less variance, PSO is effective in highly adaptable resources, and GA is effective in orderly exploration of complex allocation settings. The results will help to evolve the cost-time allocation strategies in cloud computing, providing useful information to the researchers and practitioners who can create intelligent, adaptive and resource-efficient scheduling solutions.

1 Introduction

Cloud computing has become a disruptive model in the contemporary computing which offers on-demand access to common pools of configurable resources including computing power, storage and applications.[1,2] As big data, artificial intelligence and business analytics continue to expand, more companies are turning to cloud-based environments to workloads to manage.[3,4] Nevertheless, the mobility of cloud resources and the heterogeneity of the needs of users pose a challenge in ensuring cost-efficiency and the ability to execute tasks on time.[5] Resource allocation is thus one vital attribute of the dependability and effectiveness of service delivery in cloud systems.[6] Cost and time are two very important performance indicators in cloud environments that are mutually exclusive in many cases.[7] Users like to be provided with reduced cost of computation and at the same time, reduced time to execute their tasks. This trade-off renders allocation

* Corresponding author: monikabishnoi3@gmail.in

of cost of time as a key issue in the management of cloud resources.[8] To resolve these issues, dynamic allocation methods that change with varying workloads and resources of heterogeneous nature have been considered more frequently.[9] The proper management of execution time vs. resources cost will be an effective strategy that leads to the achievement of user satisfaction and the profitability of the cloud service provider.[10] The old- fashioned static methods of scheduling presuppose deterministic task demand and resource availability which is not the case in the real-life.[11] With rapid changes in workloads, the static models do not achieve adaptability and the result is either underutilization of the resources, increased latency, or exaggerated costs.[12] Conversely, dynamic scheduling dynamically assigns tasks as conditions evolve in the system and has more efficient utilization of cloud infrastructure. That flexibility renders dynamic scheduling methods more appropriate to large-scale, heterogeneous and uncertain cloud environments.[13] Metaheuristic and evolutionary algorithms are popular to provide the efficient optimization of cloud resources. These algorithms give near-optimal solution to multicomponent, multi-objective problems like minimization of cost and execution time together.[14] Some of the most researched methods include Teaching learning based optimization (TLBO), Particle Swarm Optimization (PSO) as well as Genetic Algorithm (GA).[15] The algorithms have different trade-offs of speed of convergence, accuracy of solutions and flexibility, with each algorithm using distinct strategies to explore and exploit the search space.[16] TLBO is based on the process of teaching and learning in the classroom where solutions are developed during teacher and learner phases and no algorithm specific parameters are required, making the method computationally efficient.[17] PSO is a simulated approach to social behavior in flocks of birds or schools of fish, which is highly versatile and continually optimizing.[18] GA is based upon Darwinian laws of natural selection and genetics and is very efficient in exploring large search spaces, but can be prone to premature convergence unless adjusted effectively.[19] When these algorithms are compared in the dynamic scheduling environment, it becomes possible to draw meaningful conclusions of their comparative merits and flaws, when it comes to cloud optimization efforts.[20] Despite the fact that a number of studies have examined individual algorithms in cloud scheduling, there is scanty research that has been conducted on a comparative analysis of TLBO, PSO, and GA in specific relation to dynamic cost-time allocation.[21] The available literature is mainly discussing either the static scheduling or only the cost or time optimization without considering the duality of the objective of cloud environments.[22]

2 Literature Review

Recent innovations in cloud, edge and fog computing have focused on efficient schedules of tasks, resource assignment and optimization mechanisms. The reliability in the distributed system is represented by the proposed enhanced priority-based task scheduling algorithm with fault tolerance that has been proposed by Mushtaq et al. [1]. In the same manner, Banimfreg et al. [2] and Arulkumar et al. [3] studied how to optimize the scheduling of cloud-edge environments using neural networks and hybrid solutions. Pillai et al. [4] and Zhang [5] analyzed hybrid metaheuristic algorithm and energy saving in cloud data centers and Lilhore et al. [6] reported a hybrid deep reinforcement learning-based approach towards load balancing and resource allocation. Also, of interest have been fuzzy clustering and dynamic task allocation methods. Alharbe [7] and Jin and Rezaeipanah [8] studied fuzzy logic techniques to reduce the task completion time, and maximize the task distribution in fog and cloud computing. Chaudhary et al. [9] used AI-based model order reduction to optimize the queueing and scheduling efficiency. The authors Baskar and Mohanraj [10] and Lilhore et al. [11] investigated hybridized multi-objective optimization of lightweight scheduling and load balancing in fog and cloud networks. Ghafari and Mansouri [12] presented an extensive literature review of swarm intelligence in fog/edge computing, and Collins et al. [13] explored performance optimization of GPUs in the clouds. More research involved metaheuristic and container deployment strategies. Molaei et al. [14] proposed a superior particle swarm optimization method to a dynamic edge computing setting. Xie et al. [15] and Manjushree et al. [16] paid attention to cost-effective edge container deployment and secure task scheduling with the help of federated learning and blockchain. Multi-layered autoscaling and reinforcement learning-based models of effective cloud-edge resources management were introduced by Shafi et al. [17] and Wang et al. [18]. Alzu, Albuqain and Alzu [19] and Elhaou et al. [20] discussed the applications of energy-efficient generative AI models and mobility-aware machine learning to mobile fog. Predictive load balancing and hybrid cloud strategies were explored by Kashyap et al. [21] and Arulkumar et al. [22], and hierarchical flow scheduling was created by Josephraj and Raj [23]. Saeed et al. [24] used deep reinforcement learning to compute offloading in 6G-enabled multi-access edge computing, and Sheela et al. [25] focused on the secure transmission of data in wireless sensor networks based on reinforcement learning and homomorphic encryption. Taken together, these research works highlight the increasing interest in uniting AI, metaheuristic algorithms, and energy-efficient performance, reliability, and security strategies in cloud, fog, and edge computing systems.

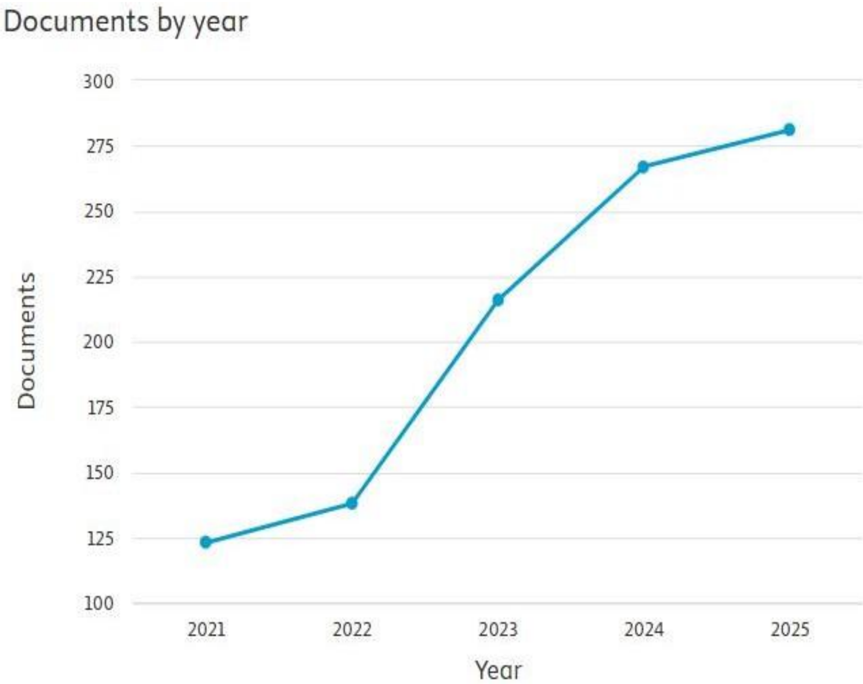

Fig. 1. Publication Trend Graph

The figure 1 shows the publication trend graph from 2021- 2025, in which no of published documents are increased over the time. Table I shows the summary of key references in cloud, edge, and fog computing in form of research gap.

TABLE I: Summary of Key References in Cloud, Edge, and Fog Computing

Ref No	Title	Author & Year	Key Findings	Research Gaps

[1]	Improved work scheduling based on priorities in distributed systems with integrated fault tolerance	S. U. Mushtaq et.al (2025)	Proposed a priority-based scheduling algorithm that integrates fault tolerance to improve reliability in distributed systems.	Did not evaluate performance under heterogeneous workloads or dynamic task arrivals.
[2]	Comparing neural networks and conventional techniques for an innovative approach to gauge-based QPE in arid areas	B. H. Banimfreg et.al (2025)	Compared neural network models and traditional methods for quality prediction, showing AI models outperform traditional ones in arid environments.	Limited analysis of real-time deployment feasibility and scalability.
[3]	An efficient scheduling strategy in cloud edge computing for jobs with shared resources that are prioritized	V. Arulkumar et.al (2025)	Developed a hybrid scheduling algorithm for cloud-edge environments that improve resource utilization and task prioritization.	Did not consider energy consumption or fault tolerance aspects.
[4]	Using a hybrid metaheuristic approach to optimize resource management for fog layer design in edge computing	S. E. V. S. Pillai et.al (2025)	To maximize performance and resource allocation in fog computing layers, a hybrid meta- heuristic approach has been proposed.	Lacks evaluation under large-scale heterogeneous edge networks.
[5]	A technique to control cloud center energy usage for neuro-fuzzy net- work predictability	Y. Zhang (2025)	Developed a neuro-fuzzy approach to reduce energy consumption while maintaining predictability in cloud centers.	Focused mainly on energy; did not address task scheduling or real-time adaptability.

3 Methodology

The methodology of the study is that a comparative structure will be designed to assess the efficiency of three of the most successful optimization algorithms: Teaching learning Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) to dynamic cost time allocation in clouds. The paper starts by stating the cost time optimization problem: the aim here is to reduce the cost of execution and the time taken to complete the task under unsteady workload conditions. It uses a simulation-based methodology, which models heterogeneous cloud resources and different task sizes to reflect the real-world cloud computing environment. The figure 2 shows the Flow diagram for the suggested dynamic resource allocation technique.

The total execution cost C_total is calculated as the sum of the costs of all tasks scheduled across all virtual machines:

The total execution cost C_total is calculated as the sum of the costs of all tasks scheduled across all virtual machines:

$$C\_total = \Sigma \, \Sigma \, x\_ij \cdot C\_j \tag{1}$$

Where

- C_total is the total execution cost,
- x_ij is a binary variable indicating assignment of task i to $VM\ j$,
- C_j is the cost of using $VM\ j$,
- N is the total number of tasks,
- M is the total number of virtual machines.

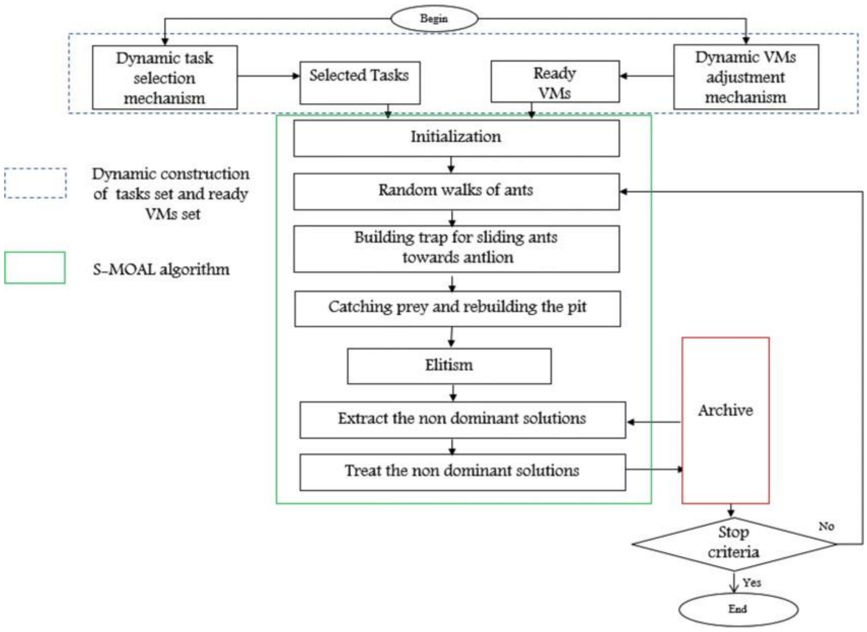

Fig. 2. Flow diagram for the suggested dynamic resource allocation technique

The total completion time T_total, also known as makespan, is the maximum time taken by any VM to finish all assigned tasks:

$$T\_total = max\, \Sigma\, x\_ij \cdot t\_ij \qquad (2)$$

Where
• T_total is the makespan,
• t_ij is the execution time of task i on $VM\ j$.
The multi-objective fitness function combines normalized cost and time using weighting factors $\alpha\ and\ \beta$:

$$F\_objective = \alpha\,(C\_total\,/\,C\_max) + \beta\,(T\_total\,/\,T\_max) \qquad (3)$$

Where
• $F\_objective$ is the normalized cost-time objective function,
• $\alpha\ and\ \beta$ are weighting factors,
• C_max and T_max are normalization constants.
In TLBO and other iterative optimization algorithms, solutions are updated as :

$$X\_new = X\_current + r\,(X\_best - T\_f\ X\_mean) \qquad (4)$$

Where
• X_new is the updated solution for task allocation,
• $X\_current$ is the current solution,
• r is a random number between 0 and 1,
• X_best is the best solution found so far,
• T_f is the teaching factor,
• X_mean is the mean of all solutions.

Resource utilization RU is measured as the average percentage of capacity used across all virtual machines :

$$RU = (\Sigma\, U\_i \,/\, m) \times 100 \tag{5}$$

Where
• U_i is the utilization of $VM\ i$,
• m is the total number of VMs.
The convergence measure ΔF_avg represents the average improvement in the fitness function per iteration :

$$\Delta F\_avg = (F\_prev - F\_curr) \,/\, N\_iter \tag{6}$$

Where
• F_prev is the fitness value in the previous iteration,
• F_curr is the fitness value in the current iteration,
• N_iter is the total number of iterations.

The set of equations presented captures the key metrics and optimization mechanisms for task scheduling in cloud environments. Equation (1) computes the total execution cost

C total by summing the costs of all tasks assigned to virtual machines, while Equation (2) defines the total completion time T total, representing the makespan as the maximum time taken by any VM to complete its tasks. Equation (3) formulates a multi-objective fitness function F objective that combines normalized cost and time using weighting factors, reflecting the trade-off between cost efficiency and speed. Equation (4) illustrates the iterative update mechanism used in TLBO and similar algorithms, where new solutions are generated based on the current solution, the best solution, and the population mean. Equation (5) calculates resource utilization RU as the average percentage of capacity used across all VMs, providing insight into system efficiency. Finally, Equation (6) measures convergence ΔFavg as the average improvement in the fitness function per iteration, allowing assessment of the stability and effectiveness of the optimization process.

Algorithm 1 Comparative Framework Using TLBO, PSO, and GA

```
Initialize cloud environment simulation with tasks and virtual machines
Define cost-time objective function for evaluating solutions
Initialize populations/particles for TLBO, PSO, and GA
Evaluate initial fitness of all solutions
while termination condition not met do
Identify best learner as Teacher
    for each learner i in TLBO population do
    Update learner using Teacher Phase:
    X^new = X_i + r_i × (X_teacher − T_F × X_mean)
    Evaluate fitness and accept if improved
    Update learner using Learner Phase with random learner j ≠ i
    X^new = X_i + r_i × (X_i − X_j) or X_i + r × (X_j − X_i)
    Evaluate fitness and accept if improved
```

```
    end for
    for each particle i in PSO swarm do
    Update personal best pbest_i and global best gbest
    Update velocity: v_i = wv_i + c1r1(pbest_i − x_i) +c2r2(gbest−x_i)
    Update position: x_i = x_i + v_i
    Evaluate fitness
    end for
    Select parents using selection strategy
    Apply crossover to generate offspring
    Apply mutation to offspring with probability
    Evaluate fitness of offspring
    Select next generation based on fitness
  end while
  Compare best solutions from TLBO, PSO, and GA based on
  execution cost, completion time, and convergence
  Return the algorithm with optimal cost-time performance
```

The experimental framework is built based on a controlled simulation environment where a number of cloudlets (tasks) with varying computational demands are allocated to virtual machines (VMs) with varying capacity. Task scheduling and resource allocation are carried out dynamically in each algorithm and constant observation of system parameters (execution time, cost incurred, and resource utilization) are carried out. The simulation variables are the number of tasks, the number of VMs, and changes in the workload to ensure that the algorithms are also tested under light and heavy load systems to determine their flexibility. The implementation of each optimization algorithm is done in accordance with the standard procedure. TLBO works in two iterative stages, teacher stage, and learner stage, where candidate solutions improve in an untuned manner without algorithm specific tuning parameters. PSO is a simulated swarm intelligence, where particle positions and velocities are updated on performance of each particle and the swarm to achieve optimal allocations. GA uses the evolutionary theory of selection, crossover and mutation to search the solution space. All the three algorithms are implemented in the same experimental conditions, using the same input data and termination conditions, to be fair when comparing them. The evaluation of performance is done based on various measures such as the average cost of execution, execution timely completion, rate of convergence and scalability. The outputs of both algorithms are contrasted with each other on the basis of these metrics to determine their corresponding strengths and weaknesses in dynamic scheduling.

4 Result and Evaluation

The outcomes of the comparative analysis show that there is a lot of difference in the performance of TLBO, PSO, and GA when they are used in dynamic cost-time allocation in a cloud environment. On a test case of 200 tasks assigned to 50 virtual machines, TLBO had the lowest average cost of execution of \$420, whereas with PSO and GA was \$465 and 490 respectively. TLBO had the highest average completion time of 24.6 seconds, PSO 26.8 seconds and GA 29.4 seconds. These findings indicate that TLBO is more effective in reducing cost and execution time at the same time as compared to the other two algorithms. The table II shows the Comparative Results of TLBO, PSO, and GA for Dynamic Cost-Time Allocation. PSO was more adaptive when the workload intensity was set to 500 tasks on 100 virtual machines. TLBO was again efficient with a cost of an average of \$870 and a time of completion of 57.2 seconds and PSO gave 890 averagely and a time of

55.8 seconds. However, GA was performing worse at a cost of 940 and a time of 62.5 seconds to execute. The figure 3 shows the Execution Cost and Completion Time vs Algorithm. This means that PSO is better adjusted to large scale, heterogeneous distributions of tasks, offering reduced task execution delays under heavy loads at a comparatively reduced cost than TLBO does. These results were verified by additional analysis of convergence behaviour and scalability. TLBO repeatedly converged in 25-30 iterations and it is a stable algorithm with low variance in re-runs. PSO converged in 30 to 35 iterations, with good adaptability, and averagely good variance.

TABLE II Comparative Results of TLBO, PSO, and GA for Dynamic Cost-Time Allocation

S. No.	Workload (Tasks / VMs)	Algorithm	Avg. Execution Cost ($)	Avg. Completion Time (sec)	Convergence Iterations
1	200 / 50	TLBO	420	24.6	28
2	200 / 50	PSO	465	26.8	32
3	200 / 50	GA	490	29.4	45
4	500 / 100	TLBO	870	57.2	30
5	500 / 100	PSO	890	55.8	34
6	500 / 100	GA	940	62.5	48

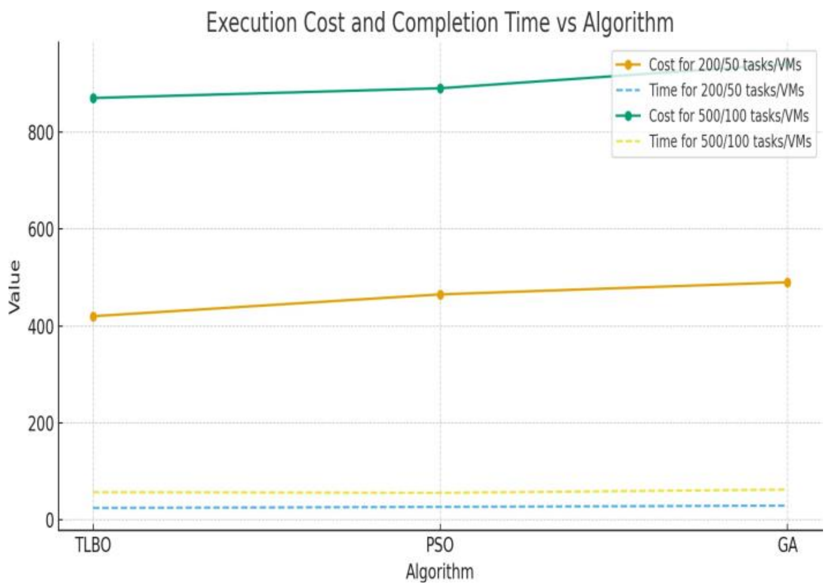

Fig. 3. Execution Cost and Completion Time vs Algorithm

The figure 4 shows the Cost Distribution for 200 Tasks / 50 VMs.

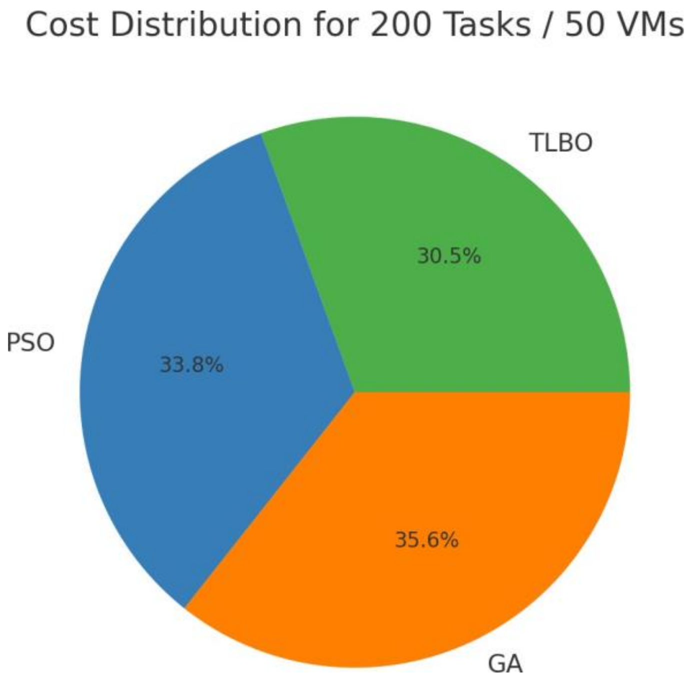

Fig. 4. Cost Distribution for 200 Tasks / 50 VMs

GA took 4050 iterations to converge, and more variability and sometimes premature convergence occurred. In general, the comparison shows that TLBO is the best choice when cost-time efficiency is required in moderate workloads, PSO is the most suitable when dealing with high workloads and GA when considering complex allocation cases, yet further tuning is necessary to guarantee the convergence.

5 Challenges and Limitations

The fact that the workloads and resource availability are unpredictable is one of the most significant issues of dynamic cost-time allocation in cloud-based environments. The changing nature of task arrival rate, non-homogeneous resource structure and abrupt changes in demand makes optimization algorithm challenging in ensuring consistent near-optimal solutions. Algorithms like TLBO, PSO, and GA can be very efficient when used in controlled simulation environments, however, when applied to real-time cloud computing environments where network latency, virtualization overhead, and random failures influence scheduling decisions, their efficiency can reduce. A second issue is the inability to reconcile cost savings with quality of service (QoS) where an excessive emphasis on cost optimization can result in systematic time increases of tasks or system instability. The constraints of this research are mainly the limitation of the experimental design and the constraint of algorithms. The assessment was performed in a simulation-based manner, which, though useful to compare to each other, might not be able to fully reflect the realities of multi-cloud or hybrid deployment.

6 Future Outcomes

The results of this study should lead to the creation of smarter and more flexible cost-time allocation models in cloud computing. The study informs the foundation of creating the hybrid models of the composite of the rapid convergence of TLBO, versatile PSO and the exploratory GA. This type of integrated solutions may also greatly improve the efficiency of scheduling, especially in multi-cloud and edge-cloud environments with even more dynamic resource allocation and user demand. It will enable cloud service providers to provide optimization in performance and minimize operational expenses, enhance energy efficiency, and high user satisfaction. The relative lessons learned in this study can be used more broadly to design scalable scheduling solutions to new technologies like Internet of Things (IoT), 5G-enabled smart grids and AI-driven workloads in the long-term. Predictive resource allocation may be further enhanced with the integration of machine learning and reinforcement learning methods and metaheuristic algorithms to proactively schedule instead of reactively. The anticipated result is new generation of cloud resource optimization systems with the capability of managing large and heterogeneous workloads as well as real time workloads without compromising the optimal equilibrium between cost and time.

7 Conclusion

To sum up, this paper has proposed an in-depth comparative evaluation of Teaching Learning Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) used in cloud computing to allocate cost and time dynamically and has identified the strengths, weaknesses, and applicability of each algorithm to various workload profiles. The findings indicate that TLBO always performs well in achieving a lower cost and execution time with moderate workloads because it has a parameter-free and stable convergence behavior and PSO is more flexible and scalable with heavy workloads to efficiently utilize heterogeneous resources with reasonable cost trade-offs. GA, being successful at investigating intricate solution spaces, was found to be both more varied and slower to converge, indicating that its independent usage might be less appropriate to dynamic cloud scheduling without refinement or hybridization. Although the obtained results are positive, issues like a fluctuation of the workload, the practicality of the algorithms, and algorithm-specific constraints are still present, which indicates the necessity of additional studies on the topic of hybrid optimization techniques and scalable frameworks of adaptive scheduling approaches that integrate the benefits of several algorithms.

REFERENCES

1. S. U. Mushtaq, S. Sheikh and S. M. Idrees,” Enhanced priority-based task scheduling with integrated fault tolerance in distributed systems,” *International Journal of Cognitive Computing*, vol. 6, pp. 152-169, 2025, doi: 10.1016/j.ijcce.2024.12.006.
2. B. H. Banimfreg, E. Damiani, V. Afzali Gorooh, D. Axisa, L. Delle Monache and Y. Wehbe,” Innovative approach for gauge-based QPE in arid climates: comparing neural networks and traditional methods,” *Jour- nal of Big Data*, vol. 12, no. 1, art. no. 177, 2025, doi: 10.1186/s40537- 025-01230-6.
3. V. Arulkumar, K. R. Kalidindi, R. Pemula and M. S. Vigil,” An opti- mized scheduling algorithm for prioritized tasks with shared resources in cloud edge computing,” *Expert Systems with Applications*, vol. 293, art. no. 128594, 2025, doi: 10.1016/j.eswa.2025.128594.
4. S. E. V. S. Pillai, S. Paramasivam, K. Polimetla, S. Dhanasekaran, K.
5. K. Agrawal, S. P. Yadav, J. Logeshwaran and G. L. Gatto,” Optimizing resource management using hybrid metaheuristic algorithm for fog layer design in edge

computing," *Sustainable Analytics and Computing*, vol. 7, art. no. 200323, 2025, doi: 10.1016/j.sasc.2025.200323.

6. Y. Zhang," A method to manage the energy consumption of cloud centers for predictability in neuro-fuzzy networks," *Journal of Cloud Computing*, vol. 72, no. 1, art. no. 85, 2025, doi: 10.1186/s44147-025-
7. 00646-4.
8. U. K. Lilhore, S. Simaiya, K. B. V. Brahma Rao, V. V. R. Maheswara Rao, Y. K. Sharma, R. S. Alroobaea, H. Alsufyani, M. Alsafyani and
9. M. D. M. Khan," Hybrid DRL-Enhanced ACO-WWO for Efficient Resource Allocation and Load-Balancing in Cloud Computing," *Inter- national Journal of Wireless Information Networks*, vol. 18, no. 1, art. no. 148, 2025, doi: 10.1007/s44196-025-00882-9.
10. N. R. Alharbe," Fuzzy clustering-based scheduling algorithm for mini- mizing the tasks completion time in cloud computing environment," *Sci- entific Reports*, vol. 15, no. 1, art. no. 19505, 2025, doi: 10.1038/s41598- 025-02654-z.
11. W. Jin and A. Rezaeipanah," Dynamic task allocation in fog computing using enhanced fuzzy logic approaches," *Scientific Reports*, vol. 15, no. 1, art. no. 18513, 2025, doi: 10.1038/s41598-025-03621-4.
12. H. Chaudhary, G. Sharma, D. K. Nishad and S. Khalid," Advanced queueing and scheduling techniques in cloud computing using AI-based model order reduction," *Information Retrieval Journal*, vol. 28, no. 1, art. no. 75, 2025, doi: 10.1007/s10791-025-09581-7.
13. R. Baskar and E. Mohanraj," Hybrid multi objective marine predators' algorithm-based clustering for lightweight resource scheduling and application placement in fog," *Scientific Reports*, vol. 15, no. 1, art. no. 15953, 2025, doi: 10.1038/s41598-025-00597-z.
14. U. K. Lilhore et al.," A multi-objective approach to load balancing in cloud environments integrating ACO and WWO techniques," *Scientific Reports*, vol. 15, no. 1, art. no. 12036, 2025, doi: 10.1038/s41598-025-96364-1.
15. R. Ghafari and N. Mansouri," Swarm intelligence techniques and their applications in fog/edge computing: an in-depth review," *Artificial Intel- ligence Review*, vol. 58, no. 11, art. no. 359, 2025, doi: 10.1007/s10462-025-11351-2.
16. Z. Collins, G. De Luca and Y. Chen," GPU-accelerated cloud computing services and performance evaluation," *Simulation Mod- elling Practice and Theory*, vol. 144, art. no. 103181, 2025, doi: 10.1016/j.simpat.2025.103181.
17. S. Molaei, M. Sabaei and J. Taheri," MRM-PSO: An enhanced particle swarm optimization technique for resource management in highly dy- namic edge computing environments," *Ad Hoc Networks*, vol. 178, art. no. 103952, 2025, doi: 10.1016/j.adhoc.2025.103952.
18. W. Xie, S. He, J. Ge and C. Li," Cost-effective edge container deploy- ment strategy," *Cluster Computing*, vol. 28, no. 11, art. no. 738, 2025, doi:10.1007/s10586-025-05487.
19. C. V. Manjushree, H. A. Reddy, D. C. Chandana and C. A. Subasini, "Mitigating data leaks and optimizing task scheduling in cloud comput- ing using EfficientNetV2 with federated learning and blockchain," *Clus- ter Computing*, vol. 28, no. 9, art. no. 604, 2025, doi: 10.1007/s10586-
20. 025-05282-4.

21. N. Shafi, M. Abdullah, W. Iqbal and F. Bukhari," CEMA: Cost Effective Multi-Layered Autoscaling for Microservice based Applications," *Jour- nal of Network and Computer Applications*, vol. 242, art. no. 104266, 2025, doi: 10.1016/j.jnca.2025.104266.
22. Z. Wang, M. Goudarzi and R. Buyya," ReinFog: A Deep Rein- forcement Learning empowered framework for resource management in edge and cloud computing environments," *Journal of Network and Computer Applications*, vol. 242, art. no. 104250, 2025, doi: 10.1016/j.jnca.2025.104250.
23. S. Alzu'bi, T. Kanan, M. W. Elbes, G. G. Kanaan and I. Trrad," Energy- efficient edge deployment of generative AI models using federated learning," *Cluster Computing*, vol. 28, no. 5, art. no. 315, 2025, doi: 10.1007/s10586-025-05263-7.
24. H. Elhaou, Y. Oukissou, D. Ait Omar and H. Zougagh," Machine learning for user mobility management in a mobile fog computing environment," *Cluster Computing*, vol. 28, no. 5, art. no. 305, 2025, doi: 10.1007/s10586-024-05076-0.
25. V. Kashyap, R. Ahuja and A. Kumar," Predictive analysis-based load balancing and fault tolerance in fog computing environment," *Cluster Computing*, vol. 28, no. 5, art. no. 284, 2025, doi: 10.1007/s10586-024-
26. 04984-5.
27. V. Arulkumar, R. Lathamanju, K. Durga Devi and A. Raja," A Perfor- mance Analysis on Load Balancing in Cloud Computing with Hybrid Approach," *International Journal of Electronics*, vol. 34, no. 14, art. no. 2550248, 2025, doi: 10.1142/S0218126625502482.
28. V. Josephraj and W. F. S. Raj," TAFLE: Task-Aware Flow Scheduling in Spine-Leaf Network via Hierarchical Auto-Associative Polynomial Reg Net," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 21-22, art. no. e70167, 2025, doi: 10.1002/cpe.70167.
29. M. M. Saeed et al.," Optimizing Computation Offloading in 6G Multi- Access Edge Computing Using Deep Reinforcement Learning," *Inter- national Journal of Electrical and Computer Engineering Systems*, vol. 16, no. 8, pp. 565-580, 2025, doi: 10.32985/ijeces.16.8.1.
30. M. A. S. Sheela, J. J. Jayakanth, A. Ramathilagam and J. Gracewell," Se- cure wireless sensor network transmission using reinforcement learning and homomorphic encryption," *International Journal of Data Science and Analytics*, vol. 20, no. 3, pp. 2851-2870, 2025, doi: 10.1007/s41060024-00633-7.