African Journal of Advanced Pure and Applied Sciences (AJAPAS)
*Online ISSN: 2957-644X*
Volume 4, Issue 4, 2025
Page No: 344-353
Website: https://aaasjournals.com/index.php/ajapas/index
ISI 2025: 1.126 SJIFactor 2024: 6.752 معامل التأثير العربي: 1.62

# Integrating Deep Reinforcement Learning (DRL) with GAACO for Resource Scheduling in Cloud Computing

Fatma Rjab Almasre *
Department of Computer Science, Faculty of Education Ghat, Sebha University, Ghat, Libya

# دمج التعلم التعزيزي العميق (*DRL*) مع *GAACO* لجدولة الموارد في الحوسبة السحابية

فاطمة رجب المصري *
قسم الحاسوب، كلية التربية غات، جامعة سبها، غات، ليبيا

*Corresponding author: fatm.almasre@sebhau.edu.ly

| Received: August 18, 2025 | Accepted: October 22, 2025 | Published: November 02, 2025 |
|---|---|---|

**Abstract:**
Efficient scheduling of resources is a major concern in cloud computing, which is mainly due to dynamic workloads, scarcity of resources and requirement for quality of service (QoS). This paper develops a hybrid dispatching model, which combines Genetic Algorithm-Ant Colony Optimization (GAACO) with Deep Reinforcement Learning (DRL). The hybrid algorithm was implemented and evaluated using CloudSim 3.0.3 to improve the adaptivity and efficiency. Experimental results demonstrate that GAACO+DRL consistently outperforms GAACO alone, reducing makespan by up to 30%, lowering average waiting time by 20–35%, improving throughput, balancing workload distribution, reducing energy consumption, and completely eliminating SLA violations. These findings highlight the effectiveness of combining met heuristic optimization with reinforcement learning to achieve stable, efficient, and scalable resource scheduling in cloud computing.

**Keywords**: Cloud Computing, Deep Reinforcement Learning, Resource scheduling, Intelligent Resource Management.

**الملخص**
تُعدّ جدولة الموارد بكفاءة مصدر قلق كبير في الحوسبة السحابية، ويعود ذلك أساسًا إلى ديناميكيات أعباء العمل، وندرة الموارد، والحاجة إلى جودة الخدمة. تُطوّر هذه الورقة نموذج توزيع هجين يجمع بين خوارزمية جينية لتحسين مستعمرات النمل (GAACO) وتعلم التعزيز العميق (DRL). طُبّقت الخوارزمية الهجينة وقُيّمت باستخدام CloudSim 3.0.3 لتحسين تكيف وكفاءة الحوسبة السحابية. تُظهر النتائج التجريبية أن خوارزمية GAACO+DRL تتفوق باستمرار على خوارزمية GAACO وحدها، حيث تُقلّل زمن الوصول بنسبة تصل إلى 30%، وتُخفّض متوسط وقت الانتظار بنسبة تتراوح بين 20% و35%، وتُحسّن الإنتاجية، وتُوازن توزيع أعباء العمل، وتُخفّض استهلاك الطاقة، وتُزيل تمامًا انتهاكات اتفاقيات مستوى الخدمة (SLA). تُسلّط هذه النتائج الضوء على فعالية الجمع بين التحسين الفوقي والتعلم التعزيزي لتحقيق جدولة موارد مستقرة وفعالة وقابلة للتطوير في الحوسبة السحابية.

**الكلمات المفتاحية**: الحوسبة السحابية، التعلم التعزيزي العميق، جدولة الموارد، إدارة الموارد الذكية.

## Introduction

In recent years, cloud computing has been widely adopted. Cloud computing pertains to centralized computing resources; users access these resources to perform calculations, and the cloud computing center delivers the program's results back to the user [1,2]. Cloud computing serves not only individual users but also enterprise clients. By leasing a cloud server, users avoid purchasing a large fleet of computers, thereby reducing computing costs. Cloud service providers can dynamically allocate computing resources according to users' access demands to enhance resource utilization efficiency [3,4].

 

Rational resource allocation is essential in cloud computing. In cloud resource allocation, the cloud computing center possesses limited resources, and users arrive sequentially. Each user requests the center to employ a certain number of cloud resources at a specific time. The allocation process must account for the diverse needs of users [5,6].

This study primarily considers assigning users to nearby servers to improve service quality. However, if users are directed to proximate servers, some servers may become congested, leading to long waiting times. Consequently, this work expands the Genetic Algorithm-Ant Colony Optimization-Deep Reinforcement Learning (GAACO-DRL) algorithm by integrating deep reinforcement learning techniques, enabling the system to adapt to dynamic workload and cloud resource changes in real-time and to enhance performance over time.

## RELATED WORK

Nguyen et al. [7] proposed a genetic algorithm (GA) for virtual machine allocation in private clouds designed to reduce energy consumption. The algorithm employed a hierarchical model connecting physical and virtual resources, with a fitness function that measures energy use. Tests demonstrated it surpassed the traditional Best-Fit Decreasing method, however it was confined to static allocation, lacking support for dynamic scheduling or live migration, suggesting room for enhancement via advanced hybrid algorithms.

A hierarchical hybrid deep reinforcement learning (DRL) framework was introduced by Cheng et al. [8] to enhance task scheduling in massive data centers. By integrating many tactics into a learning model that facilitates dynamic decision-making, the framework aims to maximize resource allocation while taking quality of service (QoS) into consideration. According to experimental results, this hybrid strategy improves overall performance in warehouse-scale data centers by reducing response time and increasing load-balancing efficiency when compared to traditional methods.

A system that combines DRL and graph neural networks (GNN) was created by Liu et al. [9] to optimize the scheduling of Directed Acyclic Graph (DAG) activities in dynamic vehicular cloud environments. In order to enable more intelligent and efficient scheduling in dynamic circumstances, the framework seeks to improve resource allocation efficiency while adjusting to temporal changes in the network. According to experimental data, in vehicular cloud situations, this hybrid technique outperforms standard methods in terms of scheduling performance, execution time, and stability.

In order to optimize resource allocation in cloud computing environments, Manavi et al. [10] proposed a hybrid strategy that combines neural networks with GA. This approach makes use of the neural network's capacity to anticipate and examine consumption trends in conjunction with the genetic algorithm's optimization capabilities. According to the results, the hybrid strategy outperforms traditional methods in terms of resource distribution efficiency, reaction time, and quality of service.

In order to improve solutions for combinatorial optimization issues, Ye et al. [11] developed a hybrid model that combines deep neural networks and ant colony systems (ACO). The model increases search efficiency and shortens the time it takes to find high-quality solutions by using deep learning to enhance the pheromone guidance mechanism in ant algorithms. According to experimental studies, DRLACO is a viable method for resource scheduling and performance optimization in computer systems since it performs better than conventional algorithms in a variety of optimization tasks.

In 2025, Wang et al [12]. created a framework that uses DRL to optimize resource scheduling between cloud and edge computing settings. The framework seeks to guarantee quality of service, decrease response times, and enhance the collaboration of distributed computing resources. According to experimental data, reinforcement learning supports optimal performance in hybrid computing systems by improving workload balancing and resource allocation efficiency when compared to traditional methods.

The application of machine learning approaches to enhance cloud computing resource scheduling and management was investigated by Zhang et al. [13]. The study concentrated on combining several machine learning algorithms to forecast demand and analyze consumption trends, which improved resource allocation, sped up response times, and improved the efficiency of using cloud infrastructure. In comparison to conventional techniques, the results showed that machine learning helps achieve a dynamic balance between performance and resource consumption.

In order to improve resource allocation and load balancing in cloud computing settings, Lilhore et al. [14] presented a hybrid strategy that combines Deep Reinforcement Learning (DRL) with Ant Colony Optimization (ACO) and Water Wave Optimization (WWO). By directing the ACO and WWO search process, DRL enhances solution exploration and lessens reliance on static parameters. According to experimental data, the suggested hybrid approach performed better than both conventional and non-hybrid algorithms in terms of energy consumption, resource utilization efficiency, and execution time reduction.

## Material and methods

### System design

In the cloud context, the intelligent scheduling function is defined by the environment, encompassing data centers, tasks, virtual machines (VMs), service-level agreements (SLAs), energy consumption, and the state, which captures unscheduled tasks, server resources, and current load [15, 16]. The action corresponds to selecting an appropriate server for each task, while the reward increases with improved service quality and reduced energy usage. To generate effective initial solutions, a hybrid GAACO approach—combining GA and ACO is applied prior to DRL training, enabling hybrid bootstrapping instead of random initialization. Subsequently, an Actor-Critic model refines these policies with real-time policy adaptation, dynamically improving resource allocation, reducing latency, preventing bottlenecks, and supporting energy-aware auto-scaling decisions [17-19].

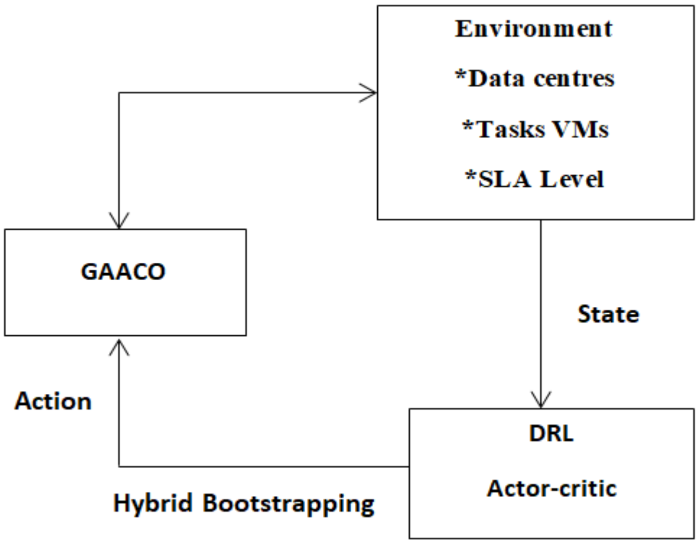


**Figure (1):** System simple design of a hybrid GAACO with DRL.

**The Hybrid Algorithm's Operation Mechanism: GA–ACO–DRL Based Allocation Approach**

This section outlines the systematic procedure for the proposed hybrid allocation algorithm, which integrates GA, ACO, and DRL to achieve efficient resource allocation in dynamic environments. The algorithm operates through the following sequential stages:

**Stage 1: Initial Population Generation (GA Initialization)**

The process begins with the generation of an initial population using GA. Each individual solution encodes a scheduling strategy that maps computational tasks to available resources, such as virtual machines (VMs) or edge nodes. The encoding structure ensures that task–resource assignments are well defined [20, 21].

Individual = {(T→ $T_1$) , (R→ $2R_2$),...}

The fitness of each candidate solution is evaluated using a multi-objective function that considers performance metrics, including makespan, energy consumption, and task failure rate.

**Stage 2: Evolutionary Refinement (Genetic Operators)**

The initial population is subsequently enhanced through evolutionary mechanisms within GA. Specifically, the following operators are applied [22]:

- *Selection*: Retaining the fittest individuals for the next geneation.
- *Crossover*: Combining genetic material from pairs of solutions to generate new offspring.
- *Mutation*: Introducing stochastic modifications to explore additional regions of the solution space.

The process iteratively improves population diversity and convergence, yielding an initial near-optimal generation of solutions [23].

**Stage 3: Local Optimization via ACO**

The best-performing solution obtained from GA is utilized as the starting point for the ACO stage. In this phase, each ant represents a candidate scheduling path. Resource assignments are made at each step based on pheromone trails and heuristic desirability, where heuristics are defined to minimize execution time and balance system load. This stage refines the solution by leveraging the collective search capability of the swarm-based method [24].

$$P_{ji} = \frac{[T_{ji}]^{\propto} * \ [\eta_{ji}]^{\propto\beta}}{\sum_k [T_{ki}]^{\alpha} * \ \eta_{ki}{}^{\beta}}$$

 

**Stage 4: Dynamic Decision-Making through DRL**

The refined solutions generated from GA and ACO are then introduced to a DRL agent. The environment is formalized as a Markov Decision Process (MDP), characterized by [25, 26]:

- *State*: System attributes, including resource utilization, task queue size, and energy consumption.
- *Action*: The allocation decision, i.e., assigning a task to a specific resource.
- *Reward*: A composite metric reflecting the quality of the allocation in terms of minimized execution time, reduced energy consumption, and lower failure rate.

The DRL model iteratively learns an optimal allocation policy, adjusting its parameters to maximize long-term system performance across diverse conditions.

$$Q(a_t, s_t) \rightarrow r_t + \gamma . \text{may} . \, {}^{max}_{a'} \, Q(s_{t+1,} \, a')$$

**Stage 5: Adaptive Feedback Loop**

Upon completing each allocation cycle, performance data are collected, encompassing average execution time, task failure probability, and power consumption. These feedback metrics are employed to dynamically fine-tune the parameters of GA (e.g., mutation probability, population size), ACO (e.g., pheromone evaporation rate, α and β factors), and DRL (e.g., learning rate, memory buffer size, and exploration parameter ε). This adaptive mechanism ensures sustained system performance improvement in response to environmental fluctuations and varying task characteristics.

## SIMULATION EXPERIMENTS

The resource allocation algorithm in this work was simulated using CloudSim-3.0.3, a Java-based library used to simulate various cloud environments. The simulation was run on a computer with the following configurations: a 2.53 GHz processor, 4 GB of memory, and a 512 GB hard drive. Eclipse was used as the environment to run the CloudSim library.

**Table 1**: shows the simulation parameter settings.

| Parameter Name | Default Value | Description |
|---|---|---|
| VM_COUNT | 5 | Number of virtual machines (VMs) |
| CLOUDLET_COUNT | 15 | Number of songs (Cloudlets) |
| Cloudlet Exec Time | 5.0 | Task execution time in seconds |
| pheromone[i] | 1.0 | Player value per VM (within ACO) |
| reward[i] | 0.5 | Bonus value per VM (within DRL) |
| load[i] | 0 | Number of allocated options per VM |
| Finish Time[i] | 0.0 | Last task completion time per VM |
| Pheromone[selectedVm] += 0.1 | 0.1 | Pheromone rate increment after allocation |
| Reward[selectedVm] += 0.05 | 0.05 | Bonus increment rate after allocation |
| Pheromone[i] *= 0.95 | 0.95 | Pheromone success rate for non-existent VMs |
| reward[i] *= 0.98 | 0.98 | Bonus rate for non-optional VMs |
| SLA Threshold = 1.2 * execTime | 6.0 | Service Level Agreement violation threshold (greater than 120% of execution time) |
| Random rand | — | Major key number generator for the main VM calculated |

## Results and discussion

### Fitness improvement

Figure (2) illustrates the comparison between the hybrid GAACO algorithm and the proposed integrated GAACO+DRL algorithm in cross-generational fitness improvement. In all experiments, GAACO+DRL consistently outperformed GAACO in terms of fitness value, exhibiting a significant and stable improvement. The differences between the experimental runs were less than 0.0044, compared to the larger variations observed with GAACO. This is because, while GAACO leverages a combination of ant colony optimization and genetic algorithms for broad exploration, it is susceptible to randomness and can get stuck in local optima. DRL, on the other hand, adds a dimension of continuous adaptive learning through interaction with the environment, guiding the search towards high-quality solutions and reducing performance fluctuations. Thus, the integration of these two approaches combines the exploration power of GAACO with the precision and stability of DRL, resulting in superior and more consistent solutions across multiple iterations.

 

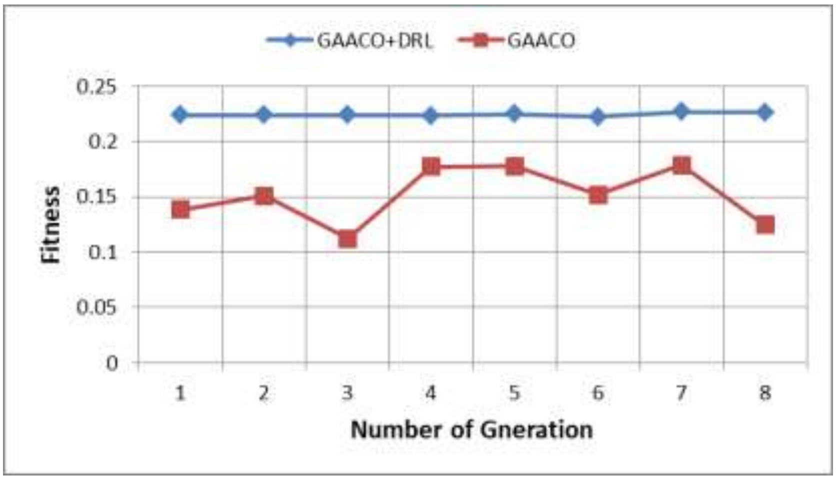


**Figure (2):** Comparison of cross-generational fitness improvement in the GAACO+DRL algorithm and the GAACO algorithm.

**Reducing the makespan**

The results of the eight experiments clearly demonstrated the superiority of the hybrid GAACO+DRL model compared to the traditional GAACO algorithm in reducing the makespan, as shown in Figure (3). In some cases, the improvement exceeded 30% (for example, in the first experiment, the value decreased from 23 to 15.4). This improvement is attributed to the ability of combining metaheuristic optimization algorithms (GAACO) with deep reinforcement learning techniques (DRL) to produce more efficient and stable scheduling solutions. Furthermore, the hybrid model's performance was consistent across all experiments, which enhances its reliability and applicability in dynamic, large-scale cloud computing environments that require highly efficient resource management and stable performance.

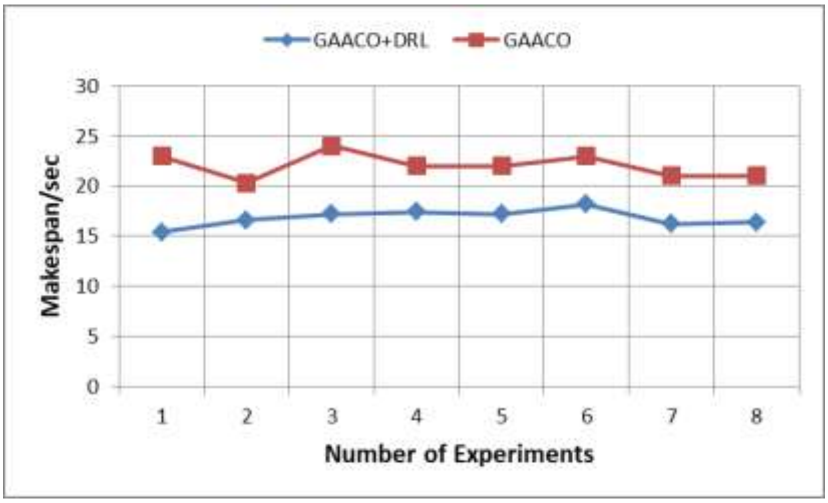


**Figure (3):** Makespan changes across experiments using the GAACO+DRL algorithm.

**Average waiting time**

Figure (4) illustrates the significant reduction in average waiting time, ranging from 20% to 35%, when using the hybrid CAACO+DRL algorithm compared to the GAACO algorithm. The integration of GAACO techniques with deep reinforcement learning enables more efficient task distribution across cloud computing resources. GAACO+DRL excels in its ability to adapt dynamically to changes in the operating environment through continuous agent learning and real-time scheduling decision optimization, whereas GAACO relies on relatively static improvements that do not adjust solutions quickly enough. This reduced waiting time reflects a decrease in task accumulation in VM queues, mitigating congestion and improving response time. Furthermore, using GAACO+DRL with initial GAACO solutions as a bootstrapping phase accelerates the development of efficient scheduling policies compared to training from scratch.

 

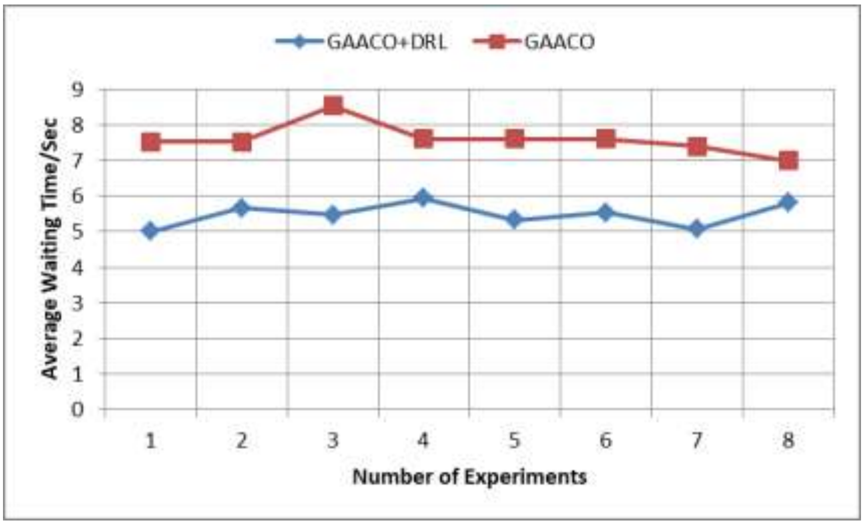


**Figure** (4): Average Waiting time comparison between GAACO+DRL algorithm and the GAACO algorithm.

**Average energy consumption**

The data in Figure (5) indicates that integrating the DRL algorithm with GAACO resulted in a significant reduction in average energy consumption compared to using GAACO alone across all eight scenarios. For example, in one case, consumption decreased from 1278.67 to 850.67, and in another from 1438 to 944, reflecting a substantial improvement in resource management efficiency. Overall, it can be observed that the GAACO+DRL values consistently remain lower than the GAACO values, demonstrating the hybrid model's ability to optimize task allocation and reduce energy consumption through adaptive decision-making during simulation, thus supporting the effectiveness of combining metaheuristic optimization (GAACO) and deep reinforcement learning (DRL) in cloud computing environments.

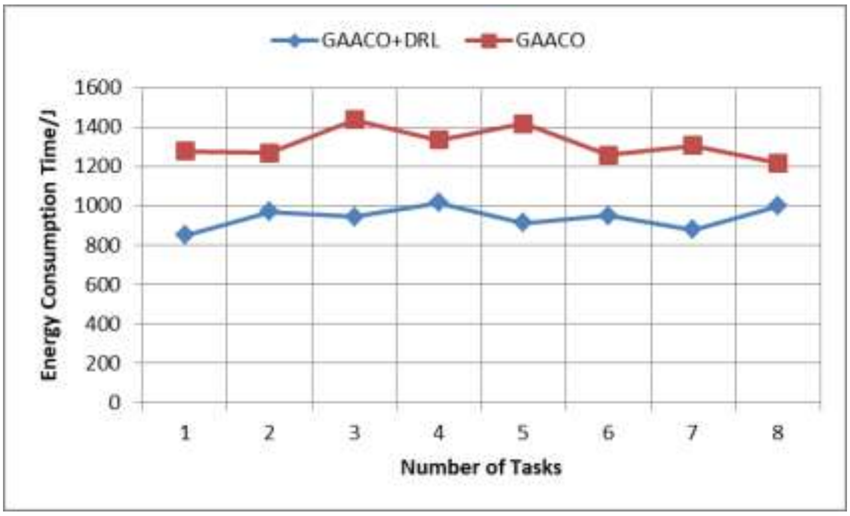


**Figure** (5): Comparison of average energy consumption between GAACO+DRL algorithm and the GAACO algorithm.

**Distribute the final load on virtual machines**

As shown in Figure (6), the GAACO algorithm resulted in an unbalanced distribution; some VMs, such as VM0, VM2, and VM4, received a high workload (4 tasks each), while VM1 received a low workload (only one task), indicating an imbalance in resource utilization. However, with the hybrid GAACO+DRL algorithm, the workload was distributed almost evenly across all VMs, with each receiving 3 tasks, reflecting a more balanced resource allocation and reducing the risk of overloading some VMs compared to others. This, in turn, contributes to improved overall system performance and reduced response time.

 

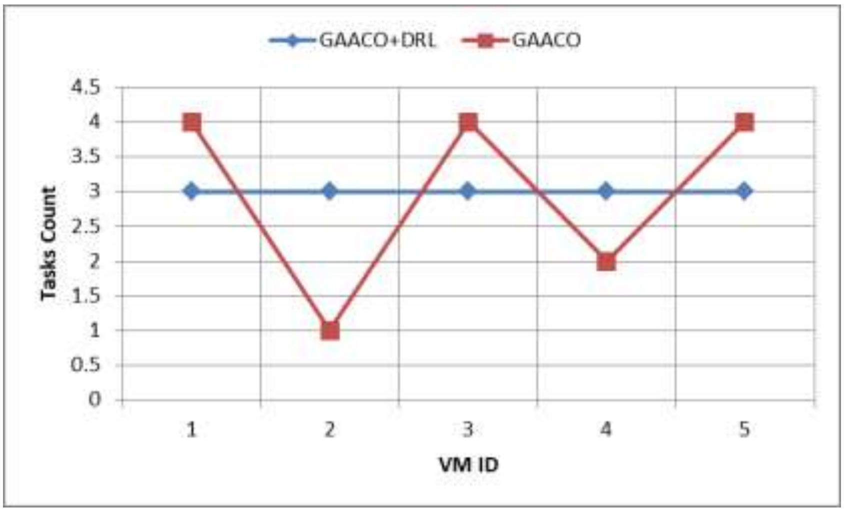


**Figure** (6): Distribute the final load on virtual machines (VMs) between GAACO+DRL algorithm and GAACO algorithm.

**Algorithm performance evaluation**

The GAACO+DRL algorithm achieved significantly higher performance than the GAACO algorithm in all cases, as shown in Figure (7). The productivity values for GAACO ranged from 0.63 to 0.75, while for GAACO+DRL they ranged from 0.82 to 0.97, reflecting a consistent improvement in resource utilization efficiency. This difference indicates that integrating Deep Reinforcement Learning (DRL) techniques with GAACO enhances the system's ability to adapt to the dynamic nature of cloud computing environments and mitigate bottlenecks, thus achieving higher throughput and more stable performance across various scenarios.

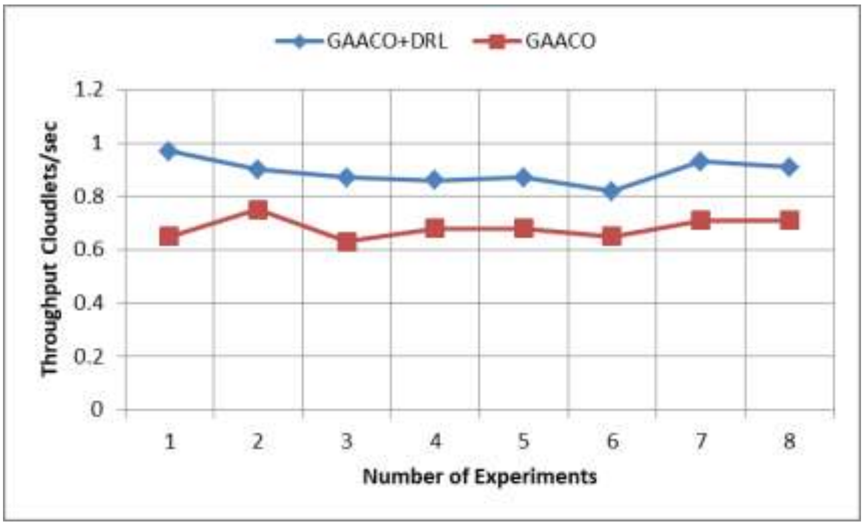


**Figure** (7): Throughput variability across experiments using the GAACO+DRL algorithm and GAACO algorithm.

**SLA Optimization**

Figure (8) shows that the violation rate for the GAACO + DRL algorithm is 0% in all experiments, while for the GAACO algorithm, it ranges from 0.87% to 46.67%, with a relatively high average in most cases (over 20%). This demonstrates that combining the improved ant colony optimization (GAACO) algorithm with deep reinforcement learning (DRL) techniques provides more efficient and responsive scheduling, virtually eliminating SLA violations compared to the hybrid genetic algorithm (GAACO). This can be attributed to DRL's ability to dynamically adapt to changes in workload and resource availability, unlike GAACO, which relies on a less flexible, static optimization approach.

 

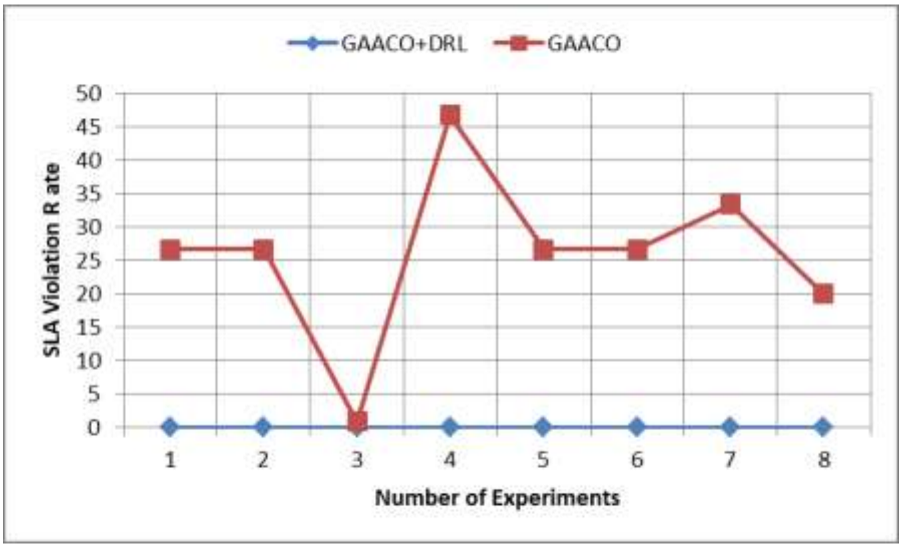


**Figure** (8): Comparison of SLA violation rate between CAACO + DRL algorithm and GAACO algorithm.

**Discussion**

The experimental results clearly demonstrate that the proposed GAACO+DRL framework significantly outperforms the conventional GAACO algorithm across multiple performance indicators, including makespan, waiting time, throughput, energy consumption, and SLA violation rates. This confirms the initial hypothesis that combining metaheuristic search with adaptive deep reinforcement learning can provide more efficient and stable scheduling solutions in dynamic cloud environments.

It is important to note, however, that the scope of the comparison in this study was limited to GAACO as the baseline. While this provides strong evidence of improvement over a well-established hybrid metaheuristic, future research could broaden the evaluation by benchmarking GAACO+DRL against other recent hybrid frameworks, such as GNN-DRL for DAG scheduling in vehicular clouds or WWO-DRL for enhanced load balancing. Incorporating such comparisons would help position GAACO+DRL more clearly within the broader research landscape and highlight its relative advantages and limitations.

Overall, the findings confirm that integrating DRL with GAACO is a promising direction, and they also open the door for future work exploring further hybridizations, real-world deployments, and cross-domain evaluations.

**Conclusion**

This work demonstrates that integrating Deep Reinforcement Learning with GAACO provides significant advantages in cloud resource scheduling compared to conventional metaheuristic approaches. The hybrid GAACO+DRL framework not only improves performance metrics such as makespan, waiting time, and throughput but also ensures balanced workload distribution, reduced energy consumption, and zero SLA violations. By combining the exploration capability of GAACO with the adaptive learning of DRL, the proposed model achieves both efficiency and stability under dynamic and large-scale cloud environments. These results suggest that the GAACO+DRL algorithm is a robust and scalable solution for future cloud computing systems, with potential applications extending to edge and hybrid computing infrastructures where adaptability and real-time decision-making are essential.

---

**Compliance with ethical standards**

*Disclosure of conflict of interest*

The authors declare that they have no conflict of interest.

**References**

[1] Mo X, Xu J. Energy-efficient federated edge learning with joint communication and computation design. *Journal of Communications and Information Networks.* 2021;6(2):110–124. DOI: **10.23919/JCIN.2021.9474620**

[2] Hussain H, Malik S U R, Hameed A, U. M, Shafiq M, Qureshi M B, Ali I, et al. A survey on resource allocation in high performance distributed computing systems. *Parallel Computing*. 2013;39(11):709–736. DOI: **10.1016/j.parco.2013.09.009**

 

[3] Bellendorf J, Mann Z Á. Classification of optimization problems in fog computing. *Future Generation Computer Systems*. 2020;107:158–176. DOI: **10.1016/j.future.2020.01.036**
[4] Zhou G, Tian W, Buyya R, Xue R, Song L. Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions. *Artificial Intelligence Review*. 2024;57(5). DOI: **10.1007/s10462-024-10717-9**
[5] Zhao Y, Liu J, Chen F, Zhang K, Wang X, Tian H. Performance analysis of cloud resource allocation scheme with virtual machine inter-group asynchronous failure. *Journal of King Saud University – Computer and Information Sciences*. 2024;36(7):102155. DOI: **10.1016/j.jksuci.2024.102155**
[6] Manzoor M F, Khan M A, Kim S, Waseem M, Nam Y, Qureshi A, et al. Resource allocation techniques in cloud computing: a review and future directions. *Elektronika ir Elektrotechnika*. 2020;26(6):40–51. DOI: **10.5755/j01.eie.26.6.26467**
[7] Nguyen Q-H, Pham D N, Nguyen H N, Thoai N. A genetic algorithm for power-aware virtual machine allocation in private cloud. *arXiv preprint*. 2013;arXiv:1302.4519. DOI: **10.1007/978-3-642-36818-9_19**.
[8] Cheng M, Li J, Bogdan P, Nazarian S. H2O-Cloud: A resource and quality of service-aware task scheduling framework for warehouse-scale data centers – a hierarchical hybrid deep reinforcement learning based approach. *arXiv preprint*. 2020;arXiv:1912.10808. DOI**: 10.1109/TCAD.2019.2930575**
[9] Liu Z, Huang L, Gao Z, Luo M, Hosseinalipour S, Dai H. GA-DRL: Graph Neural Network-Augmented Deep Reinforcement Learning for DAG Task Scheduling Over Dynamic Vehicular Clouds. *IEEE Transactions on Network and Service Management*. 2024;21(4):4226–4242. DOI: **10.1109/TNSM.2024.3380793**
[10] Manavi M, Zhang Y, Chen G. Resource allocation in cloud computing using genetic algorithm and neural network. In: *Proceedings of IEEE SmartCloud 2023*; 2023 Sep 16. IEEE. DOI**: 10.1109/SmartCloud58862.2023.00013.**
[11] H. Ye, J. Wang, Z. Cao, H. Liang, and Y. Li, *DeepACO: Neural-Enhanced Ant Systems for Combinatorial Optimization: Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, December 10–16, 2023, New Orleans, LA, USA. Red Hook, NY: Curran Associates, 2024.
[12] Wang Y, Yang X. Research on edge computing and cloud collaborative resource scheduling optimization based on deep reinforcement learning. *arXiv preprint*. 2025.
[13] Zhang Y, Liu B, Gong Y, Huang J, Xu J, Wan W. Application of machine learning optimization in cloud computing resource scheduling and management. In: *Proceedings of the 5th International Conference on Computer Information and Big Data Applications (CIBDA 2024)*; 2024 Apr 26. Association for Computing.
[14] Lilhore U K, Simaiya S, Alroobaea R, Alsafyani M, Alhazmi A, Khan M D M. Hybrid DRL-Enhanced ACO-WWO for efficient resource allocation and load-balancing in cloud computing. *International Journal of Computational Intelligence Systems*. 2025;18(1):Article 148. DOI: **10.1007/s44196-024-00541-5**
[15] Ilankumaran A, Narayanan S J. An energy-aware QoS load balance scheduling using hybrid GAACO algorithm for cloud. *Cybernetics and Information Technologies*. 2023;23(1):161–177. DOI: **10.1515/cait-2023-0009**
[16] Adhikari M, Amgoth T, Srirama S N. A survey on scheduling strategies for workflows in cloud environment and emerging trends. *ACM Computing Surveys*. 2019;52(4):1–36. DOI: **10.1145/3325097**
[17] Wang Z, Chen S, Bai L, Gao J, Tao J, Bond R R, Mulvenna M D. Reinforcement learning based task scheduling for environmentally sustainable federated cloud computing. *Journal of Cloud Computing*. 2023;12(1):174. DOI: **10.1186/s13677-023-00553-0**
[18] Cheng M, Li J, Nazarian S. DRL-Cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers. In: *Proceedings of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC 2018)*; 2018 Jan 22–25; Jeju, Korea (South). IEEE. p. 129–134. DOI: **10.1109/ASPDAC.2018.8297294**
[19] Nalini J, Khilar P M. Reinforced ant colony optimization for fault tolerant task allocation in cloud environments. *Wireless Personal Communications*. 2021;121(4):2441–2459. DOI: **10.1007/s11277-021-08583-3**
[20] Sawant S. A genetic algorithm scheduling approach for virtual machine resources in a cloud computing environment. MS.D. dissertation, Dept. Comp. Sc., San Jose State Univ., San Jose, CA, 2011. **https://doi.org/10.31979/etd.et9e-73fz**
[21] Peng Z, Wang Y, Zhao L, Zhang W, Liu J. Genetic algorithm-based task scheduling in cloud computing using MapReduce framework. *Mathematical Problems in Engineering*. 2022;2022(1):4290382. DOI: **10.1155/2022/4290382**
[22] Pirozmand P, Moghadam R A, Ghafour M Y, Yaghoubi F. Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. *Neural Computing and Applications*. 2021;33(19):13075–13088. DOI: **10.1007/s00521-021-06002-w**

 

[23] Girgis M R, Mahmoud T M, Azzam H M. GA-based QoS-aware workflow scheduling of deadline tasks in grid computing. *Knowledge and Information Systems*. 2024;66(5):2859–2884. DOI: **10.1007/s10115-023-02048-5**
[24] Abdulghani A. Hybrid task scheduling algorithm for makespan optimisation in cloud computing: a performance evaluation. *Journal of Artificial Intelligence*. 2024;6:241. DOI: **10.32604/jai.2024.046209**
[25] Ullah I, Li J, Imran M, Shoaib M, Zhang L, Khan Z A. Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. *Journal of Cloud Computing*. 2023;12(1):112. DOI: **10.1186/s13677-023-00515-4**
[26] Wang Z, Chen S, Bai L, Gao J, Tao J, Bond R R, Mulvenna M D. Reinforcement learning based task scheduling for environmentally sustainable federated cloud computing. *Journal of Cloud Computing*. 2023;12(1):174. DOI: **10.1186/s13677-023-00553-0**



 