The Scientific Temper
JULY 2026 Vol. 17 (7): 6557-6579
ISSN: 0976-8653, E-ISSN: 2231-6396
DOI: 10.58414/SCIENTIFICTEMPER.2026.17.7.2496
https://scientifictemper.com/

RESEARCH ARTICLE

A Hierarchical Multi-Agent Reinforcement Learning With a Heterogeneous Metaheuristic Aware Resource Allocation in Big Data-Cloud

A. MOHAMED ASHAR ALI[1], A.R. MOHAMED SHANAVAS[2]

[1,2]Jamal Mohamed College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli – 620 020, India.

Corresponding Author*: ama@jmc.edu

ABSTRACT

The rapid advancement of technology like Internet of Things (IoT) and Cloud computing (CC) based heterogeneous environment required dynamic resource management system. The complexity of IoT-Cloud is increasing due to abundance of dynamic data dissemination that create poor performance like high energy consumption (EC), workload imbalancing, poor adaptability, and fail to handle SLA violations. The two primary contributions of the proposed work are the Optimized Priority-Aware Hierarchical Multi-Agent Deep Q Network (OPHM-DQN) and the Adaptive Multi-Objective Dung Beetle Optimization Algorithm (ADBOA). The multi-agent strategies increase the scalability and adaptability of resource management through local and global hierarchies. The approach integrates information-based decision-making and priority-aware allocation while accounting for SLA requirements, system constraints, and job complexity to optimise resource generation, utilisation, allocation, and task scheduling. In comparison to existing optimization and Reinforcement Learning (RL) techniques, experimental results show that the proposed OPHM-DQN-ADBOA framework consistently reduces EC (up to 30 % lower), execution delay, and SLA violations while improving resource utilisation and LB. The ADBOA enhances the proposed model through optimal multi-objective training, reducing EC, SLA violation, and cost while improving resource utilization and scheduling efficiency. As a results, the model achieves high scalability and adaptability in heterogeneous IoT-Cloud resource management.

Keywords: *Resource Allocation, IoT, Cloud, SLA, DQN, Multi-Objective, Dung Beetle Optimization, Hierarchical Multi-agent decision making.*

INTRODUCTION

The fast development of Cloud Computing, IoT, and Big Data technologies has significantly transformed modern computational organizations (Qi et al., 2023). The growth of heterogeneous Big Data–cloud environments that can support intelligent services, large-scale distributed processing, and real-time applications has been made possible by the integration of IoT devices with CC platforms (Monroe & Langford, 2024). The growing prevalence of smart healthcare, intelligent transportation, industrial automation, smart cities, e-commerce, and multimedia applications has led to the generation of a tremendous volume of data of diverse types, demanding efficient computational and storage solutions (Rafiq et al., 2023). Thus, CC has emerged as a key platform to process and manage large scale dynamic workloads because of its scalability, elasticity, virtualization support and provision of resources on demand (Ghandour et al., 2023).

Although CC offers numerous benefits, the huge amount of data generated by IoT devices and evolving user requirements presents a number of problems in Resource Allocation (RA) and workload scheduling (Jeyaraj et al., 2023), (Awad et al., 2025). In order to ensure system stability and quality of service (QoS), cloud data centers must regularly manage heterogeneous resources, varying workloads, and service-level requirements

Received: 12th July, 2026 **Accepted: 17th July, 2026** **Published: 28th July, 2026**

(Kushchazli et al., 2025). In traditional RA methods, there is a lack of adaptability, scalability, and intelligent decision making in the context of heterogeneous Big Data–cloud environments. Typically, static scheduling techniques and rule-based allocation are not enough to address the demands of dynamic workload fluctuations, real-time service requirements, and multi-objective optimization constraints (Awad et al., 2025) (Lahza et al., 2024). Therefore, cloud infrastructures often suffer from issues like high energy usage, load imbalance, resource underutilization, high operational cost, and long execution time.

The RA in heterogeneous cloud environment is a multi-objective optimization problem, since multiple conflicting objective functions need to be met (Shukla & Pandey, 2024), (Hao et al., 2024). For instance, reducing energy usage may reduce the performance of the system and processing delay, while increasing energy usage may lead to increase workload imbalance. In addition, the geographically distributed cloud nodes, Virtual Machines (VMs) and dynamically arriving Big Data workloads makes optimal scheduling decisions more complex (Radhika & Duraipandian, 2025). Thus, the realization of intelligent and adaptive resource management mechanisms, which are able to optimize multiple objectives in a constantly shifting and unpredictable environment, is a key research challenge in CC systems.To tackle these problems, researchers have investigated different optimization and machine learning (ML) methods for cloud RA. Conventional metaheuristic algorithms like Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Whale Optimization Algorithm (WOA), and Grey Wolf Optimization (GWO) have been extensively used for workload scheduling and resource management challenges (Acharya et al., 2025), (Thorne, 2026), (Murali & Brindha, 2023). In static environments, these algorithms offer good optimization performance. But there are a few drawbacks in the traditional metaheuristic methods such as premature convergence, slow adaptation to the changes in the environment, the lack of scalability and an inadequate learning capability for highly dynamic cloud infrastructures. Moreover, these methods do not necessarily provide an efficient means to represent the long-term scheduling constraints and decision-making needs in large-scale distributed systems.

In recent years, deep reinforcement learning (DRL) has become a very promising paradigm for solving the problems of intelligent RA and scheduling in cloud environments (Zhou et al., 2024), (Jalali Khalil Abadi et al., 2024). RL allows agents to learn optimum policies without the need for explicit supervision through constant contact with the environment (Sivamayil et al., 2023), (Ismail et al., 2025). Specifically, scheduling models based on deep artificial neural networks, Deep Q-Network (DQN), are shown to be effective in dynamic and uncertain RA problems (Zhao & Wang, 2023). DQN models can approximate a complex relationship between the state-action by combining deep neural networks with RL mechanisms, enabling scheduling decisions based on this approximation (Momenikorbekandi & Abbod, 2023). However, single-agent RL suffers from scalability issues in heterogeneous cloud infrastructures (Jayanetti et al., 2024). Centralized learning systems encounter greater state-space complexity, slower convergence, and communication overhead as resource and increase task levels (Qadoua & Khalil, 2025). To address these challenges, Hierarchical Multi-Agent Reinforcement Learning (HMARL) has received much attention in distributed cloud resource management (Sonani, 2023).

Multiple cooperative agents are introduced in HMARL as hierarchical systems for decentralized decision making and distributed workload scheduling (Yu et al., 2025). Hierarchical architectures usually involve global agents (GAs) focusing on aggregate system-wide goals like energy efficiency, SLA satisfaction, and resource balancing, and local agents (LAs) looking at finer-grained scheduling and resource orchestration for objects within localized cloud clusters. This hierarchical coordination provides better scalability, adaptability and decision efficiency in heterogeneous settings (Wang et al., 2023). Besides, multi-agent RL supports parallel learning and joint policy optimization, it is very suitable for large-scale Big Data–cloud infrastructures (Zhang et al., 2024). Current RA models based on HMARL assisted to enhance the efficiency of scheduling in distributed systems, but there are still several problems to be addressed. Many of the current approaches do not take into account the priority of workloads while making scheduling decisions.

Workloads in real-world cloud environments may have varying priorities, urgency, computational complexities, and QoS requirements (Alam et al., 2024). When workload heterogeneity is not taken into account, it can lead to sub-optimal RU and to a higher SLA violation rate. Second, existing RL models consider as single objective or limited multi objective optimization problems and are not capable of solving the multiple conflicting system objectives. Thirdly, RL algorithms can encounter problems of unstable convergence and local optimum in very dynamic environments with large state-action spaces.For this purpose, a combination of RL with adaptive metaheuristic optimization methods has emerged as an effective research direction. The Dung Beetle Optimization (DBO) algorithm is one of the algorithms in the field of swarm intelligence and has gained significant in-

terest for its potent global search ability, its efficient exploration-exploitation balance, and its robustness in complex optimization problems (Xue & Shen, 2023). Based on the natural foraging and navigation pattern of dung beetles, DBO has shown promising optimization performance in multidimensional search space. However, standard DBO algorithms may still experience population diversity degradation and convergence instability in complex optimization conditions. Consequently, optimization accuracy and convergence performance can be greatly enhanced by including adaptive control techniques into DBO. Inspired by these abovementioned research gaps, this study proposes an Optimized Priority-Aware Hierarchical Multi-Agent Deep Q-Network (OPHM-DQN) integrated with an ADBOA for intelligent RA in heterogeneous Big Data–cloud environments. The key contribution of the proposed work is summarized below.

- We proposed a new HMARL-based adaptive, priority-aware DQN architecture for intelligent cloud RA. In this architecture, decentralised resource management and workload-based scheduling are carried out concurrently by global and local hierarchical agents in a dynamic environment of massive data sets and clouds.

- We introduced a priority-based adaptive learning mechanism that can dynamically allocate incoming Bigdata tasks based on workload urgency, Service Level Agreement (SLA) requirements, and resource availability that improves scheduling efficiency and QoS.

- A multi-objective Dung beetle optimization strategy is developed to simultaneously optimize EC, processing cost, workload balancing, and SLA violation during cloud RA. A multi-objective fitness evaluation mechanism uses a unified fitness model to make efficient decisions about RA in heterogeneous cloud environments.

- In the proposed ADBOA, three adaptive control parameters are introduced for enhancing the balance between exploration and exploitation while preventing premature convergence. Here, the adaptive convergence controller, adaptive diversity controller, and adaptive learning weight controller help to achieve self-adaptive mechanism for DBO algorithm.

RELATED WORK

In heterogeneous CC and Big Data environments, where workloads can be dynamic and resources are heterogeneous, there are a number of problems related to efficient RA and workload scheduling that need to be addressed to meet QoS requirements. Various optimization and intelligent learning techniques have been proposed for better performance in cloud resource management.

Nematollahi et al. (2024) presented an IoT-based RA and task offloading scheme based on Opposition-Based Learning combined with Moth-Flame Optimization (OBLMFO). The framework implemented adaptive scheduling and blockchain-based transactions to scale workload and alleviate load imbalance. The model was not intelligent adaptive learning however with high dynamic cloud environment. To optimize the LB of clouds, Geetha et al. (2024) presented a hybrid Intercrossed Chimp and Bald Eagle Algorithm (IC&BA). The use of Bald Eagle Search and Chimp Optimization provided an enhanced allocation of workload and utilization of resources. However, the method was predominantly based on static optimization methods without adaptive RL assistance. Gurusamy and Selvaraj (2024) proposed a task scheduling and RA system based on Hierarchical Auto-associative Polynomial Convolutional Neural Network (RA-HAPCNN-CC). Under various resources constraints, the scheduler based on starling murmuration technique achieved better makespan and throughput. But this structure was lacking in decentralized adaptive decision making for large-scale heterogeneous cloud systems.

Fan et al. (2025) proposed a Multi-Agent Deep Reinforcement Learning (MADRL)-based joint resource management scheme for cloud-edge collaborative systems. The model focused on optimizing the latency, energy usage, and RA in distributed systems. The framework was successful in enhancing distributed scheduling performance but lacked support for workload priority-aware multi-objective optimization. Lilhore et al. (2025) proposed a hybrid water wave optimization (WWO) and ACO for Cloud resource management. The method enhanced the global exploration and local search functionality to reduce the response time and operating cost. But the method did not have adaptive learning for dynamically varying cloud workloads. Huang et al. (2024) proposed DRL based RA and task offloading scheme for Hybrid Cloud and Multi-access Edge Computing (MEC) systems. The proposed multi-agent DRL model was successful in reducing EC and latency. However, the framework did not deal with heterogeneous cloud scheduling on a large scale, but rather on edge-assisted networks.

The Neural Network Task Classification (N2TC) framework was proposed by Mittal et al. (2024) that combine neural networks with genetic algorithms to provide intelligent task scheduling and RA. Although the method enhanced scheduling efficiency, it was strictly evolutionary optimization method with no hierarchical adaptive learning capability. Prabha and Rengarajan (2025)

proposed a hybrid approach of Vision Transformer, DRL, and Enhanced Shrike Optimization for adaptive workload balancing and intelligent RA. However, the model increased the scheduling efficiency and VM migration performance, and hierarchical multi-agent coordination was not taken into account. In multi-objective cloud optimization problem, Dankolo et al. (2025) proposed an integrated algorithm named as hybrid Flower Pollination Algorithm and Tabu Search (FPA-TS). Adaptive probability mechanisms and memory-guided local refinement enhanced makespan and execution cost optimisation. But the structure had no adaptive scheduling based on the RL. Nambi and Thanapal (2025) presented Enhanced Multi-Objective Optimization Algorithm for Task Scheduling (EMO-TS) which integrates Enhanced Electric Fish Optimization with DRL. The proposed framework performed dynamically optimized energy, resource, and QoS performance under different load conditions. The model did not, however, take into consideration hierarchical multi-agent coordination and workload priority-aware scheduling.

Based on the literature review, it is concluded that optimizing cloud RA has made substantial advances with the combination of optimization algorithms, ML and RL techniques. Existing research has been concentrated on energy efficiency, workload balancing, scheduling accuracy, response time and RU through hybrid optimization and intelligent scheduling techniques. However, there are still a number of research questions that have not yet been answered. However, most of the available approaches are either based on centralized learning approaches or are single-objective optimization approaches, which are difficult to scale and adapt to large-scale and heterogeneous cloud systems. Besides this, workload priority, SLA-awareness, and dynamic multi-objective optimization are not sufficiently taken into account in the RA decisions in many existing frameworks. Moreover, the traditional optimization methods tend to converge too early, and lack flexibility with the high dynamism of workloads.To address these challenges, this work proposes an OPHM-DQN integrated with ADBOA for intelligent RA in heterogeneous Big Data–cloud environments. The proposed framework introduces the concept of hierarchical multi-agent RL and adaptive multi-objective optimization to allow for scalable, decentralized and priority-based resource management. Unlike current models, the proposed model optimizes EC, execution cost, SLA violation, delay and workload imbalance in a single system, and is adaptive to dynamically adapt to varying cloud workload conditions via hierarchical global-LA coordination and adaptive optimization mechanisms.

PROPOSED METHODOLOGY

The proposed work has introduced a novel optimized Priority-Aware Hierarchical Multi-Agent Deep Q Network i.e. named as OPHM-DQN. The ultimate goal of proposed **OPHM-DQN** is enabling an intelligent RA scheme for Big Data-Cloud environments. The priority aware scheduling mechanism in the OPHM-DQN helps the system to handle to dynamic workload management problem. Moreover, the proposed model is highly suitable for managing multi-objective decision making by focusing on minimization (EC, delay, cost, SLA violations, and workload imbalancing) problems in large-scale cloud environments. This model executes hierarchical multi-agents (global and local) based mutual allocation policies where overall (energy efficiency, workload balancing, and SLA satisfaction) decision is in the hand of GA while the LAs perform fine-grained adaptive scheduling and resource orchestration within localized cloud environments.

The overall system model, the HMARL-based AP-DQN framework, the adaptive multi-objective Dung Beetle Optimization mechanism, the intelligent RA strategy, the environment feedback and reward evaluation process, and the mathematical formulation of the suggested framework (as shown in Figure 2) are all covered in this section.

PROBLEM FORMULATION

The RA problem is formulated as a multi-objective optimization problem under dynamically changing cloud workload conditions. The primary objective of the proposed framework is to identify optimal RA policies that maximize RU and SLA satisfaction while minimizing EC, execution delay, operational cost, and workload imbalance. The mathematical expression for the scheduling problem is expressed as (mapping function)

$$f: W \rightarrow C \tag{1}$$

The mapping function represents the workload tasks assigned to available cloud resources. As discussed earlier, the proposed work has considered the multi-objective problem rather than the uni-objective. Specifically, the multi-objective problem is addressed via five combination of functions such as $f_1(X_{ec})$, $f_2(X_d)$, $f_3(X_c)$, $f_4(X_{SLA})$, and $f_5(X_{lb})$. Since all these are minimization problem. The multi-objective based decision making based mathematical function is formulated as

$$min\big(f(X_{mo})\big) = \{f_1(X_{ec}), f_2(X_d), f_3(X_c), f_4(X_{SLA}), f_5(X_{lb})\} \tag{2}$$

Based on aforementioned equation, the mathematical expression like $f_1(X_{ec})$ denotes minimizing EC,$f_2(X_l)$ means minimizing delay, $f_3(X_c)$ denotes minimizing execution cost,

$f_4(X_{SLA})$ represents minimizing SLA violations, and $f_5(X_{lb})$ means minimizing the LB. Moreover, these operations are subject to multiple constrains such as

$$R_t = \{C_t, M_t, B_t, ST_t, E_t, SLA_t\} \quad (3)$$

$$C_i \leq C_i^{max}$$

$$M_i \leq M_i^{max}$$

$$B_i \leq B_i^{max}$$

$$S_i \leq S_i^{\varphi}$$

Where C_i (CPU utilization at i_{th} level), C_i^{max}(maximum CPU utilization), M_i (Memory utilization at i_{th} level), M_i^{max}(maximum Memory utilization), B_i (Bandwidth utilization at i_{th} level), B_i^{max}(maximum Bandwidth utilization), SLA_i (SLA Violation at i_{th} level), S_i^{max}(maximum SLA violation level at ($threshold\ value$)). All these constraints must be followed while allocating all the resources.

SYSTEM MODEL

This section presents the system model of the proposed intelligent RA framework for heterogeneous Big Data–Cloud environments. The proposed architecture integrates Hierarchical Multi-Agent Reinforcement Learning (HMARL)-based Adaptive Priority-Aware Deep Q-Network (DQN) with ADBOA technique to achieve efficient, scalable, and SLA-aware resource management under dynamically changing workload conditions.

A Markov Decision Process (MDP) defined by $M = (S, A, P, R, \gamma)$ describes the system. After observing the system state,$s_t \epsilon S$, the agent executes the allocation action, $a_t \epsilon A$, and gets the reward, r_t and transitions to state s_{t+1}. The objective is to choose the best policy in order to maximise projected discounted reward.

$$\pi\text{-}*.\ \pi^* = \arg\max_{\pi} E[\sum_{t=0}^{\infty} \gamma^t rt] \quad (4)$$

The overall framework is organized into multiple functional layers, including the Big Data Workload Management Layer, Resource Monitoring Layer, Adaptive Decision-Making Layer, Adaptive Multi-Objective Optimization Layer, and Cloud Resource Execution Layer. The interaction among these layers enables continuous monitoring, adaptive learning, optimization, and intelligent RA in distributed cloud infrastructures.

BIG DATA WORKLOAD MANAGEMENT LAYER

The workload management layer plays a top level role in handling large-scale heterogeneous workloads generated from multiple Big Data sources including IoT devices, enterprise applications, mobile systems, and cloud-based services. The incoming workloads consist of multiple independent and dependent tasks characterized by varying computational complexity, execution deadlines, storage requirements, and QoS constraints.

The workload queues are represented by W that associated with T_n (n_{th} workload task) submitted to the cloud environment. As mentioned in earlier section (equation 3), C_t, M_t, B_t, and ST_t are the primary attributes but task priority (p_i) and execution deadline (d_i) are highly needed for enabling better decision during dynamic scenarios. Moreover, this layer organizes the dynamic workload changes before sending them to the resource monitoring layer and adaptive decision making modules.

RESOURCE MONITORING LAYER

The operational state of diverse cloud resources and workload behaviour are continuously monitored by the Resource Monitoring Layer. Several geographically dispersed data centres with a variety of physical servers, VMs, containers, storage resources, and network components make up the cloud infrastructure. Periodically, the monitoring module gathers resource-state data, which is represented as:

$$R_t = \{C_t, M_t, B_t, ST_t, E_t, SLA_t\} \quad (5)$$

Where C_t represents the CPU utilization, M denotes memory, B denotes Bandwidth, ST denotes storage, energy is represented by E_t, and SLA_t denotes SLA status. On the other hand, some other parameters like Workload based (queue length and arrival rate), delay, VM utilization, and throughput are needed to be monitored continuously for analysing the dynamic changes. These information are considered as the input states are being analysed via proposed AP-HMAMRL-DBO framework.

AP-HMARL-DBO FRAMEWORK

The Adaptive Decision-Making Layer represents the core intelligence component of the proposed framework. This layer employs a Hierarchical Multi-Agent Reinforcement Learning (HMARL)-based AP-DQN architecture to perform intelligent scheduling and RA decisions.

In order to capture long-range dependencies in system states such as workload change, service demand, and network congestion, the AP-DQN component uses a deep neural network supplemented with an attention mechanism to estimate the best action-value function $Q(s, a; \theta)$. The mathematical expression of the temporal-difference (TD) error is shown in equation (2):

$$\delta_i = r_i + \gamma \max_{a'} Q(s_i', a'; \theta^-) - Q(s_i, a_i; \theta) \quad (6)$$

The Decentralized MDP (Dec-MDP) is utilized for formulating the RA problem by considering the RU, scheduling, and workload arrivals in dynamic procedure. In the hierarchical mechanism of proposed system, the global system plays a significant role that is mathematically denoted as g_t^s(refer equation 6):

$$g_t^s = \{R_t, W_t, SLA_t, Q_t\} \quad (7)$$

In the equation (7), workload states and resources states are respectively represented by R_t *and* W_t. And the condition of SLA is indicated by SLA_t while Q_t represents the queue information.

$$\{g_1, g_2, \dots, g_N\} = G(S_g, \lambda) \quad (8)$$

The GA"G" generates N sub-goals using the global state g_t^s and the optimization parameter λ (from AMO-DBO). Each LA is guided by these sub-goals toward coordinated behaviours that balance system-wide objectives (energy, delay, cost, SLA, LB). In contradict to g_t^s, the l_t^s represents the role of LAs. The following equation shows its mathematical computation based on i_{th}LA.

$$l_t^s = \{cpu_i, mem_i, bw_i, q_i, l_i\} \quad (9)$$

In the equation 9, q_i denotes status of local queue while the l_i represents the workload conditions. The graphical illustration of the role of Global and local agents is shown in figure 1. The proposed architecture achieves efficient IoT-Cloud resource allocation by combining hierarchical reinforcement learning and adaptive optimization. Global agents are responsible for system-wide scheduling and SLA compliance, while local agents manage task-level decisions to balance workloads. The Adaptive Multi-Objective Dung Beetle Optimizer enhances decision-making by continuously modifying RA strategies. Under heterogeneous and dynamic workload situations, this layered design guarantees scalability, robustness, and enhanced QoS.

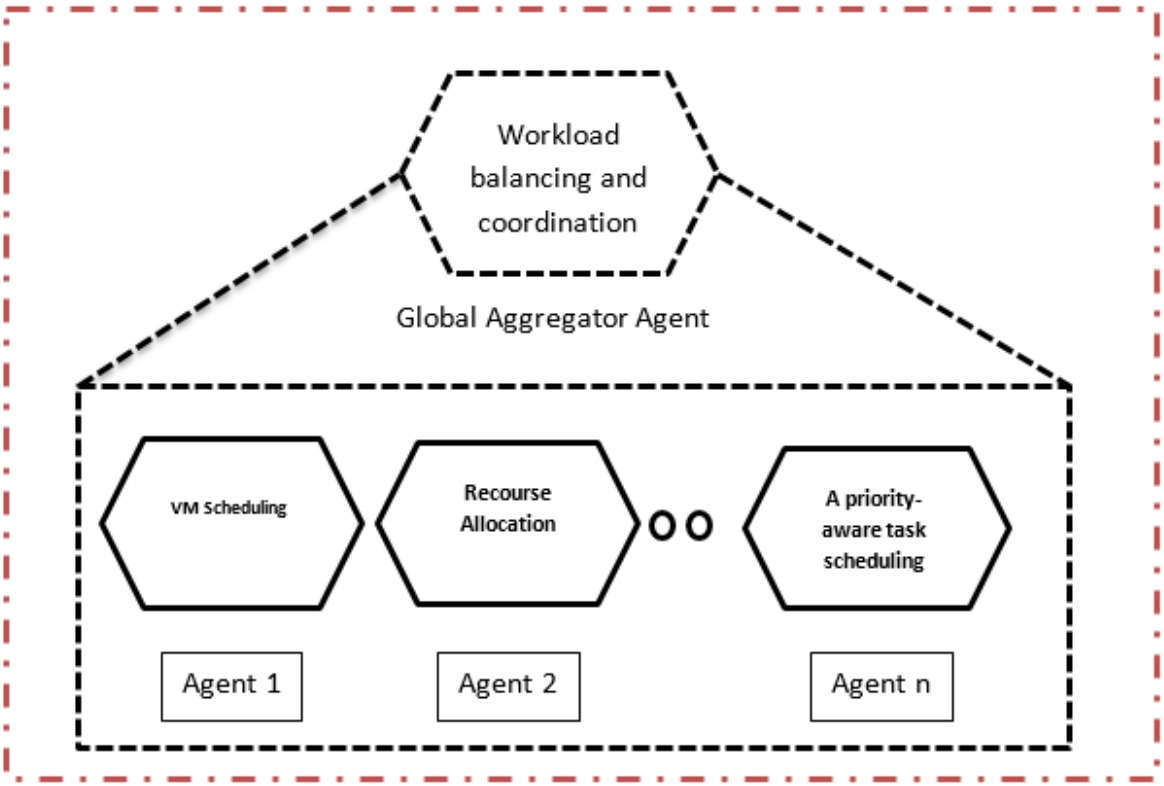


Figure 1: Graphical illustration of the role based global and local agents

GLOBAL AGENT

In the hierarchical mechanism, the GA takes aggregated decision making by considering various resources and workload states for performing high-level scheduling on heterogeneous Big Data-Cloud environment. The GAs analyse the collected global state information for generating dynamic workload demand vectors based on cumulative information obtained from LAs. This process helps to maintain efficient RU and reliable service via SLA-aware scheduling and effective LB. Moreover, the priority based resource distribution and allocation are the

primary motive of the GA where the resources are allocated in terms of urgency, availability, and condition of the environment. And the global action space targets workload balancing (w_{bal}) based on coordinative scheduling actions, priority guidance (p_{guide}), and global RA(g_{ra}) for achieving adaptive cloud resources management system. The mathematical representation of global action space (g_t^a)is shown in equation (10).

$$g_t^a = \{p_{guide}, w_{bal}, g_{ra}\} \quad (10)$$

LOCAL AGENT

In converse to GA, each LA acts independently for observing localized environmental states and performing fine-grained RA decision within its assigned cloud domain. From the local action space (equation. 11), each LA uses its Q-network to choose an action (a_i). The local action space incorporated with VM allocation (VM_{alloc}), task scheduling(t_s), scaling(s_{scale}), and Migration(m_{mig}). The mathematical expression is shown in equation (11)

$$l_t^a = \{VM_{alloc}, t_s, s_{scale}, m_{mig}\} \quad (11)$$

Once an action is chosen from the local action space (a_i). The agent performs the action, receives rewards (r_i) and the next state (s_i'), and records the transition in the replay buffer D as follows:

$$(s_i, a_i, r_i, s_i') \epsilon D \quad (12)$$

The temporal difference (TD) error is based on equation (13) and captures the difference between predicted and updated Q values, is used to quantify the learning signal for each transition. Each transition is additionally given a weight to guarantee prioritised sampling during replay.

$$\delta_i = r_i + \gamma. \max_{a'}(s_i', a') - Q(s_i, a_i) \quad (13)$$

The sample weight (w_i) is defined by equation (14) as a function of task priority, fairness, Quality of Service (Qos), and the TD-error (δ_i). This guarantees that during replay, transitions with greater learning potential or system significance are given priority for effective RA.

$$w_i = f(\delta_i, Qos, priority, fairness) \quad (14)$$

The DQN architecture uses deep neural networks to estimate Q-values for state-action pairs and adaptive policy learning to choose actions.

REWARD FUNCTION AND LEARNING MECHANISM

The Environment Feedback and Reward Evaluation Layer serves as the adaptive feedback mechanism that continuously refines the policies of the proposed priority-aware HMARL–DBO framework. Unlike conventional RL systems that focus solely on generic performance optimization, the proposed layer emphasizes priority-sensitive decision adaptation and Pareto-optimal multi-objective trade-off learning. Rather than redefining the reward formulation presented in the Reward Function section, this layer operationalizes previously defined reward signals by monitoring, aggregating, and dynamically applying them to guide both global and local AP-DQN agents toward intelligent RA policies. At each decision cycle, the environment continuously observes system-level and local-level performance indicators, including EC, execution delay, SLA satisfaction, workload imbalance, operational cost, and RU efficiency. More importantly, the framework incorporates workload priority awareness into the feedback mechanism, enabling high-priority and delay-sensitive tasks to receive adaptive scheduling preference during policy learning and optimization. These monitored metrics are transformed into weighted reward signals and propagated across hierarchical agents to ensure coordinated decision-making between global and local resource controllers.

Following the execution of the chosen actions, the cloud environment returns a reward signal $R_t$and moves from the current state (s_t)to the subsequent state (s_{t+1}). The reward function uses a variety of performance criteria to assess how well decisions about RA are working. The priority aware reward function is expressed as follows:

$$R_t = \omega_1.SLA_t + \omega_2.U_t + \omega_3.LB_t - \omega_4.EC_t - \omega_5.Delay_t - \omega_6.Cost_t \quad (15)$$

Where, SLA_t represents SLA satisfaction, U_t denotes RU, LB_t indicates LB, EC_t denotes energy consumption, $Cost_t$ denotes cost, and $\omega_1, \omega_2, \ldots\ldots, \omega_n$ are the weight coefficients.

Efficient and balanced resource deployment is rewarded, while SLA violations, excessive delays, resource overload, and excessive energy usage are penalised (S_t, A_t, R_t, andS_{t+1}) is stored in the experience replay buffer for continuous DQN learning and target network updates.

A min-batch B of transitions with probability proportional to w_i^a is sampled from the replay buffer "D" in order to stabilise training (equation. 15). The Q-network parameters are then updated using a weighted loss function (equation. 16) for each transition in the mini-batch, ensuring that prioritised experiences contribute to convergence more successfully.

The priority based Experience Replay mechanism is a key component of the prop approach. Every transition tuple is expanded as follows:

$$(s_i, a_{i,} r_{i,}, s_i', w_i) \quad (16)$$

In which $w_i \epsilon \mathbb{R}^+$ represents the weight of adaptive relevance. The definition of the sampling probability is.

$$P(i) = \frac{w_i^a}{\sum_{j=1}^{N} w_j^a}, a\epsilon[0,1] \quad (17)$$

Multi-source signals obtained from both AP-DQN learning dynamics and AMO-DBO optimization outputs are integrated by the weight function:

$$w_i = \lambda_1|\delta_i| + \lambda_2 \frac{1}{delay_i} + \lambda_3 priority_i + \lambda_4 \mathcal{F}_i \quad (18)$$

The QoS delay constraint violation factor is denoted by $delay_i$, the service-level priority indicator is denoted by $priority_i$., the fairness deviation metric is represented by $\mathcal{F}_i$. andλ_4 is an AMO-DBO-tuned adaptive weighting coefficient. This process improves learning efficiency and stability under variable workload situations by ensuring that high-impact, uncommon, and QoS-critical events are sampled more frequently.

The weighted loss function used in training is given in equation (19). In the sampled mini batch B, each transition I adds to the total loss in proportion to its relevance weight w_i. The model stabilises convergence by emphasising transitions with higher learning potential or system significance by scaling the squared TD²error (δ_i^2)with w_i.

$$L = \sum_{i \epsilon B} w_i . \delta_i^2 \quad (19)$$

The Q-network parameters θ are updated via gradient descent in equation (20). The gradient of the weighted loss $\nabla_\theta L$, guarantees that parameter adjustments are directed by prioritised experiences, while the learning rate ɳ determines the step size. The network can effectively adjust to situations involving dynamic RA thanks to this mechanism.

$$\theta \leftarrow \theta - \eta . \nabla_\theta L \quad (20)$$

The proposed feedback mechanism performs three major functions. First, it enables priority-aware reward aggregation by integrating LA rewards into a global performance objective while preserving workload importance and service criticality. This ensures that local optimization strategies remain aligned with overall system-wide QoS and SLA objectives. Second, the aggregated feedback is utilized for adaptive Q-value updating within the AP-DQN framework, accelerating convergence toward Pareto-optimal RA policies that effectively balance conflicting objectives such as energy efficiency, execution delay, operational cost, LB, and RU. Third, the layer supports adaptive environment monitoring by continuously identifying SLA violations, resource bottlenecks, workload congestion, and energy inefficiencies, thereby enabling dynamic corrective scheduling and allocation actions in highly heterogeneous IoT-cloud environments.

ADAPTIVE MULTI-OBJECTIVE DBO

An ADBOA algorithm is integrated into the proposed model to improve intelligent scheduling and RA in heterogeneous Big Data–Cloud environments. Under dynamic workload situations, the ADBOA performs as a global optimization mechanism that collaborates with the HMARL-based AP-DQN agents to find Pareto-optimal RA solutions. In contrast to traditional single-objective optimization techniques, the suggested adaptive DBO concurrently optimises several competing goals, such as LB, EC, execution delay, operational cost, SLA satisfaction, and resource utilisation. Large-scale distributed cloud infrastructures can benefit from adaptive exploration–exploitation balance, non-dominated solution search, and reliable resource orchestration thanks to the combination of DBO and hierarchical RL.

$$X_i^{t+1} = X_i^t + r \times (X_{best} - X_i^t) \quad (21)$$

The current beetle position is denoted by X_i^t, the next beetle position is indicated by X_i^{t+1}., the best solution is denoted by X_{best}, and r is a random search coefficient. In the proposed ADBOA, the adaptive convergence controller (a_{CC}), adaptive diversity controller (a_{dC}), and adaptive learning weight controller (a_{lC}) are introduced in DBO population to enhance the search capability and convergence stability of the traditional Dung Beetle Optimization (DBO) algorithm. In order to minimise premature convergence and increase global search efficiency, the adaptive convergence controller (refer equation (22)) is designed to dynamically balance exploration and exploitation during various optimization stages. When solution similarity increases, the adaptive diversity controller increases search randomness to preserve population variety is shown in equation (23). This helps the optimizer avoid local optimum stagnation and enhances solution distribution throughout the search space. Furthermore, during the search updating process, the adaptive learning weight controls (refer equation 24) the impact of elite solutions, allowing for a steady convergence toward optimal RA while preserving convergence stability. The resilience, search variety, convergence accuracy, and optimization performance of the proposed multi-objective DBO framework in dynamic Big Data–Cloud contexts are all greatly improved by the incorporation of these adaptive controllers.

$$a_{cc} = a_{min} + (a_{max} - a_{min}) e^{-\beta t/T} \quad (22)$$

$$a_{dc} = \frac{1}{N} \sum_{i=1}^{N} |X_i - X_{best}| \quad (23)$$

$$\gamma_C = (\gamma_C{}^{max} - \gamma_C{}^{min}) \frac{t}{T} \quad (24)$$

The mathematical upgradation of the proposed ADBOA is shown in equation (25) based on three adaptive controllers.

$$X_i^{t+1} = a_{cc}.X_i^t + r \times (X_{best} - X_i^t) + a_{dc}.r1 + \gamma_C.(X_{elite} - X_i^t) \quad (25)$$

Where r1 is random variable, and X_{elite} is elite solution guidance. Along with these contribution, the multi-objective problems are addressed in proposed ADBOA algorithm.

The intelligent foraging and navigational activity of dung beetles is replicated by the proposed adaptive DBO algorithm for iterative solution optimization. In the search space, each dung beetle represents a potential RA solution based on multi-objective problems such as minimization of energy, latency, cost, and SLA violations and minimization of LB and RU. As discussed in the equation (2), the proposed work focused on multi-objective problem EC$(f_1(X_{ec}))$, delay $(f_2(X_d))$, $cost\,(f_3(X_c))$, SLA violations $(f_4(X_{SLA}))$, and LB$(f_5(X_{lb}))$. LB is a maximising when compared to all other functions, however it is a minimization in our situation. In the equation 10-14, the mathematical derivations of all the functions are shown.

Energy Consumption

Cloud data centers require enormous computing power and entail high operational costs that make energy efficiency is highly significant for enabling efficient services. The energy consumption are as follows:

$$f_1(X_{ec}) = \sum_{i=1}^{N} P_i(U_i) \quad (19)$$

The power consumption of resource i is represented by the variable P_i in equation (19), the utilisation level is indicated by U_i, and N is the total number of computing resources. The expression for the power model is:

$$P_i(U_i) = P_i^{idle} + \left(P_i^{max} - P_i^{idle}\right) \times U_i \quad (20)$$

Delay estimation

To ensure QoS-aware scheduling, execution latency is minimized using the mathematical observation is

$$f_2(X_d) = \sum_{j=1}^{M} \left(T_j^{end} - T_j^{arrival}\right) \quad (21)$$

Equation (18) shows that M stands for total tasks, T_j^{end} for task completion time, and $T_j^{arrival}$ for task arrival time.

Cost estimation

The provisioning of cloud resources involves networking and computational expenses. Equation (22) is used to calculate the operational cost estimate. The utilization cost is represented by C_i^{Cpu}, the memory allocation cost by C_i^{mem}, and the bandwidth utilisation cost by C_i^{bw}.

$$f_3(X_c) = \sum_{i=1}^{N} \left(C_i^{Cpu} + C_i^{mem} + C_i^{bw}\right) \quad (22)$$

SLA violation estimation

The objective of mitigating SLA violations is calculated using $f_4(X_{SLA})$. As mentioned in equation (23), this creates a binary indication for SLA violations (0 and 1). For every job j: 1 if the task surpasses SLA restrictions, otherwise 0.

$$f_4(X_{SLA}) = \frac{1}{M}\sum_{i=1}^{N} V_j \quad (23)$$

$$V_j = \begin{cases} 1, if\ SLA\ vilated \\ 0, \quad otherwise \end{cases} \quad (24)$$

Load Balancing

The proposed framework minimizes load imbalance in order to preserve a balanced workload distribution:

$$f_5(X_{lb}) = 1 - \sqrt{\frac{\sum_{i=1}^{N}(U_i - \widehat{U}_i)^2}{N}} \quad (24)$$

Where the average resource consumption is represented by $\widehat{U}_i$. Balanced workload distribution across heterogeneous cloud resources is indirectly maximised by minimising$f_5(X_{lb})$.

The HMARL-based AP-DQN integrated with the proposed adaptive DBO. Priority-aware candidate scheduling actions are first produced by global and local AP-DQN agents based on environmental conditions, resource availability, and workload urgency. These potential solutions are then improved by the DBO using Pareto-optimal multi-objective optimization. The following is a representation of the optimal RA action:

$$A_t^{opt} = DBO\left(A_t^{AP-DQN}\right) \quad (25)$$

In equation (22), actions produced by AP-DQN agents are denoted by A_t^{AP-DQN}, whereas Pareto-optimized allocation techniques are denoted by A_t^{opt}. Finally, the workload migration, adaptive scaling, intelligent provisioning, VM allocation, and container orchestration modules receive the optimal allocation decisions.

Fitness Evaluation

The total fitness of candidate solutions is calculated using weighted target aggregation.

$$F(x) = \sum_{i=1}^{5} w_i f_i(X), \quad (26)$$

$$w.r.t \ \sum_{i=1}^{5} w_i = 1$$

The weighting coefficients of each objectives are represented by w_i. Flexible ranking based on workload urgency and SLA sensitivity is made possible by this weighted evaluation.

The proposed ADBOA algorithm finds the most effective RA solution that meets workload priority requirements, SLA constraints, execution efficiency, energy optimization, and balanced resource utilisation in heterogeneous Big Data–Cloud environments following iterative Pareto-optimal multi-objective optimization. The final optimal allocation strategy looks like this:

$$X^* = \arg\min F(X) \quad (27)$$

In this case, argminfinds the candidate solution that minimises the combined optimization objectives, $F(X)$ reflects the overall multi-objective fitness function, and X^* indicates the globally optimal Pareto-efficient RA solution.

The overall flow of the pseudocode is shown in Pseudocode 1.

Pseudocode 1: Proposed RA Scheme.

Initialization

- Cloud task (T), Virtual Machine (VM), Resources (R), max_iter
- AMO-DBO RA population p,
- Global agent g,
- local AP-DQN agents
- Set AP-DQN reply buffer D
- For each iter. T=1 to max_iter DO

Global Coordination

- Update AMO-DBO population P using equation 2.
- Select resource constraints equation 3.
- Extract global parameters (λ, α)
- Examine the global state

$g_t^s = \{R_t, W_t, SLA_t, Q_t\}$ (equation (7))

- Create sub-goals {g1, g2,...,gN} with G(Sg, λ).
- Global action space (g_t^a) is defined using equation (10)

Local Agent Interaction

- FOR each local agent DO:
- Observe the state (s_t)of Use Q-network
- Choose action (a_i) from the local action space (l_t^a).
- Execute (a_i) receives reward (r_i) and the next state (s_i').

- Priority aware reward estimation using equation (15)
- Adaptive priority based relevance weights estimation (18)
- Determine the sampling probability using equation (17).
- Replay buffer D's mini-batch B is sampled using weighted probability.

- Store transition (s_i, a_i, r_i, s_i')in replay buffer (D) (Eq. 12)
- Compute TD-error (δ_i) using Eq. 13.
- Compute sampling weight (w_i) using equation (14)
- END FOR

Learning Update	
	• Sample mini-batch (B) from replay buffer (D) using ısing equation (17) • FOR each transition $(s_i, a_i, r_i, s_i', w_i)\ \epsilon\ B$ DO: • Compute weighted loss (L) using equation (19). • Update parameters (θ) using equation (20). • END FOR
Termination	
	• END FOR • Return optimized resource solution and trained AP-DQN model.

CLOUD RESOURCE EXECUTION LAYER

The Cloud Resource Execution Layer consists of heterogeneous physical servers, VMs, containers, storage systems, and network infrastructures responsible for executing the optimized scheduling and allocation decisions generated by the HMARL–DBO framework. The optimized allocation policies are dynamically applied to the cloud environment for workload execution, VM provisioning, container deployment, resource scaling, and task scheduling operations. The environmental states are continuously updated according to workload variations and execution outcomes. The updated environmental information is returned to the monitoring layer, forming a closed feedback-driven learning loop that enables continuous adaptation and intelligent resource optimization under dynamically changing Big Data–Cloud environments.

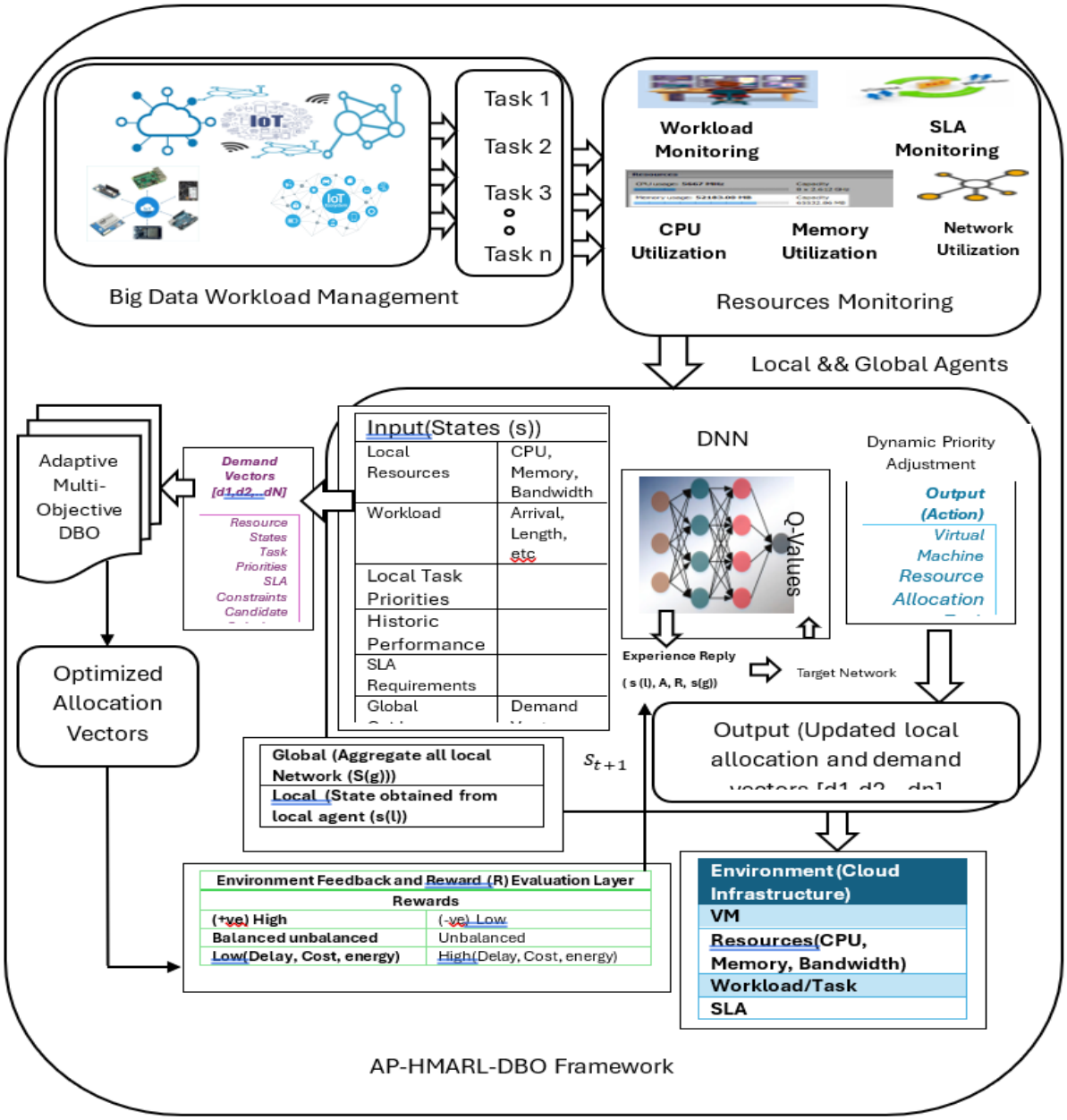


Figure 2: Architecture of the OPHM-DQN with ADBOA framework for IoT-Cloud Resource Management

RESULTS AND DISCUSSION

This section presents the experimental evaluation and performance analysis of the proposed OPHM-DQN-ADBOA framework in heterogeneous Big Data–cloud environments. The proposed model is compared with existing RA and scheduling approaches using multiple performance metrics to validate its scalability, adaptability, and optimization capability under dynamic workload conditions.

Simulation Environment and Workload Configuration

The performance of proposed OPHM-DQN-ADBOA has been tested and analysed in a simulation-based heterogeneous Big Data–Cloud environment. The simulation platform captured the distributed cloud data centres with a set of heterogeneous VMs of different computational power, memory size, bandwidth and EC. Dynamic IoT generated workloads were designed to mimic realistic CC scenarios, such as computation intensive, delay sensitive applications and real-time Big Data applications. The proposed framework was based on two RL agents: a GA and a LA, both of which were hierarchical, with the GA performing decentralized workload scheduling and the LA performing resource orchestration. The workloads generated were of different sizes, prioritization, SLA constraints and arrival rates to test the scalability and adaptability of the proposed framework under low, medium and high workload conditions. To assure reliability and stability of the obtained results, multiple simulation trials were performed.

Table 1: Simulation Environment and Workload Configuration Parameters

Parameter	Configuration
Simulation Environment	Heterogeneous Big Data–Cloud Environment
Resource Architecture	Distributed Multi-Cloud Data Centres
Scheduling Model	HMARL
Optimization Technique	ADBOA
VM Processing Power	500–5000 MIPS
Task Size	500–20,000 MI
RAM Capacity	512 MB–8192 MB
Bandwidth Capacity	100 Mbps–10 Gbps
Workload Type	IoT and Big Data Dynamic Workloads
Workload Categories	Delay-sensitive, Real-time, Computation-intensive
Workload Priority Levels	High, Medium, Low
Performance Conditions	Low, Medium, High Workload Scenarios
QoS Constraints	SLA Deadline and Resource Availability
Experimental Trials	20 ndependent simulation runs

Experimental Setup and Model Training

The proposed OPHM-DQN-ADBOA was used in a Python-based simulation environment to assess the intelligent RA performance in heterogeneous Big Data–cloud infrastructures. The experimental setup was based on assigning different cloud clusters to different distributed nodes, and controlling them using two levelRL agents, one global and the other local, which were structured hierarchically. GAs were responsible for workload coordination, energy-aware scheduling, and SLA management, while LAs were responsible for fine-grained scheduling of VMs and local orchestration of resources. An experience replay and target network update mechanism was used to increase the stability and convergence of learning for DQN model. The State Space comprised CPU utilisation, Memory utilisation, Bandwidth availability, workload priority, queue length, EC and SLA status information. The action space in this case was the allocation of resources and scheduling of workloads on different cloud resources. The goal of the reward mechanism was to maximize the utilization of the resources, the efficiency of their scheduling, and minimize EC, execution delay, operational cost, SLA violation, and workload imbalance. A RL-based framework was adopted and integrated with ADBOA to enhance policy learning process and enhance the exploration-exploitation balance while the algorithm was trained. Adaptive convergence controller, adaptive diversity controller, and adaptive learning-weight controller were used to ensure convergence and avoid early optimization. The proposed framework was trained and tested using a variety of simulation episodes under dynamically varying workloads. All baseline methods were used with the same workloads and infrastructure setup in order to be fairly comparable. The final performance results were achieved by taking the average of the simulated results with multiple independent runs.

Table 2: Experimental Setup and Training Parameters

Parameter	Configuration
Implementation Environment	Python-based Cloud Simulation
Scheduling Framework	Hierarchical multi-Agent DQN
Learning Architecture	DQN
Agent Structure	Global and Local RL Agents
Optimization Method	ADBO
Learning Rate	0.001
Discount Factor (γ)	0.95
Replay Memory Size	10,000
Batch Size	64
Training Episodes	1000
DBO Population Size	50
Maximum Optimization Iterations	100

Performance metrics

To assess scheduling efficiency and system adaptability in heterogeneous Big Data–cloud environments, the performance of the proposed OPHM-DQN-ADBOA framework was investigated using various metrics of QoS and resource optimization. The EC was measured to determine the overall energy used during workload execution and the execution delay was the meantime taken for workload to be completed. The percentage of tasks that did not meet the pre-

defined QoS constraints, called SLA violation rate, was used to measure the violation rate. The measure of the effective use of available cloud resources during scheduling was its RU. Operational cost measured total processing and infrastructure overhead while allocating resources. Moreover, to analyse workload distribution efficiency on distributed cloud resources, the LB factor was also taken into consideration. The overall evaluation measurements have confirmed the ability of the proposed framework to optimize multiple conflicting goals while operating under dynamically changing workloads.

Comparative Performance Analysis

This section presents the comparative performance assessment of OPHM-DQN-ADBOA framework against existing approaches under varying workload conditions.

Energy Consumption Analysis

The comparative EC analysis of the proposed OPHM-DQN-ADBOA framework with the baseline methods is displayed in Table 3 under different workload conditions. The experimental results showed that the proposed framework consistently reduced the EC compared to the optimization-based and RL-based approaches. The proposed framework was found to be energy efficient with only 104.27 kWh under low workload condition, while the existing methods were found to be using higher energy as it is not very efficient in the workload distribution and underutilized in the resources. Likewise, in medium workload and high workload cases, the proposed framework managed the energy usage significantly by adapting the workload, intelligently allocating VMs and hierarchically coordinating multiple agents. Incorporating the ADBOsuccessfully balanced exploration and exploitation in policy learning to enhance the energy-aware RA decisions. Moreover, the priority-based scheduling mechanism avoided unnecessary energy overhead in idle state and activation of multiple resources. The findings validate the effectiveness of the proposed framework in energy efficiency and sustainable resource management in the context of heterogeneous Big Data–Cloud environments.

Table 3: Comparative Energy Consumption Analysis under Different Workload Conditions

Method	Workload (kWh)		
	Low	Medium	High
OBLMFO	142.35	228.74	341.62
IC&BA	136.81	219.53	328.47
RA-HAPCNN-CC	131.64	210.28	315.91
MADRL	126.92	201.47	301.36
WWO-ACO	123.58	196.84	294.73
DRL-EEFO	118.46	188.29	281.54
OPHM-DQN-ADBOA	**104.27**	**167.83**	**236.48**

Execution Delay Analysis

The comparative analysis of execution delay of proposed OPHM-DQN-ADBOA algorithm and existing baseline algorithms is presented in Table 4 under different workload scenarios. The results show that the proposed framework could successfully reduce execution delay compared to the traditional optimization and RL based scheduling approaches, throughout the experiments. In the case of low workload, the proposed model has execution delay of 298.43ms which is much less than the base model. Similarly, under medium and high workload scenarios, the proposed framework preserved low task completion time while the workload was increased. The main contribution to the improved execution performance was through the use of hierarchical multi-agent coordination and adaptive priority-aware scheduling mechanisms. GAs were able to distribute workloads across distributed cloud clusters and LAs were able to schedule VMs at the fine-grained level and orchestrate them adaptively. The ADBOA algorithm also improved the convergence of the schedules and reduced contention of the resources during workload execution. The design successfully decreased queue waiting times, minimized workloads migration overhead and resource bottlenecks, resulting in a more efficient performance of real-time tasks. Results confirm the ability of the proposed OPHM-DQN-ADBOA framework to enable basic data applications with delay sensitivity and computational complexity in heterogeneous cloud environments.

Table 4: Comparative Execution Delay Analysis under Different Workload Conditions

Method	Workload		
	Low (ms)	Medium (ms)	High (ms)

OBLMFO	412.58	638.41	924.76
IC&BA	398.37	612.25	887.14
RA-HAPCNN-CC	384.62	586.73	842.58
MADRL	369.84	561.48	801.36
WWO-ACO	356.29	542.17	774.92
DRL-EEFO	341.75	519.84	738.45
OPHM-DQN-ADBOA	**298.43**	**451.26**	**624.81**

SLA Violation Analysis

The comparative analysis of SLA violation for proposed OPHM-DQN-ADBOA framework and baseline methods under various workloads are shown in Table 5. The experimental results show that the proposed framework outperformed the other two frameworks in terms of the number of SLAs violations in all workload scenarios. In cases of low workload, the proposed framework recorded an SLA violation rate of just 4.18%, while the baseline methods had comparatively higher rates of SLA violations which resulted from inefficient workload prioritisation and resource holding. Likewise, throughout the medium and high workload, the proposed framework demonstrated good QoS performance with growing workload complexity. The adaptive priority-aware scheduling mechanism and hierarchical multi-agent coordination strategy are the key factors to reduce SLA violations. GA monitored workload conditions and SLA constraints and LAs dynamically made decisions on how to allocate resources according to workload urgency and execution deadlines, as well as resource availability. In addition, the ADBOA algorithm enhanced the scheduling stability and supported an efficient workload-resource mapping in dynamically changing cloud conditions. The proposed framework effectively reduced the delays in task execution, resource bottlenecks and workload imbalance, thus enhancing the satisfaction of the SLA and reliability of QoS. The results achieved in the obtained demonstrate the effectiveness of the proposed OPHM-DQN-ADBOA framework in providing robust and adaptive SLA-aware resource management for heterogeneous Big Data–cloud environments.

Table 5: Comparative SLA Violation Analysis under Different Workload Conditions

Method	Workload		
	Low (%)	Medium (%)	High (%)
OBLMFO	8.42	13.87	21.64
IC&BA	7.95	12.94	19.83
RA-HAPCNN-CC	7.31	11.86	18.27
MADRL	6.84	10.92	16.58
WWO-ACO	6.27	10.14	15.42
DRL-EEFO	5.73	9.28	13.76
OPHM-DQN-ADBOA	**4.18**	**6.95**	**10.21**

Resource Utilization Analysis

The RU analysis of the proposed OPHM-DQN-ADBOA framework and some baseline methods is shown in Table 6 for various workloads. The results show that the proposed system always outperforms conventional optimization and RL-based scheduling methods in terms of RU. With low workloads the proposed framework realized 86.93 percent RU and the baseline methods had lower utilization efficiency because they lacked the ability to map workload to resources and they were underutilizing resources. Likewise, the proposed framework also showed good utilization performance under the medium and high workload scenarios as workloads increased. The use of resources was mainly realised by adopting hierarchical multi-agent coordination and adapting workload scheduling mechanisms. GA was used to efficiently distribute workloads across the distributed cloud clusters, while LAs flexibly allocated VMs and orchestrated the local resources in response to the real-time system conditions. Moreover, the ADBOA algorithm improved algorithm convergence and effectively allocated computational resources based on the workload priority and SLA requirements. The proposed framework proved to be efficient in reducing idle time of resources and in balancing the workloads in heterogeneous cloud infrastructures. As a result, the system was more computationally efficient and had lower operating costs in dynamic cloud environments. The results obtained confirmed the possibility of developing a scalable and resource-efficient workload management in a heterogeneous Big Data–Cloud environment using the proposed OPHM-DQN-ADBOA framework

Table 6: Comparative Resource Utilization Analysis under Different Workload Conditions

Method	Workload		
	Low (%)	Medium (%)	High (%)
OBLMFO	71.46	76.82	81.37
IC&BA	73.18	78.54	83.29
RA-HAPCNN-CC	75.62	80.17	84.93
MADRL	77.35	82.48	86.71
WWO-ACO	79.14	84.23	88.05
DRL-EEFO	81.76	86.58	90.42
OPHM-DQN-ADBOA	**86.93**	**91.47**	**94.85**

Load Balancing Analysis

Comparative load balancing analysis of the proposed framework of OPHM-DQN-ADBOA and baseline approaches is shown in Table 7 under different workloads. The results show that the proposed framework was able to consistently outperform the other optimization and RL-based scheduling methods in terms of workload distribution efficiency. The proposed approach showed a LB factor of 85.73% for low workload, while the baseline approaches achieved relatively lower LB factors because of uneven workload distribution and resource contention. Further, under medium and high workloads, the proposed framework kept the workloads balanced while the number of tasks arriving in the cloud and their computational requirements increased. Hierarchical global-local agent coordination was the key feature that contributed to the enhanced LB performance. GAs continuously monitored the utilization of all the cloud resources and dynamically distributed workloads between different cloud clusters, whereas LAs adapted VM scheduling and orchestrated workloads locally. In addition, the ADBOA based technique enhanced the scheduling stability and allowed the efficient mapping of workload and resources in an environment with dynamic cloud conditions. The proposed framework was able to reduce the resource overloading, queue congestion and workload imbalance in different cloud infrastructures effectively. The result was increased resource efficiency, decreased processing bottlenecks, and increased scheduling stability when scaling workloads through the system. The results obtained reveals that the proposed OPHM-DQN-ADBOA framework is scalable and adaptive LB for a heterogeneous Big Data–cloud environment.

Table 7: Comparative Load Balancing Analysis under Different Workload Conditions

Method	Workload		
	Low (%)	Medium (%)	High (%)
OBLMFO	68.42	72.18	75.64
IC&BA	70.37	74.26	77.83
RA-HAPCNN-CC	73.14	76.92	80.45
MADRL	75.68	79.35	82.71
WWO-ACO	77.42	81.26	84.93
DRL-EEFO	80.15	83.84	87.36
OPHM-DQN-ADBOA	**85.73**	**89.48**	**92.64**

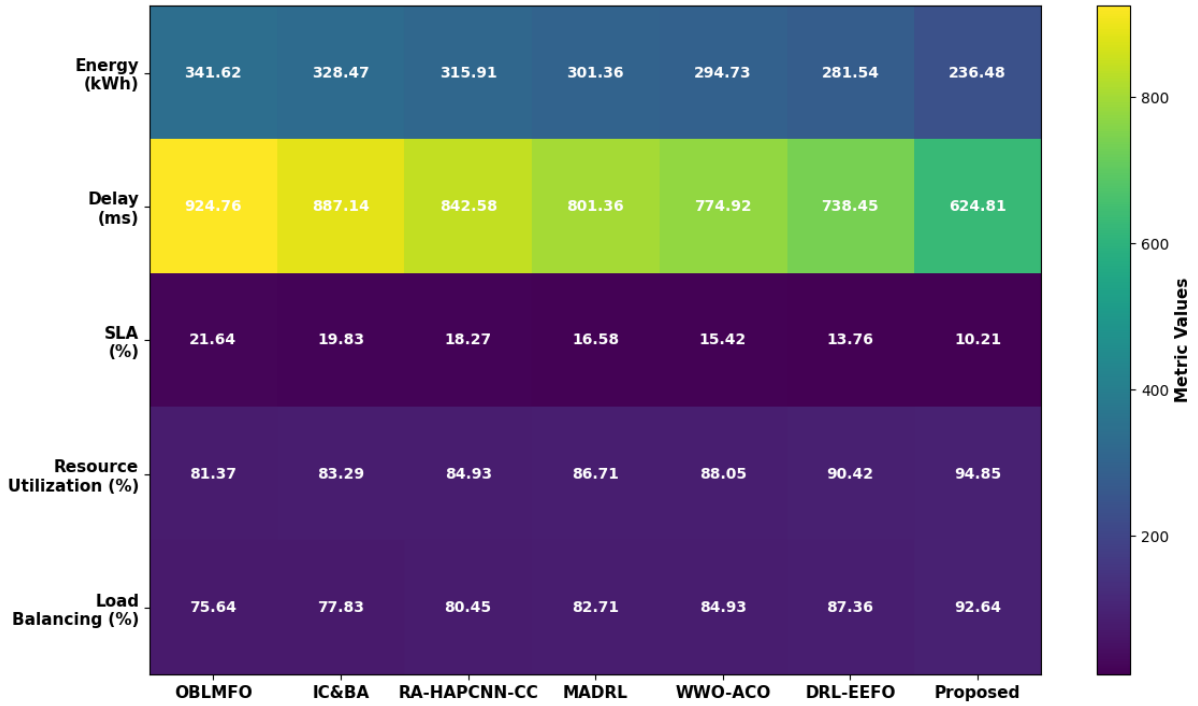


Figure 3: Heatmap analysis of proposed and baseline methods over various evaluation metrics

The heatmap in Figure 3 demonstrates how the suggested framework consistently outperforms baseline techniques in terms of various evaluation metrics. Darker areas indicate better outcomes for the suggested method, and the colour intensity draws attention to performance disparities.

Convergence Analysis

The comparison of the proposed ADBOA technique with the existing optimization algorithms is shown in Table 8. Fitness minimization ability, convergence rate, and optimization accuracy were used to evaluate the convergence performance when optimizing RA policies. Results show that the proposed ADBOA has faster and stable convergence than conventional optimization algorithms. The proposed ADBOA resulted in the lowest final fitness value of 0.214 and stable convergence within 42 iterations; the baseline optimization methods took a much higher number of iterations to be converged. Furthermore, the proposed optimization framework showed the best convergence accuracy of 96.83%, demonstrating that the framework has the best performance in terms of finding optimal workload-RA policies under dynamically varying cloud conditions.The adaptive convergence control, adaptive diversity control and adaptive learning-weight mechanisms were integrated with the DBO framework to achieve better convergence performance. The adaptive convergence controller was helpful to enhance the exploitation capability during later stages of optimization, and the adaptive diversity controller was effective in maintaining the diversity of populations and preventing them from converging on local optima. In addition, the adaptive learning-weight mechanism dynamically adapted the exploration and exploitation during optimization.The proposed ADBOA effectively enhanced the search capability and optimization stability of the search process in RL policy optimization in the world. Therefore, the proposed OPHM-DQN-ADBOA system could effectively improve the scheduling efficiency, minimize EC, optimize workload balance and decrease the number of SLA violations in cloud environments. The results obtained are found to be valid, and thus confirms the robustness and optimization efficiency of the proposed ADO strategy for intelligent cloud RA.

Table 8: Comparative Convergence Analysis of Optimization Algorithms

Optimization Method	Initial Fitness Value	Final Fitness Value	Iterations for Stable Convergence	Convergence Accuracy (%)
MFO	0.842	0.417	86	81.24
IC&BA	0.816	0.392	79	83.67
SMO	0.801	0.368	73	85.42
WWO-ACO	0.774	0.341	67	87.16

EEFO	0.752	0.319	61	89.48
Conventional DBO	0.731	0.287	56	91.35
ADBOA	**0.718**	**0.214**	**42**	**96.83**

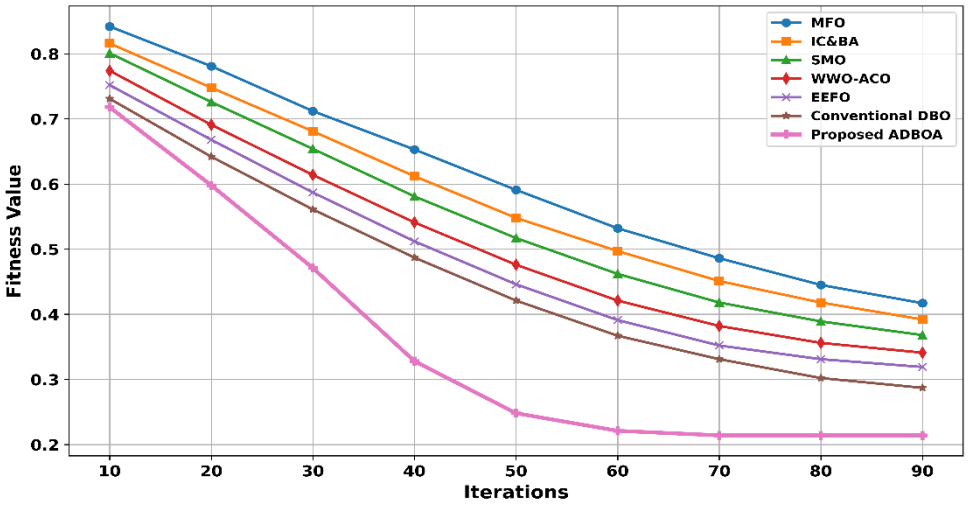


Figure 4: Convergence analysis of proposed and conventional optimization algorithms

In comparison to traditional optimization techniques, the proposed OPHM-DQN-ADBOA framework exhibits faster and more stable convergence, as illustrated in Figure 4. The curve shows that the adaptive multi-objective Dung Beetle Optimizer ensures strong performance under dynamic IoT-Cloud workloads by accelerating convergence while preventing premature stagnation.

Ablation Study Analysis

The analysis of the ablation study of the proposed OPHM-DQN-ADBOA for assessing individual contribution on overall scheduling performance is presented in Table 9. The analysis carried out was to systematically exclude certain modules from the proposed framework and assess the performance under high workload conditions. The results show that without hierarchical multi-agent coordination mechanism a high EC, execution delay, and workload imbalance occurred because of the limitations of centralized scheduling and inefficient workload distribution. Likewise, without the priority-aware scheduling module, the SLA violation percentage was the highest as delay-sensitive workloads were not given a high priority during RA. No ADBOA caused optimization instability and reduced the efficiency of the scheduling process, resulting in increased EC and delay in execution. Moreover, taking adaptive control parameters out of the optimization process degraded the convergence stability and the exploration-exploitation balancing abilities, and thus decreased the overall RU as well as LB performance. The configuration without multi-objective optimization also demonstrated the poor scheduling performance because of the absence of simultaneous optimization of conflicting objectives such as energy efficiency, minimizing delay, and satisfying the SLA. Overall, the complete OPHM-DQN-ADBOA configuration was the most effective with the lowest EC, the least execution delay, the least SLA violation rate, and the highest RU and LB efficiency, among all the configurations. The results retrieved show that the proposed framework combines hierarchical RL, adaptive priority-aware scheduling and adaptive multi-objective optimization for achieve the best results in the considered heterogeneous Big Data–Cloud environment.

Table 9: Ablation Study Analysis of the Proposed OPHM-DQN-ADBOA Framework

Model Configuration	Energy Consumption (kWh)	Execution Delay (ms)	SLA Violation (%)	Resource Utilization (%)	Load Balancing (%)
Without Hierarchical Agents	278.64	731.42	14.86	85.17	81.53
Without Priority-Aware Scheduling	269.35	702.58	16.27	86.94	84.12
Without Adaptive DBO	257.18	681.34	12.94	88.26	86.75
Without Adaptive Control Parameters	248.92	653.87	11.83	89.74	88.41
Without Multi-Objective Optimization	263.47	694.26	13.68	87.15	85.38

Proposed OPHM-DQN-ADBOA	**236.48**	**624.81**	**10.21**	**94.85**	**92.64**

Scalability and Adaptability Analysis

Table 10 shows the scalability and adaptability analysis of the proposed OPHM-DQN-ADBOA framework for the growing workload. The experimental analysis was carried out by incrementing the number of workload tasks in a heterogeneous Big Data–Cloud environment to assess the ability of the proposed framework to deal with dynamic scheduling scenarios involving a large number of tasks. The adaptability of the hierarchical multi-agent scheduling system is demonstrated in Figure 5, where scheduling efficiency and SLA satisfaction gradually decline with increasing workloads while resource usage continually improves. The results show that the proposed framework is able to schedule the workload without any noticeable impact on scheduling performance. The number of tasks has been increased by a factor of 50 from 500 to 5000, but the scheduling efficiency has been maintained consistently high, 89.74% under large workload conditions. Likewise, the RU continually increased as the workload became more intensive, thanks to the adaptive workload distribution feature of the hierarchical multi-agent scheduling mechanism. The proposed framework also managed to achieve high levels of SLA satisfaction even with higher computational requirements and contention.

The orchestration of RL agents at various levels of the hierarchy, and their efficient workload balancing and adaptive resource orchestration across distributed cloud infrastructures, were achieved. Moreover, the ADBOA technique adaptively adjusted RA policies under the influence of varying loads, enhancing adaptability and convergence stability. The observed delay was still within acceptable QoS limits at higher workload intensities; however, it did increase gradually with the intensity of the workload and efficient scheduling and workload prioritization mechanisms ensured that the delay was not excessive. The results obtained validate the performance of the proposed OPHM-DQN-ADBOA framework for scalable, adaptive, and stable resource management in heterogeneous Big Data–Cloud environments with dynamically changing workload.

Table 10: Scalability and Adaptability Analysis under Increasing Workload Conditions

Number of Tasks	Scheduling Efficiency (%)	Resource Utilization (%)	SLA Satisfaction (%)	Average Execution Delay (ms)
500	94.12	89.47	95.63	312.45
1000	93.48	90.82	94.76	384.27
2000	92.76	91.94	93.85	462.18
3000	91.83	93.16	92.71	548.64
4000	90.95	94.03	91.84	587.39
5000	89.74	94.85	89.79	624.81

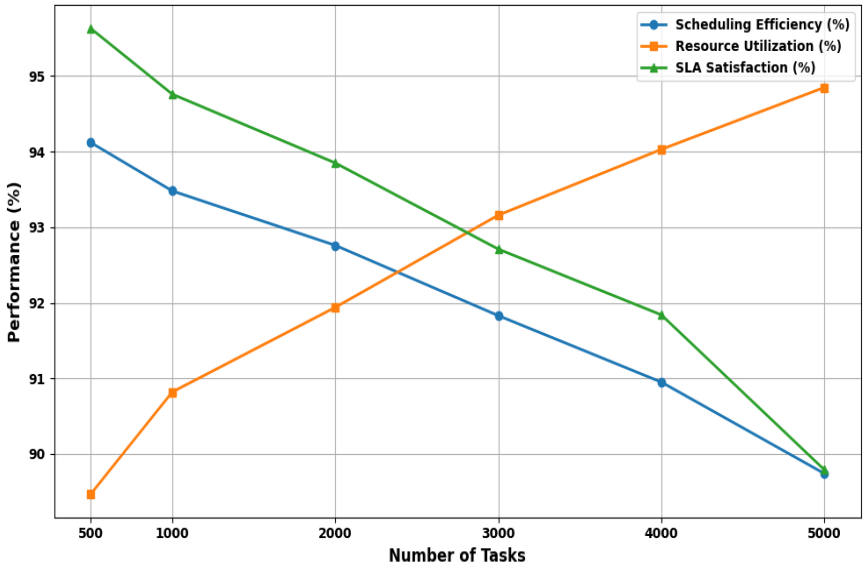


Figure 5: Performance analysis of scalability of proposed model

The proposed OPHM-DQN-ADBOA framework performs steadily as the number of IoT devices and workload intensity rise, as seen in Figure 5. In comparison to baseline techniques, the scalability analysis shows that the model effectively adjusts to bigger environments, maintaining high throughput and low latency.

Statistical Significance Analysis

Table 11 shows the statistical significance test between the OPHM-DQN-ADBOA and the average performance of the baseline RA approaches. Repeated simulation experiments were performed under the same workload to assess the reliability and consistency of the results obtained. The outcome shows the effectiveness of the proposed framework as significant improvements are achieved for all the evaluation metrics. The OPHM-DQN-ADBOA could lower the energy usage by 19.47% and execution delay by 22.75% over the existing baseline models. Likewise, the proposed framework significantly reduced the violation rate of SLA by 43.02%, which shows better satisfaction with QoS and better adaptation capability to workload management in heterogeneous cloud environments. Besides, the proposed framework could gain maximum RU and better LB thanks to the hierarchical multi-agent scheduling architecture and adaptive optimization strategy. The p-values for all the evaluation metrics obtained were found to be less than 0.05, thus, the performance gains realized were found to be significant and not only random variations in experiments. The results obtained from repeated simulation runs indicate the robustness, stability and adaptability of the proposed framework with dynamically changing workload conditions. The results from the statistical analysis demonstrates that the performance of intelligent cloud RA in the context of the heterogeneous Big Data–cloud environment can be effectively improved by the integration of hierarchical RL and ADBOA.

Table 11: Statistical Significance Analysis of the Proposed OPHM-DQN-ADBOA Framework

Performance Metric	Baseline Average	Proposed OPHM-DQN-ADBOA	Improvement (%)	p-value
Energy Consumption (kWh)	293.67	236.48	19.47	0.0031
Execution Delay (ms)	808.87	624.81	22.75	0.0024
SLA Violation (%)	17.92	10.21	43.02	0.0018
Resource Utilization (%)	85.13	94.85	11.42	0.0042
Load Balancing (%)	82.84	92.64	11.83	0.0037

DISCUSSION

The experimental results show that the proposed OPHM-DQN-ADBOA is an effective framework to enhance the intelligent RA performance of a heterogeneous Big Data–Cloud environment. The proposed framework achieved a better performance in terms of EC, execution delay, SLA violation reduction, resource usage and LB efficiency compared to the other existing optimization and RL-based scheduling methods. The hierarchical multi-agent RL architecture allowed to coordinate global and LA interactions, which led to scalable and decentralized workload scheduling. Furthermore, the priority-aware scheduling mechanism enhanced QoS performance by dynamically allocating resources depending on the urgency of the workloads and SLA requirements. The convergence stability and optimization efficiency were significantly improved by implementing adaptive explorative-exploitative balancing of ADBOA technique. The scalability analysis demonstrated the proposed framework to be able to continue the stable scheduling performance even with the increase of workload. In addition, importance of hierarchical coordination, adaptive optimization and priority-aware scheduling in improving the performance of the system was confirmed in this ablation study. In summary, the proposed OPHM-DQN-ADBOA framework is scalable, adaptive, and energy-efficient to manage the cloud resources in the dynamically changing Big Data–cloud environments.

Limitations

The proposed framework has successfully led to significant gains in scheduling efficiency and RA, but there are still some challenges to be addressed. Several adaptive learning and optimization modules are built together, which leads to higher training complexity of the model. Secondly, for large-scale real-time cloud deployments, additional optimizations can be done on the communication overhead between distributed agents. In future, further improvements in scalability and real-world applicability can be achieved by developing lighter distributed RL models, incorporating federated learning, and conducting real-world deployments analysis of the cloud.

CONCLUSION

This proposed framework OPHM-DQN-ADBOA is used for smart resource scheduling in heterogeneous Big Data-cloud environments. It was designed to solve the problems that arise in cloud resource management including high EC, execution

delay, SLA problem, workload imbalance, and inefficient use of cloud resources in dynamically changing workload. The proposed framework is a hierarchical multi-agent RL-based framework with GAs and LAs to schedule workloads in a decentralized and adaptive manner. The GA orchestrated workloads at the system level, taking care of SLA-aware resource management, while the LAs took care of fine-grained VM scheduling and localized resource orchestration. The ADBOA was also applied to the RL framework to enhance the multi-objective optimization of the EC minimization, execution delay minimization, workload balancing improvement, SLA satisfaction enhancement, and RU maximization. The proposed OPHM-DQN-ADBOA framework was evaluated experimentally in a simulation scenario with heterogeneous Big Data cloud environment, which validated the effectiveness of the proposed framework compared to other RA methods based on optimization and RL in a heterogeneous Big Data–cloud environment. The proposed OPHM-DQN-ADBOA framework offers a scalable, adaptive and energy-efficient intelligent resource management solution for next generation of heterogeneous Big Data–cloud environments. Future research can aim to combine federated RL, real-time edge-cloud collaboration and lightweight distributed optimization techniques to further enhance practical deployment capability and computation efficiency in large scale cloud infrastructures.

ACKNOWLEDGEMENT

REFERENCES

Acharya, B., Panda, S., Das, S., Majhi, S. K., Gerogiannis, V. C., & Kanavos, A. (2025). Optimizing task scheduling in cloud environments: A hybrid golden search whale optimization algorithm approach. *Neural Computing and Applications, 37*(17), 10851–10873. https://doi.org/10.1007/s00521-025-11113-9.

Alam, M., Shahid, M., & Mustajab, S. (2024). Security challenges for workflow allocation model in cloud computing environment: A comprehensive survey, framework, taxonomy, open issues, and future directions. *The Journal of Supercomputing, 80*(8), 11491–11555. https://doi.org/10.1007/s11227-024-05932-8.

Awad, W. K., Ariffin, K. A. Z., Nazri, M. Z. A., & Yassen, E. T. (2025). Resource allocation strategies and task scheduling algorithms for cloud computing: A systematic literature review. *Journal of Intelligent Systems, 34*(1). https://doi.org/10.1515/jisys-2024-0476.

Dankolo, N. M., Radzi, N. H. M., Mustaffa, N. H., Arshad, N. I., Nasser, M., Gabi, D., & Yusuf, M. N. (2025). Optimizing resource allocation for IoT applications in the edge cloud continuum using hybrid metaheuristic algorithms. *Scientific Reports, 15*(1), Article 14409. https://doi.org/10.1038/s41598-025-97648-2

Fan, W., Chen, P., Chun, X., & Liu, Y. (2025). MADRL-based model partitioning, aggregation control, and resource allocation for cloud-edge-device collaborative split federated learning. *IEEE Transactions on Mobile Computing, 24*(6), 5324–5341. https://doi.org/10.1109/TMC.2025.3530482

Geetha, P., Vivekanandan, S. J., Yogitha, R., & Jeyalakshmi, M. S. (2024). Optimal load balancing in cloud: Introduction to hybrid optimization algorithm. *Expert Systems with Applications, 237*, Article 121450. https://doi.org/10.1016/j.eswa.2023.121450

Ghandour, O., El Kafhali, S., & Hanini, M. (2023). Computing resources scalability performance analysis in cloud computing data center. *Journal of Grid Computing, 21*(4), Article 61. https://doi.org/10.1007/s10723-023-09708-2

Gurusamy, S., & Selvaraj, R. (2024). Resource allocation with efficient task scheduling in cloud computing using hierarchical auto-associative polynomial convolutional neural network. *Expert Systems with Applications, 249*, Article 123554. https://doi.org/10.1016/j.eswa.2024.123554

Hao, Y., Zhao, C., Li, Z., Si, B., & Unger, H. (2024). A learning and evolution-based intelligence algorithm for multi-objective heterogeneous cloud scheduling optimization. *Knowledge-Based Systems, 286*, Article 111366. https://doi.org/10.1016/j.knosys.2024.111366

Huang, C., Chen, G., Xiao, P., Xiao, Y., Han, Z., & Chambers, J. A. (2024). Joint offloading and resource allocation for hybrid cloud and edge computing in SAGINs: A decision assisted hybrid action space deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications, 42*(5), 1029–1043. https://doi.org/10.1109/JSAC.2024.3365899

Ismail, A. A., Khalifa, N. E., & El-Khoribi, R. A. (2025). A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study. *Cluster Computing, 28*(3), Article 184. https://doi.org/10.1007/s10586-024-04893-7

Jalali Khalil Abadi, Z., Mansouri, N., & Javidi, M. M. (2024). Deep reinforcement learning-based scheduling in distributed systems: A critical review. *Knowledge and Information Systems, 66*(10), 5709–5782. https://doi.org/10.1007/s10115-024-02185-3

Jayanetti, A., Halgamuge, S., & Buyya, R. (2024). Multi-agent deep reinforcement learning framework for renewable energy-aware workflow scheduling on distributed cloud data

centers. *IEEE Transactions on Parallel and Distributed Systems, 35*(4), 604–615. https://doi.org/10.1109/TPDS.2024.3360448

Jeyaraj, R., Balasubramaniam, A., Kumara, A. M. A., Guizani, N., & Paul, A. (2023). Resource management in cloud and cloud-influenced technologies for Internet of Things applications. *ACM Computing Surveys, 55*(12), 242:1–242:37. https://doi.org/10.1145/3571729

Kushchazli, A., Leonteva, K., Kochetkova, I., & Khakimov, A. (2025). Evaluating QoS in dynamic virtual machine migration: A multi-class queuing model for edge-cloud systems. *Journal of Sensor and Actuator Networks, 14*(3), Article 47. https://doi.org/10.3390/jsan14030047

Lahza, H., BR, S., & Lahza, H. F. M. (2024). Adaptive multi-objective resource allocation for edge-cloud workflow optimization using deep reinforcement learning. *Modelling, 5*(3), 1298–1313. https://doi.org/10.3390/modelling5030062

Lilhore, U. K., Simaiya, S., Prajapati, Y. N., Rai, A. K., Ghith, E. S., Tlija, M., Lamoudan, T., & Abdelhamid, A. A. (2025). A multi-objective approach to load balancing in cloud environments integrating ACO and WWO techniques. *Scientific Reports, 15*(1), Article 12036. https://doi.org/10.1038/s41598-025-96364-1

Mittal, P., Kumar, D. S., & Sharma, D. S. (2024). Revolutionizing cloud-based task scheduling: A novel hybrid algorithm for optimal resource allocation and efficiency in contemporary networked systems. *International Journal of Computing and Digital Systems, 15*(1), 1551–1563.

Momenikorbekandi, A., & Abbod, M. (2023). Intelligent scheduling based on reinforcement learning approaches: Applying advanced Q-learning and state–action–reward–state–action reinforcement learning models for the optimisation of job shop scheduling problems. *Electronics, 12*(23), Article 4752. https://doi.org/10.3390/electronics12234752

Monroe, J. L., & Langford, E. D. (2024). The integration of Internet of Things, big data analytics, and cloud computing technologies for real-time application development. *European Journal of Emerging Real-Time IoT and Edge Infrastructures, 1*(1), 1–11.

Murali, J. A., & Brindha, T. (2023). A meta-heuristic clustered grey wolf optimization algorithm for cloud resource scheduling. *International Journal of Advanced Technology and Engineering Exploration, 10*(107), 1336.

Nambi, S., & Thanapal, P. (2025). Emo-TS: An enhanced multi-objective optimization algorithm for energy-efficient task scheduling in cloud data centers. *IEEE Access, 13*, 8187–8200.

Nematollahi, M., Ghaffari, A., & Mirzaei, A. (2024). Task and resource allocation in the Internet of Things based on an improved version of the moth-flame optimization algorithm. *Cluster Computing, 27*(2), 1775–1797. https://doi.org/10.1007/s10586-023-04260-6

Prabha, S. P., & Rengarajan, A. (2025). Enhancing cloud resource allocation with vision transformer, deep reinforcement learning, and improved shrike optimization algorithm. *Corrosion Management, 35*(2), 233–245.

Qadoua, K. A. M., & Khalil, E. M. A. (2025). Resource allocation challenges and solutions in cloud computing for machine learning workloads: A comprehensive review. *Libyan Journal of Educational Research and E-Learning*, 290–307.

Qi, W., Sun, M., & Hosseini, S. R. A. (2023). Facilitating big-data management in modern business and organizations using cloud computing: A comprehensive study. *Journal of Management & Organization, 29*(4), 697–723. https://doi.org/10.1017/jmo.2022.21.

Radhika, D., & Duraipandian, M. (2025). Optimized dynamic task scheduling in cloud computing for big data processing. *Wireless Networks, 31*(5), 3661–3672. https://doi.org/10.1007/s11276-025-04053-z.

Rafiq, I., Mahmood, A., Razzaq, S., Jafri, S. H. M., & Aziz, I. (2023). IoT applications and challenges in smart cities and services. *The Journal of Engineering, 2023*(4), e12262. https://doi.org/10.1049/tje2.12262.

Shukla, P., & Pandey, S. (2024). MOTORS: Multi-objective task offloading and resource scheduling algorithm for heterogeneous fog-cloud computing scenario. *The Journal of Supercomputing, 80*(15), 22315–22361. https://doi.org/10.1007/s11227-024-06315-2.

Sivamayil, K., Rajasekar, E., Aljafari, B., Nikolovski, S., Vairavasundaram, S., & Vairavasundaram, I. (2023). A systematic study on reinforcement learning based applications. *Energies, 16*(3), Article 1512. https://doi.org/10.3390/en16031512.

Sonani, R. (2023). Hierarchical multi-agent reinforcement learning framework with cloud-based coordination for scalable regulatory enforcement in financial systems. *Spectrum of Research, 3*(2).

Thorne, A. (2026). Optimizing task scheduling and resource distribution in heterogeneous cloud and fog ecosystems: A comprehensive analysis of hybrid bio-inspired meta-heuristics and multi-objective frameworks. *The American Journal of Engineering and Technology, 8*(1), 271–276.

Wang, Y., Yu, J., & Yu, Z. (2023). Resource scheduling techniques in cloud from a view of coordination: A holistic survey. *Frontiers of Information Technology & Electronic Engineering, 24*(1), 1–40. https://doi.org/10.1631/FITEE.2100298.

Xue, J., & Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing, 79*(7), 7305–7336. https://doi.org/10.1007/s11227-022-04959-6.

Yu, J., Liang, L., Ye, H., & Jin, S. (2025). Hierarchical multi-agent reinforcement learning-based coordinated spatial reuse for next generation WLANs. *IEEE Transactions on Mobile Computing*. https://doi.org/10.1109/TMC.2025.3649899.

Zhang, Y., Zhou, Y., & Fujita, H. (2024). Distributed multi-agent reinforcement learning for cooperative low-carbon control of traffic network flow using cloud-based parallel optimization. *IEEE Transactions on Intelligent Transportation Systems, 25*(12), 20715–20728. https://doi.org/10.1109/TITS.2024.3452430.

Zhao, X., & Wang, G. (2023). Deep Q networks-based optimization of emergency resource scheduling for urban public health events. *Neural Computing and Applications, 35*(12), 8823–8832. https://doi.org/10.1007/s00521-022-07696-2.

Zhou, G., Tian, W., Buyya, R., Xue, R., & Song, L. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review, 57*(5), Article 124. https://doi.org/10.1007/s10462-024-10756-9.