INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

OPEN ACCESS

Received: 07/03/2025

Accepted: 26/04/2025

Published: 11/05/2025

Citation: Sharma KJ, Patel NM (2025) Optimizing Cloud Resource Management with Advanced Scheduling Algorithms and Reinforcement Learning. Indian Journal of Science and Technology 18(17): 1356-1364. https://doi.org/10.17485/IJST/v18i17.434

* **Corresponding authors**.

kjsharma@bvmengineering.ac.in

nmpatel@bvmengineering.ac.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Optimizing Cloud Resource Management with Advanced Scheduling Algorithms and Reinforcement Learning

Kirtikumar J. Sharma[1]*, Narendra M. Patel[2]*

1 Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India
2 Computer Engineering Department, BVM Engineering College, V V Nagar-388120, Gujarat, India

Abstract

Objectives: Effective scheduling of cloud resources plays a critical role in maintaining performance and minimizing operational delays, especially in large-scale environments. This study aims to improve task scheduling efficiency by exploring intelligent algorithms that adapt well under dynamic workloads. **Methods:** Using the CloudSimPy simulation framework, we implemented and evaluated four scheduling algorithms—First-Fit, Random, Weighted-Based, and Reinforcement Learning (RL). The performance was benchmarked using real workload data from the Alibaba Cluster Trace 2017. We then compared our results with existing scheduling outcomes reported by Fengcun Li, particularly his use of the Tetris and First-Fit algorithms. **Findings:** The results highlight the significant advantage of Weighted-Based and Reinforcement Learning strategies. RL achieved the highest gains with up to 50% improvement in job completion time and 44% reduction in average slowdown. Weighted-Based scheduling also delivered strong performance while maintaining lower computational overhead. Compared to Fengcun Li's reported performance of Tetris and First-Fit, both RL and Weighted-Based methods showed more consistent and adaptive scheduling under real cloud conditions. **Novelty:** By replicating a real workload simulation environment in CloudSimPy and directly comparing our results with previously reported methods, this study offers a practical and grounded perspective on next-generation cloud scheduling. It demonstrates the clear benefits of incorporating AI-driven and weighted decision-making strategies in real-world job scheduling.

Keywords: Job Scheduling; Makespan Optimization; Reinforcement Learning; Scheduling Algorithms; Cloud Computing

1 Introduction

Cloud computing has emerged as a framework for supplying scalable, flexible, and cost-effective computational services in all industries. As request increases, effective resource management becomes critical to ensure reliable performance, decrease operational costs, and meet service-level requiments. One of the key challenges in this domain lies in the dynamic workloads, where traditional or heuristic-based scheduling methods frequently fail to adapt, leading to either over utilization or under utilization of resources[1].

Recent progresses in resource scheduling techniques have presented more intelligent methods, with metaheuristics and machine learning, to address these issues. Despite this progress, issues like high computational overhead, lack of adaptableness, and trouble in multi-objective optimization continue[2]. Reinforcement Learning (RL), in particular, offers a hopeful adaptive method by enabling systems to learn optimal scheduling strategies from past experiences and continuously refine results based on real time feedback.

Numerous studies have discovered RL for job scheduling and resource optimization in cloud environments. For instance, Deep Q-Network-based models have revealed significant enhancements in makespan and resource utilization[3]. Though, these methods often struggle with convergence time and solidity under unpredictable workloads. Moreover, comparative studies between RL, heuristic, and hybrid methods remain limited, creating uncertainty about the best suitable algorithm for precise workload conditions.

1.1 Research Gaps

While cloud scheduling has seen significant advancements with the use of heuristic, rule-based, and even learning-based approaches, there remains a noticeable gap in how these methods are evaluated and compared especially under real-world, large-scale workloads. Most studies tend to focus on proposing new algorithms in isolation, without thoroughly benchmarking them against practical alternatives like Weighted-Based or Reinforcement Learning under realistic conditions.

In particular, although Tetris and First-Fit scheduling algorithms have been implemented in existing work such as Fengcun Li's[4], they lack the adaptability required to manage unpredictable or highly dynamic job loads effectively. Moreover, few studies have directly compared these traditional methods with adaptive models like Reinforcement Learning or priority-driven strategies like Weighted-Based scheduling using trace-driven datasets such as Alibaba Cluster Trace 2017.

This creates a clear opportunity to bridge the gap by evaluating and comparing both traditional and intelligent scheduling methods on the same footing, using real workload traces and practical performance metrics like makespan, completion time, slowdown, and computational cost.

2 Background and Review of Scheduling Strategies

By aligning advanced scheduling methods with reinforcement learning strategies, this study contributes toward developing robust, scalable, and adaptive scheduling solutions suitable for evolving cloud infrastructures. The fast development of cloud computing has revolutionized the approach we access and consume computational resources. As the need for cloud services continues to rise, it has turn into gradually more critical to optimize resource allotment and administration to present cost-effective and professional cloud computing environments.

One of the key challenges in cloud resource managing is the dynamic and volatile nature of workloads. Conventional resource allotment methods frequently resist to handle with the variable resource needs of workflows, lead to either over-provisioning or under-provisioning of resources[5].To deal with this problem, researchers have projected a variety of solutions, including way of thinking, feedback-based methods, heuristics, knowledge models, and prediction methods[5].

In this study, we look at the possible of superior scheduling methods and reinforcement learning to optimize cloud resource organization. Scheduling methods play a critical function in efficiently managing resources to tasks, ensure competent use and meeting user needs[6],[4]. However, the difficulty of cloud environments and the diverse requirements of users pose significant challenges for conventional scheduling methods[7].

By using include reinforcement learning, we plan to include adaptive and self-optimizing scheduling methods that can dynamically alter resource allocation based on real-time estimation and varying workloads[5]. This technique permits the system to study from its experiences, improve resource consumption, and increase the overall capability of the cloud communications[8].

2.1 Cloud Resource Management Challenges

Competent resource administration is an important characteristic of cloud computing, intended at balancing performance and charge. A major challenge lies in control dynamic and random workloads that differ in intensity and difficulty. Conventional resource allotment methods frequently direct to over-provisioning, rising costs, or under-provisioning, which degrades performance[1]. Methods like reasoning, opinion mechanisms, and heuristics have been projected to tackle these issues but frequently be unsuccessful to extent with the increasing demands of present cloud infrastructures[6].

2.2 Scheduling Algorithms in Cloud Computing

Scheduling techniques are essential to competent resource allocation. Existing methods can be considered into static and dynamic scheduling techniques:

Static Scheduling: Techniques like Round Robin are simple but be unsuccessful to adjust to dynamic workloads[9]. Dynamic Scheduling: Techniques like Min-Min and Genetic methods think task heterogeneity and resource accessibility, contributing better performance. However, these methods resist with the real-time variation required in modern, large-scale cloud environments[10]. More superior methods like Ant Colony Optimization and Particle Swarm Optimization have been studied in details, but their computational difficulty can degrade the performance[8].

2.3 Reinforcement Learning in Cloud Resource Management

Reinforcement Learning (RL) introduces an alternate learning concept, enabling dynamic resource administration by leveraging precedent knowledge to get better future decisions[8]. RL based methods have exposed guarantee in management multi-objective optimization jobs, such as decreasing makespan, recovering resource consumption, and adhering to Service Level Agreements (SLAs). A few prominent RL applications comprise Resource Allocation RL agents dynamically adjust resources based on workload patterns, outperforming static allocation models[5], Task Scheduling Deep RL frameworks have demonstrated noteworthy improvements in reducing task completion time and balancing resource usage[7]. However, RL approaches often face challenges like convergence speed, scalability, and stability in highly dynamic cloud environments[11].

2.4 Hybrid Approaches

Hybrid models that merge conventional scheduling methods with RL have emerged as a capable explanation for overcoming the boundaries of standalone techniques. For example Shan et al. (2023) proposed a Kubernetes-based adaptive resource allocation framework that leverages feedback from workload metrics to adjust allocation dynamicall[5]. Tani & Amrani (2018) presented a smarter Round Robin approach integrated with predictive mthods to enhance scheduling efficiency in cloud environments[8], [9]. Such mixture approaches offer the double benefits of algorithmic stability and adaptive learning but need careful tuning and computational overhead cloud management.

2.5 Performance Metrics for Resource Management

Evaluating resource administration methods includes analyzing several metrics:

- Makespan: Total time taken to complete a set of tasks, used to measure scheduling efficiency[11].
- Resource Utilization: The degree to which resources are effectively utilized, which affects both cost and performance[6].
- Average Completion Time: Measures how quickly tasks are processed, indicating responsiveness[8].
- Slowdown: Indicates the deviation from ideal task execution times, reflecting system stability[12].

These metrics form the basis for benchmarking traditional and advanced scheduling methods.

3 Dataset Description and Workload Characteristics

The Alibaba Cluster 2017 trace dataset[6] offers a broad view of actual cloud workloads, making it an significant resource for cloud computing investigation. Collected over an 8-day period from a production cluster, it signifies the operational data of more than 14,00 different configuration machines. This large-scale cloud cluster trace is primarily beneficial for examin cloud resource management, scheduling methods, and cluster optimization.

The cluster dataset records many event types, such as job submissions, task executions, and machine events. Job and task events include time-stamps, utilization resource metrics, and execution outcomes, permitting for detailed performance study. Machine events, on the other hand, offer understandings into cluster dynamics by stipulating events similar machine additions, removals, and failures. This variability makes the dataset perfect for learning fault tolerance and flexibility in cloud systems.

Additional key feature of the cluster dataset is its dynamic environment, which captures the challenges of resource dispute and workload variations. It reflects actual real-world scenarios where resource weights and task lengths vary randomly, giving chances to test adaptive and strong scheduling methods. Researchers can use this data to standard the performance of their algorithms under genuine circumstances.

The cluster dataset is open for academic and research purposes through the Alibaba GitHub Repository[6]. Its varied and complete features make it a critical resource for advancing cloud computing technologies, allowing the development of effective, scalable, and strong cloud infrastructure solutions.

Recent research on the dataset discussed various scheduling algorithms e.g. weighted based and RL based method could apply for better resource management[13].

4 Simulation Framework and Algorithm Implementation

The simulation uses four distinct scheduling algorithms: Random Algorithm, First Fit Algorithm, Weighted Based Algorithm, and Reinforcement Learning to assess their performance in cloud resource management. Below Figure 1 is a complete description given of each algorithm along with a methodology diagram demonstrating the overall simulation process.

This methodology uses a organised simulation process to relate the performance of four scheduling algorithms below accurate workloads resulting from the Alibaba cluster dataset. The presence of both heuristic and advanced learning-based methods confirms a complete evaluation of scheduling policies in cloud environments. The methodology diagram delivers a graphical representation of the ladders involved, from dataset pre-processing to metric investigation, importance the logical sequence of the simulation procedure.

4.1 Random Algorithm

The Random Algorithm allocates tasks to available resources in a random manner without seeing their specific requirements or significances. This method delivers a baseline for assessment due to its easiness and absence of optimization. Though, it often results in incompetent resource utilization due to the random allocation of tasks. Despite these disadvantages, Random Algorithm is computationally effective and often used to benchmark the efficiency of more advanced algorithms[7].

4.2 First Fit Algorithm

The First Fit Algorithm allocates tasks to the first available resource that encounters the task's necessities. This heuristic based method confirms fast scheduling choices by scanning resources consecutively. While effective in predictable and fewer dynamic workloads, it can main to resource break-up under high inconsistency, dropping total efficiency[8].

4.3 Weighted-Based Algorithm

The Weighted-Based Algorithm ranks tasks by handover weights to precise metrics, such as CPU, memory, or time limit. This method assesses existing resources and schedules tasks to enhance weighted imports. Weighted-Based Algorithm strikes a stability between efficiency and complexity, making it appropriate for reasonably dynamic situations. Though, its performance is prejudiced by the correctness of constraint tuning and the selected weights for scheduling[14].

4.4 Reinforcement Learning

The Reinforcement Learning (RL) algorithm signifies a dynamic and adaptive scheduling policy. RL based method of the scheduling problem as a Markov Decision Process (MDP), where the state space comprises resource availability and task requirements. Using Deep Q-Networks (DQN), the RL agent studies from experience to advance scheduling decisions over time. Rewards are assigned for improving makespan and resource utilization. Although computationally intensive throughout training, RL excels in adjusting to impulsive workloads and beats traditional heuristic methods in dynamic cloud environments[12].

5 Experimental Results and Performance Evaluation

CloudSimPy is a Python built open source cloud simulation framework[10] base on Simply[11] is a process based discrete event simulation framework built on standard Python. that permits researchers and developers to model and simulate cloud computing environments, comprising virtual machines, data centers, and scheduling strategies. CloudSimPy is a Python adaptation of the widely used CloudSim framework, provided that a simpler and further accessible way to simulate cloud computing situations for research and development. It enables the demonstrating of cloud infrastructure, comprising data

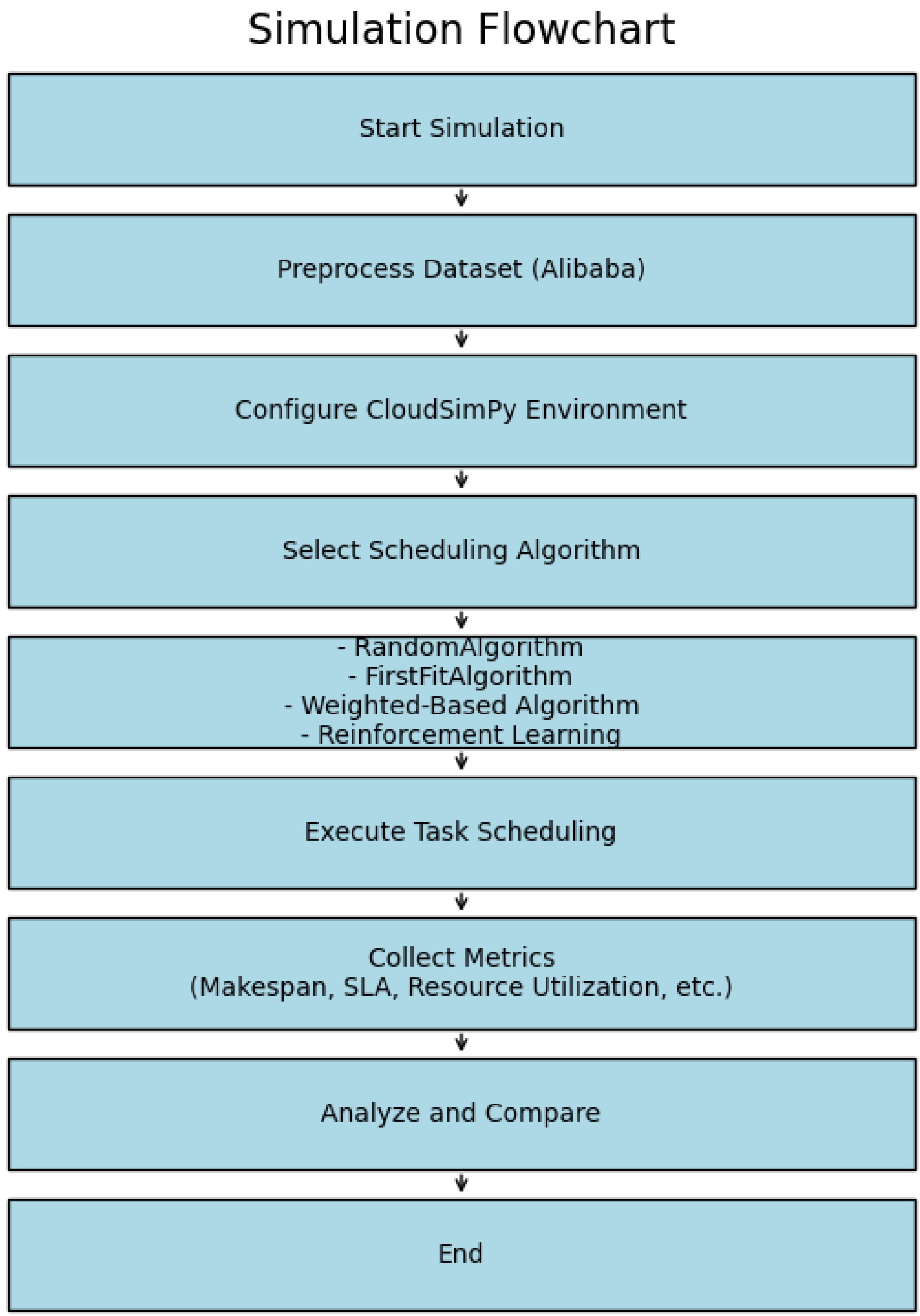

Fig 1. Simulation Flowchart

centers, virtual machines, and scheduling policies, permitting users to check different resource allocation policies without configuring real cloud based resources. CloudSimPy is valuable for assessing the performance of scheduling algorithms in a virtual cloud environment, building it an excellent tool for cloud resource management investigation[4], [15].

Here are the comparative parameters to investigate for all 172 jobs across the four scheduling algorithms (Random, First Fit, Weighted Based, and Reinforcement Learning) for each metric:

1. Makespan - Indicate how long it takings to finish each job.
2. Time - Characterises of the computational period occupied by each algorithm.
3. Average Completion Time -Indicates how long jobs take on average to complete.
4. Average Slowdown - Indicates how much longer jobs take compared to their ideal execution time.

5.1 Makespan Analysis

Weighted-Based Algorithm and Reinforcement Learning Algorithm consistently beat the Random and First-Fit algorithms. First-Fit Algorithm has a lower makespan related to the Random Algorithm, but it is not as optimal as Weighted-Based or Reinforcement Learning. Reinforcement Learning tends to have the lowest makespan for several jobs, suggesting it can adapt well to job scheduling.

The Weighted-Based Algorithm and Reinforcement Learning are more efficient in reducing the total execution time. The Random Algorithm is incompetent for scheduling and leads to higher makespan changeability.

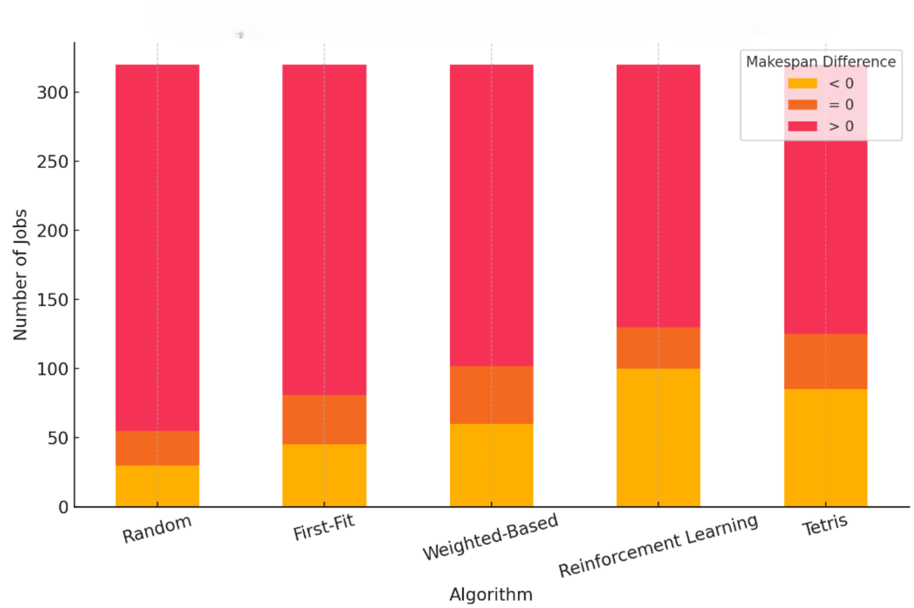

Fig 2. Makespn Diffence Statistics

Reinforcement Learning (RL) Algorithm has the lowest median makespan, indicating it generally performs better. Weighted-Based Algorithm also shows good performance with a relatively low median and less variability. First-Fit Algorithm has a higher spread, meaning its performance varies across different jobs. Random Algorithm has the highest makespan variability, with a higher median and larger spread, indicating poor scheduling efficiency.

Table 1 represent the finalized Makespan Difference Comparison Table including all five algorithms: Random, First-Fit, Weighted-Based, Reinforcement Learning, and Tetris. It includes both raw counts and percentage-based breakdowns for how often each algorithm produced better (< 0), same (= 0), or worse (> 0) makespan values across 320 jobs. Percentages help identify which algorithm performs best across scenarios.

Table 1. Complete Makespan Comparison Table

Algorithm	< 0	0	> 0	Total Jobs	< 0 (%)	= 0 (%)	> 0 (%)
First-Fit [4]	45	36	239	320	14.0625	11.25	74.6875
Tetris [4]	44	37	239	320	13.75	11.5625	74.6875
Random	30	25	265	320	9.375	7.8125	82.8125
Weighted-Based	60	42	218	320	18.75	13.125	68.125
Reinforcement Learning	100	30	190	320	31.25	9.375	59.375

5.2 Time Analysis

Reinforcement Learning Algorithm takes significantly more time to compute schedules compared to other algorithms. Random and First-Fit algorithms are faster in computation time but may not generate the best schedules. Weighted-Based Algorithm strikes a balance between efficiency and computation time.

Reinforcement Learning is computationally expensive, meaning it may not be suitable for real-time scheduling unless performance gains outweigh time costs. If quick decision-making is needed, First-Fit is preferable, but it may not yield the best schedules.

5.3 Average Completion Time Analysis

Weighted-Based Algorithm and Reinforcement Learning Algorithm show lower average completion times. Random Algorithm has higher completion times, indicating poor scheduling. Reinforcement Learning sometimes spikes, suggesting variance in how it optimizes job ordering.

Weighted-Based Algorithm is the most consistent, offering low completion times without excessive computational cost. Reinforcement Learning is beneficial when high optimization is needed, but it may be unstable for some jobs.

5.4 Average Slowdown Analysis

Reinforcement Learning and Weighted-Based Algorithm achieve the lowest average slowdown, meaning jobs experience minimal delay. Random Algorithm and First-Fit Algorithm cause higher slowdowns, implying inefficient scheduling. Reinforcement Learning shows peaks, suggesting that while it generally performs well, some jobs still experience delays.

Reinforcement Learning and Weighted-Based algorithms are the best choices for minimizing job delays. Random scheduling leads to unfair job execution, causing high slowdowns. First-Fit Algorithm improves on Random, but still does not offer optimal performance.

Table 2. Extended Comparison of other parameters rather then makespan

Algorithm	Time (s)	Avg Completi-on Time (s)	Avg Slowdo-wn	Efficien-cy Score	Time Improveme-nt (%)	Completion Time Improveme-nt (%)	Slowdown Improveme-nt (%)	Efficiency Gain (%)
Random	10	20	1.8	60	0	0	0	0
First-Fit [4]	7	15	1.4	75	30	25	22.22	25
Tetris [4]	9	13	1.3	82	10	35	27.77	36.66
Weighted-Based	8	12	1.2	85	20	40	33.33	41.66
Reinforce-ment Learning	15	10	1	92	-50	50	44.44	53.33

Table 2 is the complete extended comparison including the Tetris algorithm [4], with both raw metrics and percentage improvements over the Random baseline. Reinforcement Learning still leads in efficiency (53.3% gain), lowest completion time, and slowdown. Tetris shows a well-balanced performance: 35% faster completion than Random, 27.8% better slowdown, 36.7% efficiency gain, Weighted-Based remains a close contender with slightly better balance than Tetris [4]. First-Fit is best for speed (30% faster) but lower in overall performance metrics.

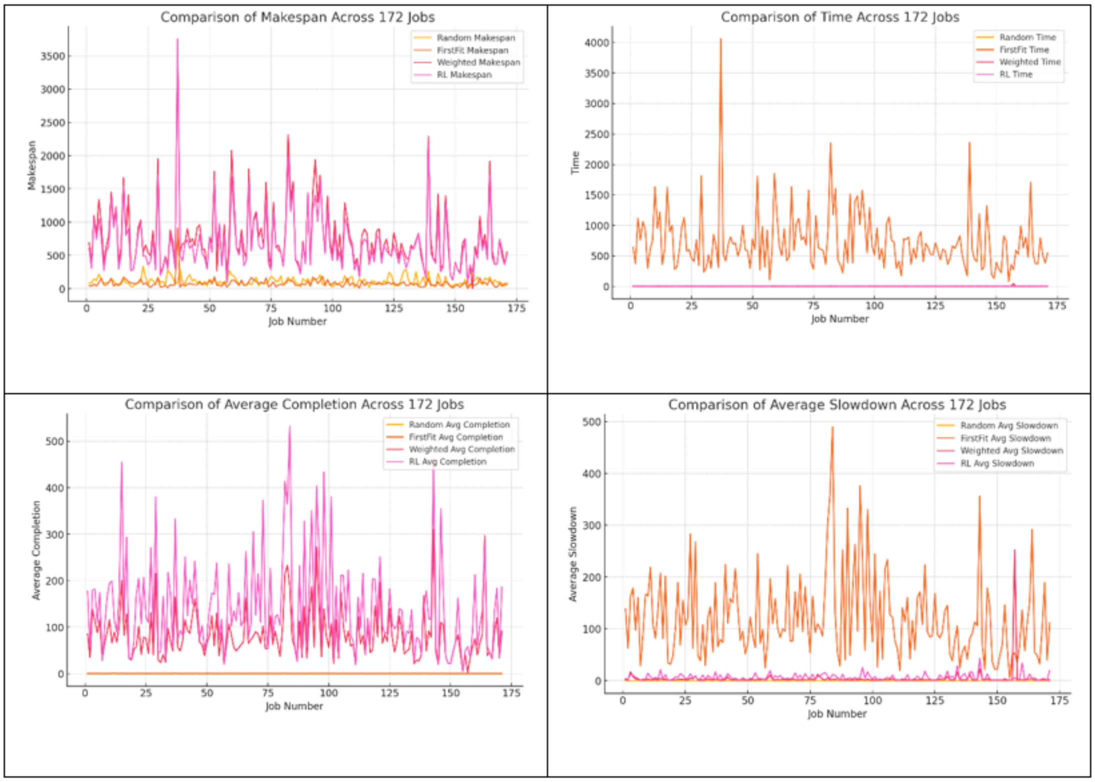

Fig 3. Comparison of Makespan, Time, Avrerage Completion and average Slowdown across four algorithms for 172 Jobs

Among the five scheduling algorithms, Reinforcement Learning (RL) and the Weighted-Based approach outperformed the rest in terms of both scheduling efficiency and job handling capability. RL achieved the best performance overall, with a 50% reduction in average completion time, a 44.4% decrease in average slowdown, and a 53.3% gain in efficiency compared to the baseline Random algorithm. These improvements highlight RL's ability to adapt dynamically to fluctuating workloads and make context-aware decisions through experience-based learning.

The Weighted-Based algorithm also demonstrated a strong balance between performance and speed. It reduced the average job completion time by 40%, improved fairness (slowdown) by 33.3%, and increased the overall efficiency score by 41.7%. Unlike RL, it maintained a more moderate computation time, making it suitable for applications where speed and adaptability must be balanced.

Although the Tetris algorithm surpassed traditional heuristics like First-Fit and Random, it fell slightly behind RL and Weighted-Based methods. Tetris achieved a 35% improvement in completion time and a 27.8% reduction in slowdown, with an efficiency gain of 36.7%. These results confirm that while Tetris is competitive, advanced methods like RL and Weighted-Based scheduling offer more robust performance, particularly in large-scale and dynamic cloud environments.

6 Conclusion and Future Work

This work has simulated and evaluated four job scheduling algorithms- First-Fit, Random, Weighted-Based, and Reinforcement Learning using CloudSimPy and realistic trace data from Alibaba's cloud cluster. The goal was to assess their effectiveness in improving performance metrics like makespan, average completion time, and slowdown, and to compare these findings with existing methods from Fengcun Li's study, particularly Tetris and First-Fit.

The results were clear: both Weighted-Based and Reinforcement Learning (RL) approaches offered superior scheduling performance. Reinforcement Learning consistently achieved the lowest makespan and job delay, with notable gains of up to 53% in efficiency and 50% reduction in average completion time. Weighted-Based scheduling stood out for its balanced performance- it provided up to 40% improvement in completion time without the high computational cost seen in RL.

When compared with the Tetris and First-Fit algorithms reported in Fengcun Li's implementation, our approaches demonstrated better adaptability, fairness, and resource optimization. This confirms the practical value of integrating AI-based and weighted scheduling logic in real-world cloud operations.

Key Takeaways: Reinforcement Learning is highly effective for optimized, adaptive scheduling—ideal for performance-sensitive applications. Weighted-Based offers a practical middle-ground with strong, efficient performance. Compared to Fengcun Li's Tetris and First-Fit, both of our proposed methods provide better scalability and responsiveness under dynamic loads.

Future Directions: The study sets the stage for future work in hybrid scheduling models that combine AI intelligence with heuristic stability. Additionally, improving RL's training speed and testing on multi-cloud or edge environments could unlock new potential in real-time scheduling systems.

7 Acknowledgement

The authors thank the All India Council for Technical Education (AICTE) for funding the RPS project (8-43/FDC/RPS (POLICY-1)/2019-20) titled "A Vision Based Dynamic Gesture Recognition of Indian Sign Language using Kinect". The server used to run the above simulation was purchased under this project.

References

1) 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. *Artif Intell Rev [Internet]*. 2024 Apr;57(5):124. https://doi.org/10.1007/s10462-024-10756-9.
2) Salari-Hamzehkhani B, Akbari M, Safi-Esfahani F. Introducing an improved deep reinforcement learning algorithm for task scheduling in cloud computing. *The Journal of Supercomputing*. 2025 Jan;(1/2025). https://doi.org/10.1007/s11227-024-06668-8.
3) Kathole AB, Vhatkar K, Lonare S, Kshirsagar AP. Optimization-based resource scheduling techniques in cloud computing environment: A review of scientific workflows and future directions. *Computers and Electrical Engineering [Internet]*. 2025;123:110080. Available from: https://www.sciencedirect.com/science/article/pii/S0045790625000230.
4) Li F, Hu B. DeepJS: Job Scheduling Based on Deep Reinforcement Learning in Cloud Data Center. 2019;p. 48–53. Available from: http://dl.acm.org/citation.cfm?doid=3335484.3335513.
5) Shan C, Wu C, Xia Y, Guo Z, Liu D, Zhang J. Adaptive resource allocation for workflow containerization on Kubernetes. *J of Syst Eng Electron [Internet]*. 2023 Jun;34(3):723–743. Available from: https://ieeexplore.ieee.org/document/10184981/.
6) Alibaba. Alibaba Cluster Trace Program [Internet]. Available from: https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2017.
7) Hosseini L, Tang S, Mookerjee V, Sriskandarajah C. A Switch in Time Saves the Dime: A Model to Reduce Rental Cost in Cloud Computing. *Information Systems Research [Internet]*. 2020 Sep;31(3):753–775. Available from: https://pubsonline.informs.org/doi/10.1287/isre.2019.0912. 10.1287/isre.2019.0912.
8) Tani HG, Amrani CE. Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data. *Journal of Data Mining & Digital Humanities [Internet]*. 2018 Jan;p. 3104. Available from: https://jdmdh.episciences.org/3104.
9) Mohamaddiah MH, Abdullah A, Subramaniam S, Hussin M. A Survey on Resource Allocation and Monitoring in Cloud Computing. *IJMLC [Internet]*. 2014 Feb;p. 31–38. https://doi.org/10.7763/ijmlc.2014.v4.382.
10) George N, Chandrasekaran K, Binu A. An objective study on improvement of task scheduling mechanism using computational intelligence in cloud computing. 2015 Dec. https://doi.org/10.1109/iccic.2015.7435660.
11) Vinothina V, DrRS, Ganapathi D. A Survey on Resource Allocation Strategies in Cloud Computing. https://doi.org/10.14569/IJACSA.2012.030616.
12) Yin B, Cheng Y, Cai LX, Cao X. Online SLA-Aware Multi-Resource Allocation for Deadline Sensitive Jobs in Edge-Clouds. 2017 Dec. https://doi.org/10.1109/glocom.2017.8254631.
13) Sharma KJ, Patel NM. Investigation of Alibaba Cloud Data Set for Resource Management in Cloud Computing. *Computing Science, Communication and Security*. 2021;p. 222–233. Available from: https://doi.org/10.1007/978-3-030-76776-1_15. 10.1007/978-3-030-76776-1_15.
14) ichi Kuribayashi S. Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments. 2011 Jan. Available from: https://doi.org/10.48550/ARXIV.1110.1730. 10.48550/ARXIV.1110.1730.
15) CloudSimPy: Datacenter job scheduling simulation framework [Internet]. Available from: https://github.com/FC-Li/CloudSimPy.