JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Advancements in Reinforcement Learning for Task and Resource Scheduling in Cloud Computing: A Comprehensive Survey

[1]**Joseph R**
[2]Student, MTech in Artificial Intelligence
[1]Reva Academy for Corporate Excellence (RACE),
[1]REVA University, Bengaluru, India

Abstract: The rapid evolution of cloud computing and the advent of quantum cloud environments have introduced new challenges and opportunities for task scheduling and resource management. This paper presents a comprehensive study of innovative approaches to scheduling and optimization in these dynamic environments. We address five key areas: (1) task scheduling in cloud computing using a novel Q-learning-based framework, (2) application of Deep Reinforcement Learning (DRL) for cloud resource scheduling, (3) online scheduling of deferrable jobs to minimize delays, (4) task placement in quantum cloud computing using DRL techniques, and (5) cost optimization for workflow scheduling with spot and on-demand instances. Each section includes a detailed mathematical formulation that supports the proposed models and methods, highlighting their efficiency and effectiveness in addressing real-world problems. Each exploring advanced methodologies for task scheduling, resource management, and optimization in cloud computing environments. The research spans various innovative applications of reinforcement learning (RL) and deep reinforcement learning (DRL) to tackle challenges posed by dynamic workloads, resource constraints, and emerging technologies like quantum computing. Each paper introduces unique frameworks, algorithms, and experimental insights, contributing to the evolving landscape of cloud computing.

***IndexTerms* - Reinforcement Learning (RL), Task Scheduling, Deep Reinforcement Learning (DRL), Resource Management, Cloud Resource Allocation, Deep Q-Network (DQN)**

I. TASK SCHEDULING IN CLOUD ENVIRONMENTS

1.1 Introduction: Cloud computing environments present challenges in task scheduling due to their dynamic nature and resource heterogeneity. This paper introduces a Q-learning-based Multi-Task Scheduling Framework (QMTSF), designed to optimize task allocation at the server and virtual machine (VM) levels. By integrating the Unified Q-learning Reinforcement Learning (UQRL) algorithm, the framework adapts to changing workloads while minimizing makespan and task processing times.

1.2 Methodology:

The QMTSF operates in two stages:

1. Server Allocation: Tasks are allocated to cloud servers based on server type and availability.

2. VM Scheduling: An enhanced Q-learning algorithm (UQRL) refines task scheduling at the VM level.

The Q-value table, iteratively updated using the Bellman equation, guides decision-making:

Here, represents the Q-value for a state-action pair, is the learning rate, is the reward, and is the discount factor. The framework demonstrated significant reductions in makespan and task processing times compared to traditional algorithms such as round-robin and particle swarm optimization. The complexity of task scheduling in cloud environments has prompted the need for more adaptive and intelligent systems. In this research, a novel Q-learning-based Multi-Task Scheduling Framework (QMTSF) is introduced to address the dynamic nature of cloud computing tasks and resources. The framework operates in two stages. In the first stage, tasks are allocated to cloud servers based on server type and availability, while the second stage implements an enhanced Q-learning algorithm, known as UQRL (Unified Q-learning Reinforcement Learning). UQRL refines task scheduling at the virtual machine (VM) level, ensuring that tasks are allocated to specific VMs in a manner that minimizes makespan and reduces task processing time.

1.3 Mathematical insight:

The decision-making process is guided by a Q-Value table, which is iteratively updated using the Bellman equation. This table stores accumulated rewards and is formulated as follows:

$$Q(s,a)\leftarrow Q(s,a)+\alpha[R+\gamma a' \max Q(s',a')-Q(s,a)]$$

Here, Q(s,a) Q(s, a) Q(s,a) represents the Q-value for state-action pair (s,a) (s, a) (s,a), α is the learning rate, R is the reward, and γ is the discount factor. The framework enables the system to adapt to changing workloads by constantly updating the Q-values and selecting actions that lead to the best long-term outcomes. When compared to traditional algorithms such as round-robin and particle swarm optimization (PSO), the Q-learning approach demonstrated a notable reduction in total makespan and average task processing time, making it highly effective in high-load cloud environments.

Conclusion:

QMTSF demonstrates that reinforcement learning can provide intelligent, adaptive scheduling solutions for cloud environments, offering improvements in efficiency and scalability. Future work may explore incorporating other machine learning techniques for further optimization.

II. Application of Deep Reinforcement Learning for Resource Scheduling

2.1 Introduction: This study explores Deep Reinforcement Learning (DRL) as a solution for resource scheduling in cloud environments. By balancing task allocation across multiple resource dimensions, the DRL-based model adapts to dynamic changes and achieves better performance than heuristic and meta-heuristic approaches.

Resource scheduling in cloud computing is complex due to the need for balancing resource utilization while minimizing task delays. Traditional approaches such as heuristic and hybrid algorithms often fail to account for the scale and variability of cloud systems. DRL, with its capacity to adapt and optimize, is proposed as a superior alternative.

In cloud environments, resource scheduling is critical for ensuring optimal performance and resource utilization, especially when managing complex multi-dimensional tasks. This research introduces a Deep Reinforcement Learning (DRL)-based Resource Scheduling Model that dynamically allocates tasks to server nodes based on resource availability across multiple dimensions, such as CPU, memory, and disk space.

The key challenge lies in balancing the load across servers while minimizing task completion times. To address this, a scheduling framework is developed using DRL, where the system learns and adapts to real-time changes in the resource environment. The model takes into account the parameters matrix For classic approaches, the most commonly utilized methods in surveyed literature are heuristic, meta-heuristic and hybrid algorithms. Thus, we mainly review these three types of algorithms to assist the later review and discussion on the application of DRL in Cloud scheduling.

1) heuristic:

Heuristic is an algorithm to solve an optimization problem based on intuitionistic or empirical construction. Due to their lower complexity, heuristic algorithms are prevalent in some scenarios with a clear evaluation function requiring rapidity but not requiring high optimization results. Additionally, the worst-case of heuristic algorithms is generally predictable hence with a lower risk of improper allocation

2) meta-heuristic:
Meta-heuristic algorithms blend heuristic techniques with randomization to solve optimization problems. Key examples include Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and others. While meta-heuristic algorithms are more flexible and capable of handling complex problems compared to traditional heuristics, they also introduce higher computational complexity and randomness. Despite simplifying calculations with ideal assumptions, these algorithms face challenges in accurately reflecting real-world scenarios.

3) hybrid algorithms: Some other classic algorithms used in Cloud scheduling mainly contain DP, Random algorithms, and hybrid algorithms (combining two or more algorithms). Among them, hybrid algorithms are also widely used in solving complex scheduling problems in Cloud computing. Hybrid algorithms can combine the advantages of multiple algorithms to produce better solutions.

The use of machine learning (ML) in cloud scheduling includes various methods such as deep learning (DL), reinforcement learning (RL), and deep reinforcement learning (DRL). Additionally, other ML techniques like K-Nearest Neighbours (KNN) and Support Vector Machines (SVM) have also been applied. Key characteristics of cloud systems, such as their large scale, complexity, and the randomness of tasks, pose challenges that traditional algorithms often cannot address effectively.

2.2 Methodology:

The methodology involves developing a Deep Reinforcement Learning (DRL)-based framework for resource scheduling in cloud environments. The model dynamically allocates tasks to server nodes by considering resource availability across dimensions like CPU, memory, and disk space. It uses an agent-environment interaction paradigm, where the agent learns optimal scheduling strategies based on feedback (rewards) from its actions. A parameter matrix is employed to map task requirements and server capabilities, enabling effective decision-making. Compared to classical heuristic, meta-heuristic, and hybrid algorithms, DRL offers superior adaptability and performance in handling the complexity and dynamism of cloud scheduling.

2.3 Mathematical insight:

In DRL, the reward Rt at time t is maximized as:

$$R_t = \sum_{i=1}^{n} (\text{Utilization}(i) - \text{Delay}(i))$$

where n is the number of tasks, utilization reflects resource allocation, and delay penalizes inefficient scheduling.

2.4 Key Insights:

- Challenges in Cloud Scheduling: The complexity and dynamism of cloud environments require scheduling algorithms that can adapt quickly and handle multiple variables, making classical methods insufficient.
- Role of ML: ML methods can map optimization objectives and evaluate solution quality, enabling more effective scheduling strategies.

- DRL Advantages: DRL combines the strengths of DL and RL, allowing for complex modeling of systems and adaptability to various optimization objectives. It can process large-scale tasks and handle dynamic environments effectively.

- Evolution of RL and DRL: The RL framework focuses on the agent-environment interaction, where agents learn optimal strategies based on feedback (or rewards) from their actions. This feedback mechanism differs from traditional optimization methods that rely on evaluation functions.

Overall, the application of DRL in cloud scheduling represents a promising intersection of emerging technologies, showing superior performance in managing complex scheduling problems.

Conclusion:
The DRL-based scheduling model effectively balances resource utilization and minimizes task delays. It outperforms classical methods by leveraging dynamic adaptability. Future enhancements could include multi-cloud integrations and consideration of additional resource constraints

III. Reinforcement Learning Framework for Online Scheduling of Deferrable Workloads

3.1 Introduction:

OSDEC, an online scheduling framework based on deep reinforcement learning, enhances resource utilization and reduces user waiting times. By leveraging auxiliary tasks and state representations, the framework outperforms traditional methods in dynamic environments. The growing demand for cloud computing resources requires efficient scheduling mechanisms. OSDEC addresses this need by integrating auxiliary predictions into its RL framework, optimizing the scheduling of deferrable workloads. Unlike traditional methods, OSDEC utilizes pre-collected data to improve scheduling decisions.
Cloud computing has become a critical infrastructure for delivering shared computing resources over the Internet, allowing users to access virtual machines provided by cloud platforms like Amazon Web Services, Microsoft Azure, and Google Cloud. However, optimizing the utilization of these resources while maintaining high service levels remains a challenge, especially with growing customer demand. One strategy is the use of deferrable virtual machine (VM) products, which allow job requests to be scheduled for future execution during valley hours. Scheduling these jobs effectively is difficult due to dynamic online job submissions and the need to continuously update scheduling plans. Traditional heuristic methods, though commonly used, are not effective in such complex, evolving environments and often fail to leverage the potential of pre-collected job data. This has led researchers to explore reinforcement learning (RL) for online scheduling, but current RL methods mainly focus on real-time job scheduling without utilizing the pre-collected job information.
This paper introduces a novel solution called OSDEC (Online Scheduling for Deferrable jobs in Cloud), a reinforcement learning framework designed to handle the scheduling of deferrable jobs in cloud computing environments. OSDEC aims to optimize both resource utilization and job waiting time by incorporating pre-collected workloads and capacity data into the scheduling process. The framework is enhanced with auxiliary tasks to improve feature extraction and policy learning, further boosting performance. Extensive experiments on real-world public datasets validate the effectiveness of OSDEC, showing its superiority over traditional scheduling methods. The proposed approach not only provides an efficient way to manage online scheduling but also addresses the unique challenges posed by dynamic cloud environments, marking a significant advancement in cloud job scheduling methodologies.
The problem of deferrable job scheduling is modelled as an online decision-making problem where jobs dynamically arrive and need to be scheduled for execution. Each job request is represented by several parameters: resource capacity {ci}, duration {di}, earliest start time {ei}, latest start time {li}, and submission time {gi}. The goal is to decide at each time step which jobs to deploy without exceeding the available capacity, which is the total platform capacity minus the resources used by on-demand jobs. The decision variable {Xit} is binary, indicating whether job i is scheduled at time t. The objective is to maximize the revenue, which is the product of requested resources and duration {ri=cidi} while minimizing delays for job execution. The delay for each job {pi=ti-ei}.
The capacity violation at any time step t is denoted as Vt, which quantifies the risk of exceeding the available capacity. The scheduling problem is formulated as a maximization problem with the objective function:

$$\max \sum_{i=1}^{N} \sum_{t=1}^{T} X_{it} \cdot (r_i - \omega_1 p_i) - \omega_2 v_t$$

where ω1 and ω2 are the coefficients that penalize delays and capacity violations, respectively. The constraints ensure that each job is scheduled at most once, within its valid time window, and the total capacity at any time step does not exceed the available capacity. In the online setting, the decision-making process is modelled using reinforcement learning (RL), where the system learns a policy to maximize the expected reward through sequential job scheduling. The RL model is augmented with auxiliary tasks to improve learning efficiency and feature extraction, and the system continuously updates the policy based on real-time job arrivals and scheduling decisions.

3.2 Methodology

The problem is modelled as an online decision-making task with dynamically arriving jobs characterized by parameters such as capacity, duration, and time windows. Key components:

- **State Representation:** Divides jobs into past, current, and future categories.
- **Action Representation:** Uses confidence scores to prioritize jobs.
- **Reward Function:** Optimizes capacity utilization while minimizing delays.

OSDEC employs Proximal Policy Optimization (PPO) with auxiliary tasks for enhanced learning and feature extraction. Experiments on real-world datasets validated its effectiveness, demonstrating superior performance over traditional heuristic methods.

3.3 Reinforcement Learning Based Scheduling

3.3.1 State

At each time step t, the system's state is represented by the following:

- **B(t)**: Jobs that have been submitted at time t.
- **Ct**: Available capacity at time t.

The submitted jobs are divided into three subsets:

- **Bhis(t)**: Jobs from previous time steps that are still running at time t.
- **Bcur(t)**: Jobs being considered at time t, between their earliest and latest start times.
- **Bfut(t)**: Jobs submitted by time t but haven't reached their earliest start time yet.

Thus, the state st at time t is:

$$s_t = \{Bhis(t), Bcur(t), Bfut(t), C_t\}$$

Action (at): The action involves choosing which jobs from Bcur(t) to deploy. This is represented by a vector of binary values, indicating whether a job is scheduled at time t or not. To handle large action spaces, a continuous "confidence score" is used to prioritize jobs based on their scores.

Policy: The scheduling policy is modelled as a Gaussian distribution, where the mean (μt) and standard deviation (Σt) are learned by the network.

Reward: The reward function is designed to optimize capacity utilization and minimize delays. It includes penalties for time delays and capacity violations.

3.4 Model Training

The model is trained using Proximal Policy Optimization (PPO), which uses a surrogate loss function for efficient training. The training involves optimizing the policy and value networks by using the calculated reward.

3.5 Experiments and Performance Evaluation

Experiments were conducted on a Microsoft Azure dataset to validate the effectiveness of the OSDEC (Reinforcement Learning-based Scheduling) approach. The following were key findings:

- **Comparison with Competitors**: OSDEC outperforms classical heuristic methods (FIFO, SJF, Tetris) and other RL variants (REINFORCE, Pointer Net, etc.) in terms of utilization, reward, and time delay penalties.

- **Deferrable vs. Real-time Scheduling**: The deferrable setting, which allows for better planning by deferring jobs, yields higher utilization compared to real-time scheduling, where jobs must be deployed immediately upon submission. OSDEC significantly benefits from this setting.

3.6 Key Insights:

1. **State Representation**: Divides jobs into past, current, and future categories.
2. **Action Representation**: Uses confidence scores to determine which jobs to schedule.
3. **Policy**: Learned using a diagonal Gaussian distribution for each job's confidence score.
4. **Auxiliary Tasks**: Enhances state representation by predicting system properties like capacity and job duration.
5. **Training with PPO**: Uses a surrogate loss function to optimize both policy and value networks.
6. **Performance**: OSDEC outperforms other methods, particularly in deferrable settings, improving capacity utilization and reducing penalties.

3.7 Mathematical Insight:

The scheduling decision minimizes the delay D while optimizing capacity utilization U:

$$D = \sum_i (t_i - e_i), \quad U = \frac{\text{Resources Used}}{\text{Total Capacity}}$$

where t_i is the execution time and e_i is the earliest start time.

Conclusion:

This paper introduces OSDEC, a deep reinforcement learning method for online scheduling of deferrable jobs in cloud computing, leveraging a Transformer-based architecture and auxiliary prediction tasks to enhance scheduling decisions under variable capacities. Extensive experiments validate its effectiveness, with future plans to extend the framework to other scheduling problems and explore safe reinforcement learning for robust cloud deployments.

OSDEC demonstrates superior performance in deferrable job scheduling, achieving higher utilization and reduced delays. Future work could involve extending the approach to real-time settings and exploring safe RL techniques for robust deployments

IV. A DEEP REINFORCEMENT LEARNING-BASED TASK PLACEMENT FOR QUANTUM CLOUD COMPUTING

4.1 Introduction:

The paper introduces **DRLQ**, a novel Deep Reinforcement Learning (DRL) technique designed for efficient task placement in quantum cloud computing environments. Traditional heuristic methods struggle to adapt to the dynamic and heterogeneous nature of quantum computing resources, necessitating advanced solutions. DRLQ utilizes the **Deep Q-Network (DQN)** architecture, enhanced by the **Rainbow DQN** approach, to develop an adaptive strategy for optimizing task completion time and scheduling efficiency in quantum cloud systems. Extensive experiments conducted with the QSimPy simulation toolkit demonstrate that DRLQ improves task execution efficiency and reduces task rescheduling needs.

DRLQ utilizes Deep Reinforcement Learning to optimize task placement in quantum cloud environments. The framework leverages the Rainbow DQN approach, achieving significant reductions in task completion time and rescheduling needs. Task placement in quantum cloud computing is complicated by resource heterogeneity and dynamic requirements. DRLQ addresses this by modelling the problem as a Markov Decision Process (MDP) and utilizing advanced DRL techniques for optimal scheduling.

4.2 Key Contributions Include:

1. Developing one of the first DRL-based solutions for quantum task placement.

2. Demonstrating a 37.81%–72.93% reduction in task completion time and minimal task rescheduling compared to heuristics.

3. Highlighting a scalable framework to optimize quantum cloud resource management, paving the way for further exploration of quantum-specific factors like error rates and circuit transpilation.

The paper identifies gaps in existing literature, such as the absence of machine learning-based solutions for quantum resource management, and positions its work as a foundation for innovative task placement strategies in quantum cloud computing.

4.3 Mathematical Insight:

The reward R for task placement is calculated as:

$$R = \frac{1}{T}\sum_{t=1}^{T}(\text{Task Efficiency} - \text{Rescheduling Penalty})$$

where T is the total number of tasks.

Conclusion:
This study introduced the DRLQ framework, leveraging deep reinforcement learning for task placement in quantum cloud computing environments. The results demonstrated that DRLQ significantly outperforms heuristic approaches, showcasing its potential as a robust solution for quantum cloud resource management. As one of the pioneering efforts in quantum cloud resource management, this work emphasizes the need for further research in this domain.
DRLQ outperforms heuristic methods, offering a scalable and efficient solution for quantum task scheduling. Future research may explore additional quantum-specific factors, such as error rates and circuit transpilation, to enhance the framework.
,

V. A DEEP REINFORCEMENT LEARNING APPROACH FOR COST-OPTIMIZED WORKFLOW SCHEDULING

5.1 Introduction:

This paper addresses the challenge of cost optimization in cloud computing environments, specifically in workflow scheduling. The use of spot instances—discounted computing resources offered by cloud providers—can reduce costs but introduces uncertainties due to their vulnerability to interruptions and price fluctuations based on supply and demand. To overcome these challenges, the paper proposes a solution using Deep Reinforcement Learning (DRL) to develop an autonomous agent that schedules workflows efficiently by combining spot and on-demand instances. The solution is implemented in the Argo workflow engine, a popular open-source tool for executing industrial workflows. Experimental results show that the proposed DRL-based scheduling method outperforms current benchmarks, achieving cost savings while meeting business requirements.

The key challenge is to optimize scheduling strategies that balance cost savings with performance requirements. Spot instances, offered at a discount but subject to interruptions and price fluctuations, are a potential solution. However, effectively managing the balance between spot and on-demand instances is critical to avoid compromising performance.

Reinforcement Learning (RL), especially Deep Reinforcement Learning (DRL), is proposed as a method to address the complexities of workflow scheduling in dynamic, uncertain environments. DRL's ability to learn optimal actions in stochastic settings makes it ideal for the problem at hand. This paper introduces a novel hierarchical action space design for DRL, distinguishing between on-demand and spot instances, and incorporates a DRL framework with multiple actor networks guided by a single critic network. The framework is applied to the Argo workflow engine, an open-source container-native tool for managing workflows, enabling advanced scheduling policies for cost optimization.

5.2 Key contributions include:

- A DRL model for cost-optimized workflow scheduling using a mix of on-demand and spot instances.
- A hierarchical cluster organization and novel action selection process.
- A DRL framework with Proximal Policy Optimization (PPO) for learning efficient scheduling strategies.
- A fully integrated approach for deploying the DRL agent in a workflow engine, marking the first attempt at embedding an intelligent agent in this environment.

5.3 Methodology

The DRL framework uses hierarchical action space design and Proximal Policy Optimization (PPO) to balance cost and performance. Key insights include:

- **Spot vs. On-Demand Instances:** Efficient allocation of spot instances reduces costs without compromising performance.
- **Integration with Workflow Engine:** Enables advanced scheduling policies for real-world applications.

5.4 Mathematical Insight

The cost C is minimized as:

$$C = \sum_{i=1}^{n} (p_i \cdot d_i)$$

where pi is the price per unit time, and di is the duration of task i.

Conclusion:

By leveraging the intelligent allocation of spot and on-demand instances, the proposed DRL framework optimizes resource utilization and minimizes costs. The DRL-based scheduling method demonstrates superior performance in cost optimization, providing a robust solution for industrial workflows. Future work may focus on scalability and integration with other workflow engines

VI. CONCLUSION

This survey highlights the transformative role of reinforcement learning and deep reinforcement learning in addressing complex scheduling challenges in cloud and quantum computing. The reviewed frameworks—ranging from Q-learning-based task scheduling and DRL resource allocation to cost-optimized workflow scheduling—demonstrate significant advancements in scalability, adaptability, and efficiency. Through mathematical models and machine learning-driven strategies, these solutions surpass traditional methods in minimizing task completion times, enhancing resource utilization, and reducing operational Costs.

Emerging technologies, such as quantum cloud computing, present unique scheduling demands that further necessitate intelligent and adaptive algorithms. Studies like DRLQ have set the stage for leveraging advanced DRL techniques in this domain. However, challenges remain, including the interpretability of DRL models, computational complexity, and the integration of uncertainty measures into scheduling decisions. Future research should focus on exploring safe reinforcement learning, hybrid algorithmic approaches, and multi-cloud scenarios to enhance the reliability and robustness of these systems. This survey underscores the vast potential of reinforcement learning in revolutionizing resource and task management across diverse computing environments.

REFERENCES

[1] Wang, Y.; Dong, S.; Fan, W. Task Scheduling Mechanism Based on Reinforcement Learning in Cloud Computing. Mathematics 2023, 11, 3364.
[2] Zhou, G., Tian, W., Buyya, R. *et al*. Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions. *Artif Intell Rev* 57, 124 (2024)
[3] Hang Dong, Boshi Wang, Bo Qiao, Wenqian Xing, Chuan Luo, Si Qin, Qingwei Lin, Dongmei Zhang, Gurpreet Virdi, and Thomas Moscibroda. Predictive job scheduling under uncertain constraints in cloud computing. In IJCAI, pages 3627–3634, 2021.
[4] Luis Fanjul-Peyro, Federico Perea, and Rubén Ruiz. Models and matheuristics for the unre lated parallel machine scheduling problem with additional resources. European Journal of Operational Research, 260(2):482– 493, 2017.
[5] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, "Advances in quantum cryptography," Adv. Opt. Photon., vol. 12, pp. 1012–1236, Dec 2020.
[6] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum computing for finance: State-of-the-art and future prospects," IEEE Transactions on Quantum Engineering, vol. 1, pp. 1–24, 2020.
[7] M. T. West, S.-L. Tsang, J. S. Low, C. D. Hill, C. Leckie, L. C. L. Hollenberg, S. M. Erfani, and M. Usman, "Towards quantum enhanced adversarial robustness in machine learning," Nature Machine Intelli gence, vol. 5, p. 581–589, May 2023.
[8] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature, vol. 549, p. 242–246, Sep 2017.
[9] H. T. Nguyen, P. Krishnan, D. Krishnaswamy, M. Usman, and R. Buyya, "Quantum Cloud Computing: A Review, Open Problems, and Future Directions," 2024. arXiv:2404.11420.
[10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.
[11] G. Zhou, W. Tian, and R. Buyya, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," Artificial Intelligence Review, vol. 57, 2024.
[12] B. Zolfaghari and S. Abrishami, "A multi-class workflow ensemble management system using on-demand and spot instances in cloud," Future Generation Computer Systems, vol. 137, pp. 97–110, 2022.
[13] D. Poola, K. Ramamohanarao, and R. Buyya, "Enhancing reliability of workflow execution using task replication and spot instances," ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 10, no. 4, pp. 1–21, 2016.
[14] T.-P. Pham and T. Fahringer, "Evolutionary multi-objective workflow scheduling for volatile resources in the cloud," IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1780–1791, 2020.
[15] D. Poola, K. Ramamohanarao, and R. Buyya, "Fault-tolerant workflow scheduling using spot instances on clouds," Procedia Computer Science, vol. 29, pp. 523–533, 2014.
[16] "New amazon ec2 spot pricing model." https://aws.amazon.com/blogs/ compute/new-amazon-ec2-spot-pricing/.
[17] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, and S. Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based heft," Future Generation Computer Systems, vol. 93, pp. 278 289, 2019.
[18] A. Belgacem and K. Beghdad-Bey, "Multi-objective workflow schedul ing in cloud computing: trade-off between makespan and cost," Cluster Computing, vol. 25, no. 1, pp. 579–595, 2022.
[19] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," Journal of Systems Architecture, vol. 100, p. 101631, 2019.

[20] G. L. Stavrinides and H. D. Karatza, “An energy-efficient, qos-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing dvfs and approximate computations,” Future Generation Computer Systems, vol. 96, pp. 216–226, 2019.

[21] G. Ismayilov and H. R. Topcuoglu, “Neural network based multi objective evolutionary algorithm for dynamic workflow scheduling in cloud computing,” Future Generation Computer Systems, vol. 102, pp. 307–322, 2020.