Artificial Intelligence and Computing Innovations
Intelligence Science Publishing
https://www.ispublishing.com
ISSN: 3070-4944
Vo.5, No.7, 2025

Adaptive Resource Scheduling in Distributed Systems via Reinforcement Learning under Dynamic Competition

Shi Shu[1*]
[1]University of Pennsylvania, Philadelphia, USA
**Corresponding author: jamesnikess@gmail.com*

Abstract: To address the challenges faced by distributed systems in dynamic resource contention environments, such as uncertain task arrival, significant node load fluctuations, and insufficient adaptive capabilities of traditional scheduling strategies, a reinforcement learning-driven adaptive scheduling method for distributed systems is proposed. This method starts with the dynamic coupling characteristics of task requirements, node states, and resource contention, modeling the scheduling process as a sequential decision-making problem oriented towards long-term reward optimization. By constructing a competition-aware state representation mechanism, a unified characterization of multi-source heterogeneous information in complex operating environments is achieved. Furthermore, a context-enhanced state encoder is designed to improve the model's ability to express temporal dependencies, global load changes, and potential competition patterns. Combined with a multi-objective reward optimization mechanism, it collaboratively constrains job completion efficiency, resource utilization, load distribution balance, and system throughput, enabling the scheduling strategy to achieve more stable and effective decisions in complex resource game processes. Results show that the proposed method maintains strong environmental adaptability and global optimization capabilities in dynamically changing distributed scenarios, providing a feasible path for distributed resource scheduling that balances efficiency, balance, and overall system service capabilities.

Keywords: Resource orchestration; sequence decision-making; load coordination; intelligent scheduling.

1 Introduction

With the continuous development of cloud computing, edge computing, and large-scale data processing platforms, distributed systems have become the infrastructure supporting key scenarios such as intelligent services, online transactions, the industrial internet, and high-performance computing. Faced with the ever-expanding scale of business, increasingly diverse task types, and continuously improving service quality requirements, distributed systems not only need high throughput, low latency, and high reliability, but also need to efficiently coordinate various resources such as computing, storage, and networking in complex environments[1,2]. However, in actual operation, the supply and demand

relationship of resources often exhibits significant dynamism and uncertainty. Competition and conflict are common among different tasks in terms of execution time, resource occupancy intensity, and priority. Traditional scheduling methods relying on static rules or fixed heuristic strategies are increasingly unable to adapt to rapidly changing system states and multi-objective optimization needs. Therefore, exploring more intelligent and adaptive scheduling methods for dynamic resource competition scenarios has significant theoretical and practical value[3].

Dynamic resource competition is one of the core challenges in distributed system scheduling problems. Its essence lies in the continuous game and coupled evolution of multiple tasks in a shared heterogeneous resource environment. When task arrival is random, resource load fluctuates, and node states are heterogeneous, the internal scheduling decision-making of the system is no longer an isolated local choice, but a sequence optimization problem closely related to global performance, long-term benefits, and service stability. If the scheduling strategy lacks continuous perception and self-adjustment capabilities to environmental changes, it can easily lead to resource allocation imbalances, congestion of hot nodes, increased task waiting times, and a decline in overall service efficiency, further impacting system throughput and user experience[4]. Against this backdrop, how to characterize the complex relationship between tasks and resources from a dynamic interaction perspective, and how to achieve accurate modeling and efficient decision-making of system states in competitive environments, have become crucial areas requiring breakthroughs in intelligent scheduling research for distributed systems[5].

Reinforcement learning offers a new research approach to solving these problems[6]. Compared to traditional methods, reinforcement learning can learn decision-making strategies adapted to complex scenarios through continuous interaction with the environment, in a cyclical process of state perception, action selection, and feedback optimization[7]. This makes it more suitable for handling high-dimensional, dynamic, nonlinear, and long-term dependent scheduling tasks. Introducing reinforcement learning into adaptive scheduling of distributed systems helps to overcome the strong dependence of manual rule design on prior experience, improving the scheduling strategy's ability to model complex resource competition relationships and its responsiveness to sudden load changes. Simultaneously, reinforcement learning can comprehensively consider resource utilization, task completion efficiency, load balancing, and service stability from the perspective of maximizing long-term benefits, providing a methodological foundation for building intelligent scheduling frameworks with continuous optimization capabilities. This integration not only expands the research paradigm of distributed system scheduling problems but also provides a new technical path for the autonomous operation of intelligent computing infrastructure.

Research on reinforcement learning-driven adaptive scheduling of distributed systems focusing on dynamic resource competition has significant academic and applied value. From a theoretical perspective, this research helps promote the cross-integration of resource competition modeling, sequence decision optimization, and intelligent system control, enriching the research connotation of distributed system scheduling problems in complex dynamic environments. From a methodological perspective, exploring scheduling mechanisms that can adapt to multi-task concurrency, heterogeneous resource collaboration, and continuously changing environments helps improve the robustness,

generalization, and self-organization capabilities of systems in complex scenarios. From an application perspective, the research findings can serve various key scenarios such as cloud platform resource management, edge node collaborative computing, data center task orchestration, and real-time service assurance. They play a significant role in improving infrastructure operational efficiency, reducing resource waste, and enhancing system service capabilities. Therefore, research on reinforcement learning-driven adaptive scheduling methods for distributed systems in the context of dynamic resource competition not only responds to the urgent need for intelligent scheduling capabilities in new computing environments but also demonstrates important research significance for the evolution of future intelligent infrastructure.

Recent studies on representation and retrieval also indicate that adaptive distributed scheduling should not be isolated from broader intelligent system design. Keyword-constrained retrieval-augmented generation shows how external constraints can guide model outputs under knowledge-intensive conditions[8], while multi-source feature fusion for delay-risk prediction demonstrates the value of integrating heterogeneous signals before making operational decisions[9]. Distributed anomaly detection based on reconstruction residuals and latent distribution shifts, together with temporal state modeling for anomaly detection, further suggests that robust schedulers need to detect not only current load levels but also abnormal changes in hidden system states[10,11].

Knowledge-structured and semantic-alignment methods provide another methodological reference for modeling task-resource relationships. Graph-based knowledge representation, knowledge-graph semantic fusion, and logistic similarity learning all emphasize that relational structure and calibrated similarity can improve matching quality under sparse or heterogeneous observations[12,13,14]. Frequent subgraph mining with structural prompt injection and structured pruning-driven low-rank adapter learning further show that compact structural knowledge and parameter-efficient adaptation can improve model flexibility without excessive computational overhead[15,16].

These ideas are consistent with the scheduling problem considered in this paper, where task urgency, node status, resource contention, and temporal workload evolution must be jointly represented. Knowledge-graph-enhanced sentiment classification, frequency-guided Transformer forecasting, cross-modal graph neural recommendation, instruction-driven classification, and Transformer-based Kubernetes anomaly detection illustrate complementary strategies for combining semantic structure, temporal signals, graph relations, and sequence modeling in intelligent decision systems[17,18,19,20,21]. They motivate a scheduling framework that treats resource allocation as a dynamic representation-learning and policy-optimization problem rather than a static rule-matching process.

2 Methodology

To address the intrinsic coupling between task arrival uncertainty and resource contention in distributed environments, the scheduling process is formulated as a sequential decision problem in which the scheduler continuously observes system variations and produces adaptive allocation actions. Rather than relying on fixed heuristics that are insensitive to temporal load transitions, the proposed method establishes an actor-critic reinforcement learning-driven control loop that explicitly captures the interaction among task urgency, node load, and inter-resource competition, thereby enabling the

scheduler to pursue long-horizon efficiency instead of myopic local optimality. This paper also presents the overall model architecture, as shown in Figure 1.

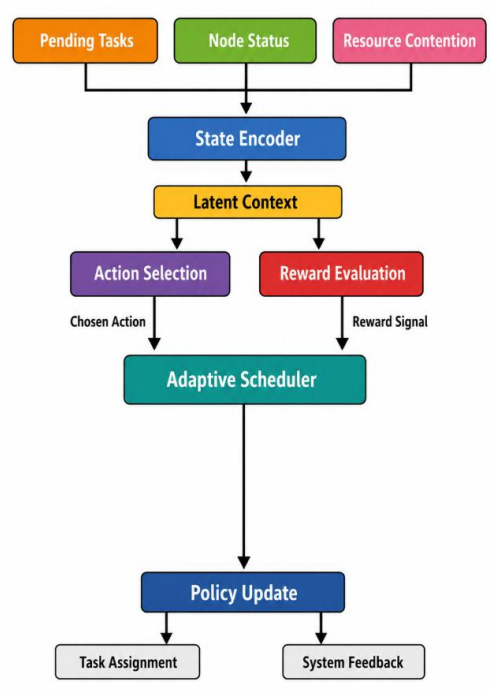


Figure 1. Overall model architecture

At decision step t, the distributed system is represented by a system-wide state composed of task features, node features, and contention-aware relational descriptors:

$$s_t = \Phi\left(\boldsymbol{Q}_t, \boldsymbol{N}_t, \boldsymbol{C}_t\right) \tag{1}$$

where $\boldsymbol{Q}_t$ denotes the pending task set at step t, $\boldsymbol{N}_t$ characterizes the current condition of available worker nodes, and $\boldsymbol{C}_t$ summarizes the instantaneous competition intensity among heterogeneous resources through an integrated state mapping function $\Phi(\cdot)$. Such a design is introduced to ensure that the policy is not only aware of isolated resource usage but also sensitive to the structural dependency between competing demands. For each candidate task i and node j, a contention-aware affinity score is further constructed to quantify the suitability of assigning the task to that node under dynamic congestion:

$$\psi_{ij}^{(t)} = \alpha \frac{d_i}{d_t + \epsilon} + \beta \frac{l_j^{(t)}}{l_t + \epsilon} + \gamma \frac{\kappa_{ij}^{(t)}}{\kappa_t + \epsilon} \tag{2}$$

in which d_i reflects the resource demand of task i, $l_j^{(t)}$ measures the current load of node j, $\kappa_{ij}^{(t)}$ evaluates the interference caused by placing task i on node j, and α, β, and γ control the relative importance of demand pressure, load accumulation, and competition intensity. By embedding these normalized factors into a unified score, the scheduler gains a more discriminative view of hidden bottlenecks that are often overlooked when only raw utilization is considered. Based on this representation, the admissible scheduling action is defined as:

Accordingly, the contention-aware score can be understood as a feature-fusion layer over task demand, node capacity, and interference intensity. Multi-source fusion and semantic similarity learning provide useful design principles for this step, since both emphasize that heterogeneous signals should be normalized, aligned, and jointly evaluated before final classification or matching[9,14]. Graph representation and knowledge-fusion studies further support treating relations among tasks, resources, and contention factors as structured context rather than independent scalar indicators[12,13,19].

$$a_t \in \mathcal{A}(s_t) = \left\{(i,j) | i \in \boldsymbol{Q}_t,\ j \in \boldsymbol{N}_t,\ \chi_{ij}^{(t)} = 1\right\} \tag{3}$$

where the feasibility indicator $\chi_{ij}^{(t)}$ equals 1 only when capacity, locality, and service constraints are all satisfied, which prevents the policy from exploring invalid allocations and improves decision stability in highly constrained operating conditions.

Because dynamic resource competition evolves across time rather than within isolated snapshots, a context-enhanced encoder is introduced to compress high-dimensional system observations into a compact latent representation that preserves both instantaneous status and temporal dependency. This mechanism is meaningful for distributed scheduling because short-term fluctuations may not immediately reveal future congestion, whereas historical trends often contain strong signals about impending contention escalation. The latent state is obtained by:

$$\boldsymbol{z}_t = f_\theta(s_t, \boldsymbol{h}_{t-1}) \tag{4}$$

where $f_\theta(\cdot)$ denotes a parameterized state encoder and $\boldsymbol{h}_{t-1}$ carries decision-relevant historical context from preceding steps. On top of the latent representation, the adaptive scheduling policy outputs a distribution over feasible assignments:

$$\pi_\theta(a_t | s_t) = \frac{\exp\left(g_\theta(\mathbf{z}_t, a_t)\right)}{\sum_{a' \in \mathcal{A}(s_t)} \exp\left(g_\theta(\mathbf{z}_t, a')\right)} \tag{5}$$

in which $g_\theta(\boldsymbol{z}_t, a_t)$ produces the preference score of the action a_t conditioned on the encoded system context. This probabilistic formulation is adopted to maintain sufficient exploration under nonstationary workloads while preserving the ability to emphasize assignments that are beneficial to global scheduling quality.

The context encoder also functions as a drift-sensitive representation module. Anomaly detection based on residual-distribution discrimination, temporal state modeling, frequency-guided Transformer forecasting, and Kubernetes sequence modeling all indicate that temporal deviations and latent workload changes should be encoded before they become visible performance degradation[10,11,18,21].

From the perspective of system objectives, merely minimizing one metric cannot fully reflect the operational requirements of distributed platforms, since throughput, delay, load balance, and contention mitigation are tightly coupled and often mutually restrictive. For that reason, the reward function is designed to encourage balanced optimization across multiple dimensions so that the learned policy remains aligned with practical scheduling goals. The immediate reward at step t is written as:

$$r_t = \lambda_1 \Delta u_t - \lambda_2 \Delta \tau_t - \lambda_3 \Delta b_t - \lambda_4 \Delta c_t \tag{6}$$

where Δu_t denotes the improvement in resource utilization, $\Delta \tau_t$ represents the increase in task completion delay, Δb_t measures load imbalance among nodes, Δc_t captures the growth of contention cost, and λ_1 through λ_4 are nonnegative trade-off coefficients. Through this construction, the scheduler is encouraged to allocate resources efficiently without sacrificing service timeliness or causing excessive concentration of workload on a small subset of nodes. The long-term objective is then defined by the discounted cumulative return:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{7}$$

where the discount factor $\gamma \in (0,1)$ regulates the balance between immediate gain and future system quality, which is essential for avoiding shortsighted actions that appear beneficial locally but amplify downstream congestion.

In order to continuously refine the scheduling strategy under changing system conditions, the policy parameters are optimized by maximizing the expected return while preserving adaptation capability in the presence of dynamic workload drift. Unlike static optimization schemes that implicitly assume a stable environment, the update process here is designed to absorb feedback from successive interactions and reshape the allocation preference according to evolving contention patterns. The parameter update follows the gradient of the objective:

$$\theta \leftarrow \theta + \eta \nabla_\theta J(\theta) \tag{8}$$

where η is the learning rate controlling the adaptation pace of the scheduler. This iterative optimization is meaningful because distributed systems rarely operate under fixed demand distributions, and a policy that cannot revise its behavior online will gradually lose effectiveness as resource competition intensifies or shifts across nodes. By integrating contention-aware state construction, context-sensitive policy learning, and multi-objective return optimization into one unified framework, the method improves the capability of distributed scheduling to respond to heterogeneous workloads with stronger autonomy, higher flexibility, and better alignment with long-horizon system utility.

For deployment-oriented adaptation, the policy update can be combined with lightweight structural refinement. Structural prompt injection, low-rank adapter learning, and instruction-driven semantic alignment provide references for maintaining model adaptability while controlling parameter growth and preserving decision constraints[15,16,20]. Keyword-constrained retrieval and knowledge-graph-enhanced classification also suggest that external constraints can be used to improve interpretability and reduce unsupported decisions in complex intelligent systems[8,17].

3 Experimental Results and Analysis

3.1 Dataset

This study uses the Alibaba Cluster Trace Program as the source of experimental data. This dataset consists of cluster operation logs from a real production environment, publicly released within the Alibaba Cluster Data project. It is characterized by its open accessibility, large scale, significant dynamic changes in task load, and rich resource contention, making it well-suited for research on adaptive scheduling in distributed systems facing dynamic resource competition. The data covers multi-

dimensional information such as job and task submission, machine status, resource usage, and mixed deployment, providing reliable support for modeling task arrival fluctuations, node load evolution, CPU and memory usage changes, and competitive behavior under multi-task resource sharing. Therefore, constructing scheduling scenarios based on this open-source cluster trajectory data can more realistically characterize the dynamics, uncertainty, and heterogeneity of distributed systems in complex operating environments, thus providing a representative data foundation for research on reinforcement learning-driven adaptive scheduling methods.

3.2 Experimental results compared with other models

To more clearly define the performance characteristics of our proposed method in the field of reinforcement learning-driven adaptive scheduling of distributed systems, we selected representative studies closely related to dynamic task scheduling, cluster resource allocation, Kubernetes scheduling, and distributed resource management for horizontal comparison. The comparison dimensions revolve around job completion latency, resource utilization level, load imbalance, and system throughput, thus presenting the differences in scheduling capabilities of different methods in dynamic resource contention scenarios from a unified metric perspective. The experimental results are shown in Table 1.

Table 1. Experimental results compared with other models

Method	JCT↓	UTL↑	IMB↓	THR↑
Mao et al.[22]	13.92	78.41	0.23	142.37
Mao et al.[23]	12.64	81.76	0.20	149.58
Chen et al.[24]	12.31	82.48	0.19	151.24
Liang et al.[25]	11.88	83.95	0.18	154.63
Fan et al.[26]	11.47	84.82	0.17	157.91
Huang et al.[27]	11.21	85.36	0.16	159.74
Jian et al.[28]	10.94	86.08	0.15	162.53
Ours	10.42	87.61	0.13	166.28

Table 1 uses four evaluation metrics to characterize the overall performance of the scheduling method. JCT (Job Completion Time) measures the total time cost from job submission to completion. A lower JCT indicates that the scheduling strategy can shorten task waiting and execution cycles, improving system response efficiency. UTL (Resource Utilization) reflects the sufficiency of computing resource utilization during system operation. A higher UTL indicates less resource idleness and waste, resulting in more efficient overall scheduling. IMB (Load Imbalance) measures the load distribution differences between different nodes. A lower IMB indicates more balanced task allocation, helping to alleviate local congestion and hotspot clustering. THR (Throughput) reflects the system's task processing capacity per unit time. A higher THR indicates stronger overall support for the scheduling method in ensuring execution efficiency and resource coordination.

The algorithm proposed in this paper demonstrates strong comprehensive advantages across multiple key metrics, indicating that it can achieve more effective scheduling decisions in complex resource contention environments. On one hand, the method can better shorten job completion time and improve system throughput, demonstrating high efficiency in task sequencing, resource matching, and execution advancement. On the other hand, the improved resource utilization and reduced load imbalance further demonstrate that this method not only focuses on the rapid completion of individual tasks but also takes into account global resource coordination and node load distribution, thereby maintaining a more stable and efficient scheduling state during the operation of the distributed system. This indicates that the proposed method can adapt more fully to the complex changes brought about by dynamic resource competition and achieve a better coordination between efficiency, balance, and overall service capacity.

3.3 Ablation test results

To verify the roles of each key component in the reinforcement learning-driven adaptive scheduling framework for dynamic resource contention in distributed systems, ablation analysis was conducted, and the results are summarized in Table 2. The ablation settings strictly adhere to the contention-aware state modeling, context-enhanced state encoding, and multi-objective reward optimization mechanisms in the method design, thereby examining the actual contributions of different modules to job latency control, resource utilization efficiency, load balancing capabilities, and system throughput performance.

Table 2. Ablation test results

Ablation Setting	JCT↓	UTL↑	IMB↓	THR↑
w/o Contention-Aware State Modeling	11.36	84.29	0.17	158.46
w/o Context-Enhanced State Encoder	11.09	85.02	0.16	160.11
w/o Multi-Objective Reward Optimization	10.87	85.74	0.15	161.84
Ours	10.42	87.61	0.13	166.28

Ablation analysis results demonstrate that the proposed algorithm achieves superior overall performance while maintaining the integrity of the overall structure, indicating that each component plays an irreplaceable role in the scheduling decision-making process. Specifically, competition-aware state modeling enhances the method's ability to characterize the relationship between task requirements, node load, and resource conflicts, enabling the scheduling process to more accurately perceive complex operating environments. The context-enhanced state encoder improves the ability to express temporal dependencies and global state information, thus contributing to the formation of more stable and effective decision representations. The multi-objective reward optimization mechanism further coordinates the relationship between job completion efficiency, resource utilization level, load balancing capability, and system throughput, ensuring that the scheduling objective is no longer limited to a single

performance dimension. Because these key designs form a mutually supportive synergy within a unified framework, the proposed method can achieve higher-quality resource allocation and task scheduling in dynamic resource competition scenarios, demonstrating stronger overall adaptability and system optimization capabilities.

3.4 The impact of state coding dimension on JCT

The state encoding dimension is a crucial hyperparameter affecting the representational ability and decision-making quality of scheduling models. Its setting directly relates to the compression method of system state information and the degree to which key features are preserved. Too low a dimension may limit the model's ability to fully express task requirements, node load, and resource contention, while too high a dimension can easily introduce redundant information and increase the burden of representation learning. Therefore, it is necessary to conduct a dedicated analysis around the state encoding dimension to further examine the model's ability to characterize the complex operating states of distributed systems under different representation scales. The experimental results are shown in Figure 2.

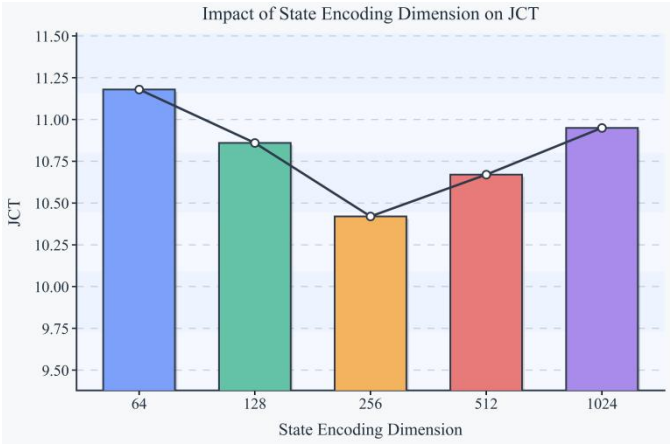


Figure 2. The impact of state coding dimension on JCT

The state encoding dimensionality has a significant impact on the model's scheduling performance. A moderate representation size can more effectively balance information representation capability and decision stability. When the encoding dimensionality is low, task characteristics, node load, and resource contention relationships in the system state are difficult to fully preserve, thus weakening the model's ability to characterize complex scheduling scenarios. Conversely, as the encoding dimensionality continues to increase, although the representation space becomes more abundant, redundant information is more easily introduced, increasing the difficulty of feature extraction and affecting the effectiveness of policy learning. In comparison, the method presented in this paper performs better with a moderate dimensionality setting, indicating that this setting achieves a more reasonable balance between state compression and key information preservation, enabling the model to better model dynamic resource contention environments and make adaptive scheduling decisions.

4 limitation

Although this paper proposes a reinforcement learning-driven adaptive scheduling method for distributed systems in dynamic resource contention scenarios and makes targeted designs in state

modeling, context encoding, and multi-objective optimization, this research still has certain limitations. First, the proposed method mainly focuses on the joint modeling of task requirements, node load, and resource contention relationships, and has not fully characterized the more complex real-world constraints in distributed systems, such as cross-node communication overhead, task dependency structures, heterogeneous hardware differences, and scheduling robustness under fault disturbances. In real-world production environments, these factors are often deeply coupled with the resource allocation process, further affecting the transferability and generalization ability of the scheduling strategy. Therefore, while the current method can adapt well to dynamic resource contention problems, there is still room for further expansion in covering scenarios with higher complexity and multiple coupled constraints.

On the other hand, reinforcement learning methods themselves are highly dependent on the construction of the training environment, the quality of reward design, and the state representation method, which may affect the model's adaptability under different system scales and load distributions. Especially under conditions of rapid changes in business models or frequent extreme bursts of load, whether existing strategies can maintain stable decision-making capabilities requires further in-depth research combining richer online update mechanisms and security constraint mechanisms. Furthermore, this paper currently focuses more on the overall optimization of system-level scheduling performance, and there is still insufficient discussion on interpretability analysis, transparent expression of policy decision-making criteria, and lightweight modifications for practical deployment. Future research can further combine causal modeling, robust optimization, and interpretable reinforcement learning to improve the reliability, availability, and engineering application value of scheduling methods in real distributed environments.

5 Conclusion

This paper addresses the challenges of complex resource coupling, frequent state changes, and insufficient adaptability of traditional static strategies in distributed system scheduling under dynamic resource contention scenarios. It proposes a reinforcement learning-driven adaptive scheduling method. Starting from the dynamic interaction between task requirements, node states, and resource contention in actual distributed system operation, this method constructs a more environment-aware scheduling modeling framework. Through context-enhanced state representation and multi-objective collaborative optimization decision design, the scheduling strategy achieves more flexible and effective resource allocation under complex system conditions. Related research shows that the proposed method effectively balances job completion efficiency, resource utilization, load balancing, and system throughput, demonstrating its adaptability and application potential in dynamic, variable, and highly competitive distributed computing environments. Overall, this work not only provides new research ideas for the in-depth application of reinforcement learning methods in distributed system scheduling problems but also lays the methodological foundation for building intelligent scheduling frameworks with continuous awareness, autonomous decision-making, and long-term optimization capabilities.

From an application perspective, this research has significant reference value for cloud computing platform resource orchestration, data center task management, edge computing collaborative processing,

and real-time service assurance for complex business loads. As new computing infrastructure continues to evolve towards large-scale, heterogeneous, and intelligent architectures, the resource competition patterns faced by scheduling systems will become more complex, and the requirements for adaptive decision-making and global coordination capabilities during task execution will continue to increase. Therefore, future research can further expand scheduling models by incorporating cross-node communication overhead, task dependencies, heterogeneous device collaboration, and fault recovery mechanisms. Based on this, lightweight policy update mechanisms and more robust decision-making frameworks for online deployment can be explored. Simultaneously, reinforcement learning scheduling methods can be promoted from theoretical research to deeper engineering applications in real-world scenarios such as intelligent computing platforms, the industrial internet, real-time edge services, and high-concurrency intelligent applications. This will provide stronger technical support for improving the operational efficiency, resource coordination level, and service stability of next-generation distributed systems.

References

[1] Y. Han, S. Shen, X. Wang, et al., "Tailored learning-based scheduling for kubernetes-oriented edge-cloud system," in Proceedings of the IEEE International Conference on Computer Communications, 2021, pp. 1-10.

[2] Y. Mao, Y. Fu, W. Zheng, et al., "Speculative container scheduling for deep learning applications in a kubernetes cluster," IEEE Systems Journal, vol. 16, no. 3, pp. 3770-3781, 2021.

[3] Z. Lin, C. Li, L. Tian, et al., "A scheduling algorithm based on reinforcement learning for heterogeneous environments," Applied Soft Computing, vol. 130, p. 109707, 2022.

[4] J. Wang, S. Li, X. Zhang, et al., "Deep reinforcement learning task scheduling method based on server real-time performance," PeerJ Computer Science, vol. 10, p. e2120, 2024.

[5] J. Pan and Y. Wei, "A deep reinforcement learning-based scheduling framework for real-time workflows in the cloud environment," Expert Systems with Applications, vol. 255, p. 124845, 2024.

[6] P. Choppara and S. Mangalampalli, "An efficient deep reinforcement learning based task scheduler in cloud-fog environment," Cluster Computing, vol. 28, no. 1, p. 67, 2025.

[7] Z. Yao, L. Ding, H. Zhang, et al., "Deep reinforcement learning for job scheduling on load-aware heterogeneous cluster," Journal of Big Data, vol. 12, no. 1, p. 276, 2025.

[8] W. Ma and W. Zhong, "Keyword-Constrained Retrieval-Augmented Generation for Large Language Models," 2025.

[9] R. Jiang and Y. Shi, "SCDF-Net: Research on a Multi-Source Feature Fusion Binary Classification Prediction Network Algorithm for Supply Chain Delay Delivery Risk," 2025.

[10] Z. Liu and D. Abdurehim, "Distributed System Anomaly Detection via Collaborative Discrimination of Reconstruction Residuals and Latent Distribution Shifts," 2025.

[11] Y. Zhan, "Adaptive Representation Learning and Temporal State Modeling for Robust Anomaly Detection in Distributed Systems," 2024.

[12] J. Zhou, "Graph-Based Knowledge Representation and Semantic Alignment for Large Language Model-Driven Entity Recognition," 2025.

[13] Z. Zhu, "Knowledge Graph-Augmented Semantic Fusion for Large Language Model-Based Long-Text Classification," Transactions on Computational and Scientific Methods, vol. 4, no. 10, 2024.

[14] R. Hu, "Unified Semantic Matching with Logistic Similarity Learning for Cold-Start Advertising Recommendation," 2024.

[15] Z. Xiao, C. Moretti, and Q. Li, "Frequent Subgraph Mining and Structural Prompt Injection for Large Language Model Fine-Tuning," 2025.

[16] O. Lin, G. Brennan, and X. Xu, "Structured Pruning-Driven High-Expressive Low-Rank Adapter Learning for Large Language Models," 2025.

[17] Z. Gu and L. Nie, "Knowledge Graph-Enhanced Large Language Models for Opinion Target Sentiment Classification," 2025.

[18] Y. Lyu, J. Jiang, and J. Hu, "Frequency-Guided Transformer Modeling with Spectral Enhancement for Accurate Time Series Forecasting," 2024.

[19] Z. Ke, "MCA-TAN: Cross-Modal Graph Neural Recommendation with Temporal Behavior and Social Network Fusion," 2025.

[20] Y. Jiang and Y. L. Peng, "Instruction-Driven Language Model for Text Classification via Cross-Attention Fusion and Semantic Alignment," 2024.

[21] F. Chang, "Intelligent Anomaly Detection in Kubernetes Clusters Through Transformer-Based Sequence Modeling," 2024.

[22] H. Mao, M. Alizadeh, I. Menache, et al., "Resource management with deep reinforcement learning," in Proceedings of the ACM Workshop on Hot Topics in Networks, 2016, pp. 50-56.

[23] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, et al., "Learning scheduling algorithms for data processing clusters," in Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 2019, pp. 270-288.

[24] W. Chen, Y. Xu, and X. Wu, "Deep reinforcement learning for multi-resource multi-machine job scheduling," arXiv preprint arXiv:1711.07440, 2017.

[25] S. Liang, Z. Yang, F. Jin, et al., "Data centers job scheduling with deep reinforcement learning," in Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Cham: Springer International Publishing, 2020, pp. 906-917.

[26] Y. Fan, B. Li, D. Favorite, et al., "Dras: Deep reinforcement learning for cluster scheduling in high performance computing," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 12, pp. 4903-4917, 2022.

[27] J. Huang, C. Xiao, and W. Wu, "Rlsk: A job scheduler for federated kubernetes clusters based on reinforcement learning," in Proceedings of the IEEE International Conference on Cloud Engineering, 2020, pp. 116-123.

[28] Z. Jian, X. Xie, Y. Fang, et al., "DRS: A deep reinforcement learning enhanced Kubernetes scheduler for microservice-based system," Software: Practice and Experience, vol. 54, no. 10, pp. 2102-2126, 2024.