

Deep Reinforcement Learning for Intelligent Resource Allocation in Dynamic Cloud Computing Environments

Dr. William Turner

Professor, Manchester Institute of Advanced Computing Systems, United Kingdom

Abstract

Cloud computing environments support dynamically changing computational workloads through shared processing, memory, storage, and network resources. Efficient resource allocation remains challenging because incoming tasks vary in computational demand, execution duration, priority, and service-level requirements. Conventional rule-based and heuristic allocation procedures can respond effectively under predictable operating conditions but may become less suitable when workloads change rapidly. Deep reinforcement learning offers a methodological alternative in which an intelligent scheduling agent learns resource-allocation policies through repeated interaction with a simulated cloud environment. This hypothetical methodological study examines deep reinforcement learning for intelligent resource allocation under dynamic cloud workloads. The proposed framework represents cloud allocation as a sequential decision problem containing system states, scheduling actions, and reward functions. A conceptual workload consisting of 5,000 scheduling intervals is used to compare three hypothetical allocation strategies: static threshold allocation, heuristic adaptive allocation, and a deep reinforcement learning policy.

Resource utilization, service-level agreement violations, average job waiting time, and normalized energy consumption are used as evaluation indicators. The illustrative analysis assigns the strongest overall performance to the deep reinforcement learning configuration, with higher average utilization and lower job waiting time than the comparison strategies. A second analysis examines the same hypothetical policy under low, medium, high, and volatile workloads. Performance becomes weaker as environmental uncertainty increases, illustrating why a policy trained under one workload condition should not automatically be assumed to generalize to another. The study concludes that deep reinforcement learning provides a useful methodological framework for adaptive resource management because it can represent sequential decisions and delayed performance consequences. However, practical deployment requires stable training, safe exploration, reproducible simulation, workload generalization, and comparison with strong conventional allocation approaches. The numerical results reported in this manuscript are illustrative rather than experimentally validated.

Keywords: Deep Reinforcement Learning, Cloud Computing, Resource Allocation, Dynamic Workloads, Deep Q-Network, Resource Scheduling, Cloud Resource Management, Virtualization, Intelligent Scheduling, Service-Level Agreement.

1. Introduction

Cloud computing provides computing infrastructure, platforms, and applications through dynamically provisioned shared resources. Its operational model allows computing capacity to be allocated according to changing service requirements rather than maintaining permanently dedicated infrastructure. Armbrust et al. (2010) identified elasticity and resource availability as central characteristics of cloud computing. However, the ability to provide flexible capacity also creates a continuous resource-management problem because processing demand, storage requirements, memory utilization, and service workloads can vary considerably across time.

Resource allocation determines how available infrastructure is assigned among competing applications, virtual machines, and computational tasks. Poor allocation can produce overloaded servers, delayed job execution, underutilized infrastructure, unnecessary energy consumption, or service-level agreement violations. Buyya et al. (2009) emphasized market-oriented and service-driven cloud resource management, while Beloglazov et al. (2012) examined energy-aware allocation in data centers. These studies demonstrate that allocation decisions must often balance several objectives rather than optimize a single performance indicator.

Conventional cloud-management techniques commonly use threshold rules, optimization procedures, scheduling heuristics, and predictive provisioning. Such methods can be effective when workload characteristics and optimization objectives are reasonably stable. However, dynamic cloud environments may involve changing task arrivals, heterogeneous resource requirements, varying execution durations, and unpredictable utilization patterns. A fixed policy designed for one workload may therefore perform poorly after environmental conditions change. Adaptive allocation methods seek to modify resource decisions according to observed system behavior.

Reinforcement learning provides a framework in which an agent learns decisions by interacting with an environment and receiving feedback through reward signals. Tesauro (2005) demonstrated reinforcement learning for online resource allocation in autonomic computing. Developments in deep learning later enabled reinforcement-learning methods to represent more complex state spaces. Mnih et al. (2015) demonstrated deep Q-network learning from high-dimensional states, while Mao et al. (2016) applied deep reinforcement learning directly to systems resource management through the DeepRM framework.

The present hypothetical study examines a deep reinforcement learning architecture for dynamic cloud-resource allocation. The proposed methodology treats task scheduling as a sequential decision problem in which system state includes resource availability, job requirements, and queue conditions. Alternative scheduling strategies are compared using illustrative performance values. The objective is not to claim implementation of a production cloud scheduler but to demonstrate how a reproducible DRL-based resource-allocation study can be structured, evaluated, and interpreted.

2. Objectives of the Study

The primary objective is to examine deep reinforcement learning as a methodological approach for adaptive resource allocation in dynamic cloud environments. The study focuses on policy learning, workload-responsive allocation, resource-utilization efficiency, service-quality indicators, and major implementation challenges.

2.1 Examination of DRL-Based Resource Allocation

The first objective is to examine how resource allocation can be represented as a reinforcement-learning problem. Cloud conditions are conceptually represented through available CPU capacity, memory availability, active jobs, queued tasks, and resource demands. The agent selects scheduling actions and receives rewards representing desirable or undesirable system outcomes. This formulation allows resource allocation to be modeled as a sequence of related decisions rather than independent scheduling events, providing a methodological basis for adaptive cloud management.

The study also considers how deep neural networks can approximate action values when the system contains many possible workload conditions. Instead of constructing a separate scheduling rule manually for every state, a deep model can conceptually learn useful representations from observed resource patterns. However, increased representational flexibility introduces training instability and interpretability limitations. The objective therefore examines deep reinforcement learning as a potential adaptive allocation mechanism while recognizing that complex policies must be evaluated against simpler and more transparent alternatives.

2.2 Evaluation of Resource Allocation Performance

The second objective is to establish a transparent procedure for comparing alternative allocation policies. The hypothetical analysis evaluates static threshold allocation, heuristic adaptive allocation, and a deep reinforcement learning policy under an identical conceptual workload. Performance measures include average resource utilization, job waiting time, service-level agreement violation rate, and normalized energy consumption. These indicators reflect different aspects of cloud-system performance and discourage reliance on one metric as a complete measure of allocation effectiveness.

A further objective is to assess the hypothetical DRL policy under different workload intensities. Low, medium, high, and volatile workloads are considered separately because resource-allocation behavior may change significantly as system pressure increases. A policy demonstrating good average performance may still produce unacceptable delays or violations during extreme conditions. Workload-sensitive evaluation therefore provides additional evidence concerning policy robustness and highlights the need to test learned allocation strategies under conditions different from those represented during training.

2.3 Identification of Deployment Challenges

The third objective is to identify methodological and operational difficulties affecting DRL-based cloud allocation. Important concerns include state-space complexity, reward-function design, unstable learning, costly exploration, changing workload distributions, and computational overhead. A poorly specified reward can encourage behavior that improves one performance metric while reducing overall system quality. Effective DRL research therefore requires careful definition of system objectives and transparent reporting of the relationships between rewards and operational performance measures.

The investigation also considers the gap between simulation and production deployment. Reinforcement-learning policies are commonly developed in controlled environments where failed actions have limited consequences. In operational clouds, unsuitable exploratory decisions may cause substantial latency or service disruption. Simulation can reduce such risk but may not represent every characteristic of real infrastructure. The objective consequently emphasizes validation, workload

diversity, policy monitoring, and safe deployment procedures as essential requirements for practical DRL resource management.

3. Review of Literature

3.1 Cloud Computing and Resource Management

Buyya et al. (2009) presented a market-oriented vision of cloud computing emphasizing service-level requirements and dynamically provisioned resources. Armbrust et al. (2010) discussed the broader opportunities and technical challenges associated with cloud infrastructure. Calheiros et al. (2011) subsequently introduced CloudSim, enabling researchers to simulate cloud systems, virtual machines, data centers, and resource-provisioning policies. Together, these studies established important conceptual and experimental foundations for evaluating alternative resource-allocation procedures without requiring large physical data-center deployments.

Beloglazov et al. (2012) investigated energy-aware resource-allocation heuristics designed to reduce power consumption while maintaining quality-of-service requirements. Manvi and Shyam (2014) reviewed resource-management techniques for Infrastructure-as-a-Service environments, including provisioning, allocation, mapping, and adaptation. Jennings and Stadler (2015) further reviewed resource management in cloud environments and identified major challenges concerning scalability and dynamic control. This literature demonstrates that resource allocation involves interconnected performance, energy, workload, and service-management objectives.

3.2 Reinforcement Learning for Computing Resource Allocation

Tesauro (2005) investigated reinforcement learning for online server allocation in a distributed computing environment characterized by changing application loads. The approach demonstrated that resource management can be formulated as sequential decision-making in which allocation actions influence future utility. Tesauro et al. (2006) further investigated reinforcement-learning approaches to autonomic resource allocation. These studies established an important methodological link between adaptive computing systems and reinforcement learning before the emergence of modern deep neural architectures.

Resource-management research also investigated adaptive control and virtualization. Kusic et al. (2009) examined power and performance management in virtualized computing environments through control-based procedures. Padala et al. (2009) developed automated control for multiple virtualized resources, while Islam et al. (2012) examined adaptive resource provisioning through empirical prediction. Collectively, such work demonstrates that dynamic resource management can benefit from feedback-driven decisions, although reinforcement learning represents only one among several available adaptive-control paradigms.

3.3 Deep Reinforcement Learning and Resource Scheduling

Mnih et al. (2015) demonstrated how deep neural networks and Q-learning could be combined to learn control policies from complex state representations. Their work significantly expanded interest in deep reinforcement learning by showing that a common learning architecture could develop successful policies across multiple decision environments. Silver et al. (2014) also developed deterministic policy-gradient methods relevant to continuous action spaces, while van Hasselt et al. (2016) addressed value overestimation through Double Q-learning.

Mao et al. (2016) applied deep reinforcement learning directly to computing resource management through DeepRM. Their formulation translated multidimensional job scheduling into a reinforcement-learning environment and demonstrated that a policy could learn resource-management strategies from experience. This work is especially relevant to cloud allocation because it connects DRL concepts with task packing and scheduling. Nevertheless, resource-management performance remains dependent on workload representation, state encoding, reward definition, training stability, and the quality of comparison baselines.

4. Research Methodology

4.1 Research Design and DRL Framework

The present study adopts a hypothetical methodological research design centered on sequential resource allocation. The cloud environment is represented conceptually as a collection of virtualized computing nodes offering CPU and memory capacity. Tasks arrive over time with different resource demands and execution durations. At each scheduling interval, the agent observes available capacity, queued jobs, and currently allocated tasks before selecting a scheduling action. A reward signal summarizes the desirability of resulting resource-management behavior.

The conceptual deep reinforcement learning agent uses a deep Q-network to estimate action values. Each state is mapped to predicted values for eligible scheduling actions, and the policy selects actions according to the learned estimates. Experience replay and a separate target network would conceptually be used to reduce learning instability. No actual DQN implementation was executed for this manuscript. The architecture is included to establish a plausible experimental design for subsequent simulation-based research.

4.2 Hypothetical Cloud Workload and Allocation Strategies

The hypothetical workload contains 5,000 scheduling intervals and 2,400 conceptual computational jobs. Each job is assigned CPU demand, memory demand, duration, and service-priority attributes. The simulated infrastructure contains twenty computational nodes with normalized resource capacities. Workload intensity varies throughout the conceptual experiment to represent periods of low, medium, high, and volatile demand. All jobs, resources, and workload distributions are illustrative and are not obtained from an actual cloud provider.

Three resource-allocation strategies are evaluated conceptually. Static threshold allocation assigns resources according to fixed utilization thresholds. Heuristic adaptive allocation adjusts scheduling according to current queue and capacity conditions. The DRL allocation strategy selects actions through learned state-action values. All strategies conceptually receive the same workload sequence and infrastructure capacity. This common evaluation setting is necessary because performance comparisons would otherwise be confounded by different job-arrival patterns or resource availability.

4.3 Reward Design and Evaluation Measures

The hypothetical reward function combines job waiting time, service-level agreement violations, unused capacity, and normalized energy consumption. Penalties are assigned to excessive waiting and SLA violations, while efficient utilization contributes positively to the reward. The purpose is to demonstrate multi-objective policy design rather than define a universally optimal cloud reward function. In a

genuine study, reward weights would require sensitivity analysis because their values can strongly influence learned scheduling behavior.

Evaluation is based on average utilization, average job waiting time, SLA violation rate, and normalized energy index. Resource utilization indicates the proportion of available capacity actively used. Job waiting time measures delay before execution begins. SLA violation rate represents the proportion of tasks exceeding a conceptual service threshold. The normalized energy index expresses energy use relative to a fixed reference configuration. These measures allow policy behavior to be assessed from efficiency, service, and infrastructure perspectives.

5. Analysis

5.1 Comparative Resource Allocation Performance

The first hypothetical analysis compares three allocation strategies across the same conceptual workload. Static threshold allocation is assigned average resource utilization of 68.4%, while heuristic adaptive allocation reaches 76.8%. The DRL policy is assigned utilization of 84.6%. Average waiting time decreases across the three configurations from 12.8 to 9.1 and 6.4 scheduling units. These values illustrate how adaptive allocation could be evaluated across multiple operational performance dimensions rather than solely through resource utilization.

The DRL policy also records the lowest illustrative SLA violation rate at 3.8%, compared with 7.6% for static allocation and 5.4% for heuristic allocation. Its normalized energy index is assigned a value of 0.81 relative to 1.00 for the static baseline. These differences are entirely hypothetical and do not establish empirical DRL superiority. Actual conclusions would require implementation, repeated simulation runs, workload replication, statistical testing, and comparison against optimized heuristic baselines.

Table 1: Hypothetical Performance of Cloud Resource-Allocation Strategies

Allocation Strategy	Avg. Resource Utilization (%)	Avg. Job Waiting Time	SLA Violations (%)	Normalized Energy Index
Static Threshold Allocation	68.4	12.8	7.6	1.00
Heuristic Adaptive Allocation	76.8	9.1	5.4	0.90
Deep Reinforcement Learning	84.6	6.4	3.8	0.81

Note: All performance values are illustrative and do not represent measurements from an implemented cloud platform or DRL agent.

5.2 DRL Performance Under Changing Workload Conditions

The second analysis examines the hypothetical DRL policy across four workload conditions. Under low demand, resource utilization is assigned at 72.5% with a 1.8% SLA violation rate. Under medium and high workloads, utilization increases to 84.1% and 91.3%, while SLA violations also increase. The volatile workload records lower utilization than the steady high-load condition but produces the largest illustrative response delay because rapidly changing job arrivals create additional scheduling uncertainty.

The hypothetical results demonstrate why average performance should not be interpreted independently of workload structure. A DRL agent trained primarily under moderate demand may behave differently during sharp workload transitions. Volatility can challenge state representations and learned action values because recent observations may provide weaker information about upcoming resource requirements. These values are designed solely to illustrate workload-specific evaluation. Empirical robustness would require multiple workload traces and testing on conditions excluded from policy training.

Table 2: Hypothetical DRL Performance Across Dynamic Workload Conditions

Workload Condition	Resource Utilization (%)	SLA Violations (%)	Avg. Waiting Time	Normalized Reward
Low Demand	72.5	1.8	3.6	0.89
Medium Demand	84.1	3.1	5.2	0.84
High Demand	91.3	5.7	8.4	0.73
Volatile Demand	82.7	6.4	9.3	0.68

Note: Workload scenarios and results are fictional and do not constitute empirical cloud-computing performance measurements.

6. Discussion

6.1 Adaptive Allocation and Resource Utilization

The hypothetical analysis illustrates why deep reinforcement learning can be considered for dynamic cloud-resource allocation. Unlike a fixed threshold rule, an RL policy can conceptually associate current system conditions with long-term consequences of scheduling actions. Mao et al. (2016) demonstrated this principle through DeepRM, where resource management was formulated as an online decision problem. Such a formulation can be useful when immediate allocation efficiency must be considered alongside future job arrivals and queue conditions.

The assigned utilization improvement in this paper should nevertheless not be interpreted as evidence that DRL automatically outperforms conventional scheduling. Well-designed heuristics can provide strong performance with lower training and computational overhead. The relative advantage of a learned policy depends on workload complexity and the quality of baseline methods. Future experiments should therefore compare DRL against optimized heuristics, control methods, and alternative learning procedures under identical workload traces and resource constraints.

6.2 Service Quality and Multi-Objective Optimization

Cloud scheduling often involves competing objectives. Increasing utilization may improve infrastructure efficiency, but aggressive consolidation can create contention and service degradation. Similarly, reducing waiting time may require maintaining spare capacity that increases energy consumption. Beloglazov et al. (2012) demonstrated the importance of balancing energy use and quality-of-service requirements in cloud data centers. A reinforcement-learning reward must therefore represent several operational priorities without allowing one objective to dominate unintentionally.

The hypothetical reward in the present study includes waiting time, SLA violations, unused capacity, and energy consumption. Such a design illustrates multi-objective optimization but raises an important methodological concern: changing reward weights can change the learned policy substantially. A model reporting strong reward performance may therefore reflect favorable weighting rather than objectively superior resource management. Researchers should report individual operational measures alongside aggregate reward and perform sensitivity analysis across alternative reward specifications.

6.3 DRL Learning and Dynamic Workloads

The workload comparison demonstrates the importance of evaluating learned policies under environmental variation. Reinforcement learning assumes that experience obtained during training provides useful information for future decision-making. If workload characteristics change substantially, previously learned action values may become less appropriate. Dynamic cloud systems may experience sudden demand surges, changing job mixes, and variations in available resources. These conditions create a generalization challenge for policies trained in restricted simulation settings.

Policy adaptation introduces additional complexity because online exploration can affect production workloads. An RL agent that intentionally tests uncertain actions may cause avoidable service degradation. One possible methodological approach is extensive offline simulation followed by constrained online adaptation. However, simulation models must accurately represent important infrastructure behavior. Evaluation should therefore examine both policy performance and the difference between the simulated training environment and the operational system in which the policy is intended to run.

7. Challenges

7.1 State-Space Complexity and Training Stability

Cloud environments can contain many servers, virtual machines, jobs, queue positions, and possible allocation actions. The number of system states can therefore increase rapidly with infrastructure scale. Deep networks can approximate value functions without storing each state explicitly, but large state spaces still create learning difficulties. State representation must preserve important scheduling information while remaining computationally manageable. Poor representations may hide information required for appropriate decisions or make training unnecessarily complex.

Training stability presents a second challenge because deep reinforcement learning combines function approximation with bootstrapped value updates. Estimated values can change as network parameters are updated, producing unstable learning behavior. Techniques such as target networks and experience replay were introduced to reduce such instability in deep Q-learning. Nevertheless, cloud resource allocation adds workload nonstationarity and multi-objective rewards. Training procedures

should therefore report convergence behavior, random seeds, sensitivity to initialization, and variability across independent runs.

7.2 Reward Engineering and Safe Exploration

Reward engineering is critical because the reinforcement-learning agent optimizes the numerical objective it receives rather than the broader intention of system administrators. A reward emphasizing utilization too strongly may encourage excessive consolidation, whereas a reward dominated by service penalties may maintain unnecessary spare capacity. Reward terms and their scaling should therefore correspond clearly to operational priorities. Researchers should evaluate whether policies achieve desirable individual metrics rather than reporting only cumulative reward.

Exploration presents a related operational challenge. Reinforcement-learning agents require information about alternative actions, but testing uncertain allocations within a production environment can create delays or service failures. Training in simulation can reduce direct operational risk, yet simulation may omit hardware failures, network contention, or complex application dependencies. Safe deployment may therefore require constrained action spaces, policy validation, rollback mechanisms, and monitoring systems capable of detecting unacceptable behavior before significant service degradation occurs.

7.3 Workload Generalization and Computational Overhead

Workload generalization is important because cloud systems may experience conditions that were not represented during training. Job distributions, service priorities, resource demands, and arrival rates can change over time. A policy optimized for one workload may therefore perform poorly under a different application mixture. Robust evaluation should use multiple workload traces and should reserve some scenarios exclusively for out-of-distribution testing. This distinction helps determine whether the agent has learned transferable allocation principles or memorized characteristics of the training environment.

Deep reinforcement learning also introduces computational overhead associated with model training, experience storage, and action inference. Although inference may be rapid after training, repeated policy updating can require significant resources. These costs should be compared with the performance improvements obtained over heuristic allocation. A complex DRL system may not be justified when a simpler approach achieves similar service quality with substantially lower development and maintenance requirements. Operational suitability therefore requires both predictive and systems-level evaluation.

8. Conclusion

Deep reinforcement learning provides a methodological framework for adaptive resource allocation in cloud computing environments where workloads and infrastructure conditions change over time. The present hypothetical study represented scheduling as a sequential decision problem combining resource states, allocation actions, and multi-objective reward signals. It also demonstrated how DRL could be compared with static and heuristic resource-management approaches through resource utilization, job delay, SLA violations, and energy-related indicators.

The illustrative analysis assigned stronger overall performance to the hypothetical DRL policy and showed that its effectiveness varied across workload conditions. These numerical results are entirely constructed and do not constitute experimental evidence. Real performance claims would require an

implemented agent, validated simulation environment, multiple workload traces, independent runs, statistical analysis, and strong benchmark policies. Reward sensitivity and policy generalization would also require explicit evaluation.

The study concludes that DRL is a promising research direction for intelligent cloud-resource management but should not be treated as an automatically superior replacement for conventional scheduling. Future work should emphasize stable learning, safe exploration, transparent reward design, workload generalization, computational overhead, and deployment validation. Such evaluation can determine when learned allocation policies offer practical advantages in dynamic cloud computing systems.

References

1. 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, Article 124. <https://doi.org/10.1007/s10462-024-10756-9>
2. Neelakantan, P., Gangappa, M., Rajasekar, M., Kumar, T. S., & Reddy, G. S. (2024). Resource allocation for content distribution in IoT edge cloud computing environments using deep reinforcement learning. *Journal of High Speed Networks*, 30(3). <https://doi.org/10.3233/JHS-230165>
3. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks* (pp. 50–56). Association for Computing Machinery. <https://doi.org/10.1145/3005745.3005750>
4. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533. <https://doi.org/10.1038/nature14236>
5. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press. <http://incompleteideas.net/book/the-book-2nd.html>
6. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, 8, 279–292. <https://doi.org/10.1007/BF00992698>
7. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*. <https://arxiv.org/abs/1509.02971>
8. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*. <https://arxiv.org/abs/1707.06347>
9. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning* (pp. 1861–1870). PMLR. <https://proceedings.mlr.press/v80/haarnoja18b.html>
10. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
11. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616. <https://doi.org/10.1016/j.future.2008.12.001>

12. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. <https://doi.org/10.1016/j.future.2011.04.017>
13. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23–50. <https://doi.org/10.1002/spe.995>
14. Zhan, Z.-H., Liu, X.-F., Gong, Y.-J., Zhang, J., Chung, H. S.-H., & Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys*, 47(4), Article 63. <https://doi.org/10.1145/2788397>
15. Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM Special Interest Group on Data Communication* (pp. 270–288). Association for Computing Machinery. <https://doi.org/10.1145/3341302.3342080>
16. Vázquez-Poletti, J. L., & Buyya, R. (2010). Energy-efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 IEEE International Conference on Cluster Computing Workshops and Posters*. IEEE. <https://doi.org/10.1109/CLUSTERWKSP.2010.5613094>
17. Harchol-Balter, M. (2013). *Performance modeling and design of computer systems: Queueing theory in action*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139226424>
18. Khani, M., et al. (2024). Resource allocation in 5G cloud-RAN using deep reinforcement learning algorithms: A review. *Transactions on Emerging Telecommunications Technologies*, 35(1), Article e4929. <https://doi.org/10.1002/ett.4929>
19. Energy efficient task scheduling based on deep reinforcement learning in cloud environment: A specialized review. (2024). *Future Generation Computer Systems*, 151, 214–231. <https://doi.org/10.1016/j.future.2023.10.002>
20. RAPID: Enabling fast online policy learning in dynamic public cloud environments. (2024). *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2023.126737>