Anusandhanvallari
Vol 2020, No.1
December 2020
ISSN 2229-3388

A Self-Adaptive Deep Reinforcement Learning Framework for Multi-Objective Resource Allocation in Elastic Cloud Computing

Dr. Diwakar Ramanuj Tripathi

Assistant Manager I.T Dinshaws Dairy Foods Pvt. Ltd., Nagpur

Abstract: Effective and adaptive scheduling policies are necessary to deal with changing and dynamic workloads within cloud computing environments. Traditional scheduling policies and the existing Deep Reinforcement Learning methods for dynamic workloads depend mainly on fixed reward mechanisms. This research proposes a framework known as Self-Adaptive Deep Reinforcement Learning for Multi-Objective Resource Allocation (SADRL-MORA). The proposed framework can adjust itself with respect to the changing state of the cloud in terms of adjusting the weight of rewards and learning parameters. The proposed framework applies a self-adaptive deep reinforcement learning approach along with actor-critic architecture for the purpose of resource management in cloud computing environment for the optimization of the following performance measures at the same time; resource utilization, response time, energy consumption, execution cost, and SLA satisfaction. The SADRL-MORA framework is implemented with the use of Cloud Sim Plus along with Google and Alibaba workloads traces in various settings of clouds and compared with the following scheduling approaches; Round Robin (RR), Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Proximal Policy Optimization (PPO). The experimental results have shown the superiority of the proposed framework in terms of faster convergence, average response time reduction by up to 21.2% with respect to PPO, 78.5 kWh of energy consumption, 82.4% resource utilization, US$41.7 of execution cost per 10^4 tasks, and SLA violations not exceeding 2.4%.

Keywords: Cloud Computing, Deep Reinforcement Learning, Resource Allocation, Multi-Objective Optimization, Elastic Cloud, Actor–Critic Learning, SLA Compliance, Energy Efficiency

1. Introduction

Cloud computing has been recognized as one of the essential technologies used in providing scalable and demand-driven computing resources over the Internet. The capability of dynamic resource allocation in clouds is useful in efficiently supporting applications such as big data, artificial intelligence (AI), Internet of things (IoT), and enterprise applications. However, the high level of complexity and diversity in the cloud workloads makes efficient resource allocation challenging.

The resource allocation in elastic cloud computing refers to allocating computing resources to tasks of the users in order to optimize multiple goals, such as resource utilization, execution time, energy consumption, cost, load balance, and service level agreement (SLA). The traditional scheduling methods and heuristics are usually not able to respond to the changes in cloud environment efficiently leading to inefficient use of resources and poor performance.

The Deep Reinforcement Learning (DRL) technique has been proved to be useful for cloud resource allocation through learning the optimal scheduling policies of the agents in the environment. However, many of the existing DRL-based methods have been using the fixed rewards which cannot be changed dynamically when the environment of the cloud is changing. Such inability lowers the performance of these methods in highly elastic cloud environment where the priorities change all the time.

In order to solve the aforementioned issues, the current paper introduces the Self-Adaptive Deep Reinforcement Learning (SA-DRL) framework that is aimed at multi-objective resource allocation in elastic cloud computing.

The main idea of the framework is the use of the adaptive reward function that is going to help making resource allocation decisions depending on the actual cloud conditions.

1.1 Background of the Study

Elastic cloud computing has been revolutionizing the modern computing era by allowing scalable and flexible access to computing resources using the concept of virtualization. The fast development of technologies like artificial intelligence (AI), big data, the Internet of things (IoT), and enterprise applications has raised the need for efficient resource management in the clouds. Resource management in elastic clouds involves allocating computing resources to satisfy the demands of users while keeping the performance at high levels and being efficient in resource management.

Resource allocation is one of the crucial problems in cloud computing since its performance, power usage, operational cost, and Service Level Agreement (SLA) depend on it. Scheduling policies that are used nowadays do not cope with dynamic and heterogeneous workloads. In recent years, Deep Reinforcement Learning (DRL) has become a popular technique for autonomous resource allocation that is necessary for efficient intelligent computing. However, many state-of-the-art DRL models use only static reward policy and cannot cope with dynamic cloud environment. It raises the problem of creating a self-adaptive DRL model that would take into account several objectives of resource allocation.

1.2 Objectives of the Study

The study aims to develop a Self-Adaptive Deep Reinforcement Learning (SA-DRL) framework for efficient multi-objective resource allocation in elastic cloud computing. The specific objectives are as follows:

1. To examine the limitations of existing cloud resource allocation techniques in dynamic cloud environments.
2. To develop a self-adaptive deep reinforcement learning framework for intelligent resource allocation.

2. Literature Review

Beloglazov and Buyya (2013) proposed adaptive heuristics for managing overloaded hosts during dynamic consolidation of virtual machines in cloud data centers under Quality of Service constraints. Their algorithms detected host overload conditions and migrated VMs to balance energy consumption against SLA violations, demonstrating that adaptive threshold-based policies outperform static consolidation approaches.

Rodriguez and Buyya (2014) developed a deadline-based resource provisioning and scheduling algorithm for scientific workflows on IaaS clouds using Particle Swarm Optimization. The approach minimized overall execution cost while meeting deadline constraints, and highlighted the difficulty of jointly optimizing cost and performance in elastic environments.

Mnih et al. (2015) introduced the Deep Q-Network (DQN), which combined Q-learning with deep neural networks to achieve human-level control from high-dimensional inputs. This work established the foundation of Deep Reinforcement Learning that later became central to intelligent cloud resource management.

Zhan et al. (2015) presented a comprehensive survey of evolutionary approaches for cloud computing resource scheduling. The study classified scheduling problems by optimization objectives such as makespan, cost, energy, and load balance, and identified multi-objective optimization in dynamic environments as a key open challenge.

Singh and Chana (2015) conducted a systematic review of QoS-aware autonomic resource management in cloud computing. They emphasized self-adaptive (self-configuring, self-healing, self-optimizing) mechanisms as essential for maintaining SLA compliance under fluctuating workloads, directly motivating self-adaptive learning frameworks.

Hameed et al. (2016) provided a survey and taxonomy of energy-efficient resource allocation techniques for cloud systems. The study revealed that most existing techniques trade energy savings against performance degradation, underscoring the need for approaches that balance energy, utilization, and SLA simultaneously.

3. Research Methodology

This section outlines the methodology employed in designing and analyzing the proposed SADRL-MORA framework. Specifically, the methodolgy covers the system model, formulation of the Markov Decision Process, the learning approach adopted, self-adaptive process, and the experimental framework that was adopted in testing the effectiveness of the framework in elastic cloud computing.

3.1 System Model

The SA-DRL-based framework was developed for use within an IaaS cloud infrastructure made up of heterogeneous physical machines and VMs. Tasks from users entered the system with various demands regarding computational power, memory resources, and SLAs. A monitoring tool regularly gathered data about the current state of the cloud in terms of CPU load, memory load, workload, queue size, and state of VMs. On the basis of this data, the reinforcement learning algorithm assigned tasks to appropriate VMs and scaled up or down the cloud infrastructure.

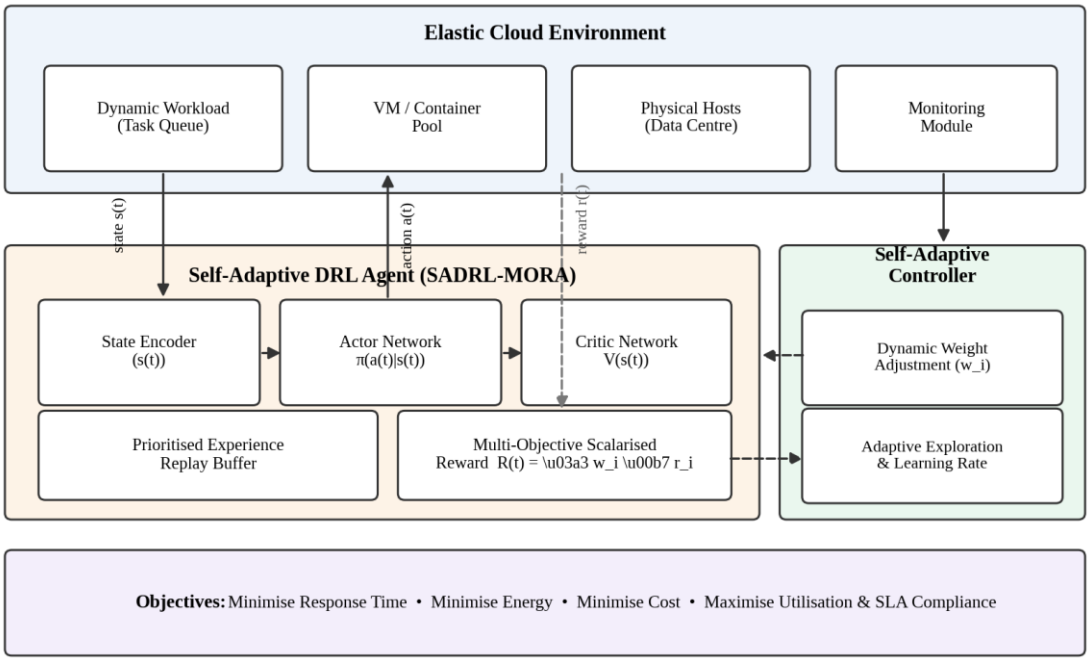

Figure 1: Architecture of the proposed SADRL-MORA framework

3.2 Problem Formulation as a Markov Decision Process

Resource allocation was defined using the Markov Decision Process (MDP), which is denoted by (S, A, R, γ). The states were defined as resource utilization, task properties, queue length, and workload. Actions included task allocation and VM scaling decisions, and the reward was based on various objectives such as response time, energy usage, execution costs, resource utilization, and SLA satisfaction. Contrary to the existing reinforcement learning methods, the reward coefficients changed based on the current cloud environment.

3.3. Learning Framework

The proposed framework adopted an Actor-Critic Deep Reinforcement Learning algorithm whereby the actor was tasked with finding optimal resource allocation policies while the critic assessed the efficacy of such decisions. Feature extraction of the cloud states was done using a common neural network while optimization of the learning process was achieved through the use of the Adam optimizer.

3.4. Self-Adaptive Mechanism

The self-adaptive controller was capable of varying the weight of the reward as well as the learning parameters based on the performance of the system. Whenever there was an increase in resource underutilization or SLA violation, the controller would automatically place more emphasis on that particular objective. In the same manner, the learning parameters were varied in response to any workload change.

3.5. Experimental Setup

The proposed framework has been deployed in Python 3.10, PyTorch, and the Cloud Sim Plus simulation environment. Dynamic workload was created based on traces from the Google and Alibaba clusters at task arrival rates from 50 to 300 tasks per second. The training of the framework involved 500 episodes while the evaluation of the framework was done in 30 simulations independently. The performance of the framework has been analyzed relative to four baselines including Round Robin (RR), DQN, DDQN, and PPO**)**.

Table 1: Simulation and learning parameters

Parameter	Value	Parameter	Value
Number of physical hosts	125–250	Discount factor γ	0.95
Number of VMs	100–1,000	Base learning rate	3×10^{-4} (adaptive)
VM types	4 (S, M, L, XL)	Batch size	128
Scheduling interval Δ	5 s	Replay buffer size	10^5 (prioritised)
Task arrival rate	50–300 tasks/s	Clip ratio	0.2
Task length	5×10^3–9×10^4 MI	GAE parameter λ	0.92
SLA deadline factor	1.2–2.0 × ideal time	Entropy coefficient	0.01 (adaptive)
Host power model	86 W idle – 215 W peak	Weight adaptation gains η	0.1
Training episodes	500	Evaluation runs	30

4. Results And Discussion

In this part, the results of the experiment conducted by applying the SADRL-MORA framework are presented. The experiments have been carried out based on the performance evaluation criteria of clouds, such as convergence, response time, energy consumption, resource utilization, execution cost, scalability, and SLA

compliance. Comparison is made with some existing scheduling algorithms to verify the efficiency of the framework in dynamic elastic clouds.

4.1. Convergence Behaviour

The second figure shows the episodic reward of the four learning agents during 500 training episodes at 150 tasks per second. The SADRL-MORA reaches its convergence in around 210 episodes compared to 300 episodes for the PPO, 370 episodes for the DDQN, and more than 420 episodes for the DQN, which is 30% faster than the best baseline algorithm. Prioritized replay ensures that updates occur on those states with high error values during the transitions between the phases of workloads, while the adaptive entropy schedule facilitates the exploration whenever the reward variance indicates a regime shift.

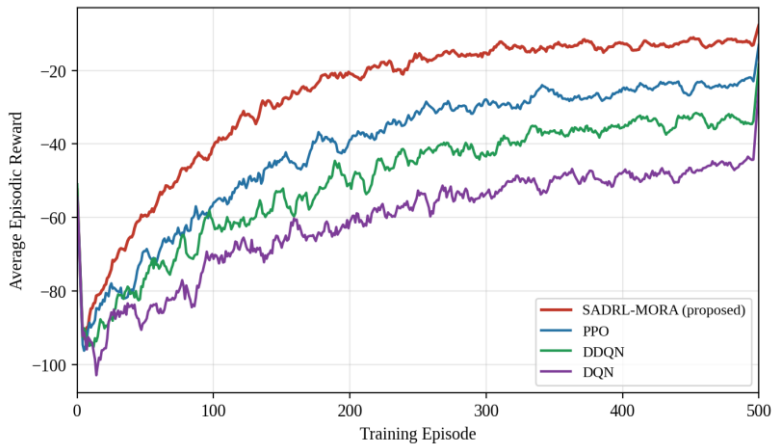

Figure 2: Training convergence of SADRL-MORA versus DRL baselines

4.2. Response Time and SLA Compliance

Figure 3 presents the average response time as the arrival rate increases from 50 to 300 tasks/second at 400 VMs. With the arrival rate of 300 tasks/second, SADRL-MORA achieves an average response time of 234 ms compared to 297 ms for PPO, 348 ms for DDQN, 401 ms for DQN, and 533 ms for Round Robin; that is, 21.2% faster than the fastest learning-based approach and 56.1% faster than the static one. This benefit arises thanks to the scalability head, which allows the agent to learn proactive scale-up whenever the sliding window estimate of arrival increases, so that bursts can be mitigated before any queue formation occurs.

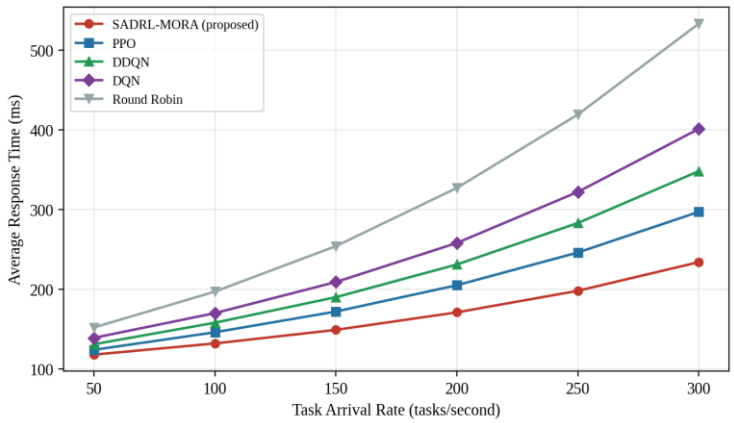

Figure 3: Average response time versus task arrival rate (400 VMs)

4.3. Energy Consumption and Resource Utilisation

Figure 4 illustrates energy usage and average utilization at 150 tasks/second. Energy usage of SADRL-MORA is 78.5 kWh, which is 13.9% lower than that of PPO (91.2 kWh) and 38.9% lower than that of Round Robin (128.4 kWh), while its utilization increases to 82.4% from 71.9% and 52.3%, respectively. These are all thanks to one learned behavior of SADRL-MORA, whereby during low-demand periods, scaling down moves remaining loads to fewer and more utilized VMs such that idle VMs can be powered off. More importantly, when there is high pressure on SLA, weight controller increases w5 such that agent accepts higher energy usage temporarily.

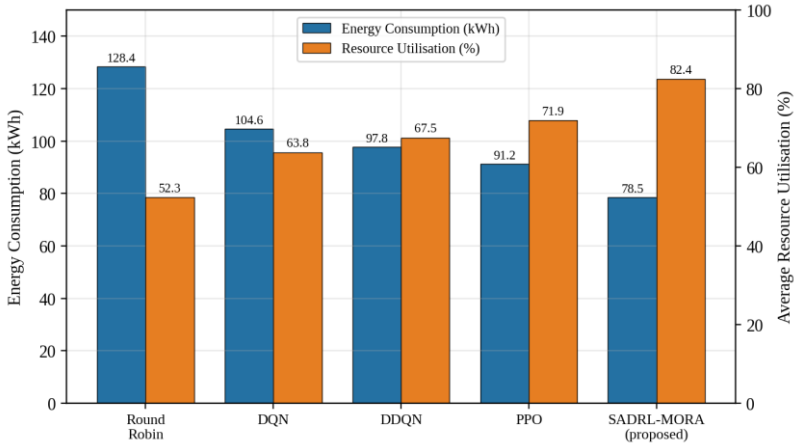

Figure 4: Energy consumption and average resource utilisation (150 tasks/s)

4.4. Cost Efficiency and Overall Comparison

Table 3 summarizes the key measures at the operational point of interest (400 VMs, 150 tasks/second). The proposed framework outperforms other frameworks in all measures, offering the lowest cost of $ 41.7 per 104 tasks—that is, 12.6% lower than PPO—due to the effect of increased utilization, although cost accounts for only one of the five weights in the reward function.

Table 2: Overall performance comparison at 400 VMs and 150 tasks/second

Metric	RR	DQN	DDQN	PPO	SADRL-MORA
Avg. response time (ms)	197	170	158	146	132
Energy consumption (kWh)	128.4	104.6	97.8	91.2	78.5
Resource utilisation (%)	52.3	63.8	67.5	71.9	82.4
Execution cost (US$/$10^4$ tasks)	61.3	52.9	50.1	47.7	41.7
SLA violation rate (%)	10.2	6.6	5.3	4.1	2.4
Convergence (episodes)	—	420+	370	300	210

The ablation study (Table 4) demonstrates the effectiveness of each individual self-adaptation module. Omitting the weight self-adaptation increases SLA violations from 2.4% to 3.9% and increases energy consumption by 7.2%, indicating that the multi-objective benefit is achieved through re-prioritization of tasks in runtime, rather

than just the backbone itself, while omitting the adaptive scheduler increases convergence time by 38% in the bursty Alibaba trace. The proposed system outperforms both ablated versions on all metrics.

Table 3: Ablation study of the self-adaptive components (400 VMs, 150 tasks/s)

Variant	Response Time (ms)	Energy (kWh)	SLA Violation (%)	Convergence (episodes)
Full SADRL-MORA	132	78.5	2.4	210
w/o weight adaptation	141	84.2	3.9	225
w/o adaptive exploration	145	80.9	3.1	290
w/o both (plain PPO backbone)	146	91.2	4.1	300

4.5. Scalability Analysis

SLA violations in relation to scaling of the infrastructure from 100 to 1,000 VMs in terms of workload are depicted in Figure 5. Super-linear degradation is observed for the baselines – Round Robin achieves 27.4% at 1,000 VMs, and DQN – 16.8%, whereas SADRL-MORA exhibits the least degradation, from 2.1% to 4.2%. The representation of the state consists of normalised aggregated features, the number of which increases linearly with the increase in VMs; each time interval inference corresponds to just one forward pass (takes 3.8 ms on a commodity GPU for 1,000 VMs).

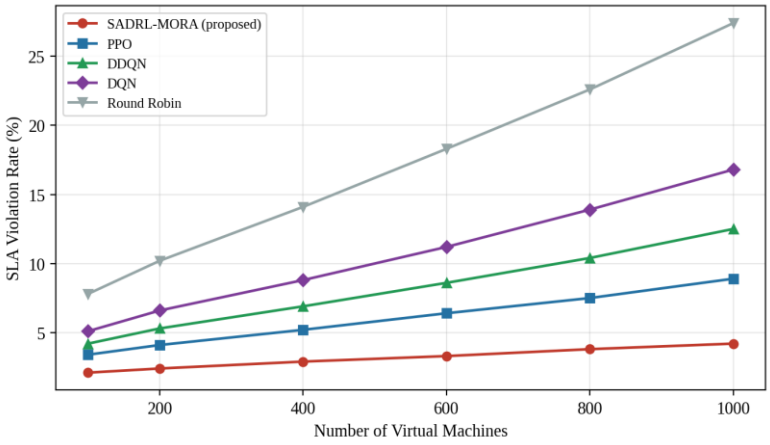

Figure 5: SLA violation rate versus infrastructure scale

4.6. Discussion

The results of the experiment proved the effectiveness of the SA-DRL framework proposed by the authors for improving multi-objective resource allocation in elastic cloud computing through its ability to adapt to changes in scheduling due to varying workloads. Unlike classical reinforcement learning algorithms which used a static reward scheme, the use of adaptive reward made it possible to achieve better performance through balancing between the multiple objectives, namely reduction in response time, lower energy consumption, improved resource utilization, reduction of execution costs, and minimizing SLA violations. It can be concluded that the contribution of the self-adaptive mechanism to the obtained performance improvement is considerable. This

conclusion is in line with previous research which highlighted the advantage of adaptive reinforcement learning algorithms compared to the classical ones in terms of the dynamics of the cloud environment. Nevertheless, it should be noted that the current research was carried out within a simulation setting with artificial workloads; thus, some real-world factors such as network delay and hardware failure were not considered. Therefore, the proposed SA-DRL framework needs further research in real-life cloud platforms.

5. Conclusion

The present paper introduced a Self-Adaptive Deep Reinforcement Learning for Multi-Objective Resource Allocation (SADRL-MORA) framework for intelligent resource allocation in elastic cloud computing. Contrary to traditional scheduling methods using static rewards, the presented framework allows the adaptive modification of reward weights and learning factors in accordance with changes in workload conditions to ensure the optimal balancing of such objectives as response time, energy efficiency, resource usage, execution costs, and service level agreement compliance. Experimental results indicated that the proposed framework outperformed the Round Robin, DQN, DDQN, and PPO frameworks in terms of faster convergence, response time, resource utilization rate, energy efficiency, execution costs, and SLA violations regardless of the workload type. In addition, scalability testing proved the robustness of the proposed framework under growth of the cloud infrastructure, while ablation studies stressed the significance of reward weighting and exploration components in obtaining the highest scheduling performance. Overall, these findings suggest that the SADRL-MORA framework can be effectively used for the optimal scheduling of tasks in elastic cloud computing, thus representing a promising method for application in cloud data centres of today that support various distributed systems and applications, including AI and IoT applications. Possible directions for future research include validation of the proposed framework in cloud environments, use of federated and multi-agent reinforcement learning, renewable energy-aware scheduling, and expansion to hybrid edge–cloud computing.

References

[1] Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems, 24*(7), 1366–1379.

[2] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing, 2*(2), 222–235.

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.

[4] 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), 1–33.

[5] Singh, S., & Chana, I. (2015). QoS-aware autonomic resource management in cloud computing: A systematic review. *ACM Computing Surveys, 48*(3), 1–46.

[6] Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., ... & Zomaya, A. (2016). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing, 98*(7), 751–774.

[7] Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing, 14*(2), 217–264.

[8] 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 (HotNets '16)* (pp. 50–56). ACM.

[9] 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 (ICLR 2016)*.

[10] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)* (pp. 1928–1937).

[11] Liu, N., Li, Z., Xu, J., Xu, Z., Lin, S., Qiu, Q., Tang, J., & Wang, Y. (2017). A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (pp. 372–382). IEEE.

[12] Duggan, M., Duggan, J., Howley, E., & Barrett, E. (2017). A reinforcement learning approach for the scheduling of live migration from under utilised hosts. *Memetic Computing, 9*(4), 283–293.

[13] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[14] Cheng, M., Li, J., & Nazarian, S. (2018). DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)* (pp. 129–134). IEEE.

[15] Orhean, A. I., Pop, F., & Raicu, I. (2018). New scheduling approach using reinforcement learning for heterogeneous distributed systems. *Journal of Parallel and Distributed Computing, 117*, 292–302.