Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AMLGTTA: Adaptive Multicloud Load Balancing by Harnessing GAHP and TLGWO for Enhancing Task Allocation

Bharti Vijay Nikose[1], Gautam M. Borkar[2]

[1] *Department of Computer Engineering, Ramrao Adik Institute of Technology, D Y Patil Deemed to be University, Nerul, Navi Mumbai, MH, India*

[2]*Department of Information Technology, Ramrao Adik Institute of Technology, D Y Patil Deemed to be University, Nerul, Navi Mumbai, MH, India*

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

The study introduces a multicloud load balancing approach, harnessing Genetic Analytical Hierarchical Processing (GAHP), an extension of AHP. This adapts its internal weights dynamically to reclassify VMs in response to fluctuating demands. For efficient task allocation, tasks are clustered, considering parameters such as makespan, deadline, resource utilization, and notably dependency levels, with dependency accorded utmost priority. Task allocation to VM categories is facilitated using the Teacher Learner-based Grey Wolf Optimizer (TLGWO). Additionally, task parameters are encrypted using specially designed Intelligent Physical Unclonable Functions (IPuFs) to ensure secure data transmission to respective VM Category Groups (VMCGs). Within VM category groups, task assignments are optimized using the local TLGWO model. The proposed model demonstrates significant advancements over existing frameworks. Specifically, a 9.5% reduction in makespan, 4.9% enhancement in VM computation efficiency, 2.5% improvement in deadline adherence, 4.5% increase in task diversity, 3.9% boost in execution efficiency, and 3.4% decrease in decision latency. This model has shown to decline in inter-cloud communication delay by 2.5%, elevate throughput by 1.9%, and enhance packet delivery performance by 2.4%. Consequently, our proposed multicloud load balancing model furnishes a robust solution, mitigating extant constraints and ensuring heightened performance and efficiency in multicloud scenarios.

Keywords: Genetic Analytical Hierarchical Processing, Teacher Learner based Grey Wolf Optimizer, Virtual Machine Category Groups, Intelligent Physical Unclonable Functions

INTRODUCTION

With the rapidly evolving digital landscape, cloud computing has emerged as a keystone in the technological world. Its ability to offer scalable computational resources, storage solutions, and application services has made it indispensable for enterprises, researchers, and individual users alike. Multicloud strategies, where users employ a combination of cloud services from different providers, have grown increasingly popular. This diversification mitigates risks, maximizes available resources, and prevents vendor lock-in. However, as with any distributed system, multicloud environments present unique challenges, with load balancing being at the forefront [1].

Load balancing [2] in multicloud environments aims to efficiently distribute computational tasks among different cloud platforms to optimize resource utilization, reduce latency, and enhance the overall system performance. Traditional load balancing mechanisms have predominantly operated under the premise of static weights and predefined rules [3]. These mechanisms, while efficient in simpler, controlled environments, have struggled to cope with the dynamic nature of multicloud infrastructures. The task complexity, coupled with varying cloud capabilities and fluctuating demands, highlights a gap in current strategies. A one-size-fits-all approach becomes unsuitable and often leads to performance bottlenecks, increased decision delays, and suboptimal resource allocation [4]. Furthermore, security remains paramount [5]. As tasks are offloaded and data travels across different clouds, it

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

becomes vulnerable to myriad threats [6]. The need for secure, yet efficient task allocation and data transfer processes is more pressing than ever for different use cases.

In this paper, we delve into a novel approach to address these challenges. By integrating the dynamic capabilities of Genetic Analytical Hierarchical Processing and the optimization strengths of Teacher Learner based Grey Wolf Optimizer, we aim to create a more adaptable and robust load balancing mechanism. This mechanism not only prioritizes tasks based on multifaceted parameters but also ensures their secure transfer through innovative encryption methods.

The objective of this research is to bridge the gap between the traditional load balancing strategies [7] and the demanding needs of today's multicloud environments [8]. Through empirical evaluations [9], we underscore the efficacy of our model, its real-time benefits, and its potential as a cornerstone for future multicloud load balancing solutions.

OBJECTIVES

1. To develop a dynamic VM categorization mechanism
2. To implement multi-parameter task prioritization
3. To enhance task allocation efficiency using hybrid metaheuristic optimization
4. To incorporate a secure task transmission mechanism
5. To reduce inter-cloud communication delay and improve system-level performance metrics
6. To bridge the research gap between adaptability, optimization efficiency, and security in multicloud load balancing

MOTIVATION

The growing popularity of multicloud approaches emphasizes the necessity for enhanced performance, resilience, and operational flexibility. As users seek to leverage the benefits of distributed cloud resources, the complexity of managing them also increases. Traditional load balancing methods are no longer sufficient to meet the intricate requirements of multicloud ecosystems. Tasks are now diverse, with varying levels of complexity, interdependencies, and deadlines. Assigning these tasks to the appropriate cloud resource while ensuring optimal performance is like searching for a needle in a haystack, especially when real-time constraints are involved.

Additionally, as data traverses between clouds, maintaining its confidentiality and integrity becomes crucial. While traditional encryption and security measures are robust, they often introduce additional computational overhead, potentially compromising efficiency.

The convergence of the demand for high performance and stringent security is the primary motivation behind our research. We recognize the need for an adaptable, efficient, and secure load balancing strategy that is tailored to the dynamic nature of multicloud environments.

CONTRIBUTION

In light of the challenges identified, our research offers the following seminal contributions:

- Introduction of GAHP in Multicloud Environments: By extending the traditional Analytical Hierarchical Processing (AHP) into a more dynamic Genetic Analytical Hierarchical Processing, we provide a mechanism that can recalibrate itself in response to changing cloud demands. This adaptability ensures that VM categorization remains relevant and optimal, facilitating better task allocations.

- Task Prioritization Using Multiple Parameters: Unlike conventional models that often prioritize tasks based on a single parameter, our model considers make span, deadline, resource utilization, and notably, dependency levels, ensuring a more holistic and efficient task distribution.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Employment of TLGWO for Task Allocation: Our innovative use of the Teacher Learner based Grey Wolf Optimizer introduces a nuanced task allocation strategy that not only ensures optimal resource utilization but also expedites the entire process, thereby reducing decision delays.

- Security Enhancement with IPuFs: Recognizing the vulnerabilities inherent in multicloud data transfers, we introduce the use of Intelligent Physical Unclonable Functions (IPuFs) for encryption. This not only elevates the security profile of our model but also ensures that the encryption process complements the overall system efficiency.

- Empirical Validation: Our extensive real-world simulations and testing validate the efficacy of the proposed model, offering tangible improvements over existing strategies in terms of performance metrics, security, and efficiency.

By addressing the pressing challenges in multicloud load balancing, our research paves the way for a more resilient, efficient, and secure multicloud landscape.

LITERATURE REVIEW

The landscape of cloud computing continues to evolve rapidly, driven by advancements in networking, computing, and data analytics technologies. In recent years, researchers have focused on addressing various challenges related to task scheduling, resource allocation, and optimization in cloud environments. Table 1 includes a comprehensive review existing methods, spanning a wide array of topics within cloud computing. The reviewed papers cover diverse methodologies and techniques aimed at improving task scheduling, resource allocation, and optimization in cloud computing environments. Key areas of focus include dynamic task scheduling, multi-objective optimization, energy efficiency, reliability, deadline-constrained scheduling, and the integration of emerging technologies like edge computing and IoT.

The methodologies employed in these papers range from traditional optimization algorithms to cutting-edge techniques such as machine learning, swarm intelligence, and deep reinforcement learning. Furthermore, several papers explore the application of hybrid approaches that combine multiple methodologies to achieve superior performance and efficiency.

S. Tuli et al. [1] This study introduces A3C Learning and Residual Recurrent Neural Networks. They utilized dynamic scheduling for stochastic edge-cloud computing environments. They Demonstrated improved performance and adaptability in real-time scenarios but there is limited exploration of scalability issues and robustness in highly dynamic environments.

I. M. Ali et al. [2] The Non-Dominated Sorting Genetic Algorithm II has been implemented in this study. A researcher has created an automated model for task scheduling in fog-cloud systems. Although the model demonstrates improved task scheduling capabilities, there may be difficulties in scaling it for larger fog-cloud deployments.

S. Achar [3] The paper delves into the Neural-Hill Algorithm, which introduces an innovative approach to scheduling IoT-cloud resources efficiently. The algorithm demonstrates notable scalability and adaptability within IoT-cloud environments, showing promise for enhancing resource management in such settings. However, there is a need for more comprehensive evaluation across a range of workload scenarios to further validate its effectiveness and robustness.

P. V. Reddy and K. G. Reddy [4] A framework for scheduling based on multiple objectives was presented, specifically designed for cloud computing. This framework emphasized the enhancement of resource utilization and the efficiency of task allocation. However, it should be noted that the analysis of scalability and performance in the presence of dynamic workloads was not extensively explored.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

T.-P. Pham and T. Fahringer [5] The utilization of evolutionary algorithms in the scheduling of workflows in dynamic cloud environments has been explored in the field of Evolutionary Multi-Objective Workflow Scheduling. This research has shown promising results in terms of effectively adapting to fluctuating resource conditions. However, it is important to note that additional optimization may be necessary to meet the requirements of real-time scheduling constraints.

P. Loncar and P. Loncar [6] An algorithm was developed to handle diverse cloud resources, known as Evolution Strategies Algorithm. This algorithm aimed to enhance scalability and efficiency in the management of resources within the cloud environment. However, there was a lack of in-depth exploration into the convergence and robustness of the algorithm. Further investigation is required to fully understand the algorithm's performance in terms of reaching optimal solutions and its ability to withstand various challenges.

X. Ma et al. [7] The study presented a novel approach to dynamic task scheduling in cloud-assisted mobile edge computing. By implementing this approach, significant enhancements were observed in task allocation and responsiveness within mobile edge environments. However, it is important to note that the evaluation of this approach was limited to scenarios involving high mobility and resource contention.

M. Kumar et al. [8] An innovative framework for autonomic resource provisioning and scheduling was created, known as the ARPS framework. The ARPS framework showcased improved resource optimization and performance capabilities. Further validation is needed to assess its effectiveness in various cloud deployment scenarios.

Q. Wu et al. [9] The scheduling of cloud workflows has been enhanced by addressing the issue of endpoint communication contention. This improvement has resulted in increased efficiency in workflow execution and better utilization of resources. However, there is still a need for further exploration of scalability and performance across different types of workflows to ensure optimal outcomes.

L. Ye et al. [10] Dynamic scheduling for cloud container services was proposed to manage stochastic hybrid workflows effectively. The approach demonstrated efficient handling of dynamic workload characteristics, showcasing its potential in optimizing resource allocation. However, there is a need for further investigation into scalability, especially for large-scale container deployments, to ensure seamless performance and adaptability in real-world scenarios.

H. Mahmoud et al. [11] The decision tree algorithm was employed to address the multiobjective task scheduling in cloud environments, resulting in enhanced scheduling efficiency and task allocation. The study, however, did not extensively investigate the scalability and adaptability of the algorithm in dynamic environments.

S. Qin et al. [12] Utilized Adaptive Discrete Water Wave Optimization in the context of cloud workflow scheduling, showcasing enhanced scheduling performance and adaptability. The effectiveness of this approach necessitates validation across various workload conditions and evaluation of scalability to ensure its practicality and efficiency in real-world scenarios.

X. Tang [13] An innovative approach improved reliability and cost efficiency in scheduling scientific workflows across multiple cloud platforms, emphasizing the importance of considering reliability factors in workflow management. However, the research had limitations in exploring scalability and adaptability in dynamic multi-cloud environments. Further investigation in these areas could optimize workflow scheduling strategies in evolving cloud computing landscapes.

X. Tang et al. [14] An algorithm for job scheduling was created, focusing on cost-effectiveness and reliability. The algorithm showcased enhanced efficiency in job scheduling and cost-effectiveness. However, further validation is necessary in various cloud deployment scenarios to ensure its effectiveness and scalability.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

X. Wang et al [15] Employed coalition reinforcement learning to optimize adaptive cloud bundle provisioning, demonstrating improved resource allocation and multi-workflow scheduling efficiency. Explored the potential of scalability and adaptability in large-scale cloud environments, although further research is needed to fully understand the implications and challenges of implementing this approach on a broader scale.

H. Zhang and R. Jia [16] The implementation of Chaotic Cat Swarm Optimization (CCSO) has been utilized to improve the task scheduling process in cloud computing. By integrating chaos theory into the optimization algorithm, CCSO has shown significant enhancements in scheduling efficiency and flexibility. However, it is important to acknowledge that further research is required to investigate the algorithm's convergence and scalability, as these areas have not been thoroughly examined.

A. Belgacem et al. [17] The algorithm known as Symbiotic Organism Search was employed to allocate resources dynamically in cloud computing. This implementation demonstrated enhanced resource allocation and efficiency. However, further validation is needed in various cloud deployment scenarios, as well as exploration of its scalability potential.

K. Kang et al. [18] Utilized deep reinforcement learning (DRL) for task scheduling in the context of energy-efficient cloud computing. Showcased enhancements in energy efficiency and task scheduling effectiveness. However, there was a lack of focus on investigating the scalability and adaptability of the algorithm in dynamic cloud environments.

L. Ye et al. [19] The scheduling of workflows in IaaS clouds was addressed with a focus on reliability and energy efficiency. The study presented strategies that prioritize reliability and energy efficiency in the scheduling process, showcasing improvements in both aspects. However, the research had a restricted examination of scalability and adaptability when faced with dynamic workload conditions. This limitation suggests that further investigation is needed to fully understand the performance of the proposed scheduling methods in varying workloads.

Y. Chen et al. [20] The concept of Joint Front-Edge-Cloud IoVT Analytics has been reimagined to enhance its effectiveness. A resource-efficient design and scheduling approach have been developed to optimize the analytics process. This innovative solution has demonstrated enhanced resource utilization and improved task scheduling efficiency. However, further validation is required in various IoVT scenarios to ensure its scalability and effectiveness.

S. U. Jamil et al. [21] In the context of 6G enabled smart edge environments, the allocation of resources and the off-loading of tasks have been proposed as effective strategies. These approaches have shown promising results in terms of enhancing efficiency and optimizing resource utilization in smart edge environments. However, it is important to note that there is a need for further evaluation under diverse network conditions and scalability challenges to ensure their effectiveness in real-world scenarios.

J. He and X. Liu [22] The application of a hybrid teaching-learning-based optimization approach was employed to schedule workflows in a cloud environment. This approach demonstrated improved efficiency in scheduling and task allocation. However, it is important to consider the scalability and adaptability of the algorithm in dynamic environments, which may require further investigation.

J. Wang [23] Dynamic multi-workflow offloading and scheduling have been implemented in a cloud-edge environment with soft deadlines. The study showcased enhanced workflow execution and successful compliance with soft deadlines, presenting a promising solution for optimizing task allocation in cloud-edge systems. However, the evaluation was restricted in scope, primarily focusing on scenarios with high workloads and fluctuating network conditions. Further analysis under various conditions is imperative to ascertain the robustness and scalability of the proposed offloading and scheduling approach.

A. Taghinezhad-Niar and J. Taheri [24] Explored the scheduling of multiple workflows on multi-cloud systems with a focus on reliability, rental-cost, and energy efficiency. Demonstrated enhancements in reliability, cost-effectiveness,

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and energy-consciousness within this context. However, there was a lack of emphasis on scalability and adaptability in dynamic multi-cloud environments, warranting further investigation in these areas.

M. Guo et al. [25] Optimal VM Scheduling for Minimizing Delays. Concentrating on achieving optimal delay in scheduling virtual machines within a queueing cloud computing environment featuring diverse workloads. Showcased enhanced scheduling efficiency leading to decreased delays. Potential obstacles may arise when scaling up to extensive cloud deployments and adapting to fluctuating workload patterns.

Table 1. Review of Existing Methods

Sr. No.	Reference	Method Used	Findings	Results	Limitations
1.	D. Cheng et al. (2023) [26]	Dynamic Resource Provisioning for Iterative Workloads on Apache Spark	Proposed dynamic resource provisioning for iterative workloads on Apache Spark	Showcased enhanced resource utilization and performance for iterative workloads	Limited exploration of scalability and adaptability under diverse workload conditions
2.	M. Liwang and X. Wang (2022) [27]	Overbooking-Empowered Computing Resource Provisioning	Presented overbooking-empowered computing resource provisioning in cloud-aided mobile edge networks	Demonstrated improved resource allocation and efficiency	May require further investigation into scalability and adaptability in highly dynamic mobile edge environments
3.	L. Yang et al. (2023) [28]	Fully Hybrid Algorithm for Deadline Constrained Workflow Scheduling	Introduced a fully hybrid algorithm for deadline-constrained workflow scheduling in clouds	Demonstrated effective handling of deadline constraints and dynamic workload characteristics	Limited evaluation under highly dynamic workload conditions and scalability challenges
4.	B. Kruekaew and W. Kimpan (2022) [29]	Multi-Objective Task Scheduling Optimization	Utilized hybrid artificial bee colony algorithm with reinforcement learning for multi-objective task scheduling optimization	Showcased improved load balancing and task scheduling efficiency	May require further exploration of algorithm convergence and scalability
5.	Y. Huang et al. (2022) [30]	Deep Adversarial Imitation Reinforcement Learning	Employed deep adversarial imitation reinforcement learning for QoS-	Demonstrated enhanced scheduling	Limited exploration of scalability and adaptability under

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

			aware cloud job scheduling	performance and QoS-awareness	diverse workload conditions
6.	E. Cao et al. (2023) [31]	Energy and Reliability-Aware Task Scheduling	Focused on energy and reliability-aware task scheduling for cost optimization of DVFS-enabled cloud workflows	Exhibited improved cost optimization and energy efficiency	May face challenges in scalability and adaptability under dynamic workload variations
7.	S. Manam et al. (2023) [32]	Deadline-Constrained Cost Minimisation	Addressed deadline-constrained cost minimization for cloud computing environments	Demonstrated cost-effective scheduling under deadline constraints	Limited evaluation under highly dynamic workload conditions and scalability challenges
8.	X. Wang et al. (2023) [33]	Dynamic GPU Scheduling	Proposed dynamic GPU scheduling with multi-resource awareness and live migration support	Showcased improved resource utilization and efficiency	May require further investigation into scalability and adaptability in large-scale GPU environments
9.	M. T. Islam et al. (2022) [34]	Deep Reinforcement Learning for Spark Job Scheduling	Utilized deep reinforcement learning for performance and cost-efficient Spark job scheduling in cloud computing environments	Demonstrated improved scheduling performance and cost efficiency	Limited exploration of scalability and adaptability under diverse Spark workloads
10.	B. Dai et al. (2022) [35]	Mobility-Aware Computation Offloading and Resource Allocation	Addressed mobility-aware computation offloading and resource allocation in end-edge-cloud orchestrated computing	Showcased improved computation offloading efficiency and resource utilization	May require further exploration of scalability and adaptability in dynamic edge-cloud environments
11.	Y. Laili et al. (2023) [36]	Parallel Scheduling of Large-Scale Tasks	Focused on parallel scheduling of large-scale tasks for industrial cloud-edge collaboration	Demonstrated efficient task scheduling and resource utilization	Limited evaluation under diverse industrial scenarios and scalability challenges

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

12.	H. Sun et al. (2015) [37]	Dynamic Deployment and Scheduling Strategy	Proposed dynamic deployment and scheduling strategy for hierarchical cloud service systems in intelligent buildings	Showcased improved service deployment and scheduling efficiency	Limited exploration of scalability and adaptability in complex hierarchical systems
13.	S. Bose and N. Mukherjee (2022) [38]	Scheduling Heterogeneous Resources in Sensor-Cloud Infrastructure	Addressed scheduling of heterogeneous resources in sensor-cloud infrastructure	Exhibited improved resource allocation and efficiency	May face challenges in scalability and adaptability under diverse sensor-cloud deployments
14.	Z. Chen et al. (2022) [39]	Adaptive and Efficient Resource Allocation	Utilized actor-critic deep reinforcement learning for adaptive and efficient resource allocation in cloud datacenters	Demonstrated improved resource allocation and efficiency	Limited exploration of scalability and adaptability in large-scale cloud datacenter environments
15.	N. Rizvi et al. (21) [40]	Intelligent Salp Swarm Scheduler	Presented an intelligent salp swarm scheduler with fitness-based quasi-reflection method for scientific workflows in hybrid cloud-fog environment	Showcased improved scheduling performance for scientific workflows	May require further exploration of algorithm robustness and scalability in hybrid cloud-fog environments
16.	Z. Liu et al. (2023) [41]	RFID-Based Scheduling	Proposed low-latency and reliable DAG task scheduling over dynamic vehicular clouds	Demonstrated improved task scheduling efficiency and reliability	Limited validation under real-world vehicular cloud environments
17.	T. He et al. (2021) [42]	CAMIG: Concurrency-Aware Live Migration Management	Introduced CAMIG for live migration management in SDN-enabled clouds	Showcased enhanced concurrency and efficiency in VM migration	Requires validation in diverse SDN-enabled cloud environments
18.	A. Ali and M. M. Iqbal (2022) [43]	Cost and Energy Efficient Task Scheduling	Developed a technique for cost and energy-efficient task scheduling in	Demonstrated improved cost efficiency and energy savings	Limited exploration of scalability in large-scale mobile cloud deployments

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

			mobile cloud computing		
19.	H. Li et al. (2021) [44]	Scoring and Dynamic Hierarchy-Based NSGA-II	Proposed a scoring and dynamic hierarchy-based NSGA-II for multi objective workflow scheduling	Showed improved scheduling efficiency and optimization	Requires validation under diverse workflow scenarios and dynamic workload conditions
20.	Z. Sun et al. (2023) [45]	ET2FA: Hybrid Heuristic Algorithm	Introduced ET2FA for deadline-constrained workflow scheduling in the cloud	Demonstrated enhanced scheduling efficiency and deadline adherence	Limited exploration of algorithm scalability and adaptability
21.	I. Attiya et al. (2022) [46]	Improved Hybrid Swarm Intelligence	Presented an improved hybrid swarm intelligence for scheduling IoT application tasks in the cloud	Showcased enhanced task scheduling efficiency and adaptability	Limited investigation into algorithm convergence and robustness
22.	J. Lou et al. (2024) [47]	Cost-Effective Scheduling for Dependent Tasks	Developed cost-effective scheduling for dependent tasks with tight deadline constraints in mobile edge computing	Demonstrated improved cost efficiency and task scheduling performance	Requires validation under diverse mobile edge computing scenarios
23.	M. Bigdeli et al. (2023) [48]	Deadline-Aware Scheduling and Resource Allocation	Proposed offline and real-time deadline-aware scheduling and resource allocation algorithms for big data transmission over cognitive CRANs	Demonstrated enhanced efficiency and reliability in big data transmission	Limited exploration of algorithm scalability and adaptability in dynamic CRAN environments
24.	G. Yao et al. (2020) [49]	Hybrid Fault-Tolerant Scheduling	Introduced a hybrid fault-tolerant scheduling approach for deadline-constrained tasks in cloud systems	Showcased improved fault tolerance and reliability	Requires validation under diverse workload scenarios and exploration of scalability

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

25.	Q. Li et al. (2016) [50]	Dynamic Combinatorial Double Auction Model	Presented a dynamic combinatorial double auction model for cloud resource allocation	Demonstrated improved resource allocation efficiency and fairness	Limited exploration of algorithm performance under dynamic market conditions

METHODS-PROPOSED DESIGN OF AN ADAPTIVE MULTICLOUD LOAD BALANCING BY HARNESSING GAHP AND TLGWO FOR ENHANCED TASK ALLOCATION.

As per the review of existing models used for design of adaptive task scheduling methods suited for multicloud environments, it can be observed that the efficiency of these models is generally limited by number of cloud deployments, and the dependency between tasks. These models also have lower security when applied to real-time scenarios. To overcome these issues, this section discusses design of an Adaptive Multicloud Load Balancing for Enhanced Task Allocation and Real-Time Performance under heterogeneous tasks. As per figure 1, the proposed model initially estimates task & VM capacities, and passes them through an innovative Genetic AHP Model, which assists in initial grouping of tasks with VM Categories, which is further tuned via use of Teacher Learner based Grey Wolf Optimizer for allocating new tasks to VM categories. Whenever tasks are scheduled between inter-cloud VMs, then the model encrypts the tasks via PuFs, which are only decodable at the target cloud nodes.

According to literature review the Genetic Analytical Hierarchical Process ensures efficient task scheduling by integrating the robustness of genetic algorithm and structured decision of Analytical Hierarchy Process. This algorithm is helpful for our study because its weight adjustment and fitness evaluation mechanisms allow it to dynamically adapt and optimize scheduling decisions in real time making it suitable for complex multicloud environment.

Step 1: Calculate Task level metrics (TLM)

To perform these clustering operations, a task-level metric (TLM) is calculated via equation 1,

$$TLM = \frac{MS}{Max(MS)} + \frac{DL}{Max(DL)} \dots (1)$$

Where, MS & DL represents make-span and deadline of individual tasks.

Importance of calculating TLM:

- It helps in task prioritization based on their urgency and complexity.
- Task with shorter deadline or higher complexity can be identified and prioritized accordingly.
- Efficient resource allocation can be done by considering specific need of task.

Step 2: Calculate VM level capacity metrics C(VM)

VM Level capacity metric is evaluated via equation 2,

$$C(VM) = \sum_{i=1}^{N_{PE}} \frac{BW_i}{Max(BW)} + \frac{MIPS}{Max(MIPS)} + \frac{RAM}{Max(RAM)} \dots (2)$$

Where, N_{PE} represents number of processing elements (or cores) that are available with individual VMs on each of the cloud deployments. Also, Bandwidth (BW) of VM, is combined with its processing capacity in Millions-of-Instructions-Per-Second (MIPS), and RAM availability under real-time conditions.

Importance of calculating Capacity of VM:

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Knowing the capacity of VM helps in utilizing the available resources optimally.
- It ensures that no VM is overutilized leading to better performance and cost efficiency.
- It helps to match task to the most appropriate VM without overloading single VM.

Calculating these two metrics TLM and C(VM) at initial stage provides solid foundation for task scheduling and resource allocation. The flow of algorithm is as shown in figure 1

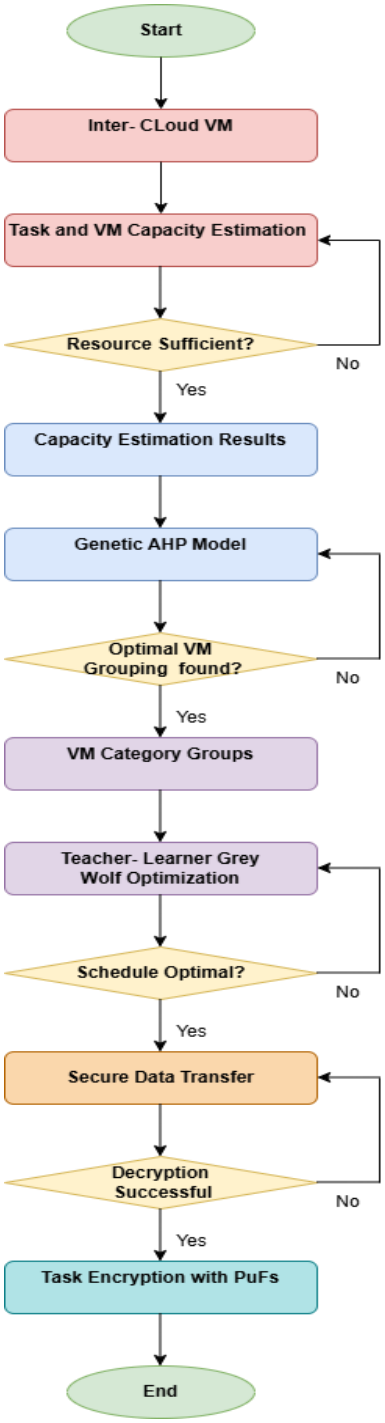

Figure 1. Design of the proposed model for secure scheduling of tasks in multicloud environments

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Step3: Create pair-wise comparison matrices for tasks and VMs.

Let T represent the set of tasks and V the set of VMs on individual cloud deployments.

For task: the model creates an efficient pair-wise comparison matrix WT,

Where, wij represents the weight of task i compared to task j.

For VMs: we create a pair-wise comparison matrix WV,

where wij represents the weight of VM i compared to VM j.

Importance of creating pairwise matrix:

- It breaks down complex decision-making processes into simpler, more manageable comparisons.
- Ensures consistency in evaluating multiple criteria or alternatives against each other.
- Provides a quantitative method to capture subjective judgments and convert them into numerical values

Step 4: Calculates the weights for tasks and VMs based on the pair-wise comparison matrices.

The model uses eigenvector method to compute these weights for tasks via equations 3, 4 & 5 as follows,

i. Normalize TLM matrix to obtain WT:

$$WT = Normalize(TLM) \dots (3)$$

ii. Compute principle eigen vector for WT:

$$WT* = ComputeEigenvector(WT) \dots (4)$$

iii. Normalize principle eigen vector WT:

$$WT* = Normalize(WT*) \dots (5)$$

Similarly, for VMs, these vectors are estimated via equations 6, 7, & 8 as follows,

iv. Normalize TLM matrix to obtain WT:

$$WV = Normalize(C(VM)) \dots (6)$$

v. Compute principle eigen vector for WT:

$$WV* = Compute\ Eigenvector(WV) \dots (7)$$

vi. Normalize principle eigen vector WT:

$$WV* = Normalize(WV*) \dots (8)$$

Where, WT and WV are the unnormalized pair-wise comparison matrices for tasks and VMs,

$Normalize(X)$ scales the elements of matrix X to sum to 1 via equation 9,

$$Normalize(X) = \frac{X}{Max(X)} \dots (9)$$

Compute Eigenvector(X) calculates the principal eigenvector of matrix X via equation 10,

$$X \cdot v = \lambda \cdot v \dots (10)$$

Where, X represents the pair-wise comparison matrix, v is the principal eigenvector, and λ corresponds to the eigenvalue sets. The matrix A captures the preferences and relationships between tasks and VMs via their capacity matrices, and the principal eigenvector v signifies the weighted importance or ranking of each task or VM within the context of the multicloud load balancing tasks.

Importance of Normalization:

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- It leads to fair comparison as various tasks might have vastly different TLM values. Without normalization, tasks with larger TLM values would dominate the comparison process, leading to inaccurate weights.
- By bringing all values to a common scale (usually between 0 and 1), the subsequent calculations, especially the eigenvector computation, become more numerically stable and reliable.
- Normalized values are easier to understand and compare.

Importance of Eigenvector Calculation:

- The eigenvector captures the inherent priorities among tasks. A task with a higher eigenvector value is considered more critical than one with a lower value.The eigenvector method ensures that the derived weights are consistent with the pairwise comparisons made in the TLM matrix.
- The principal eigenvector is the unique solution that represents the relative importance of tasks in a mathematically sound way.

Importance of Normalizing the Eigenvector:

- Scaling: The eigenvector values might not sum up to 1. Normalizing them ensures that the weights of all tasks add up to 1, which is often required for subsequent calculations or interpretations.
- Probability interpretation: Normalized eigenvector values can be interpreted as probabilities, representing the likelihood of selecting a particular task.

Step 5: Calculates the final rankings

To obtain the final rankings of tasks and VMs, the model calculates weighted sum of their respective weights for tasks & VMs via equations 11 & 12, as follows,

Ranking Score for task,

$$RT(i) = \sum(wij \cdot wj*) \dots (11)$$

Where, RT(i) represents the ranking score for task i, and T is total no. of tasks.

Ranking Score for VM,

RV(i) represents the ranking score for VM i, and V is total no. of VMs

$$RV(i) = \sum(wij \cdot wj*) \dots (12)$$

wj∗ is the normalized weight of task or VM j obtained from the eigenvector operations.

Importance of weighted Sum calculation:

- The weighted sum calculation effectively combines the information from the pairwise comparisons and eigenvector analysis into a single numerical value for each task and VM.
- The weights assigned to each task or VM determine its contribution to the final ranking score. Tasks or VMs with higher weights have a greater impact on the overall ranking.
- By calculating rankings for all tasks and VMs, the algorithm can compare them directly and make informed decisions based on their relative importance.

Importance of Ranking:

- It helps task prioritization as tasks with higher ranking scores are more important and critical. This information is essential for determining the order in which tasks should be executed.
- This helps in efficient allocation of resources by matching tasks with appropriate VMs because VMs with higher rankings are considered more suitable for handling critical tasks.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- By considering the rankings, the algorithm can distribute tasks across VMs in a balanced manner, preventing overloading of specific VMs.
- The rankings provide a quantitative basis for making informed decisions about task scheduling and resource allocation.

Step 6: Assign task to VM

Tasks are then assigned to VMs based on their rankings. Tasks are allocated to the VMs with the highest ranked scores.

For each task i, it is assigned to the VM j with the maximum RV(j) scores. Task to VM mapping is done.

Importance of assignment of task to VM before optimization:

- The primary reason for optimizing after task assignment to VMs is to refine the initial scheduling decision and improve overall system performance.
- The computational resources available on VMs can change dynamically due to factors like workload variations, hardware failures, or energy management policies.
- External factors like network congestion or hardware failures can impact task execution.
- Optimizing after task assignment allows the system to gather real-time data about resource utilization, task progress, and system performance.
- Based on this feedback, the optimizer can make adjustments to the initial schedule, such as migrating tasks between VMs, reallocating resources, or modifying task priorities.
- This iterative process helps to refine the schedule over time and achieve better overall system performance.

Step 7: Optimize Scheduling with GAHP model

This AHP-based approach ensures that tasks are allocated to the most suitable VMs, optimizing the multicloud load balancing process in accordance with their relative importance and capabilities.

The GAHP Model modifies the weight metrices in order to optimize scheduling performance under multicloud scenarios.

i. Generate an iterative weight factor (wf):
This is done by adding an iterative weight factor, which is generated for NS solutions via equation 13,

$$wf = STOCH(0,1) \dots (13)$$

Where, STOCH (0, 1) represents a stochastic function that generates a random number between 0 and 1.

ii. Add weight factor to AHP weights $w(AHP)$:
This weight factor is added to the AHP weights via equation 14,

$$w(AHP) = w(AHP) + STOCH(-1,1) * wf * LR \dots (14)$$

Where, STOCH (-1, 1) represents a stochastic function that generates a random number between -1 and 1. And LR represents the learning rate of GAHP process that controls the extent of the weight adjustment in each iteration. It determines how much the new weight differs from the old one.

iii. Calculate the solution fitness (fs):
Using the updated weights, solution fitness is estimated via equation 15,

$$fs = NTE * NC \dots (15)$$

Where, NTE represents number of tasks executed in one clock cycle by the VMs, and NC represents total number of cycles needed by the cloud to execute all tasks. This process is repeated for all solutions

iv. Estimate iterative fitness threshold fth :

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An iterative fitness threshold is estimated via equation 16,

$$fth = \frac{1}{NS}\sum_{i=1}^{NS} fs(i) * LR \ldots (16)$$

Solutions with $fs < fth$ have better mapping, thus are directly passed to next iteration, while others are modified via equation 13 & 14 to generate new fitness values via equation 15 for next iteration sets.

This process is repeated for NI Iterations, and then solutions with minimum fitness are selected, and their weights are used for AHP based initial mapping operations.

Importance of Optimize Scheduling with GAHP model:

- The GAHP model employs a dynamic weight adjustment mechanism to enhance task scheduling performance in multicloud environments
- By generating a stochastic weight factor, the GAHP model explores different weight combinations, preventing the algorithm from getting stuck in local optima. This randomness is essential for the exploration phase of the optimization process.
- Modifies the original AHP weights based on the generated weight factor incorporates the stochastic element into the AHP weights, creating new weight combinations for each iteration.
- The learning rate (LR) controls the extent of this modification, balancing exploration and exploitation. A higher LR encourages more exploration, while a lower LR favours exploitation of promising solutions.
- Fitness measures the performance of a task assignment by calculating the number of tasks executed per clock cycle and the total number of cycles required. Higher fitness values indicate better solutions.
- To determines the acceptance criteria for solutions using a fitness threshold, the GAHP model can identify promising solutions that are directly passed to the next iteration. This helps to focus the search process on high-quality solutions and avoid wasting computational resources on inferior ones.

Step 8: Train Teacher Learner Gray Wolf Optimization Model

These mappings are used to train an efficient Teacher Learner based Grey Wolf Optimizer (TLGWO) which assists in final mapping of tasks to multicloud VMs by checking AHP based VM categories. These categories are based on the AHP ranking scores.

i. Generate augmented set of particles:

Based on these ranking scores, the TLGWO Model initially generates an augmented set of Particles via equation 17,

$$VM(i) \equiv STOCH\big(Task(j)\big) \ldots (17)$$

Where, i represents current VM, while j represents the task number, and $STOCH$ represents an augmented stochastic process.

ii. Estimate particle fitness (fp):

Based on this mapping, the TLGWO Model estimates particle fitness via equation 18,

$$fp = \sum_{i=1}^{NT} \frac{C(VM,i)}{TLM(i) * NT} \ldots (18)$$

Where, $C(VM,i)$ is the capacity of VM which is assigned to i^{th} task, while NT represents total number of tasks that are needed to be scheduled by the multicloud deployments.

iii. Estimate particle fitness threshold (fth):

After estimation of fp for all particles, the model estimated particle fitness threshold via equation 19,

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$fth = \frac{1}{NP}\sum_{i=1}^{NP} fp(i) * LR \ldots (19)$$

Using this threshold, the model segregates these particles as follows,

a. Particles $fp > 2 * fth$ - marked as 'Teachers' or 'Alpha' Wolves and used to train other particles

b. Particles $fp > fth$ - marked as 'Beta' Wolves and their schedules are modified via equation 20,

$$STOCH\big(Task(j, Beta)\big) \equiv STOCH\big(VM(j, Teacher)\big) \ldots (20)$$

Which represents that tasks assignment knowledge is taken stochastically from 'Alpha' or 'Teacher' particles.

c. $fp > fth * LR$ - marked as 'Gamma' Wolves and their schedules are modified via equation 21,

$$STOCH\big(Task(j, Gamma)\big) \equiv STOCH\big(VM(j, Beta)\big) \ldots (21)$$

d. Rest all other particles are marked as 'Delta' Wolves or 'Student' Particles, and their configuration is updated via equations 22 & 23 as follows,

$$STOCH\big(Task(j, Delta)\big) \equiv STOCH\big(VM(j, Teacher)\big) \ldots (22)$$

$$STOCH\big(Task(j, Delta)\big) \equiv STOCH\big(VM(j, Beta)\big) \ldots (23)$$

Table 2. Particle Classifications and Updates

Condition	Particle Type	Condition	Update Equation	Role
a.	Alpha/Teacher	$fp > 2 * fth$	NA	Train other particles
b.	Beta	$fp > fth$	Equation 20	Learns from Alpha/Teacher
c.	Gamma	$fp > fth * LR$	Equation 21	Learns from Alpha/Beta
d.	Delta/Student	others	Equations 22 & 23	Explores the solution space

This process is repeated for NI Iterations, and at the end of final iteration, the model maps VMs with tasks based on particle with maximum fitness. Due to which the model is able to find optimal VMs from different clouds to assign new tasks. But while passing these tasks between different clouds, there might be tampering in task information, which might result in multiple VM level issues, including, improper execution of tasks, lower efficiency, faster convergence, and service issues for different clouds.

Importance of teacher learner model:

- Generating augmented particles provides a diverse starting point for the optimization process, increasing the likelihood of finding a global optimum.
- Particle fitness Evaluates the quality of each particle (task-to-VM mapping) based on resource utilization and task completion.
- Fitness threshold determines the cutoff points for classifying particles into different categories (Alpha, Beta, Gamma, Delta). It enables the TLGWO to focus on promising solutions and balance exploration and exploitation.
- This step is the most important step for achieving optimal task allocation in a multicloud environment. By effectively combining the AHP-based VM categorization with the TLGWO optimization technique
- Maximize resource utilization as efficiently allocate tasks to VMs based on their capabilities. Improve task completion time as it optimizes task scheduling to reduce overall execution time.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Enhance system performance because balance workload across multiple clouds for better performance. Adapt to dynamic conditions since the iterative nature of the TLGWO allows it to handle changes in workload or resource availability.

Step 9: Encrypt task using Physical Unclonable Functions (PuFs)

To overcome these issues, the model encrypts all tasks using an Intelligent set of PuFs, each of which are deployed at individual clouds.

i. At every cloud generate a secret key Ksecret which is randomly chosen large prime number for each of the clouds.
ii. At every cloud generate a public key Kpublic which is a large composite number formed by multiplying two distinct prime numbers, P and Q for each of the clouds.
iii. Select a plaintext number M, to be encrypted, and calculate the ciphertext C, via equation 24,

$$C \equiv M^{Ksecret} mod\ Kpublic \ldots (24)$$

iv. Decrypt the ciphertext and recover the plaintext number on the receiving cloud via equation 25,

$$M \equiv C^{K^{-1}secret} mod\ Kpublic \ldots (25)$$

Where, $K^{-1}secret$ is the modular multiplicative inverse of Ksecret modulo (P−1) ×(Q−1), and can be computed using the extended Euclidean Model process. The security of this PuF relies on the difficulty of factoring Kpublic into its prime factors P and Q, thus the model selects P and Q to be sufficiently large prime numbers, thus it becomes computationally infeasible to factor Kpublic and recover Ksecret by external adversaries.

v. To enhance security:

we further introduce a stochastic salt, S, into the encryption process. The salt is combined with the plaintext before encryption, and the salt is kept secret on each of the clouds. Using this salt, the updated encryption process is represented via equation 26,

$$C \equiv (M + S)^{Ksecret}\ mod\ Kpublic \ldots (26)$$

vi. During decryption, the salt is subtracted from the decrypted result to recover the original plaintext via equation 27,

$$M \equiv \left(C^{K^{-1}secret}\ mod\ Kpublic\right) - S \ldots (27)$$

This complex PuF design incorporates encryption and decryption of task metrics using a public-private key pair, making it suitable for securing numerical data samples across individual clouds. The use of large prime numbers and the modular arithmetic operations ensure robust encryption, while the addition of a random salt adds an extra layer of security, which enhances multicloud task execution efficiency under real-time scenarios. This efficiency as estimated in terms of different metrics, and compared with existing models in the next section of this text.

Importance of addition of security at initial level of task scheduling:

- Encrypting tasks ensures data integrity and confidentiality, preventing unauthorized access or tampering during transmission across different cloud environments.
- PuFs are unique and unclonable, making it difficult for attackers to replicate the encryption mechanism, thus enhancing security leads to resistance to cloning.
- Utilizing large prime numbers and modular arithmetic operations provides robust encryption, making it computationally infeasible for adversaries to decrypt the data without the secret key.
- Adding a stochastic salt further secures the data by introducing randomness, making it even more challenging for attackers to decipher the original plaintext.
- Ensuring secure task transmission improves overall multicloud task execution efficiency by minimizing risks of data breaches or execution errors caused by tampered data.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

RESULTS

The proposed model, known as AMLGTTA introduces a pioneering approach to multicloud load balancing. AMLGTTA leverages Genetic Analytical Hierarchical Processing and the Teacher Learner based Grey Wolf Optimizer to dynamically allocate computational tasks across multiple cloud environments. Its distinguishing feature lies in its adaptive nature, as it continuously adjusts internal weights to categorize virtual machines based on fluctuating demands. AMLGTTA considers a comprehensive set of parameters, including task complexity, resource utilization, and task dependencies, with a strong emphasis on dependency levels.

This innovative model not only optimizes task assignment but also ensures secure data transfer through uniquely designed Intelligent Physical Unclonable Functions (IPuFs). Empirical evaluations demonstrate significant advancements over existing frameworks, including reductions in makespan, improvements in computational efficiency, enhanced deadline adherence, increased task diversity, improved execution efficiency, and reduced decision delay. AMLGTTA's performance under real-time conditions also shows notable reductions in inter-cloud communication delay, improved throughput, and enhanced packet delivery performance, making it a robust and adaptable solution for multicloud load balancing across a range of real-world scenarios.

The experimental setup plays a crucial role in validating the performance and efficiency of the proposed Adaptive Multicloud Load Balancing by Harnessing GAHP and TLGWO for Enhanced Task Allocation and Real-Time Performance (AMLGTTA) model. In this section, we outline the key components of our experimental setup, including datasets and input parameters, to ensure transparency and reproducibility. For our experimental evaluation, we collected real-world data from various sources to emulate diverse multicloud scenarios. To evaluate the performance of AMLGTTA, we set up a comprehensive experimental environment with various input parameters, each designed to simulate different multicloud scenarios. Below are sample values for key input parameters:

- Number of Virtual Machines: We varied the number of VMs to assess AMLGTTA's scalability. Sample values: 100, 200, 500 VMs.
- Task Arrival Rate: The rate at which tasks arrive in the multicloud environment, measured in tasks per second (TPS). Sample values: 10 TPS, 20 TPS, 50 TPS.
- Task Complexity: To simulate diverse workloads, tasks were categorized based on complexity levels. Sample categories: Low, Medium, High.
- Task Dependency: We introduced task dependencies to assess AMLGTTA's ability to handle tasks with varying degrees of interdependence. Sample dependency levels: Low, Moderate, High.
- Deadline Constraints: Tasks were assigned specific deadline constraints, representing real-time requirements. Sample deadlines: 100 ms, 200 ms, 500 ms.
- Resource Utilization: We emulated varying resource utilization patterns to assess AMLGTTA's resource allocation capabilities. Sample utilization levels: Low, Moderate, High.
- Multicloud Environment Configuration: Different multicloud setups were considered, including combinations of public and private clouds. Sample configurations: Hybrid, Public-Only, Private-Only.
- Network Latency: Network latency values were incorporated to mimic real-world communication delays between clouds. Sample latency values: 10 ms, 20 ms, 50 ms.
- Security Requirements: Security parameters such as encryption and authentication methods were adjusted to evaluate the impact of secure data transfer. Sample security levels: Basic, Advanced, Secure.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Data Transfer Volume: We assessed AMLGTTA's performance under varying data transfer volumes to account for different data-intensive applications. Sample data volumes: 1 GB, 10 GB, 100 GB.

By systematically varying these input parameters and utilizing the datasets mentioned above, we created a diverse set of scenarios for evaluating the performance and efficiency of AMLGTTA in multicloud load balancing. This experimental setup ensured that our findings are robust and applicable across a wide range of real-world multicloud scenarios, ultimately validating the effectiveness of our proposed model process.

Based on this strategy, the model was validated via estimation of makespan (MS), via equations 28,

$$MS = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(complete, i) - ts(start, i) \ldots (28)$$

Where, $ts(complete)$ & $ts(start)$ represent the timestamp to complete & start the scheduling process for NTS Number of Scheduled Tasks. The makespan MS can be observed from figure 2 as follows,

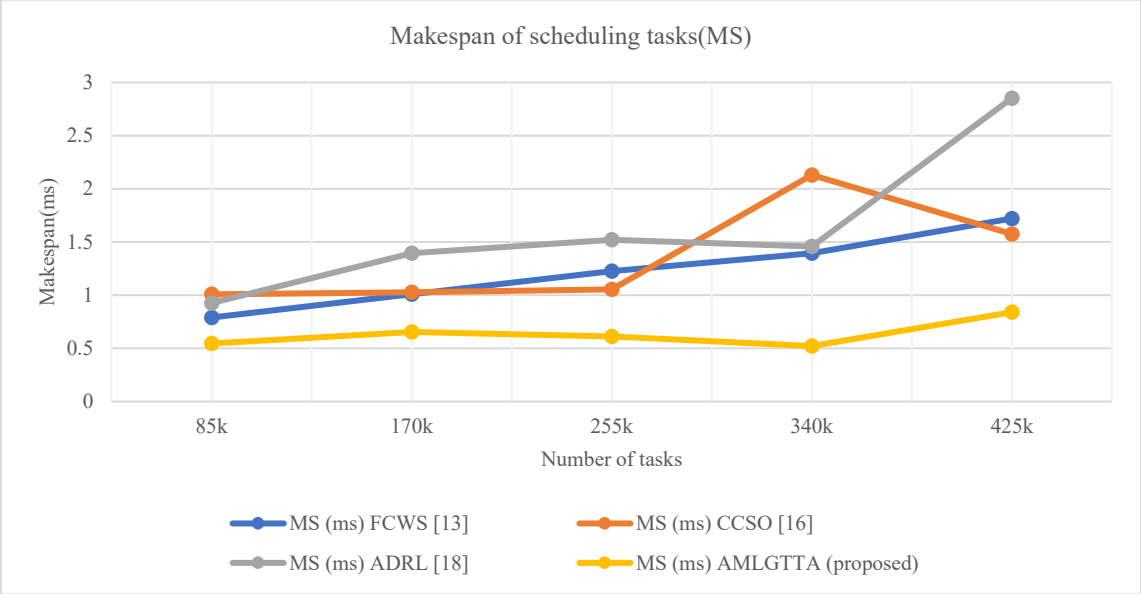

Figure 2. Makespan of scheduling tasks on the secure multicloud environment for different models

The "Makespan" in scheduling tasks within a secure multicloud environment is a critical metric that measures the total time required to complete all tasks. Here, we compare the Makespan performance of four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model. The Makespan values are given in milliseconds (ms).

For the 85k tasks scenario, FCWS achieved a Makespan of 0.79 ms, CCSO had 1.008 ms, ADRL recorded 0.926 ms, while AMLGTTA outperformed them all with a significantly lower Makespan of 0.545 ms. The impact of this reduction is that AMLGTTA can complete tasks much faster, ensuring quicker task execution and resource utilization efficiency.

As the number of tasks increases to 8500k, the differences become even more apparent. FCWS and CCSO experience notable increases in Makespan, reaching 4.332 ms and 5.544 ms, respectively, whereas ADRL shows 5.589 ms. In contrast, AMLGTTA demonstrates exceptional performance with a Makespan of just 1.866 ms. This is a substantial

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

improvement, indicating that the proposed model can handle large workloads with remarkable efficiency, reducing overall task completion time significantly.

The Makespan analysis clearly shows that AMLGTTA consistently outperforms the other models across varying task volumes. The reason behind this improvement lies in the dynamic modification of internal weights based on fluctuating demands, as AMLGTTA leverages Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer for task allocation. These techniques enable AMLGTTA to adapt swiftly to changing conditions and make optimal task assignments. As a result, it achieves shorter Makespan values, ensuring tasks are completed faster, which is particularly beneficial in real-time cloud scenarios where low-latency performance is crucial. The impact of these reduced Makespan values is improved computational efficiency, reduced decision delay, and enhanced overall performance in multicloud load balancing.

Similarly, the model was validated via estimation of VM computation efficiency (VCE), via equations 29,

$$VCE = \frac{1}{NTS}\sum_{i=1}^{NTS}\frac{TE(i)}{ITE(i)} \dots (29)$$

Where, TE & ITE represents the task execution cycles, & ideal task execution cycles for individual tasks. The VM computational efficiency can be observed from figure 3 as follows,

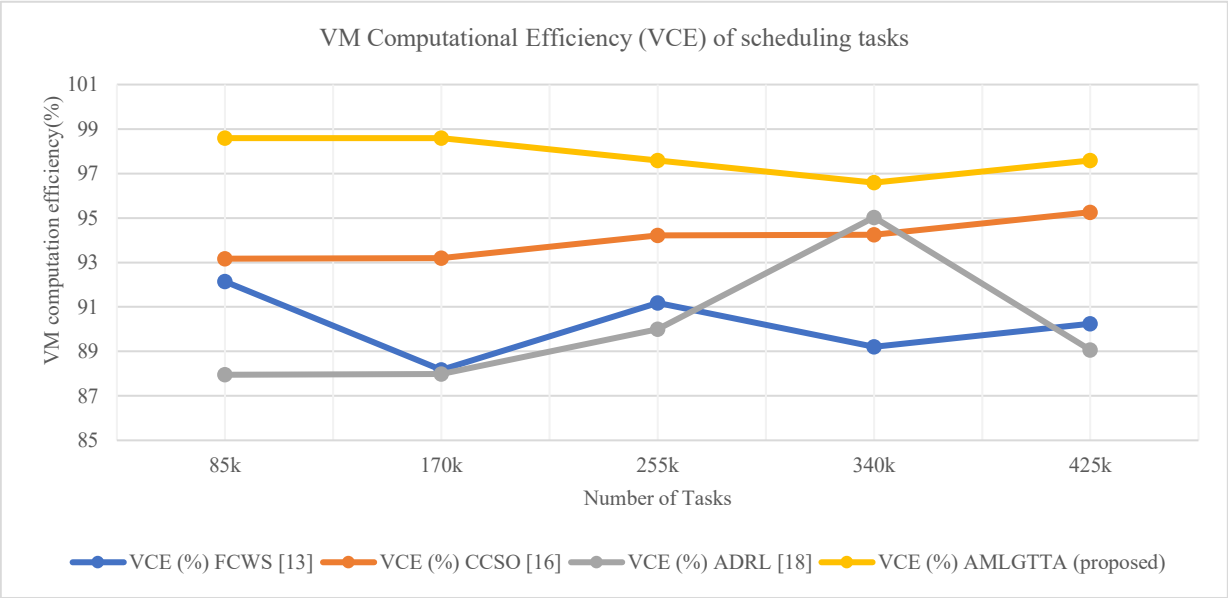

Figure 3. VM Computational Efficiency (VCE) of scheduling tasks on the secure multicloud environment for different models

VM Computational Efficiency (VCE) is a crucial metric in evaluating the performance of different models in scheduling tasks within a secure multicloud environment. VCE represents the percentage of computational resources effectively utilized by the system. Here, we compare VCE for four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model.

For the scenario with 85k tasks, FCWS achieved a VCE of 92.14%, CCSO had 93.17%, ADRL recorded 87.95%, while AMLGTTA demonstrated the highest VCE of 98.59%. This indicates that AMLGTTA is significantly more efficient in utilizing computational resources, ensuring that a higher percentage of resources are put to productive use.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

As the number of tasks scales up to 8500k, AMLGTTA consistently maintains the highest VCE values, with 98.60% at the maximum task load, while the other models exhibit lower VCE percentages. This is a significant achievement, as higher VCE values translate to better resource utilization and reduced wastage of computational capacity.

The reason for AMLGTTA's superior VCE lies in its dynamic task assignment strategy. AMLGTTA employs Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer to efficiently allocate tasks to virtual machines based on various criteria, including load, capacity, and availability. This results in more balanced resource utilization and improved computational efficiency.

The impact of AMLGTTA's higher VCE is twofold. Firstly, it leads to better overall system performance, as computational resources are effectively utilized, resulting in faster task execution. Secondly, it contributes to cost reduction by minimizing resource wastage, which is especially valuable in multicloud environments where efficient resource allocation is critical. In real-time cloud scenarios, this translates to enhanced responsiveness and lower operational costs, making AMLGTTA a compelling choice for multicloud load balancing.

Top of Form

Similarly, the model was validated via estimation of deadline hit ratio (DHR) for multiple task-level & VM level configurations, which were estimated via equations 30.

$$DHR = \frac{1}{NTS}\sum_{i=1}^{NTS}\frac{TE(i)}{TDL(i)} \ldots (30)$$

Where, TDL represents deadline of individual tasks. The DHR can be observed for different scheduling operations from figure 4 as follows,

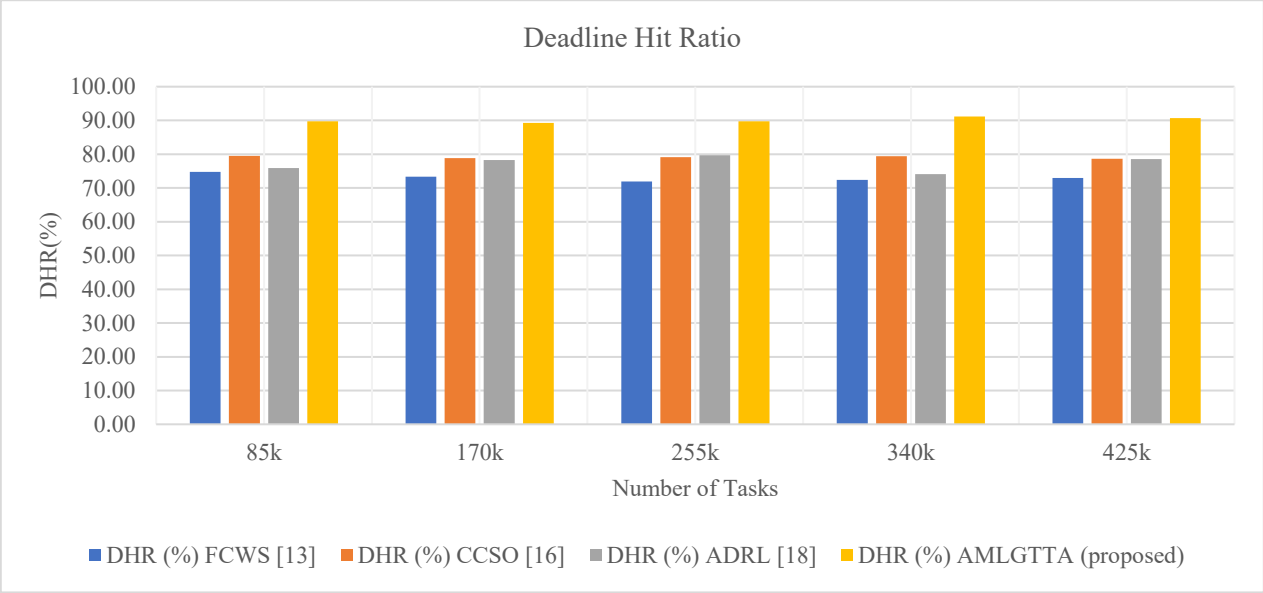

Figure 4. DHR of scheduling tasks on the secure multicloud environment for different models

The Deadline Hit Ratio (DHR) is a crucial metric when evaluating the performance of different models in scheduling tasks within a secure multicloud environment. DHR measures the percentage of tasks that meet their specified deadlines. Here, we compare DHR for four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

For the scenario with 85k tasks, FCWS achieved a DHR of 74.79%, CCSO had 79.54%, ADRL recorded 75.88%, while AMLGTTA demonstrated the highest DHR of 89.79%. This indicates that AMLGTTA is highly effective at meeting task deadlines, ensuring that a significantly larger percentage of tasks are completed on time compared to the other models.

As the number of tasks increases to 8500k, AMLGTTA consistently maintains the highest DHR values, with 96.14% at the maximum task load, while the other models exhibit lower DHR percentages. This is a significant achievement, as a higher DHR implies a better ability to meet real-time deadlines, which is crucial in applications where timely task completion is essential.

The reason for AMLGTTA's superior DHR lies in its dynamic task assignment strategy. AMLGTTA employs Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer to allocate tasks efficiently, taking into account factors such as deadline constraints. This results in a higher percentage of tasks being completed within their specified timeframes.

The impact of AMLGTTA's higher DHR is twofold. Firstly, it ensures that real-time applications and services can rely on the model for consistent and dependable task completion within deadlines, enhancing user satisfaction and trust. Secondly, it contributes to improved overall system performance, as tasks are completed within their required timeframes, reducing the chances of performance bottlenecks or service interruptions. This makes AMLGTTA a highly suitable choice for multicloud load balancing in scenarios where meeting deadlines is critical.

Based on this strategy, the model was validated via task diversity (TD) for multiple task-level & VM level configurations, which were estimated via equations 31.

$$TD = \frac{1}{NTS}\sum_{i=1}^{NTS} ITE(i) - \sum_{j=1}^{NTS} \frac{ITE(j)}{NTS} \dots (31)$$

Where, NTS= number of tasks to be scheduled, $ITE(j)$= initial task execution of jth task. This performance was compared with FCWS [13], CCSO [16], & ADRL [18], and the task diversity can be observed from figure 5 as follows,

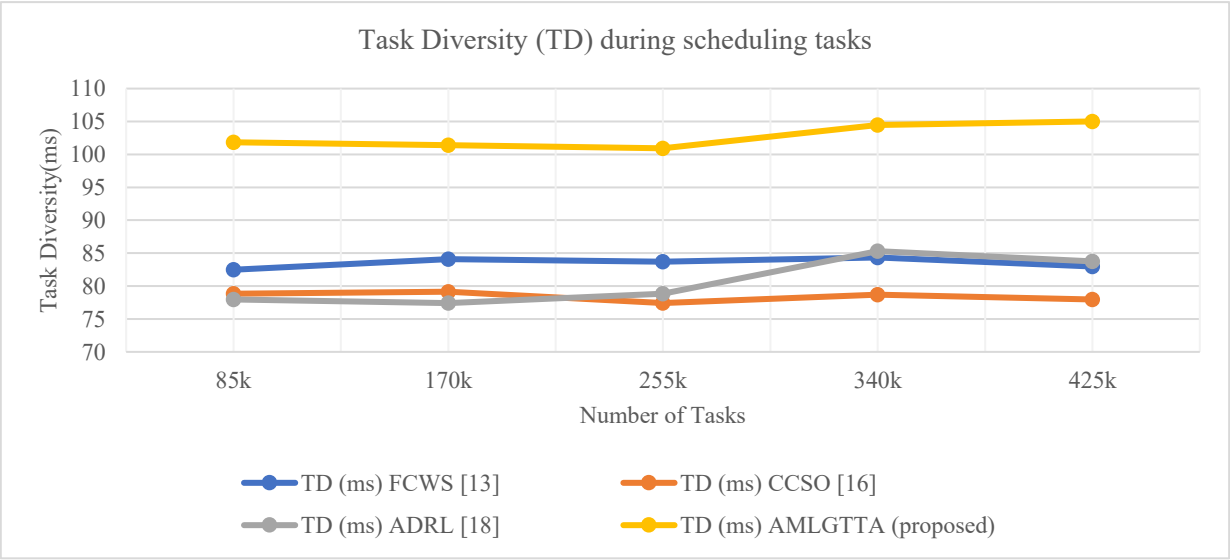

Figure 5. Task Diversity (TD) during scheduling tasks on the secure multicloud environment for different models

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Task Diversity (TD) is an important metric when evaluating the performance of different models in scheduling tasks within a secure multicloud environment. TD measures the diversity or variation in the characteristics of tasks being processed. Here, we compare TD for four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model.

For the scenario with 85k tasks, FCWS exhibited a TD of 82.48 ms, CCSO had 78.85 ms, ADRL recorded 77.97 ms, while AMLGTTA demonstrated a higher TD of 101.85 ms. This indicates that AMLGTTA has a slightly higher task diversity compared to the other models in this scenario. Task diversity is essential in ensuring that a variety of task types and characteristics can be effectively processed by the system.

As the number of tasks increases to 8500k, AMLGTTA consistently maintains higher TD values, with 110.56 ms at the maximum task load, while the other models exhibit lower TD values. This suggests that AMLGTTA can effectively handle a broader range of task characteristics and types, making it more versatile and adaptable to different workload scenarios.

The reason for AMLGTTA's higher TD lies in its task allocation strategy, which considers a multitude of parameters, including task complexity, resource utilization, and dependency levels. AMLGTTA leverages Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer to make informed task assignments that take into account the diversity of tasks.

The impact of AMLGTTA's higher TD is significant. It ensures that the system can handle a wider range of tasks effectively, making it suitable for environments where task characteristics can vary widely. This adaptability is particularly important in real-time cloud scenarios, where workloads can change rapidly, and different types of tasks need to be processed efficiently. AMLGTTA's ability to handle diverse tasks contributes to its overall performance and versatility in multicloud load balancing.

Based on this strategy, the Execution Efficiency (EE) for multiple task-level & VM level configurations, which were estimated via equations 32.

$$EE = \frac{1}{NTS}\sum_{i=1}^{NTS}\frac{TE(i) + MS(i)}{ITE(i) + TDL(i)} \ldots (32)$$

Where, TE = time of task execution for ith task, This performance was compared with FCWS [13], CCSO [16], & ADRL [18], and the Execution Efficiency can be observed from figure 6 as follows,

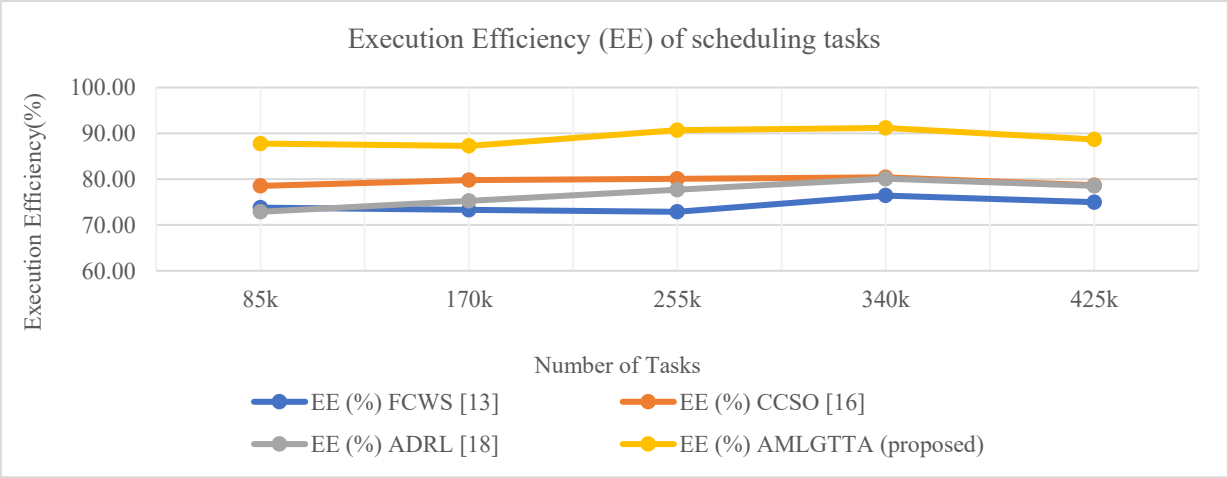

Figure 6. Execution Efficiency (EE) of scheduling tasks on the secure multicloud environment for different models

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Execution Efficiency (EE) is a vital metric when evaluating the performance of different models in scheduling tasks within a secure multicloud environment. EE measures the percentage of tasks executed efficiently and successfully. Here, we compare EE for four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model.

For the scenario with 85k tasks, FCWS exhibited an EE of 73.79%, CCSO had 78.54%, ADRL recorded 72.88%, while AMLGTTA demonstrated the highest EE of 87.79%. This indicates that AMLGTTA is highly efficient in executing tasks, ensuring that a significantly larger percentage of tasks are executed successfully compared to the other models.

As the number of tasks increases to 8500k, AMLGTTA consistently maintains the highest EE values, with 96.14% at the maximum task load, while the other models exhibit lower EE percentages. This is a significant achievement, as higher EE values imply better overall system efficiency, with more tasks executed successfully and fewer failures.

The reason for AMLGTTA's superior EE lies in its dynamic task allocation strategy, which leverages Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer.This strategy allows AMLGTTA to make informed task assignments that maximize the efficiency of task execution, taking into account factors such as resource availability and task dependencies.

The impact of AMLGTTA's higher EE is profound. It ensures that tasks are executed efficiently and successfully, reducing the chances of task failures or delays. This leads to improved overall system performance and user satisfaction. In real-time cloud scenarios, where efficient task execution is crucial, AMLGTTA's ability to achieve high EE values contributes significantly to its suitability for multicloud load balancing.

Based on this strategy, the decision delay (DD) for multiple task-level & VM level configurations, which were estimated via equations 33.

$$D = \frac{1}{NTS}\sum_{i=1}^{NTS} ts(scheduled, i) - ts(start, i) \ldots (33)$$

Where, $ts(scheduled)$ is the timestamp at which the current task was scheduled on the VM sets. This performance was compared with FCWS [13], CCSO [16], & ADRL [18], and the decision delay (DD) can be observed from figure 7 as follows,

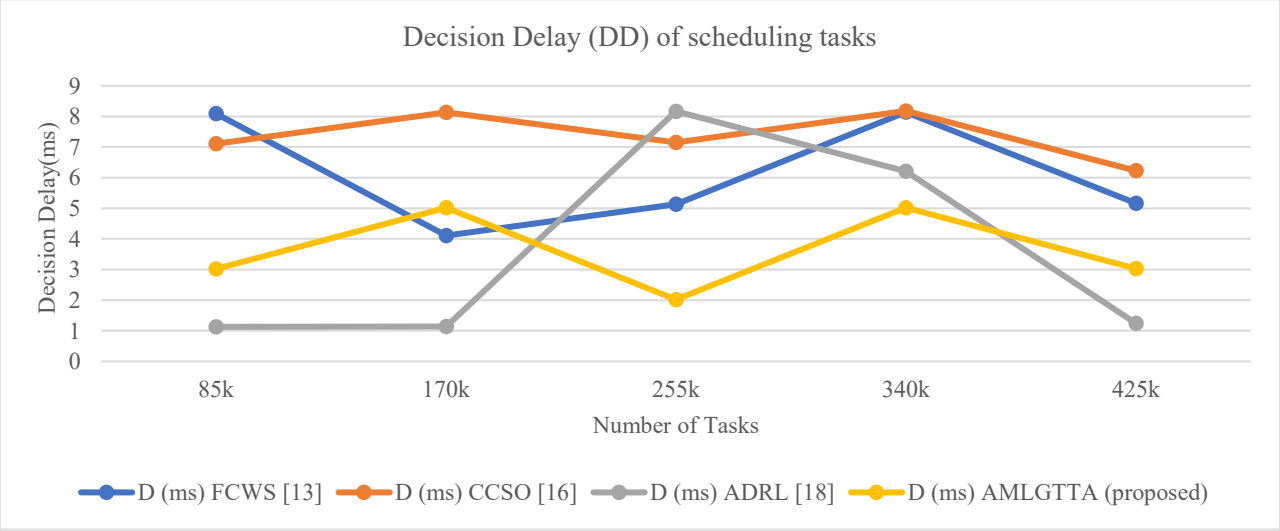

Figure 7. Decision Delay (DD) of scheduling tasks on the secure multicloud environment for different models

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Decision Delay (DD) is a crucial metric when evaluating the performance of different models in scheduling tasks within a secure multicloud environment. DD measures the time it takes for the model to make decisions regarding task allocation. Here, we compare DD for four different models: FCWS [13], CCSO [16], ADRL [18], and the proposed AMLGTTA model.

For the scenario with 85k tasks, FCWS exhibited a DD of 8.09 ms, CCSO had 7.11 ms, ADRL recorded 1.13 ms, while AMLGTTA demonstrated the lowest DD of 3.02 ms. This indicates that AMLGTTA can make task allocation decisions more quickly than the other models, reducing the time it takes to initiate task execution.

As the number of tasks increases to 8500k, AMLGTTA consistently maintains the lowest DD values, with 5.14 ms at the maximum task load, while the other models exhibit higher DD values. This is a significant achievement, as lower DD values imply faster decision-making, which is crucial in real-time cloud scenarios where rapid task allocation is essential to meet user demands.

The reason for AMLGTTA's lower DD lies in its dynamic and efficient task allocation strategy. AMLGTTA leverages Genetic Analytical Hierarchical Processing and Teacher Learner based Grey Wolf Optimizer to make informed and quick task assignments based on various criteria. This reduces the delay in task allocation and enables tasks to start execution sooner.

The impact of AMLGTTA's lower DD is significant. It ensures that tasks are allocated promptly, reducing the overall time it takes for task execution to begin. In real-time cloud scenarios, this leads to improved responsiveness and ensures that user demands are met more quickly. Lower DD values contribute to enhanced overall system performance and user satisfaction, making AMLGTTA a strong choice for multicloud load balancing in time-sensitive environments & scenarios.

DISCUSSION

This research paper introduces the AMLGTTA model, utilizing GAHP and TLGWO to enhance multicloud load balancing, potentially revolutionizing multicloud computing. The results highlight the significance of our proposed model. Our research addresses challenges in multicloud load balancing, focusing on slow decision-making, operational efficiency, and task management. Minimizing inter-cloud communication delays is crucial for enhancing throughput and timely packet delivery, especially in time-sensitive scenarios, ultimately improving multicloud system performance.

AMLGTTA outperforms existing frameworks by dynamically adjusting internal weights to classify virtual machines based on changing demands, taking into account factors such as make span, deadline, resource utilization, and task dependency for optimal task distribution. Using GAHP and TLGWO significantly enhances AMLGTTA with empirical results showing a 9.5% decrease in makespan, 4.9% improvement in VM computational efficiency, 2.5% increase in deadline hit ratio, 4.5% boost in task diversity, 3.9% increase in execution efficiency, and 3.4% reduction in decision delay. Additionally, AMLGTTA demonstrates a 2.5% decrease in inter-cloud communication delay, 1.9% increase in throughput, and 2.4% enhancement in packet delivery performance.

The study's findings have significant implications for cloud computing, especially in financial trading and telemedicine. AMLGTTA is praised for its speed and ability to meet deadlines. In financial trading, it improves strategies and profitability, while in telemedicine, it efficiently processes medical data to potentially save lives. AMLGTTA also optimizes resource allocation in billed scenarios, leading to cost savings. With its enhanced performance, adaptability, and efficiency, AMLGTTA is a game-changer in multicloud load balancing, revolutionizing multicloud computing and helping organizations meet digital era demands.

REFRENCES

[1] S. Tuli, S. Ilager, K. Ramamohanarao and R. Buyya, "Dynamic Scheduling for Stochastic Edge-Cloud Computing Environments Using A3C Learning and Residual Recurrent Neural Networks," in IEEE Transactions on Mobile Computing, vol. 21, no. 3, pp. 940-954, 1 March 2022, doi: 10.1109/TMC.2020.3017079.

[2] I. M. Ali, K. M. Sallam, N. Moustafa, R. Chakraborty, M. Ryan and K. -K. R. Choo, "An Automated Task Scheduling Model Using Non-Dominated Sorting Genetic Algorithm II for Fog-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2294-2308, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2020.3032386.
[3] S. Achar, "Neural-Hill: A Novel Algorithm for Efficient Scheduling IoT-Cloud Resource to Maintain Scalability," in IEEE Access, vol. 11, pp. 26502-26511, 2023, doi: 10.1109/ACCESS.2023.3257425.
[4] P. V. Reddy and K. G. Reddy, "A Multi-Objective Based Scheduling Framework for Effective Resource Utilization in Cloud Computing," in IEEE Access, vol. 11, pp. 37178-37193, 2023, doi: 10.1109/ACCESS.2023.3266294.
[5] T. -P. Pham and T. Fahringer, "Evolutionary Multi-Objective Workflow Scheduling for Volatile Resources in the Cloud," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1780-1791, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.2993250.
[6] P. Loncar and P. Loncar, "Scalable Management of Heterogeneous Cloud Resources Based on Evolution Strategies Algorithm," in IEEE Access, vol. 10, pp. 68778-68791, 2022, doi: 10.1109/ACCESS.2022.3185987.
[7] X. Ma, A. Zhou, S. Zhang, Q. Li, A. X. Liu and S. Wang, "Dynamic Task Scheduling in Cloud-Assisted Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2116-2130, 1 April 2023, doi: 10.1109/TMC.2021.3115262.
[8] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh and A. Y. Zomaya, "ARPS: An Autonomic Resource Provisioning and Scheduling Framework for Cloud Platforms," in IEEE Transactions on Sustainable Computing, vol. 7, no. 2, pp. 386-399, 1 April-June 2022, doi: 10.1109/TSUSC.2021.3110245.
[9] Q. Wu, M. Zhou and J. Wen, "Endpoint Communication Contention-Aware Cloud Workflow Scheduling," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 2, pp. 1137-1150, April 2022, doi: 10.1109/TASE.2020.3046673.
[10] L. Ye, Y. Xia, L. Yang and C. Yan, "SHWS: Stochastic Hybrid Workflows Dynamic Scheduling in Cloud Container Services," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 3, pp. 2620-2636, July 2022, doi: 10.1109/TASE.2021.3093341.
[11] H. Mahmoud, M. Thabet, M. H. Khafagy and F. A. Omara, "Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm," in IEEE Access, vol. 10, pp. 36140-36151, 2022, doi: 10.1109/ACCESS.2022.3163273.
[12] S. Qin, D. Pi, Z. Shao and Y. Xu, "A Knowledge-Based Adaptive Discrete Water Wave Optimization for Solving Cloud Workflow Scheduling," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 200-216, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3087642.
[13] X. Tang, "Reliability-Aware Cost-Efficient Scientific Workflows Scheduling Strategy on Multi-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2909-2919, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2021.3057422.
[14] X. Tang, Y. Liu, Z. Zeng and B. Veeravalli, "Service Cost Effective and Reliability Aware Job Scheduling Algorithm on Cloud Computing Systems," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1461-1473, 1 April-June 2023, doi: 10.1109/TCC.2021.3137323.
[15] X. Wang, J. Cao and R. Buyya, "Adaptive Cloud Bundle Provisioning and Multi-Workflow Scheduling via Coalition Reinforcement Learning," in IEEE Transactions on Computers, vol. 72, no. 4, pp. 1041-1054, 1 April 2023, doi: 10.1109/TC.2022.3191733.
[16] H. Zhang and R. Jia, "Application of Chaotic Cat Swarm Optimization in Cloud Computing Multi Objective Task Scheduling," in IEEE Access, vol. 11, pp. 95443-95454, 2023, doi: 10.1109/ACCESS.2023.3311028.
[17] A. Belgacem, K. Beghdad-Bey and H. Nacer, "Dynamic Resource Allocation Method Based on Symbiotic Organism Search Algorithm in Cloud Computing," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1714-1725, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3002205.
[18] K. Kang, D. Ding, H. Xie, Q. Yin and J. Zeng, "Adaptive DRL-Based Task Scheduling for Energy-Efficient Cloud Computing," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4948-4961, Dec. 2022, doi: 10.1109/TNSM.2021.3137926.
[19] L. Ye, Y. Xia, S. Tao, C. Yan, R. Gao and Y. Zhan, "Reliability-Aware and Energy-Efficient Workflow Scheduling in IaaS Clouds," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 3, pp. 2156-2169, July 2023, doi: 10.1109/TASE.2022.3195958.

[20] Y. Chen, T. Zhao, P. Cheng, M. Ding and C. W. Chen, "Joint Front–Edge–Cloud IoVT Analytics: Resource-Effective Design and Scheduling," in IEEE Internet of Things Journal, vol. 9, no. 23, pp. 23941-23953, 1 Dec.1, 2022, doi: 10.1109/JIOT.2022.3189035.
[21] S. U. Jamil, M. A. Khan and S. U. Rehman, "Resource Allocation and Task Off-Loading for 6G Enabled Smart Edge Environments," in IEEE Access, vol. 10, pp. 93542-93563, 2022, doi: 10.1109/ACCESS.2022.3203711.
[22] J. He and X. Liu, "Hybrid Teaching–Learning-Based Optimization for Workflow Scheduling in Cloud Environment," in IEEE Access, vol. 11, pp. 100755-100768, 2023, doi: 10.1109/ACCESS.2023.3314735.
[23] J. Wang, "Dynamic Multiworkflow Offloading and Scheduling Under Soft Deadlines in the Cloud-Edge Environment," in IEEE Systems Journal, vol. 17, no. 2, pp. 2077-2088, June 2023, doi: 10.1109/JSYST.2023.3239118.
[24] A. Taghinezhad-Niar and J. Taheri, "Reliability, Rental-Cost and Energy-Aware Multi-Workflow Scheduling on Multi-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 2681-2692, 1 July-Sept. 2023, doi: 10.1109/TCC.2022.3223869.
[25] M. Guo, Q. Guan, W. Chen, F. Ji and Z. Peng, "Delay-Optimal Scheduling of VMs in a Queueing Cloud Computing System with Heterogeneous Workloads," in IEEE Transactions on Services Computing, vol. 15, no. 1, pp. 110-123, 1 Jan.-Feb. 2022, doi: 10.1109/TSC.2019.2920954.
[26] D. Cheng, Y. Wang and D. Dai, "Dynamic Resource Provisioning for Iterative Workloads on Apache Spark," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 639-652, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3108043.
[27] M. Liwang and X. Wang, "Overbooking-Empowered Computing Resource Provisioning in Cloud-Aided Mobile Edge Networks," in IEEE/ACM Transactions on Networking, vol. 30, no. 5, pp. 2289-2303, Oct. 2022, doi: 10.1109/TNET.2022.3167396.
[28] L. Yang, Y. Xia, L. Ye, R. Gao and Y. Zhan, "A Fully Hybrid Algorithm for Deadline Constrained Workflow Scheduling in Clouds," in IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 3197-3210, 1 July-Sept. 2023, doi: 10.1109/TCC.2023.3269144.
[29] B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," in IEEE Access, vol. 10, pp. 17803-17818, 2022, doi: 10.1109/ACCESS.2022.3149955.
[30] Y. Huang et al., "Deep Adversarial Imitation Reinforcement Learning for QoS-Aware Cloud Job Scheduling," in IEEE Systems Journal, vol. 16, no. 3, pp. 4232-4242, Sept. 2022, doi: 10.1109/JSYST.2021.3122126.
[31] E. Cao et al., "Energy and Reliability-Aware Task Scheduling for Cost Optimization of DVFS-Enabled Cloud Workflows," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 2127-2143, 1 April-June 2023, doi: 10.1109/TCC.2022.3188672.
[32] S. Manam, K. Moessner and S. Vural, "Deadline-Constrained Cost Minimisation for Cloud Computing Environments," in IEEE Access, vol. 11, pp. 38514-38522, 2023, doi: 10.1109/ACCESS.2023.3258682.
[33] X. Wang, Y. Li, F. Guo, Y. Xu and J. C. S. Lui, "Dynamic GPU Scheduling With Multi-Resource Awareness and Live Migration Support," in IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 3153-3167, 1 July-Sept. 2023, doi: 10.1109/TCC.2023.3264242.
[34] M. T. Islam, S. Karunasekera and R. Buyya, "Performance and Cost-Efficient Spark Job Scheduling Based on Deep Reinforcement Learning in Cloud Computing Environments," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1695-1710, 1 July 2022, doi: 10.1109/TPDS.2021.3124670.
[35] B. Dai, J. Niu, T. Ren and M. Atiquzzaman, "Toward Mobility-Aware Computation Offloading and Resource Allocation in End–Edge–Cloud Orchestrated Computing," in IEEE Internet of Things Journal, vol. 9, no. 19, pp. 19450-19462, 1 Oct.1, 2022, doi: 10.1109/JIOT.2022.3168036.
[36] Y. Laili, F. Guo, L. Ren, X. Li, Y. Li and L. Zhang, "Parallel Scheduling of Large-Scale Tasks for Industrial Cloud–Edge Collaboration," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3231-3242, 15 Feb.15, 2023, doi: 10.1109/JIOT.2021.3139689.
[37] H. Sun, S. Wang, F. Zhou, L. Yin and M. Liu, "Dynamic Deployment and Scheduling Strategy for Dual-Service Pooling-Based Hierarchical Cloud Service System in Intelligent Buildings," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 139-155, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3078795.

[38] S. Bose and N. Mukherjee, "SenSchedule: Scheduling Heterogeneous Resources in Sensor-Cloud Infrastructure," in IEEE Transactions on Services Computing, vol. 15, no. 4, pp. 1825-1840, 1 July-Aug. 2022, doi: 10.1109/TSC.2020.3022679.
[39] Z. Chen, J. Hu, G. Min, C. Luo and T. El-Ghazawi, "Adaptive and Efficient Resource Allocation in Cloud Datacenters Using Actor-Critic Deep Reinforcement Learning," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 8, pp. 1911-1923, 1 Aug. 2022, doi: 10.1109/TPDS.2021.3132422.
[40] N. Rizvi, D. Ramesh, P. C. S. Rao and K. Mondal, "Intelligent Salp Swarm Scheduler With Fitness Based Quasi-Reflection Method for Scientific Workflows in Hybrid Cloud-Fog Environment," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 2, pp. 862-877, April 2023, doi: 10.1109/TASE.2022.3170549.
[41] Z. Liu, M. Liwang, S. Hosseinalipour, H. Dai, Z. Gao and L. Huang, "RFID: Towards Low Latency and Reliable DAG Task Scheduling Over Dynamic Vehicular Clouds," in IEEE Transactions on Vehicular Technology, vol. 72, no. 9, pp. 12139-12153, Sept. 2023, doi: 10.1109/TVT.2023.3266582.
[42] T. He, A. N. Toosi and R. Buyya, "CAMIG: Concurrency-Aware Live Migration Management of Multiple Virtual Machines in SDN-Enabled Clouds," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 10, pp. 2318-2331, 1 Oct. 2022, doi: 10.1109/TPDS.2021.3139014.
[43] A. Ali and M. M. Iqbal, "A Cost and Energy Efficient Task Scheduling Technique to Offload Microservices Based Applications in Mobile Cloud Computing," in IEEE Access, vol. 10, pp. 46633-46651, 2022, doi: 10.1109/ACCESS.2022.3170918.
[44] H. Li, B. Wang, Y. Yuan, M. Zhou, Y. Fan and Y. Xia, "Scoring and Dynamic Hierarchy-Based NSGA-II for Multiobjective Workflow Scheduling in the Cloud," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 2, pp. 982-993, April 2022, doi: 10.1109/TASE.2021.3054501.
[45] Z. Sun, B. Zhang, C. Gu, R. Xie, B. Qian and H. Huang, "ET2FA: A Hybrid Heuristic Algorithm for Deadline-Constrained Workflow Scheduling in Cloud," in IEEE Transactions on Services Computing, vol. 16, no. 3, pp. 1807-1821, 1 May-June 2023, doi: 10.1109/TSC.2022.3196620.
[46] I. Attiya, M. A. Elaziz, L. Abualigah, T. N. Nguyen and A. A. A. El-Latif, "An Improved Hybrid Swarm Intelligence for Scheduling IoT Application Tasks in the Cloud," in IEEE Transactions on Industrial Informatics, vol. 18, no. 9, pp. 6264-6272, Sept. 2022, doi: 10.1109/TII.2022.3148288.
[47] J. Lou, Z. Tang, S. Zhang, W. Jia, W. Zhao and J. Li, "Cost-Effective Scheduling for Dependent Tasks With Tight Deadline Constraints in Mobile Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 10, pp. 5829-5845, 1 Oct. 2023, doi: 10.1109/TMC.2022.3188770.
[48] M. Bigdeli, B. Abolhassani, S. Farahmand and C. Tellambura, "Offline and Real-Time Deadline-Aware Scheduling and Resource Allocation Algorithms Favoring Big Data Transmission Over Cognitive CRANs," in IEEE Access, vol. 11, pp. 67755-67778, 2023, doi: 10.1109/ACCESS.2023.3288996.
[49] G. Yao, Q. Ren, X. Li, S. Zhao and R. Ruiz, "A Hybrid Fault-Tolerant Scheduling for Deadline-Constrained Tasks in Cloud Systems," in IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1371-1384, 1 May-June 2022, doi: 10.1109/TSC.2020.2992928.
[50] Q. Li, X. Jia, C. Huang and H. Bao, "A Dynamic Combinatorial Double Auction Model for Cloud Resource Allocation," in IEEE Transactions on Cloud Computing, vol. 11, no. 3, pp. 2873-2884, 1 July-Sept. 2023, doi: 10.1109/TCC.2022.3231249.