**Artificial Intelligence and Computing Innovations**

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.5, No.2, 2025

# *Value-Iteration Reinforcement Learning for Distributed Cloud Workflow Scheduling: A Critical-Task-Prioritized Execution Optimization Framework*

**Chuan He[1*], Kuan-Yu Chen[2]**

[1]New York University, New York, USA

[2]Northeastern University, Boston, USA

**Corresponding author: hechuan2g@gmail.com*

***Abstract:*** Addressing the complex workflow scheduling problem in distributed cloud environments, this paper proposes a value-iterative reinforcement learning method prioritizing critical tasks. This method addresses the limitations of traditional approaches in handling dynamic resource states, task dependency constraints, and critical task assurance, including insufficient global coverage, limited long-term benefit characterization, and constrained execution optimization capabilities. The method models cloud workflows as task execution structures with dependencies, integrating workflow states, resource states, and dependency information within a unified scheduling framework to construct a state representation mechanism for complex execution processes. Furthermore, it strengthens the priority assurance capability for core task links through criticality evaluation. Building upon this foundation, the method combines value iteration with reinforcement learning decision-making mechanisms to jointly model the immediate benefits and subsequent state evolution during task allocation. This enables the scheduling strategy to achieve global optimization that balances execution efficiency, energy consumption control, resource utilization, and system stability in a multi-node heterogeneous resource environment. Moreover, this paper designs a reward modeling approach that matches the scheduling objectives, allowing critical task delay suppression, resource coordination and allocation, and long-term execution quality improvement to be synergistically driven under a unified goal. The results show that the proposed method has good overall performance in complex cloud scheduling scenarios, effectively improving the execution guarantee capability of critical tasks and the overall scheduling quality, and providing a technical path with application value for intelligent cloud resource management and efficient workflow execution.

***Keywords:*** Cloud resource orchestration; mission criticality modeling; long-term value decision making; heterogeneous node scheduling

## 1 Introduction

As cloud computing infrastructure continues to evolve towards distributed, service-oriented, and elastic architectures, an increasing number of complex applications for data processing, intelligent analysis, industrial control, and online business collaboration are being organized into workflows with dependencies and constraints[1,2]. Distributed cloud workflow scheduling not only requires task allocation in environments with multiple nodes, resource types, and coupled tasks, but also necessitates comprehensive consideration of multiple factors such as task dependencies, resource heterogeneity, network transmission overhead, and dynamic load fluctuations. In this context, scheduling systems are no longer simply underlying support modules pursuing throughput improvements, but have gradually become core components affecting business continuity, service latency, resource utilization efficiency, and overall operating costs. Especially in large-scale distributed cloud environments, the execution relationships between workflow tasks are more complex, and the latency propagation effect of critical tasks is more significant. If scheduling decisions lack a global and forward-looking perspective, it can easily lead to local resource congestion, critical path blockage, and overall service performance degradation. Therefore, building a methodology for distributed cloud workflow scheduling that combines dynamic awareness, strategy optimization, and critical task prioritization capabilities has significant theoretical value and practical necessity[3].

Traditional scheduling methods typically rely on static heuristics, fixed priority ordering, or local greedy search to allocate resources. While these methods apply to scenarios with small task scales and limited environmental changes, they often struggle to achieve optimal long-term benefits in distributed cloud workflow systems characterized by dynamic arrival, strong dependencies, and multi-objective constraints. Particularly in complex execution scenarios where critical and general tasks coexist, a lack of systematic characterization of the global value of tasks, state transition benefits, and subsequent execution impacts can lead to schedulers overemphasizing currently executable tasks and neglecting priority guarantees for critical path tasks, resulting in increased overall completion time and decreased service quality[4]. Table 1 summarizes several typical challenges in distributed cloud workflow scheduling research and their corresponding research significance. It can be seen that prioritizing critical tasks, global value assessment, and dynamic state adaptation have become important directions for further development in this field.

Table 1. Major Challenges and Research Significance of Distributed Cloud Workflow Scheduling

| Main Issue | Specific Manifestation | Research Significance |
|---|---|---|
| Complex task dependency relationships | Predecessor-successor constraints are significant, and local decisions can easily trigger cascading waiting. | It is necessary to improve the modeling capability of scheduling strategies for the global execution chain. |
| Dynamic resource environment | Node load, bandwidth status, and resource availability fluctuate continuously. | It is necessary to enhance the real-time adaptability of scheduling methods to dynamic environments. |
| Prominent impact of | Delays in critical path tasks can amplify the | It is necessary to establish an execution |

| | | |
|---|---|---|
| critical tasks | overall workflow completion time. | guarantee mechanism that prioritizes critical tasks. |
| Multi-dimensional coupling of optimization objectives | Latency, cost, utilization, and stability conflict with one another. | It is necessary to form a unified decision framework that balances multiple objectives. |
| Difficulty in characterizing long-term returns | The current allocation result will continuously affect subsequent state transitions and resource competition. | It is necessary to introduce a scheduling philosophy oriented toward long-term value optimization. |

Driven by these problems, introducing the concept of value iteration and reinforcement learning mechanisms into distributed cloud workflow scheduling can provide new theoretical support for sequential decision-making in complex environments. Value iteration focuses on the recursive updating of state value, characterizing the continuous impact of current scheduling actions on future execution results from a long-term benefit perspective[5]. Reinforcement learning, on the other hand, continuously refines strategies through environmental interactions, enhancing the model's adaptability to uncertain state changes and complex scheduling feedback. For distributed cloud workflows, this paradigm not only helps overcome the limitations of static rules in handling dynamic environments but also establishes a tighter coupling modeling approach between task execution chains, resource competition relationships, and critical path propagation mechanisms. Furthermore, prioritizing critical tasks in scheduling decisions can more effectively ensure the execution efficiency of core task chains, reducing the cascading impact of critical node blockages on the overall business process, thus providing more robust execution support for high-efficiency, high-reliability, and high-collaboration applications in the cloud environment[6].

From a research perspective, the value iteration reinforcement learning method for distributed cloud workflow scheduling not only responds to the development trend of intelligent resource management but also aligns with the practical needs of modern cloud platforms evolving towards self-adaptation, self-optimization, and business value-oriented collaborative development. Research in this area helps to shift workflow scheduling from experience-based rules to value-driven approaches, and from local reactive allocation to global predictive optimization. It also delves deeper into the differentiated assurance of critical tasks from a unified resource perspective[7]. For typical scenarios such as cloud manufacturing, edge-cloud collaboration, intelligent operation and maintenance, the industrial internet, and large-scale data processing, constructing a task-priority execution optimization framework can not only improve the scheduling reliability and timeliness of complex business chains but also provide a scalable theoretical basis and methodological references for the efficient and stable operation of distributed cloud systems. Therefore, systematic research on this issue has significant academic value and broad engineering application prospects.

## 2 Methodology

In distributed cloud workflow scheduling, the core challenge extends far beyond the simple assignment of ready tasks to currently available computing nodes. The essential difficulty lies in

maintaining a coordinated balance among dependency consistency, critical-path preservation, and long-horizon execution quality under continuously changing resource conditions. In practical cloud environments, workflow instances are usually composed of multiple interdependent tasks with heterogeneous computational demands, different communication costs, and diverse execution priorities. Once the scheduler focuses excessively on immediate task feasibility or local resource idleness, it may easily overlook the downstream impact of current decisions on subsequent task release, resource contention, and end-to-end workflow completion. As a consequence, local scheduling choices that appear beneficial at the current stage may still produce delayed critical tasks, intensified node congestion, and reduced global execution efficiency in later stages. This makes distributed cloud workflow scheduling a typical sequential decision problem in which future system evolution must be considered together with present allocation feasibility.

Another important source of complexity arises from the fact that distributed cloud infrastructures are inherently dynamic and heterogeneous. Computing nodes often differ in processing capacity, available memory, communication bandwidth, and current workload status, while workflow execution itself continuously changes the system state through task completion, dependency release, and resource occupation. Under such conditions, a scheduler that relies only on static rules or one-step greedy matching is often unable to capture the long-range influence of task placement decisions on the overall execution chain. For this reason, the proposed framework formulates workflow scheduling as a value-iterative reinforcement learning process, in which the scheduler repeatedly estimates the future utility of current placement actions before making execution decisions. By embedding critical-task awareness into state perception, action generation, and long-term value updating, the framework aims to strengthen decision consistency and improve the coordination between immediate scheduling requirements and future workflow progression. This paper presents the overall model architecture, as shown in Figure 1.

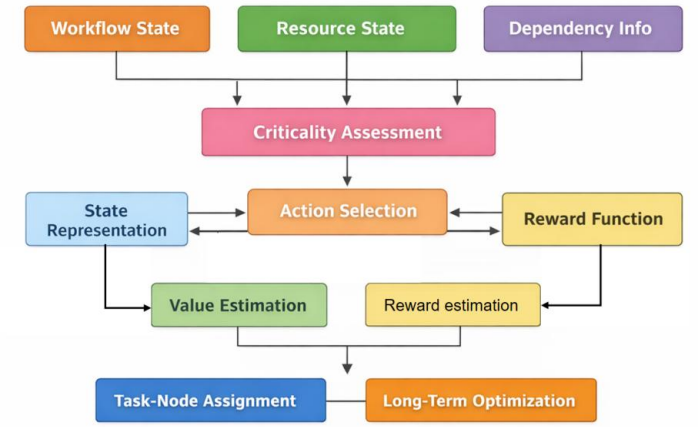



Figure 1. Overall model architecture

A workflow instance is represented as a directed acyclic graph so that precedence constraints, execution attributes, and communication relations can be modeled within a unified structure, which is essential for preventing locally optimal decisions from damaging downstream completion efficiency. Formally, the workflow is defined as follows.

$$\mathcal{W} = \left(\mathcal{T}, \mathcal{D}, \boldsymbol{F}, \boldsymbol{C}\right) \tag{1}$$

Here, denotes the task set, denotes the dependency edges, while and respectively characterize task-side features and inter-task communication costs, allowing the scheduler to jointly perceive computational demand and dependency-induced transfer overhead. Since scheduling quality is determined not only by the current ready queue but also by the remaining execution structure, the environment state is constructed by fusing workflow progression with distributed resource observations, thereby enabling the agent to reason over both execution semantics and infrastructure dynamics. $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_N\} \mathcal{D} \subseteq \mathcal{T} \times \mathcal{T} \boldsymbol{F} \boldsymbol{C}$

$$s_t = \phi\left(\boldsymbol{h}_t^{wf}, \boldsymbol{h}_t^{res}, \boldsymbol{h}_t^{dep}\right) \tag{2}$$

In this expression, summarizes the status of released and unfinished tasks, encodes node-level load, available capacity, and bandwidth-related conditions, and captures the structural pressure induced by unfinished predecessors and successor chains, so that state abstraction remains sensitive to both short-term feasibility and long-term workflow continuity. $\boldsymbol{h}_t^{wf} \boldsymbol{h}_t^{res} \boldsymbol{h}_t^{dep}$

Heterogeneous dynamic dependency graph learning for anomaly detection in large-scale backend systems provides a complementary structural principle for representing evolving task relations and node conditions in the workflow state[8]. Causal-aware anomaly detection for backend operational-state evolution further motivates distinguishing temporary resource fluctuations from state transitions that can propagate along critical task chains[9].

Unlike conventional schedulers that treat all ready tasks with similar urgency, the present method explicitly strengthens the execution priority of structurally important tasks because delays on critical branches often propagate to the entire workflow and amplify completion loss. For this reason, a task criticality score is introduced to quantify structural importance from the perspective of execution cost, successor influence, and path-level urgency, thus guiding the agent to preserve the most delay-sensitive segments of the workflow under limited resources.

$$\kappa_i = \alpha \frac{l_i}{\sum_{\tau_j \in \mathcal{T}} l_j} + \beta \frac{\left|\mathrm{Succ}\left(\tau_i\right)\right|}{\max_{\tau_j \in \mathcal{T}} \left|\mathrm{Succ}\left(\tau_j\right)\right|} + \gamma \frac{p_i}{\max_{\tau_j \in \mathcal{T}} p_j} \tag{3}$$

Within this formulation, denotes the estimated remaining path length initiated by the task , denotes the successor set of , and measures its critical-path proximity, while coefficients , , and regulate the relative emphasis on temporal depth, structural influence, and urgency. Once criticality is quantified, the action space assigns each schedulable task to an admissible resource node, and feasibility is restricted by dependency satisfaction and instantaneous capacity constraints so that the learning process remains aligned with real execution logic. $l_i \tau_i \mathrm{Succ}\left(\tau_i\right) \tau_i p_i \alpha \beta \gamma$

$$a_t = \left(\tau_i, n_j\right), \quad \tau_i \in \mathcal{Q}_t, \quad n_j \in \mathcal{N}_t^{\mathrm{avail}} \tag{4}$$

Under this design, denotes the set of ready tasks at time , and denotes the set of currently available computing nodes, which ensures that policy outputs remain executable rather than purely theoretical. $\mathcal{Q}_t t \mathcal{N}_t^{\mathrm{avail}}$

Beyond priority-aware action generation, the core of the framework lies in using value iteration to estimate the long-term contribution of each scheduling decision, because a placement that appears beneficial at the current step may still trigger downstream congestion, communication delay, or critical-path blocking in later stages. Instead of optimizing only immediate dispatch gain, the scheduler repeatedly refines the state value through recursive evaluation so that future execution consequences are explicitly internalized into current decision making.

$$V_{k+1}(s) = \max_{a \in \mathcal{A}(s)} \left[ r(s,a) + \lambda \sum_{s' \in \mathcal{S}} P(s'|s,a) V_k(s') \right] \tag{5}$$

Here, denotes the estimated value of state at iteration , denotes the immediate scheduling return generated by action , denotes the state transition probability toward , and is the discount factor controlling the influence of future utility. On the basis of this recursive valuation, the policy favors the action with the highest expected long-range benefit rather than the action with only the lowest instantaneous cost, which is particularly important in distributed workflow systems where transient resource relief may conflict with global completion efficiency.$V_k(s)skr(s,a)aP(s'|s,a)s'\lambda$

Time-varying operational graph reasoning with risk-constrained repair planning strengthens the value update process by making structural dependencies and execution feasibility explicit when evaluating future scheduling consequences[10]. Hierarchical reinforcement learning for resource-orchestration optimization in distributed microservices likewise supports decomposing global workflow objectives into coordinated decisions across resource and task-control levels[11].

$$\pi(s) = \arg\max_{a \in \mathcal{A}(s)} \left[ r(s,a) + \lambda \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s') \right] \tag{6}$$

Such a policy structure allows the scheduler to integrate immediate execution feasibility, critical-task protection, and future workflow evolution within a single optimization objective, thereby improving decision coherence across successive scheduling stages.

To make the learned value landscape reflect the actual objective of distributed workflow execution, the reward mechanism is designed to encourage fast completion, balanced utilization, and critical-task preservation at the same time, since any isolated emphasis on a single factor would weaken overall system coordination. A composite reward is therefore constructed to penalize makespan growth, resource imbalance, and critical-task delay while rewarding effective task progress and structurally beneficial placements.

$$r_t = \eta_1 \Delta Prog_t - \eta_2 \Delta Time_t - \eta_3 Imb_t - \eta_4 CritDelay_t \tag{7}$$

In this reward, measures the execution advancement achieved after the current action, reflects the induced temporal cost, describes resource utilization imbalance across distributed nodes, and quantifies the additional delay imposed on critical tasks, while the coefficients regulate the optimization preference of the scheduler. By embedding structural urgency into both value updating and reward design, the proposed method transforms workflow scheduling from a reactive matching problem into a globally aware sequential optimization process, which is meaningful for maintaining execution

continuity, controlling critical-path risk, and improving decision robustness in large-scale distributed cloud environments. $\Delta \mathrm{Prog}_t\, \Delta \mathrm{Time}_t\, \mathrm{Imb}_t\, \mathrm{CritDelay}_t\, \eta_1, \eta_2, \eta_3, \eta_4$

Reinforcement-learning-based adaptive scheduling for throughput improvement provides an additional rationale for using feedback-sensitive rewards to balance critical-path protection with system-wide execution progress under changing distributed conditions[12].

## 3 Experimental Results and Analysis

### 3.1 Dataset

This paper uses Azure Traces for Packing from Microsoft Azure Traces as its data foundation. This dataset originates from a portion of Azure Compute workloads in a real cloud computing platform. It is publicly released cloud server resource request tracking data, open for use in research on resource allocation and placement algorithms, and has a clear data source and accessibility. Unlike data focused on e-commerce business or internet application behavior, this dataset directly depicts the virtual machine request process in a cloud server environment, making it more suitable for research topics such as distributed cloud workflow scheduling, resource orchestration, and task placement optimization. In terms of data content, this dataset includes virtual machine requests and their priorities, lifecycle information of requesting virtual machines, and normalized resource allocation information corresponding to various virtual machine types. Therefore, it can effectively reflect the task arrival characteristics, resource consumption differences, and duration changes in server cloud computing scenarios, providing reliable support for scheduling modeling that prioritizes critical tasks.

Although the dataset is originally released for cloud resource allocation and placement research, the request records, lifecycle information, and normalized resource profiles it provides can still serve as an effective basis for constructing workflow-oriented scheduling states and resource decision contexts. In this work, virtual machine requests are treated as workload units with differentiated importance and execution duration, thereby supporting the modeling of critical-task-aware scheduling behavior under dynamic cloud conditions. Therefore, the adopted dataset remains consistent with the objectives of distributed cloud workflow scheduling, resource orchestration, and execution optimization considered in this paper.

### 3.2 Experimental setup

All experiments were conducted on a unified server-side cloud computing platform to ensure the reproducibility and fairness of workflow scheduling evaluation. The proposed value-iterative reinforcement learning framework was implemented in Python and trained with the PyTorch deep learning library under a GPU-accelerated environment. During training, the scheduler interacted with the workflow execution environment in an episodic manner, and the model parameters were optimized using the Adam optimizer. To improve training stability and maintain long-horizon decision consistency, the learning process adopted a fixed discount factor, a controlled target-network update interval, and a replay-based sampling strategy. The detailed hardware configuration, software environment, and main hyperparameter settings are summarized in Table 2.

Table 2. Hardware, software, and hyperparameter settings

| Category | Item | Setting |
|---|---|---|
| Hardware | CPU | Intel Xeon Gold 6330 |
| Hardware | GPU | NVIDIA GeForce RTX 3090 24 GB |
| Hardware | Memory | 128 GB RAM |
| Software | Operating system | Ubuntu 22.04 LTS |
| Software | Python version | Python 3.10 |
| Software | Deep learning library | PyTorch 2.1 |
| Software | CUDA version | CUDA 12.1 |
| Training hyperparameter | Optimizer | Adam |
| Training hyperparameter | Initial learning rate | 1e-4 |
| Training hyperparameter | Batch size | 64 |
| Training hyperparameter | Discount factor | 0.95 |
| Training hyperparameter | Replay buffer size | 100000 |
| Training hyperparameter | Target update interval | 200 steps |
| Training hyperparameter | Maximum training episodes | 500 |

### 3.3 Experimental Results and Analysis

To enhance the completeness of the comparative analysis, this paper selects representative studies closely related to distributed cloud workflow scheduling, cloud task scheduling, virtual machine placement, and deep reinforcement learning resource orchestration as comparison objects. The corresponding quantitative comparisons are shown in Table 3. Table 3 uniformly uses four indicators: MS, EC, RU, and SR, to comprehensively characterize different methods from four aspects: task completion efficiency, energy consumption control, resource utilization level, and scheduling stability, to more clearly present the performance differences of various methods in cloud computing scheduling scenarios.

Table 3. Experimental results compared with other models

| Method | MS | EC | RU | SR |
|---|---|---|---|---|
| Pan et al.[13] | 128.46 | 96.25 | 84.37 | 92.14 |
| Wei et al.[14] | 132.18 | 93.84 | 83.62 | 91.78 |
| Jayanetti et al.[15] | 124.73 | 90.56 | 85.94 | 93.21 |
| Chen et al.[16] | 126.05 | 94.17 | 84.86 | 92.63 |
| Siddesha et al.[17] | 121.88 | 89.94 | 86.75 | 94.06 |
| Rjoub et al.[18] | 123.41 | 91.22 | 85.68 | 93.58 |
| Zhou et al.[19] | 120.94 | 88.73 | 87.19 | 94.32 |

| | | | | |
|---|---|---|---|---|
| Ours | 116.27 | 84.35 | 90.84 | 96.71 |

Overall, the proposed algorithm demonstrates superior comprehensive performance across multiple evaluation metrics, indicating that it can more effectively coordinate the relationship between task execution efficiency, resource utilization, system stability, and energy consumption control in distributed cloud workflow scheduling. This is because the constructed critical task priority mechanism enhances the scheduling process's ability to guarantee the core task chain, and the value iterative reinforcement learning strategy further improves the decision-making process's ability to characterize the global state and long-term benefits. This allows the scheduling results to maintain good consistency and robustness even under complex resource constraints and dynamic execution environments. This demonstrates that the proposed method not only improves overall scheduling quality but also possesses strong application potential and practical value.

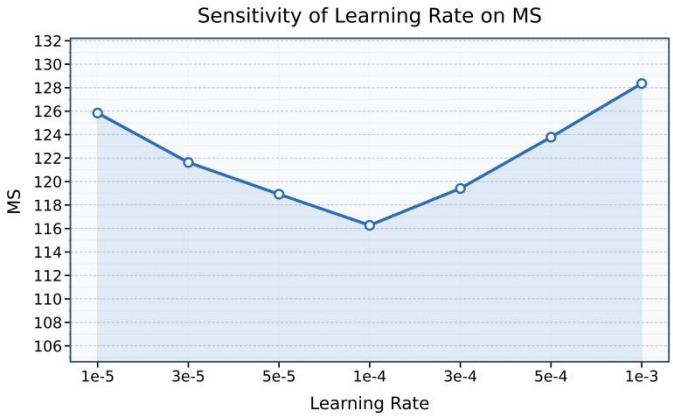



Figure 2. Sensitivity analysis of the learning rate to MS

As shown in Figure 2, the proposed algorithm maintains good scheduling quality and optimization capability during parameter tuning, indicating strong synergy between the constructed key task prioritization mechanism and the value iterative reinforcement learning strategy. Under different learning rate settings, this method can form relatively effective decision responses around the workflow scheduling objective, demonstrating the model's stable ability to characterize task dependencies, resource state changes, and long-term reward information in a distributed cloud environment, thus exhibiting high overall usability and adaptability.

Furthermore, the learning rate, as a crucial factor affecting the parameter update rhythm, directly impacts the accuracy of state perception and the magnitude of decision correction during policy convergence. The proposed method still achieves good optimization results, indicating good matching between the designed state representation, key task evaluation, and reward modeling during training. These results demonstrate that the proposed algorithm not only effectively supports the execution optimization needs in complex cloud workflow scenarios but also exhibits strong training stability and method reliability, providing a solid foundation for subsequent model deployment and parameter setting.

To further examine the adaptability of the proposed method under varying training configurations, it is necessary to analyze the role of optimizer selection, a key training factor, in cloud workflow scheduling tasks. Since different optimizers differ in parameter update methods, convergence rhythms,

and stability control, their impact on the critical task priority scheduling framework deserves separate discussion. Therefore, this section conducts a sensitivity analysis focusing on optimizer settings to observe the performance response characteristics of the proposed algorithm under different training strategies. The experimental results are shown in Figure 3.

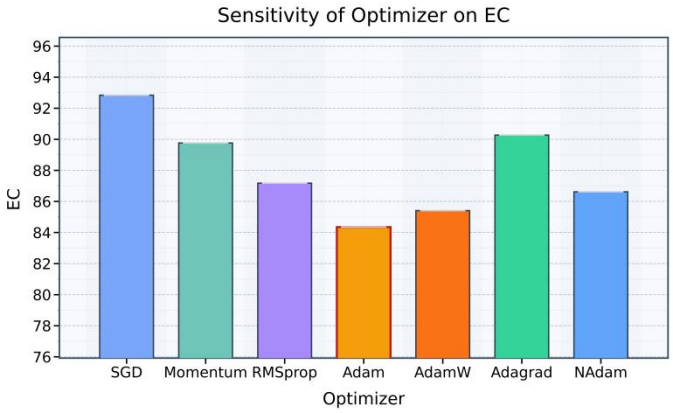



Figure 3. Sensitivity analysis experiment of Optimizer on EC

The optimizer sensitivity analysis shows that the proposed algorithm maintains good energy consumption control under different optimizer configurations, indicating strong adaptability and stability during training. The core reason lies in the fact that the constructed key task prioritization mechanism effectively constrains the resource allocation direction during scheduling, while the value-iterative reinforcement learning strategy further enhances the perception of long-term execution costs, enabling the model to maintain high decision quality even when facing different parameter update methods. This demonstrates that the proposed algorithm not only effectively meets the synergistic requirements of execution efficiency and resource conservation in distributed cloud workflow scenarios but also possesses good training reliability and practical deployment value.

## 4 Conclusion

This paper addresses the challenges of characterizing long-term benefits in distributed cloud workflow scheduling, insufficient guarantees for critical tasks, and limited decision-making stability in dynamic resource environments. It proposes a value-iterative reinforcement learning method focused on prioritizing critical task execution. This method starts with joint modeling of workflow dependency structure and the dynamic state of cloud resources, organically combining critical task importance assessment, state representation learning, action selection mechanisms, and long-term value update processes. Within a unified framework, it achieves global optimization of complex scheduling processes. Compared to traditional methods relying on static rules or local heuristics, this method emphasizes the continuous impact of scheduling decisions on future execution chains, enabling resource allocation to balance current feasibility with subsequent overall benefits. This research provides a new modeling approach for intelligent scheduling of complex task flows in distributed cloud environments and offers systematic theoretical support for the in-depth application of reinforcement learning methods in cloud resource management scenarios.

From the overall research content, the critical task priority execution optimization framework constructed in this paper aligns well with the development needs of modern cloud computing platforms evolving towards intelligence, adaptability, and high-reliability collaboration. By introducing critical path awareness and long-term value assessment mechanisms into the scheduling process, the proposed method can more accurately identify core tasks that decisively impact the overall workflow completion quality and achieve more forward-looking scheduling decisions under the constraints of heterogeneous resources across multiple nodes. This not only improves resource organization capabilities during workflow execution but also enhances the system's overall coordination capabilities in the face of dynamic load fluctuations, task dependency coupling, and complex communication overhead. Related research results demonstrate that this method exhibits significant comprehensive advantages across multiple evaluation dimensions, further illustrating the effectiveness and feasibility of value-iterative reinforcement learning in distributed cloud workflow scheduling scenarios.

At the application level, this research has strong practical significance and promotional value. With the rapid development of scenarios such as cloud manufacturing, industrial internet, edge-cloud collaborative computing, large-scale data analysis, intelligent operation and maintenance, and online service orchestration, workflow scheduling has become a crucial fundamental issue affecting system timeliness, stability, and resource utilization efficiency. The proposed method provides more flexible and globally-aware decision support for task organization and execution control in such scenarios, thereby helping to reduce the risk of congestion in critical business links, improve the continuous operation capability of complex service processes, and enhance the adaptability of cloud infrastructure to high-priority tasks and multi-objective constraints. Therefore, this research not only has academic value but also provides a methodological foundation for intelligent resource orchestration and scheduling optimization oriented towards practical business needs, playing a positive role in promoting high-quality development in related application areas.

Future research can be further developed in several directions. On the one hand, it can continue to combine more complex heterogeneous resource constraints, cross-regional collaborative environments, and multi-tenant competition scenarios to further improve the model's adaptability to the complex operating conditions of real cloud platforms, making it closer to the scheduling needs of large-scale production systems. On the other hand, task priority modeling can be more deeply integrated with service level objectives, energy consumption constraints, reliability assurance, and security isolation requirements to construct a unified intelligent orchestration framework that takes into account multi-dimensional business objectives, thereby enhancing the scalability and engineering adaptability of the method in practical deployments. Furthermore, with the continuous development of cloud-edge-device collaborative architectures and autonomous resource management mechanisms, scheduling methods oriented towards online decision-making, adaptive migration, and continuous learning will become an important direction worthy of in-depth attention. Therefore, research on distributed cloud workflow scheduling focused on prioritizing critical tasks and driving long-term value still has broad development prospects and is expected to play a more important supporting role in future intelligent cloud computing systems.

## References


[1] A. Jayanetti, S. Halgamuge, and R. Buyya, "Multi-agent deep reinforcement learning framework for renewable energy-aware workflow scheduling on distributed cloud data centers," IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 4, pp. 604-615, 2024.

[2] A. Jayanetti, S. Halgamuge, and R. Buyya, "A deep reinforcement learning approach for cost optimized workflow scheduling in cloud computing environments," in Proceedings of the 2024 Asia Pacific Conference on Computing Technologies, Communications and Networking, pp. 74-82, 2024.

[3] S. Zhang, Z. Zhao, C. Liu, et al., "Data-intensive workflow scheduling strategy based on deep reinforcement learning in multi-clouds," Journal of Cloud Computing, vol. 12, no. 1, p. 125, 2023.

[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] D. Cui, Z. Peng, K. Li, et al., "An novel cloud task scheduling framework using hierarchical deep reinforcement learning for cloud computing," Plos one, vol. 20, no. 8, p. e0329669, 2025.

[6] A. Agrawal and A. K. Pal, "Cluster and reinforcement learning-based multi-objective evolutionary algorithm for joint scheduling of virtual machines and prioritize tasks in cloud computing," Swarm and Evolutionary Computation, vol. 99, p. 102156, 2025.

[7] L. Cheng, H. He, Y. Gu, et al., "Mars: multi-agent deep reinforcement learning for real-time workflow scheduling in hybrid clouds with privacy protection," in Proceedings of the 2024 IEEE 30th International Conference on Parallel and Distributed Systems, pp. 657-666, 2024.

[8] F. Chen, "Heterogeneous Dynamic Dependency Graph Learning for Anomaly Detection in Large-Scale Backend Systems," 2024.

[9] I. H. Lai, "Causal Aware Anomaly Detection for Modeling the Evolution of Backend Operational States in Ultra Large Scale Online Services," 2025.

[10] Y. Tang, "Time-Varying Operational Graph Reasoning and Risk-Constrained Repair Planning for Tool-Augmented Large Language Model Agents," Transactions on Computational and Scientific Methods, vol. 4, no. 6, 2024.

[11] C. C. Peng, "Joint Scheduling Method for Resource Orchestration Optimization in Distributed Microservices via Hierarchical Reinforcement Learning," 2025.

[12] Y. Wang, "Reinforcement Learning Based Adaptive Scheduling Method for Throughput Improvement in Distributed Computing Environments," 2025.

[13] 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.

[14] P. Wei, Y. Zeng, B. Yan, et al., "VMP-A3C: Virtual machines placement in cloud computing based on asynchronous advantage actor-critic algorithm," Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 5, p. 101549, 2023.

[15] A. Jayanetti, S. Halgamuge, and R. Buyya, "Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments," Future Generation Computer Systems, vol. 137, pp. 14-30, 2022.

[16] G. Chen, J. Qi, Y. Sun, et al., "A collaborative scheduling method for cloud computing heterogeneous workflows based on deep reinforcement learning," Future Generation Computer Systems, vol. 141, pp. 284-297, 2023.

[17] K. Siddesha, G. V. Jayaramaiah, and C. Singh, "A novel deep reinforcement learning scheme for task scheduling in cloud computing," Cluster computing, vol. 25, no. 6, pp. 4171-4188, 2022.

[18] G. Rjoub, J. Bentahar, O. Abdel Wahab, et al., "Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems," Concurrency and Computation: Practice and Experience, vol. 33, no. 23, p. e5919, 2021.

[19] G. Zhou, R. Wen, W. Tian, et al., "Deep reinforcement learning-based algorithms selectors for the resource scheduling in hierarchical cloud computing," Journal of Network and Computer Applications, vol. 208, p. 103520, 2022.