Reinforcement Learning Techniques for Optimizing System Configuration on the Cloud: A Taxonomy and Open Problems

Theodoros Aslanidis, Andreas Chouliaras, Dimitris Chatzopoulos
School of Computer Science, University College Dublin
{theodoros.aslanidis, andreas.chouliaras}@ucdconnect.ie, dimitris.chatzopoulos@ucd.ie

Abstract

Efficient resource management (RM) is paramount for achieving high performance and utilization of computing resources in cloud computing environments. Conventional approaches, such as rule-based heuristics and optimization algorithms, face challenges in adapting to the dynamic and intricate nature of these environments. In this work, we investigate the utilization of reinforcement learning (RL) techniques for RM on the cloud. We provide a comprehensive taxonomy that categorizes RL-based approaches according to various facets of RM, encompassing resource allocation, auto-scaling, load balancing, and energy efficiency. By conducting an extensive literature review, we analyze and compare diverse RL algorithms employed in RM, highlighting the strengths and limitations of each approach. Last, we identify potential research directions in the context of RL-based resource management methods on the cloud.

Keywords
reinforcement learning, cloud resource management

1 Introduction

Resource management is a critical aspect of achieving high performance and efficient utilization of cloud resources. Over the years, numerous approaches have been proposed to address this challenge. These approaches include rule-based methods, heuristics, meta-heuristics, control or queuing theory methods, and traditional machine learning methods like supervised and unsupervised learning.

Heuristics [6, 23], provide simple and intuitive decision-making rules, while meta-heuristic algorithms [55], aim to find near-optimal solutions. Control and queuing-theoretic methods [41, 59], utilize mathematical models to allocate resources based on system performance metrics. Game-theoretic approaches [52, 66] model interactions between entities as strategic games, providing insights into optimal decision-making and equilibrium solutions. However, heuristics and control and queuing theoretic methods face limitations in adapting to changing conditions and handling the dynamic and complex nature of cloud environments. On the other hand, game-theoretic and meta-heuristic methods may encounter challenges in capturing the full complexity of interactions and require substantial computational resources.

Machine learning (ML) techniques offer notable advantages over traditional resource management methods, particularly when dealing with the non-convex properties of resource management. As highlighted by [27], they have the potential to generate superior solutions in real time, which suits the dynamic nature of cloud environments. However, the deployment of ML methods requires substantial data gathering and maintenance, incurring additional resource costs. Moreover, when considering temporal changes and interactions among multiple systems, efficient control becomes a formidable challenge for traditional ML approaches. This challenge has sparked a shift towards RL techniques that excel in *(i)* handling temporal changes, *(ii)* adapting to uncertain environments, and *(iii)* conducting long-term planning, utilizing exploration strategies to discover new optimal solutions. Consequently, RL opens up new avenues for achieving optimal solutions in resource management scenarios.

Contributions. This taxonomy offers a roadmap for further advancements in RL-based cloud resource management (CRM) methods. The contributions of this work are threefold: *(i)* We provide a comprehensive taxonomy that categorizes the literature based on the RL methods deployed, action space, and reward function used. *(ii)* We survey a wide range of RL-based techniques used for CRM, justify their effectiveness, and discuss their limitations. *(iii)* We identify open problems and future research directions in RL-based CRM.

2 Background

Before introducing the proposed taxonomy, we shortly discuss the required background for understanding the metrics we employed to categorize the state-of-the-art.

CRM. Cloud computing has transformed the way organizations utilize and manage their resources. In particular, the development of virtual machines (VMs) and containers have revolutionized the infrastructure models offered by cloud service providers (e.g., Infrastructure as a Service provides). These technologies enable sharing of physical resources among multiple entities while ensuring efficient utilization and isolation. VMs and containers operate as self-contained units, encapsulating the required software.

Infrastructure as a Service (IaaS) providers gain revenue by providing access to their resources while minimizing their operational costs and not violating service-level agreements (SLAs) by employing load-balancing algorithms, capacity planning strategies, and resource optimization techniques.In contrast, IaaS customers prioritize cost-effectiveness while maintaining application performance. They evaluate pricing models, rental costs, storage expenses, and associated fees to find economical options while meeting QoS requirements. By assessing performance metrics like response time and throughput, they seek solutions that strike a balance between cost savings and performance targets.

Notably, although both parties strive for a balance between performance and cost, they evaluate and measure these aspects differently. IaaS providers emphasize system-level performance, energy efficiency, and meeting SLA commitments, whereas IaaS customers focus on application-level performance, cost optimization, and aligning the services with their specific QoS requirements.

Reinforcement Learning. RL provides a framework for an agent to learn optimal resource allocation policies through trial-and-error interactions directly with the system. RL algorithms, such as Q-learning and policy gradient methods, have demonstrated their effectiveness in various domains. Cloud service providers can leverage the adaptability, learning capabilities, and dynamic decision-making offered by RL agents to address the limitations of existing methods for optimizing resource allocation, improving system performance, and reducing energy consumption.

The RL paradigm involves two major components: the *agent* and the *environment*. The environment represents the decision-making problem that the agent needs to solve and is usually modeled as a Markov decision process [56]. It captures the dynamics of the system by defining the *state space*, which represents all possible configurations of the environment and its transitions at any given time. The agent interacts with the environment by selecting actions from an *action space*, a set of all possible actions the agent can take. The agent's goal is to learn a *policy* that maps states to actions, directing its decision-making process. The performance of the agent is evaluated based on a *reward* signal provided by the environment. The reward serves as a feedback mechanism to guide the agent towards desirable behaviors. It quantifies the immediate desirability or utility of being in a particular state or taking a specific action. The reward can be defined based on various factors and objectives via an RL method (e.g., value-based, policy-based, or actor-critic).

A common categorization in RL is the distinction between model-based and model-free methods. In model-based reinforcement learning, an explicit model of the environment is created, allowing the agent to simulate and plan ahead. In contrast, in model-free methods, the agent directly learns a policy or value function without relying on an explicit model of the environment. Most techniques found in the literature, primarily employ model-free approaches.

RL methods. Value-based methods, focus on estimating a value function to determine the optimal action. They may struggle, however, with high-dimensional state spaces, where the number of possible states becomes prohibitively large. Despite this limitation, value-based methods are a powerful approach to learning optimal policies based on value estimates. Policy-based methods directly parameterize or represent the policy function, determining the action selection at each state. However, these methods often encounter the challenge of high variance in the estimated state value, which can negatively impact the stability and convergence of the training processes. Despite this challenge, policy-based methods offer a direct and flexible approach to learning optimal policies. Actor-critic methods leverage the strengths of both value-based and policy-based approaches by incorporating two distinct components: an actor and a critic. The actor learns the policy and makes action selections based on the current policy, while the critic estimates the reward function and provides feedback on the quality of the chosen actions. This combination allows for more efficient learning and improved decision-making in dynamic environments. These methods are beneficial in continuous environments where actions are represented as real values by amortizing the variance in the state value estimations.

RL Method	References
Value-based	[1,2,4,5,7–13,15,16,18–22,30,31,33,38–40,45–51,53,57,58,60–62,64,65,67,71,72,75–77,79,80]
Policy-based	[24,34–36,42–44,54,74]
Actor-critic	[3,14,17,29,37,63,73,78]

Table 1. RL methods in the CRM literature

Table 1 provides a categorization of the RL methods in the literature. By utilizing value estimation and policy optimization together, actor-critic methods offer a balanced and effective approach to reinforcement learning. The choice of which RL approach to use depends on various factors, such as the complexity of the problem and the available data.

Deep Reinforcement Learning. Deep Reinforcement Learning (DRL) has significantly boosted the performance of RL methods by leveraging the capabilities of deep learning. DRL has been successful in complex and dynamic environments where traditional RL approaches faced challenges with scalability. By employing deep learning models, DRL agents benefit from powerful function approximations and the ability to handle large state-action spaces. One advantage of DRL is its capacity for representation learning, allowing agents to extract meaningful features from raw sensory data and capture underlying patterns and structures. However, it is important to acknowledge the challenges and costs associated with DRL. Computationally, DRL models can be resource-intensive. Moreover, the complexity introduced by DRL can reduce interpretability, making it challenging to understand the decision-making processes.

3 RL-based techniques for CRM

In this section, we discuss and categorize the RL-based techniques for CRM. Most of the proposed techniques are tailor-made variations of RL models addressing specific resource management problems. They often incorporate other techniques, such as queuing methods, or machine learning models for handling the workload prediction [12, 21, 39], or anomaly detection [30, 49], to enhance the intelligence and effectiveness of the approach.

An approach well-suited to cloud environments that require intelligence at multiple levels is hierarchical reinforcement learning (HRL), where multiple RL agents are deployed at different levels of the system [30,39]. For instance, one RL agent may be responsible for making high-level decisions, such as selecting the cluster for job scheduling, while another RL agent within the cluster decides the exact node for job allocation [38]. HRL is closely related to the concept of Semi-Markov Decision Processes in RL that allows for actions to persist for a duration of time, enabling the modeling of higher-level actions and subtasks. HRL leverages this idea by decomposing complex tasks into a hierarchy of subtasks, each governed by its own RL agent. This hierarchical structure helps in managing computational complexity and enables agents to focus on specific levels of decision-making, leading to more efficient resource management strategies.

In contrast, multi-agent reinforcement learning (MARL) techniques involve multiple RL agents operating on the same task. Each agent may have a different objective, but collectively, they handle the multi-objective problem of resource management in an intelligent collaborative manner. Unlike hierarchical RL, where actions are taken sequentially, MARL allows for simultaneous action selection, reward observation, and coordination among agents [2, 4, 8, 16, 64, 65].

Different approaches have been employed to handle the challenge of large state and action spaces. Some works discretize the state and action space to reduce the dimensionality, making the problem more manageable [58]. However, this discretization may introduce noise and lose the continuous nature of the problem. In contrast, others preserve the continuity of the problem, arguing that it reflects the dynamic and continuous reality of the cloud environment [3, 14].

Additionally, there are emerging approaches that incorporate meta-reinforcement learning (Meta-RL) to tackle the dynamic changes in the cloud environment more effectively. Meta-RL techniques focus on learning how to learn, enabling the RL agent to adapt quickly to new and unseen situations. This adaptive capability proves beneficial in resource management scenarios where the environment undergoes frequent fluctuations and variations [69].

State space of RL agents. The state space in RL-based resource management approaches captures the current system state, including server resource utilization and pending job descriptions. It serves as a monitoring system, enabling the RL agent to observe and make informed decisions. Typical state space metrics include CPU and memory utilization, network bandwidth, and storage availability. The design of the state space is tailored to the agent's requirements and the specific problem, allowing for variations. While the core elements of the state space remain consistent, these variations reflect the diverse objectives, constraints, and operational characteristics of the systems being addressed. In certain resource management scenarios, the agent may lack complete visibility into the system state [42]. This challenge is addressed by employing partially observable Markov decision processes, where the agent receives partial observations rather than complete information about the state. The introduction of partial observations adds complexity and poses challenges for decision-making, as the agent must contend with the inherent uncertainty in the observed state.

Action	References
Task-to-resource mapping	[1–3, 7, 11, 13–19, 24, 28, 29, 31, 33–40, 42–45, 47, 48, 50, 54, 60–63, 65, 68, 69, 72–74, 77, 78]
Horizontal scaling	[4, 8, 9, 12, 20, 30, 49, 58]
Vertical scaling	[10, 30, 49, 51, 80]
Power management	[39, 50, 57, 76]
VM consolidation and load balancing	[5, 21, 22, 26, 32, 46, 48, 63, 64, 70, 75, 77]

Table 2. RL action spaces in the CRM literature.

Action space of RL agents. We organize the actions of RL-based CRM, into task-to-resource mapping, horizontal scaling, and load balancing. IaaS providers, with access to physical servers, can also perform VM consolidation, vertical scaling, and power management. Table 2 presents a taxonomy of various actions identified in the literature.

Task-to-resource mapping involves mapping or allocating tasks or VMs that are awaiting execution, without involving any migration of running jobs, to specific physical resources or servers. The goal is to efficiently assign tasks to available resources based on criteria like load balancing, resource utilization, or minimizing SLA violations. *VM consolidation & load balancing* on the other hand, involves migrating or consolidating tasks or VMs from underutilized or over-utilized resources to achieve better resource utilization, reduce energy consumption, and improve system performance. Notably, although most works in the literature focus on finding which task to schedule and where to place it, there are some approaches that only focus on the selection of which task to schedule. These approaches assume a global resource pool where task placement is not a concern [24, 42, 74].

Horizontal scaling refers to dynamically increasing or decreasing the number of instances or replicas of a task to handle varying workloads to maintain performance, meet demand, or optimize resource usage. *Vertical scaling*, on the other hand, involves adjusting the capacity or configuration of individual resources assigned to a task or VM.

Power management focuses on optimizing energy consumption in data centers by dynamically powering off underutilized resources or reducing power consumption during low-demand periods. While most works in the literature primarily focus on simple techniques such as turning resources ON or OFF, there are works that introduce more advanced strategies. Some studies explore adjusting the airflow level as a means to optimize energy consumption [50], while others incorporate dynamic voltage and frequency scaling techniques into the decision-making process [79]. These ap-

Reward Parameter	References
Performance metrics	[2,3,5,8,10–12,14,16–19,24,29, 30,34,37–39,42,43,45,47–51,53, 54,58,60–63,65,67,72–75,78]
SLA	[1,4,5,10,20,21,35,49,58,63, 64,70,77,81]
VM price cost	[4,9,12,13,15,20,28,31,33,40, 46,65,71,80]
Energy cost	[2,5,11,21,29,38,39,48,50,57, 63,64,67,72,73,76,77,79]
System metrics	[17,18,22,30,44,49,60,69,70]

Table 3. RL reward parameters in the CRM literature.

proaches aim to achieve finer-grained control over resource utilization and enhance energy efficiency in data centers.

Reward functions of RL agents. The reward function in an RL model is constructed by integrating a set of parameters that collectively define the optimization goal. In Table 3, we summarize various reward parameters commonly employed in the literature. A reward parameter represents a specific component that contributes to the overall reward function.

Performance metrics refer to individual tasks or jobs within the system. They include QoS metrics such as latency, average waiting time, throughput, etc. *System metrics*, on the other hand, focus on the overall resource utilization and workload distribution in the system. These metrics are related to the utilization levels of different resources, such as CPU, memory, and network bandwidth, as well as the balance or imbalance of workload among servers. *Energy costs* are integrated as a reward parameter to incentive the selection of energy-efficient resource allocation strategies, leading to reduced power consumption. *SLAs* are incorporated as a reward parameter to ensure that resource management decisions align with the agreed-upon service commitments. *VM price* represents the costs associated with provisioning and utilizing virtual machines. By considering VM price cost in the reward function, resource management algorithms can optimize resource allocation while minimizing costs.

Exploration strategies of RL agents. Most works in the literature utilize the ε-greedy exploration strategy. This simple yet effective approach balances the exploration vs exploitation trade-off by selecting the action with the highest estimated value most of the time (exploitation), while occasionally choosing a random action (exploration) to discover new possibilities and avoid getting stuck in sub-optimal solutions.

4 Open problems

Despite significant progress in RL-based CRM, there are still several open problems. Below we identify six of them.

1) RL in heterogeneous computing environments. One open challenge is to explore the placement and migration of application components in diverse, heterogeneous environments. Application components are parts that make up an application, which may need to run on different types of computing nodes, including smart edge nodes, edge data centers, and cloud data centers. This arrangement across various layers creates a dynamic, interconnected environment, the cloud-edge continuum. Existing literature focuses on small-scale testbeds or simulations, limiting the understanding of how RL models can be effectively deployed in such complex, real-world environments. Therefore, there is a significant knowledge gap related to the deployment of RL techniques in such evolving environments.

2) Computer vision techniques for CRM. Integrating RL models with computer vision for CRM is a highly promising research area. One possible direction is to provide visual representations to RL agents, resembling heatmaps, that capture the resource usage status of the system. This visual snapshot highlights system areas experiencing high demand. By using computer vision algorithms, the RL agent can learn abstract features to better optimize system resources.

3) Model-based RL for CRM. Previous works have faced challenges in utilizing model-based techniques, due to the complexity of creating accurate models of dynamic cloud environments. However, with the emergence of transformer architectures and generative models, there is an opportunity to explore more efficient ways of modeling such environments [25]. By creating models based on past system experiences, the complex and dynamic environment can be framed in a way that facilitates faster adaptation and convergence, enabling improved performance and resource optimization.

4) XAI techniques for RL agents in CRM. To the best of our knowledge, there is no work in the literature that incorporates explainability into RL-based systems in CRM. As DRL models get more complex, it is crucial to avoid treating them as black boxes. Understanding how these models make decisions reduces concerns over errors or biases caused by limited comprehension. Transparency helps build trust and understanding and aids in system debugging by identifying and analyzing potential issues. We envision XAI-assisted RL agents to be associated with SLA monitoring.

5) Model drifting of RL agents. Limited research exists to detect performance degradation in the behavior of RL agents due to long-term changes in the highly dynamic cloud systems. Techniques like continual RL and meta-RL have the potential to resolve *model drifting* with efficient retraining.

6) Generalization of RL agents. Lastly, Traditional RL relies on static reward functions to drive agent behavior. By employing dynamic reward functions through *reward machines*, agents can adapt and generalize across diverse scenarios, dynamically changing their behavior to achieve specific goals, based on system needs.

5 Conclusion

In this work, we discuss state-of-the-art methods for addressing CRM using RL-based agents. We provide a review of the solved problems and common methods employed. Also, we identify six challenges that require further investigation and have promising directions for future research.

6 Acknowledgements

This work has received funding from the Horizon Europe research and innovation program of the European Union, under grant agreement no 101092912, project MLSysOps.

7 References

[1] A. Alsarhan, A. Itradat, A. Y. Al-Dubai, A. Y. Zomaya, and G. Min. Adaptive resource allocation and provisioning in multi-service cloud environments. *IEEE Transactions on Parallel and Distributed Systems*, 29(1):31–42, 2017.

[2] A. Asghari, M. K. Sohrabi, and F. Yaghmaee. Online scheduling of dependent tasks of cloud's workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents. *Soft Computing*, 24:16177–16199, 2020.

[3] Y. Bao, Y. Peng, and C. Wu. Deep learning-based job placement in distributed machine learning clusters. In *IEEE INFOCOM-IEEE conference on computer communications*, pages 505–513, 2019.

[4] E. Barrett, E. Howley, and J. Duggan. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Concurrency and computation: practice and experience*, 25(12):1656–1674, 2013.

[5] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan. Learn-as-you-go with megh: Efficient live migration of virtual machines. *IEEE Trans. on Parallel and Distributed Systems*, 30(8):1786–1801, 2019.

[6] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012.

[7] R. Bianchini, M. Fontoura, E. Cortez, A. Bonde, A. Muzio, A.-M. Constantin, T. Moscibroda, G. Magalhaes, G. Bablani, and M. Russinovich. Toward ml-centric cloud platforms. *Communications of the ACM*, 63(2):50–59, 2020.

[8] J. Bibal Benifa and D. Dejey. Rlpas: Reinforcement learning-based proactive auto-scaler for resource provisioning in cloud environment. *Mobile Networks and Applications*, 24:1348–1363, 2019.

[9] C. Bitsakos, I. Konstantinou, and N. Koziris. Derp: A deep reinforcement learning cloud system for elastic resource provisioning. In *2018 IEEE international conference on cloud computing technology and science (CloudCom)*, pages 21–29. IEEE, 2018.

[10] X. Bu, J. Rao, and C.-Z. Xu. Coordinated self-configuration of virtual machines and appliances using a model-free learning approach. *IEEE Trans. on parallel and distributed systems*, 24(4):681–690, 2012.

[11] L. Caviglione, M. Gaggero, M. Paolucci, and R. Ronco. Deep reinforcement learning for multi-objective placement of virtual machines in cloud datacenters. *Soft Computing*, 25(19):12569–12588, 2021.

[12] X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong. Resource allocation with workload-time windows for cloud-based software services: a deep reinforcement learning approach. *IEEE Transactions on Cloud Computing*, 2022.

[13] X. Chen, F. Zhu, Z. Chen, G. Min, X. Zheng, and C. Rong. Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Transactions on Cloud Computing*, 10(2):1117–1129, 2020.

[14] Z. Chen, J. Hu, and G. Min. Learning-based resource allocation in cloud data center using advantage actor-critic. In *2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[15] F. Cheng, Y. Huang, B. Tanpure, P. Sawalani, L. Cheng, and C. Liu. Cost-aware job scheduling for cloud instances using deep reinforcement learning. *Cluster Computing*, pages 1–13, 2022.

[16] D. Cui, Z. Peng, J. Xiong, B. Xu, and W. Lin. A reinforcement learning-based mixed job scheduler scheme for grid or iaas cloud. *IEEE Transactions on Cloud Computing*, 8(4):1030–1039, 2017.

[17] X. Deng, J. Zhang, H. Zhang, and P. Jiang. Deep reinforcement learning-based resource allocation for cloud gaming via edge computing. *IEEE Internet of Things Journal*, 2022.

[18] D. Ding, X. Fan, Y. Zhao, K. Kang, Q. Yin, and J. Zeng. Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Generation Computer Systems*, 108:361–371, 2020.

[19] T. Dong, F. Xue, C. Xiao, and J. Li. Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. *Concurrency and Computation: Practice and Experience*, 32(11):e5654, 2020.

[20] X. Dutreilh, S. Kirgizov, O. Melekhova, J. Malenfant, N. Rivierre, and I. Truck. Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow. In *ICAS 2011, The Seventh International Conference on Autonomic and Autonomous Systems*, pages 67–74, 2011.

[21] F. Farahnakian, P. Liljeberg, and J. Plosila. Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning. In *22nd Euromicro Int. Conference on Parallel, Distributed, and Network-Based Processing*, pages 500–507. IEEE, 2014.

[22] A. Ghasemi and A. Toroghi Haghighat. A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning. *Computing*, 102:2049–2072, 2020.

[23] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella. Multi-resource packing for cluster schedulers. *ACM SIGCOMM Computer Communication Review*, 44(4):455–466, 2014.

[24] W. Guo, W. Tian, Y. Ye, L. Xu, and K. Wu. Cloud resource scheduling with deep reinforcement learning and imitation learning. *IEEE Internet of Things Journal*, 8(5):3576–3586, 2020.

[25] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[26] O. Houidi, D. Zeghlache, V. Perrier, P. T. A. Quang, N. Huin, J. Leguay, and P. Medagliani. Constrained deep reinforcement learning for smart load balancing. In *Annual Consumer Communications & Networking Conference (CCNC)*, pages 207–215. IEEE, 2022.

[27] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain. Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges. *IEEE communications surveys & tutorials*, 22(2):1251–1275, 2020.

[28] G. Ismayilov and H. R. Topcuoglu. Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing. *Future Generation computer systems*, 102:307–322, 2020.

[29] J. Jin and Y. Xu. Optimal policy characterization enhanced proximal policy optimization for multitask scheduling in cloud computing. *IEEE Internet of Things Journal*, 9(9):6418–6433, 2021.

[30] S. Kardani-Moghaddam, R. Buyya, and K. Ramamohanarao. Adrl: A hybrid anomaly-aware deep reinforcement learning-based resource scaling in clouds. *IEEE Transactions on Parallel and Distributed Systems*, 32(3):514–526, 2020.

[31] K. Karthiban and J. S. Raj. An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm. *Soft Computing*, 24(19):14933–14942, 2020.

[32] P. R. Kaveri and P. Lahande. Reinforcement learning to improve resource scheduling and load balancing in cloud computing. *SN Computer Science*, 4(2):188, 2023.

[33] A. Kontarinis, V. Kantere, and N. Koziris. Cloud resource allocation from the user perspective: A bare-bones reinforcement learning approach. In *Web Information Systems Engineering–WISE 2016: 17th International Conference, Shanghai, China, November 8-10, 2016, Proceedings, Part I 17*, pages 457–469. Springer, 2016.

[34] C. Li, W. Gao, L. Shi, Z. Shang, and S. Zhang. Task scheduling based on adaptive priority experience replay on cloud platforms. *Electronics*, 12(6):1358, 2023.

[35] F. Li and B. Hu. Deepjs: Job scheduling based on deep reinforcement learning in cloud data center. In *Proceedings of the 4th International Conference on Big Data and Computing*, pages 48–53, 2019.

[36] J. Li, X. Zhang, J. Wei, Z. Ji, and Z. Wei. Garlsched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems. *Future Generation Computer Systems*, 135:259–269, 2022.

[37] S. Liang, Z. Yang, F. Jin, and Y. Chen. Data centers job scheduling with deep reinforcement learning. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14*, pages 906–917. Springer, 2020.

[38] J. Lin, D. Cui, Z. Peng, Q. Li, and J. He. A two-stage framework for the multi-user multi-data center job scheduling and resource allocation. *IEEE Access*, 8:197863–197874, 2020.

[39] N. Liu, Z. Li, J. Xu, Z. Xu, S. Lin, Q. Qiu, J. Tang, and Y. Wang. 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)*, pages 372–382. IEEE, 2017.

[40] K. Lolos, I. Konstantinou, V. Kantere, and N. Koziris. Adaptive state space partitioning of markov decision processes for elastic resource

management. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 191–194. IEEE, 2017.

[41] E. Makridis, K. Deliparaschos, E. Kalyvianaki, A. Zolotas, and T. Charalambous. Robust dynamic cpu resource provisioning in virtualized servers. *IEEE Trans. on Services Computing*, 15(2):956–969, 2020.

[42] H. Mao, M. Alizadeh, I. Menache, and S. Kandula. Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM workshop on hot topics in networks*, pages 50–56, 2016.

[43] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng, and M. Alizadeh. Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM SIGCOMM*, pages 270–288, 2019.

[44] S. S. Mondal, N. Sheoran, and S. Mitra. Scheduling of time-varying workloads using reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

[45] S. Mostafavi and V. Hakami. A stochastic approximation approach for foresighted task scheduling in cloud computing. *Wireless Personal Communications*, 114:901–925, 2020.

[46] S. M. R. Nouri, H. Li, S. Venugopal, W. Guo, M. He, and W. Tian. Autonomic decentralized elasticity based on a reinforcement learning controller for cloud applications. *Future Generation Computer Systems*, 94:765–780, 2019.

[47] Z. Peng, D. Cui, J. Zuo, Q. Li, B. Xu, and W. Lin. Random task scheduling scheme based on reinforcement learning in cloud computing. *Cluster computing*, 18:1595–1607, 2015.

[48] A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser, and A. W. Mohamed. Intelligent decision-making of load balancing using deep reinforcement learning and parallel pso in cloud environment. *IEEE Access*, 10:76939–76952, 2022.

[49] H. Qiu, S. S. Banerjee, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer. Firm: An intelligent fine-grained resource management framework for slo-oriented microservices. In *Proceedings of The 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2020.

[50] Y. Ran, H. Hu, X. Zhou, and Y. Wen. Deepee: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 645–655. IEEE, 2019.

[51] J. Rao, X. Bu, C.-Z. Xu, L. Wang, and G. Yin. Vconf: a reinforcement learning approach to virtual machines auto-configuration. In *Proceedings of the 6th international conference on Autonomic computing*, pages 137–146, 2009.

[52] K. Rzadca, D. Trystram, and A. Wierzbicki. Fair game-theoretic resource management in dedicated grids. In *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*, pages 343–350. IEEE, 2007.

[53] L. Schuler, S. Jamil, and N. Kıdiduhl. Ai-based resource allocation: Reinforcement learning for adaptive auto-scaling in serverless environments. In *IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 804–811, 2021.

[54] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao. Deep reinforcement learning-based task scheduling in iot edge computing. *Sensors*, 21(5):1666, 2021.

[55] P. Singh, M. Dutta, and N. Aggarwal. A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems*, 52:1–51, 2017.

[56] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[57] G. Tesauro, R. Das, H. Chan, J. Kephart, D. Levine, F. Rawson, and C. Lefurgy. Managing power consumption and performance of computing systems using reinforcement learning. *Advances in neural information processing systems*, 20, 2007.

[58] G. Tesauro et al. Online resource allocation using decompositional reinforcement learning. In *AAAI*, volume 5, pages 886–891, 2005.

[59] R. Tolosana-Calasanz, J. Diaz-Montes, O. F. Rana, and M. Parashar. Feedback-control & queueing theory-based resource management for streaming applications. *IEEE Transactions on parallel and distributed systems*, 28(4):1061–1075, 2016.

[60] Z. Tong, H. Chen, X. Deng, K. Li, and K. Li. A scheduling scheme in the cloud computing environment using deep q-learning. *Information Sciences*, 512:1170–1191, 2020.

[61] Z. Tong, X. Deng, H. Chen, J. Mei, and H. Liu. Ql-heft: a novel machine learning scheduling scheme base on cloud computing environment. *Neural Computing and Applications*, 32:5553–5570, 2020.

[62] Z. Tong, F. Ye, B. Liu, J. Cai, and J. Mei. Ddqn-ts: A novel bi-objective intelligent scheduling algorithm in the cloud environment. *Neurocomputing*, 455:419–430, 2021.

[63] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya. Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE transactions on mobile computing*, 21(3):940–954, 2020.

[64] B. Wang, F. Liu, and W. Lin. Energy-efficient vm scheduling based on deep reinforcement learning. *Future Generation Computer Systems*, 125:616–628, 2021.

[65] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie. Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning. *IEEE access*, 7:39974–39982, 2019.

[66] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *The journal of supercomputing*, 54:252–269, 2010.

[67] Y. Wei, L. Pan, S. Liu, L. Wu, and X. Meng. Drl-scheduling: An intelligent qos-aware job scheduling framework for applications in clouds. *IEEE Access*, 6:55112–55125, 2018.

[68] Y. Wei, F. R. Yu, M. Song, and Z. Han. Joint optimization of caching, computing, and radio resources for fog-enabled iot using natural actor–critic deep reinforcement learning. *IEEE Internet of Things Journal*, 6(2):2061–2073, 2018.

[69] X. Xiu, J. Li, Y. Long, and W. Wu. Mrlcc: an adaptive cloud task scheduling method based on meta reinforcement learning. *Journal of Cloud Computing*, 12(1):1–12, 2023.

[70] C.-Z. Xu, J. Rao, and X. Bu. Url: A unified reinforcement learning approach for autonomic cloud management. *Journal of Parallel and Distributed Computing*, 72(2):95–105, 2012.

[71] Z. Xu, Z. Zhong, and B. Shi. Deep reinforcement learning based resource allocation strategy in cloud-edge computing system. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[72] J. Yan, Y. Huang, A. Gupta, A. Gupta, C. Liu, J. Li, and L. Cheng. Energy-aware systems for real-time job scheduling in cloud data centers: A deep reinforcement learning approach. *Computers and Electrical Engineering*, 99:107688, 2022.

[73] Y. Yang and H. Shen. Deep reinforcement learning enhanced greedy optimization for online scheduling of batched tasks in cloud hpc systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):3003–3014, 2021.

[74] Y. Ye, X. Ren, J. Wang, L. Xu, W. Guo, W. Huang, and W. Tian. A new approach for resource scheduling with deep reinforcement learning. *arXiv preprint arXiv:1806.08122*, 2018.

[75] C. Ying, B. Li, X. Ke, and L. Guo. Raven: Scheduling virtual machine migration during datacenter upgrades with reinforcement learning. *Mobile Networks and Applications*, 27(1):303–314, 2022.

[76] H. Yu and M. Tong. An energy optimization algorithm for data centers based on deep q-learning with multi-source energy. In *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pages 382–388. IEEE, 2022.

[77] J. Zeng, D. Ding, K. Kang, H. Xie, and Q. Yin. Adaptive drl-based virtual machine consolidation in energy-efficient cloud data center. *IEEE Tran. on Parallel and Distributed Systems*, 33(11):2991–3002, 2022.

[78] D. Zhang, D. Dai, Y. He, F. S. Bao, and B. Xie. Rlscheduler: an automated hpc batch job scheduler using reinforcement learning. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.

[79] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, and P. Li. Energy-efficient scheduling for real-time systems based on deep q-learning model. *IEEE transactions on sustainable computing*, 4(1):132–141, 2017.

[80] Y. Zhang, J. Yao, and H. Guan. Intelligent cloud resource management with deep reinforcement learning. *IEEE Cloud Computing*, 4(6):60–69, 2017.

[81] T. Zheng, J. Wan, J. Zhang, and C. Jiang. Deep reinforcement learning-based workload scheduling for edge computing. *Journal of Cloud Computing*, 11(1):3, 2022.