Library Progress International *Print version ISSN 0970 1052*
Vol.44 No. 3, Jul-Dec 2024: P. 15815-15822 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

HDRLGA: Hybrid deep reinforcement learning and genetic algorithm task scheduling approach in cloud computing.

[1]Sushil Kumar Jain*, [2] Dr. Jyoti

[1]Research Scholar, Department of Computer Science & Applications, Baba Mastnath University, Rohtak. Email id: jainsushil1507@gmail.com
[2]Assistant Professor, Department of Computer Science & Applications, Baba Mastnath University, Rohtak. Email id: jyotivij@bmu.ac.in

How to cite this article: Sushil Kumar Jain, Dr. Jyoti (2024) HDRLGA: Hybrid deep reinforcement learning and genetic algorithm task scheduling approach in cloud computing.. *Library Progress International,* 44(3), 15815-15822

ABSTRACT
Task scheduling in cloud computing is a critical challenge that directly impacts the performance and cost-effectiveness of cloud environments. This paper presents a hybrid task scheduling algorithm combining Deep Reinforcement Learning (DRL) and Genetic Algorithm (GA) to achieve multiple optimization objectives including minimizing makespan, reducing overall cost, average turnaround and degree of imbalance. HDRLGA method leverages DRL to predict optimal task-to-resource mappings dynamically, while GA fine-tunes the schedule by exploring a wide range of possible configurations. Experimental results demonstrate that the hybrid approach outperforms traditional scheduling methods shortest job first, largest job first, first come first serve, max-min and min-min in terms of makespan and degree of imbalance. The algorithm's effectiveness is validated through simulations using various cloud scenarios, highlighting its potential for real-world cloud environments.
KEYWORDS: Hybrid Task Scheduling, Deep Reinforcement Learning, Genetic Algorithm, Cloud Computing, Makespan Minimization, Degree of imbalance reduction

1. Introduction

Cloud computing combines distributed and parallel computing to enable the sharing of resources like software and hardware on a "pay-as-you-go" basis. Users don't need to buy platforms or software; they simply need an internet connection to access these resources and pay based on their usage [3,11]. Cloud computing is a model that allows for ubiquitous, convenient, and on-demand access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services. These resources can be rapidly provisioned and released with minimal management effort or service provider interaction. Scheduling plays a critical role in this model by ensuring the efficient allocation and utilization of these resources. Effective scheduling enables optimal performance, reduces latency, and balances workloads, making sure that resources are used efficiently while maintaining service quality [8].
Computing has revolutionized the way computing resources are provisioned, offering elastic, scalable, and cost-effective solutions to meet the demands of modern applications. By utilizing virtualization technologies, cloud service providers can allocate resources such as CPU, memory, and storage to users on an on-demand basis. However, managing these resources efficiently is a significant challenge, especially as the number of users and tasks increases in large-scale data centers. Task scheduling is at the core of this challenge, as it determines how tasks are assigned to Virtual Machines (VMs) to optimize overall performance. An effective scheduling algorithm can drastically improve system performance by minimizing makespan (the total time to complete all tasks), reducing operational costs, improving resource utilization, and optimizing the average turnaround time of tasks. Traditional scheduling methods like First Come First Serve (FCFS), Round Robin (RR), and Min-Min have been widely used but often fail to meet the multi-objective optimization needs of modern cloud systems. These conventional techniques are limited in their ability to adapt to dynamic environments, leading to inefficient resource allocation, higher costs, and suboptimal performance in terms of makespan and turnaround time.
To address these limitations, there has been growing interest in hybrid scheduling techniques that combine machine learning and evolutionary algorithms. In particular, Deep Reinforcement Learning (DRL) has demonstrated its potential in learning from dynamic environments by enabling agents to take actions that maximize cumulative rewards. DRL's ability

to learn and adapt to changing workloads makes it a strong candidate for task scheduling in cloud computing. However, DRL alone may not always find the global optimum due to the complexity of the search space. On the other hand, Genetic Algorithms (GA) are effective in exploring a wide search space by applying evolutionary principles such as mutation, crossover, and selection to find high-quality solutions. By combining DRL and GA, a hybrid scheduling algorithm can benefit from the strengths of both methods—adaptability from DRL and optimization from GA. The remainder of this paper is structured as follows: Section 2 provides the problem definition and objective of paper. Section 3 reviews the related work on cloud task scheduling algorithms. Section 4 presents the design and methodology of the proposed hybrid scheduling algorithm. Section 5 provides experimental results and performance evaluation. Finally, Section 6 concludes the paper and future scope of this research.

2. Problem Definition and Formulation of objective Function

In a cloud environment, numerous users rely on cloud resources to perform their tasks, making scheduling a crucial factor in optimizing resource utilization, response time, latency, and load balancing. The task scheduling problem can be defined as follows:

Input:

- Let **V** = $\{v_1, v_2, v_3, ..., v_m\}$ represent a set of virtual machines (VMs), where **m** is the total number of VMs in the cloud network. Each VM has distinct resources such as CPU, RAM, and bandwidth, with varying usage costs and computing capabilities. $\mathbf{v_i}$ represents the i^{th} VM.
- Let **T** = $\{t_1, t_2, t_3, ..., t_n\}$ denote a set of tasks, where **n** is the number of independent tasks assigned to the VMs. $\mathbf{t_j}$ represents the j^{th} task, and each task has specific requirements such as the number of instructions, memory, and CPU.

Output:

- The goal is to optimize the scheduling process by mapping **n** tasks to **m** VMs, aiming to minimize the makespan, reduce costs, maximize resource utilization, and improve load balancing across resources.

Hybrid task scheduling algorithm that integrates Deep Reinforcement Learning (DRL) with Genetic Algorithms (GA) to address the shortcomings of existing scheduling methods. The primary objectives of the proposed algorithm are to:

1. Minimize makespan: Reduce the total time required to complete all tasks in the cloud system with different size of cloudlets and virtual machines.
2. Decrease Degree of imbalance: Ensure Degree of imbalance must be reduced. It improves resource utilization in cloud computing environment.

3. Related Work

Cloud computing needs task scheduling to improve the utilization of resources.

Thanka, M.R., Maheswari, P.U. and Edwin, E.B., [2] proposed a hybrid of PSO and ABC task scheduling algorithms. The proposed ABPS algorithms reduces makespan and cost in comparison of ABC and PSO algorithm. Hybrid algorithm also increased resource utilization. The hybrid algorithm does better in comparison of both algorithms alone.

Zhang, J., Cheng, L., Liu, C., Zhao, Z. and Mao, Y. [1] presented a GA-DQN combines Genetic Algorithm and Deep Reinforcement Learning a to reduce both execution cost and response time. Proposed method outperforms in comparison of round robin, earliest and random methods.

Singh, S. and Chana, I., studied 110 research articles regarding resource scheduling in cloud computing to identify the effective ness if existing research work in term of makespan, cost, turnaround time and degree of imbalance. Allocation of resources on the nature of work improve resource utilization. Work nature may be heterogeneous or homogenous. Proper allocation of resource to the task reduce cost and make the response time less.[4]

Genetic Algorithms (GAs) are a powerful class of metaheuristic techniques inspired by the principles of natural selection. They excel at navigating large solution spaces to uncover optimal solutions effectively. Through key processes such as selection, crossover, and mutation, GAs can generate diverse populations of potential solutions, ultimately leading to better options for optimal outcomes. While they demonstrate remarkable robustness, it is important to note that GAs can be computationally intensive and necessitate careful parameter tuning to maximize their performance [5] [10]. This careful calibration is essential for leveraging their full potential in finding superior solutions.

Modified shortest job first algorithm optimized the resource when compared with Traditional SJF, Round Robin and First-

Come-First-Serve task scheduling in cloud computing [6]. Makespan, task completion time and resource utilization are considered for test the performance of proposed algorithm. Modified shortest job first took less makespan and task completion time and resource utilization improved.
Energy efficiency of different algorithms using
deep reinforcement learning are studied which include state space, action space, reward functions and base lines. Different data sets are also considered google cluster trace and Alibaba trace data and auto generated simulation data.[7].
MOABCQ method is a multi-objective task scheduling optimization that integrates the Artificial Bee Colony (ABC) algorithm with Q-learning, a reinforcement learning technique that enhances the efficiency and speed of the ABC algorithm to increase the resource utilization [9]. Three datasets i.e. randomly generated, Google Cloud Job and Synthetic workload.

4. Design and Methodology

The proposed hybrid algorithm works by initially utilizing DRL to model the cloud environment and learn optimal scheduling policies through continuous interaction and feedback.

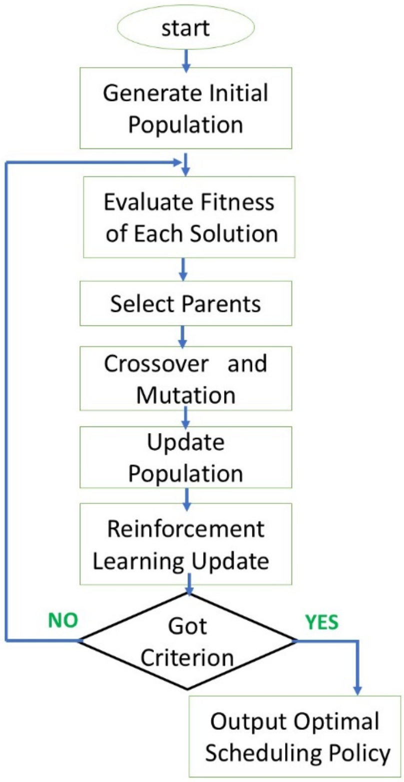


Figure 1: DRL and GA Model

The DRL agent selects actions (task-VM assignments) based on the current state of the system, with the goal of maximizing a reward function that balances multiple objectives. Once the DRL-based scheduling strategy is learned, a Genetic Algorithm is applied to further refine the task allocation by searching for better task-resource mappings. This two-phase process allows the algorithm to balance exploration (discovering new solutions) and exploitation (refining existing solutions) effectively.
To validate the effectiveness of the hybrid approach, extensive simulations are conducted in a cloud computing environment using real-world workloads. The results demonstrate that the hybrid algorithm significantly outperforms traditional scheduling techniques in terms of makespan, cost, resource utilization, and average turnaround time. By leveraging the combined strengths of DRL and GA, the algorithm dynamically adapts to changing workloads, providing a robust and scalable solution for multi-objective task scheduling in cloud computing.

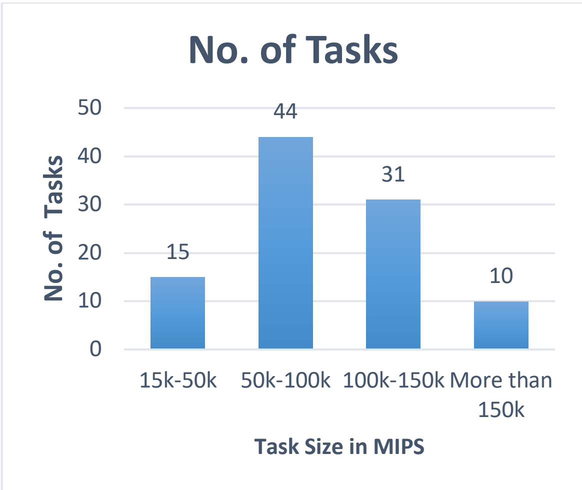
5. Simulation and Data Sets

This section describes an experiment conducted to assess the performance of the proposed method (DRLGA) in comparison to other task scheduling approaches in a heterogeneous cloud computing environment. A task scheduling simulation is developed using the Python libraries PyTorch, TensorFlow, numpy, keras and random. The experiment was performed on

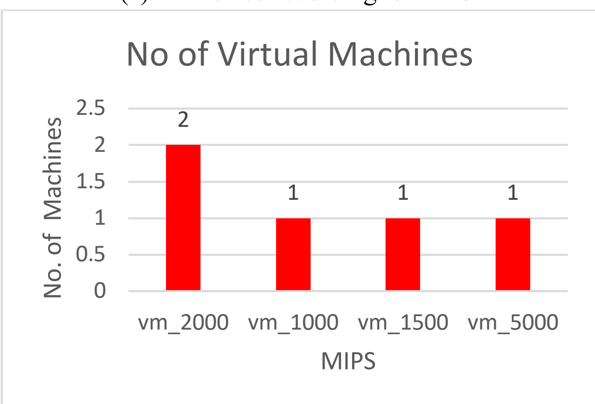
a system with 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz and 8GB RAM.
A virtual environment was simulated to demonstrate the efficiency of the proposed method in terms of makespan, average turnaround time, cost and degree of imbalance in cloud computing. In this, two data sets are used for evaluation of the effectiveness of proposed algorithm with other algorithms. First is Random data set that is generated randomly with tasks of 10k to 50k. The randomly generated dataset contains a total of 100 tasks with task size in mega instructions per second. Other data set is Google Cloud Jobs (GOCJ) dataset.

Table 1: Experimental Setup

No. of Tasks	100/100
Source	GoCJ/Randomly generated
No of VMs	5
Tasks Length	15K-900K
System	Core (TM) i3-1115G4 @ 3.00GHz, 8 GB RAM
Simulation Language	Python- Libraries as: TensorFlow, PyTorch, Numpy and pandas.



(a) Data sizes according to MIPS.



b) No. of VMS according to MIPS
Figure 2. Task Sizes and VM sizes (MIPS)

PyTorch and TensorFlow widely-used deep learning frameworks, can be highly effective in implementing advanced task scheduling algorithms for cloud computing environment. PyTorch and TensorFlow models can learn optimal load-balancing strategies to distribute tasks evenly across cloud resources, ensuring efficient utilization of resources and reduce degree of imbalance [7].

6. Experimental Results

This section offers a comprehensive experimental evaluation of the proposed scheduling approach, utilizing a benchmark dataset for performance comparison. The evaluation focuses on makespan, average turnaround time and degree of imbalance. Through these metrics, we aim to assess how well the proposed method performs in optimizing resource usage and minimizing execution time. The experiment is designed to rigorously measure the effectiveness of the approach in two scenarios, providing a thorough comparison with existing techniques to highlight its potential advantages. Results of various scheduling algorithms on the 100 tasks ranges from 15k to 900 k. Virtual machine ranges from 1000 MIPs to 5000MIPS

Algorithm	Make Span Time
FCFS	1268
Shortest Job First	1446
Largest Job First	1221
Min-Min	1448
Max-Min	1221
HDRLGA	1186

Table 2: Makespan Comparisons (GOCJ)

The performance analysis of different scheduling algorithms, measured in terms of makespan, reveals notable percentage differences. The **First Come First Serve (FCFS)** algorithm achieved a makespan of **1268**, which is **6.9%** higher than the lowest makespan.

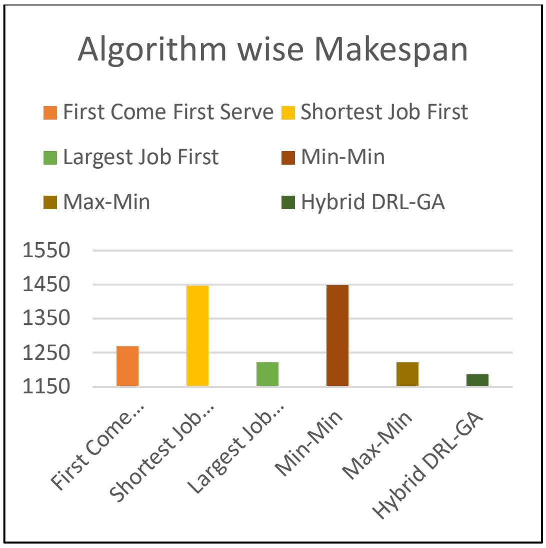


Figure 3: Makespan comparison using GOCJ.

The Shortest Job First (SJF) algorithm resulted in a makespan of 1446, making it 21.9% higher than the lowest value. The Min-Min algorithm showed a similar makespan of 1448, which is also 22.1% higher. In contrast, both the Largest Job First (LJF) and Max-Min algorithms had a makespan of 1221, reflecting a 2.9% increase over the lowest makespan. Finally, the Hybrid DRL-GA approach, outperformed all other algorithms.

2. Degree of imbalance: It is the measurement that quantifies how unevenly tasks are distributed across available resources, such as processors, virtual machines, or servers.

Algorithm	Degree of Imbalance
First Come First Serve	255
Shortest Job First	353
Largest Job First	518
Min-Min	353
Max-Min	518
HDRGA	315

Table 3: Degree of imbalance (GOCJ)

The moderate level of imbalance is found in the hybrid approach reflects its ability to adapt and optimize resource usage, making it a robust solution for dynamic and complex scheduling environments. By using more data for training, degree of imbalance can be reduced.

Algorithm	Make Span Time	Degree of imbalance
First Come First Serve	282	42
Shortest Job First	284	55
Largest Job First	275	54
Min-Min	278	55
Max-Min	275	54
HDRLGA	252	40

Table 4 Makespan (Random Data Set)

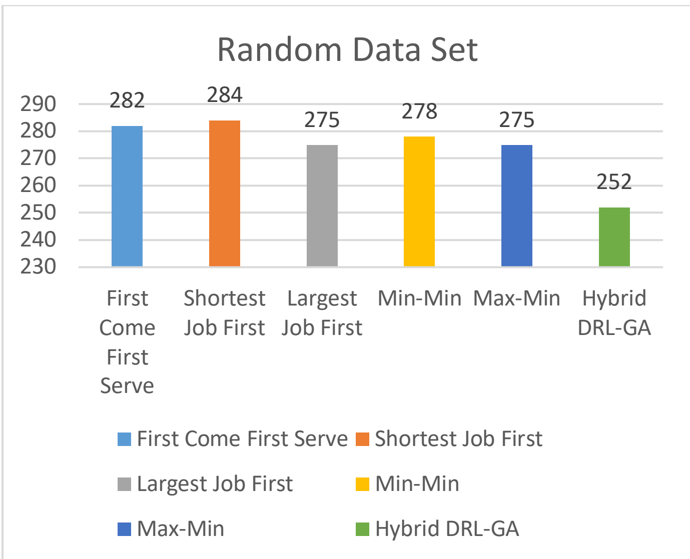


Figure 4: Makespan (Random Data Set)

In case of random data set generated manually of 100 tasks range from 10k to 50k. Make span of proposed hybrid of deep reinforcement learning with genetic algorithm is less in comparison of shortest job first, largest job first, first cum first serve, max-min and min-min traditional scheduling algorithms. Max-min and largest job has same makespan and shortest job first has largest makespan in this comparison. Degree of imbalance calculated in shortest job first, largest job first, first cum first serve, min-min, max-min and proposed algorithm. Hybrid algorithm has less degree of imbalance as shown in figure 5.

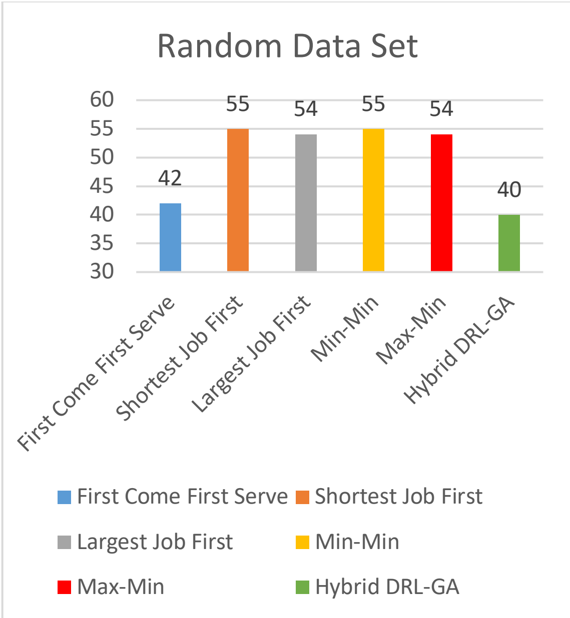


Figure 5: Degree of imbalance (Random Data)

7. Conclusion

Hybrid of Deep reinforcement learning and Genetic algorithm provides the better result in comparison of First come first serve, Max-min and shortest job first task, max-min and min-min scheduling algorithms. By using this hybrid, make span and degree of imbalance reduced in the context of gocj data set and randomly generated data set, In future, proposed hybrid HDRLGA algorithm can be tested on more real datasets. the makespan and average turnaround time reduces.

References

[1] Zhang, J., Cheng, L., Liu, C., Zhao, Z. and Mao, Y., 2023. Cost-aware scheduling systems for real-time workflows in cloud: An approach based on Genetic Algorithm and Deep Reinforcement Learning. *Expert Systems with Applications*, *234*, p.120972.

[2] Thanka, M.R., Maheswari, P.U. and Edwin, E.B., 2019. A hybrid algorithm for efficient task scheduling in cloud computing environment. *International Journal of Reasoning-based Intelligent Systems*, *11*(2), pp.134-140.

[3] Yang, S., Pan, L., Wang, Q., Liu, S. and Zhang, S., 2018, July. Subscription or pay-as-you-go: Optimally purchasing IaaS instances in public clouds. In *2018 IEEE International Conference on Web Services (ICWS)* (pp. 219-226). IEEE.

[4] Singh, S. and Chana, I., 2016. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing*, *14*, pp.217-264.

[5] Goudarzi, M., Zamani, M., & Pedram, M. (2012). A multi-objective cloud resource provisioning framework for energy efficiency and SLA compliance. Proceedings of the International Conference on Cloud Computing and Services Science (CLOSER), 1-8.

[6] Pachipala, Y., Sureddy, K.S., Kaitepalli, A.S., Pagadala, N., Nalabothu, S.S. and Iniganti, M., 2024. Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm. *Procedia Computer Science*, *233*, pp.604-613.

[7] Hou, H., Jawaddi, S.N.A. and Ismail, A., 2024. Energy efficient task scheduling based on deep reinforcement learning

in cloud environment: A specialized review. *Future Generation Computer Systems*, *151*, pp.214-231.

[8] Mell, P., 2011. The NIST Definition of Cloud Computing. NIST Special Publication, pp.800-145.

[9] Kruekaew, B. and Kimpan, W., 2022. Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning. *IEEE Access*, *10*, pp.17803-17818.

[10] Zhu, K., Song, H., Liu, L., Gao, J. and Cheng, G., 2011, December. Hybrid genetic algorithm for cloud computing applications. In *2011 IEEE Asia-Pacific Services Computing Conference* (pp. 182-187). IEEE.

[11] T. S. George and V. P. S. Kumar, ''Multicloud computing for on-demand resource provisioning using clustering,'' in Proc. 3rd Int. Conf. Sustain. Energy Intell. Syst. (SEISCON), 2012, pp. 435–440.

[12] S. Yang, L. Pan, Q. Wang, S. Liu, and S. Zhang, ``Subscription or pay-as- you -go: Optimally purchasing IaaS instances in public clouds," in Proc. IEEE Int. Conf. Web Services (ICWS), Jul. 2018, pp. 219-226