Transactions on Computational and Scientific Methods | Vo. 5, No. 10, 2025
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Task Scheduling Research for Cloud Computing Infrastructures: A Reinforcement Learning Method Integrating Resource Occupancy Prediction, Dynamic Node State Encoding, and Placement Profit Maximization

Ruoxuan Wei[1], Daniel Qiao[2]
[1]Trine University, Virginia Reston, USA
[2]University of Kentucky, Lexington, USA
*Corresponding author: Ruoxuan Wei; ruoxuanwei.roxanne@gmail.com

Abstract: To address the challenges of random task arrival, fluctuating resource consumption, continuous node state evolution, and the difficulty of balancing immediate feasibility with long-term placement benefits in cloud computing facilities, this paper proposes a task scheduling method that integrates resource consumption prediction, dynamic node state encoding, and reinforcement learning decision-making. First, a resource consumption prediction mechanism is constructed based on historical task demand information to enhance the scheduling process's ability to perceive future resource pressure. Then, node load, remaining capacity, and runtime context are dynamically represented to improve the accuracy of state modeling in complex environments. Furthermore, an adaptive matching between tasks and nodes is achieved through a reinforcement learning strategy aimed at maximizing placement benefits, thereby improving resource utilization efficiency and scheduling stability. This method unifies prediction, representation, and decision-making within a single framework, enabling a more effective characterization of the coupling relationship between changing task demands and node state evolution. Comparative experimental results demonstrate that the proposed method exhibits superior overall performance in resource utilization, service assurance, energy consumption control, and scheduling response, validating its effectiveness and applicability in cloud computing task scheduling scenarios.

Keywords: Cloud task scheduling; resource usage modeling; dynamic state representation; placement reward optimization

1. Introduction

With the continuous development of the digital economy, artificial intelligence services, and large-scale online applications, cloud computing infrastructure has gradually become a core foundational platform supporting data processing, model training, business deployment, and elastic service provision. In multi-tenant, high-concurrency, and heterogeneous resource environments, task scheduling is no longer a simple resource allocation problem, but a critical link affecting computing performance, resource utilization, service latency, energy consumption, and platform stability. Especially given the realities of sudden task arrivals, dynamic resource demands, and time-varying node loads, traditional scheduling methods relying on static rules or single-step decisions are insufficient to meet the comprehensive requirements of modern cloud computing infrastructure for high efficiency, adaptability, and robustness [1, 2]. Therefore, conducting systematic research on task scheduling in cloud computing infrastructure not only has significant theoretical value but also substantial engineering application implications [3].

From the perspective of actual operating mechanisms, task scheduling in cloud computing infrastructure faces multiple coupled challenges. On the one hand, different tasks exhibit significant differences in computing, storage, network bandwidth, and execution time, leading to complex and non-stationary fluctuations in resource demands [4]. On the other hand, node states evolve continuously with historical load, resource consumption, queue backlog, and task migration, requiring scheduling decisions to possess the ability to continuously perceive and dynamically respond to environmental changes. Simultaneously, scheduling objectives have expanded from simply allocating tasks to a combined consideration of resource fragmentation control, node balancing, revenue enhancement, and long-term operational efficiency optimization. Table 1 summarizes several key issues and their significance in current cloud computing task scheduling research. These issues collectively demonstrate that building an intelligent scheduling framework oriented towards dynamic environments has become an important direction in cloud computing resource management research.

Table 1. Key issues and research significance in task scheduling of cloud computing facilities

Key Issue	Specific Manifestation	Research Significance
Dynamic changes in resource demand	Task arrival times, resource occupancy intensity, and execution duration exhibit significant fluctuations.	Improves adaptability to complex workload patterns.
Time-varying evolution of node states	Residual resources, historical load, and congestion conditions of nodes change continuously.	Enhances perception of operational context in scheduling decisions.
Multi-dimensional coupling of scheduling objectives	Throughput, latency, energy consumption, load balancing, and placement revenue constrain one another.	Promotes the shift from single-objective optimization to multi-objective collaborative optimization.
Conflict between local decisions and long-term revenue	A placement that is optimal at the current moment may not lead to globally optimal system operation.	Strengthens sequential decision-making capability oriented toward long-term revenue.
Complexity of heterogeneous infrastructure environments	Computing, storage, and network capacities differ across nodes.	Improves the generalization ability and engineering applicability of scheduling methods.

Furthermore, in cloud computing facilities, simply placing tasks based on the instantaneous resource availability at the current moment often fails to accurately reflect future load evolution trends and potential scheduling benefits. Resource occupancy prediction can characterize changes in task demand in advance over time, providing the scheduling system with a more forward-looking decision-making basis. Dynamic encoding of node states helps transform scattered, heterogeneous, and continuously changing operational information into a unified state representation that can be modeled, thereby improving the scheduling model's ability to express complex environmental characteristics. Building on this, introducing reinforcement learning methods oriented towards maximizing placement benefits can break through the dependence of traditional heuristic strategies on manual rules, shifting the scheduling process from static matching to adaptive optimization oriented towards long-term rewards. This research approach helps to organically unify predictive ability, state awareness ability, and decision optimization ability, providing new theoretical support for high-quality task scheduling in cloud computing facilities [5].

From a research perspective, research on task scheduling focusing on resource occupancy prediction, dynamic encoding of node states, and reinforcement learning methods for maximizing placement benefits can have a strong driving effect on both theoretical and applied levels. At the theoretical level, this research direction helps deepen the understanding of the intrinsic relationship between dynamic resource management, time-varying environment modeling, and long-term decision optimization, and promotes cloud computing scheduling research from experience-driven to data-driven and decision-intelligent collaborative driving. At the application level, this research direction helps improve the resource utilization efficiency and service carrying capacity of cloud computing facilities, alleviate practical problems such as node hotspot concentration, resource fragmentation accumulation, and insufficient scheduling benefits, and provide support for the stable operation of intelligent computing infrastructure, cloud-edge collaborative platforms, and large-scale model service systems. Therefore, conducting the above-mentioned research on task scheduling of cloud computing facilities not only aligns with the evolution trend of intelligent infrastructure but also has clear and far-reaching practical value.

2. BackGround

The task scheduling problem in cloud computing facilities is essentially a combinatorial optimization problem for resource-constrained environments. Its core lies in determining the appropriate execution location and scheduling timing under given constraints, based on task attributes and infrastructure status. Compared to traditional static computing environments, cloud computing platforms typically feature resource pooling, on-demand provisioning, virtualization isolation, and multi-tenant sharing, making the scheduling process simultaneously influenced by factors such as computing power, memory capacity, network bandwidth, storage access, and service level requirements [6]. Due to significant differences in task types, arrival frequencies, lifecycles, and resource utilization patterns, the scheduling system not only needs to match resources but also handle issues such as inter-task contention, inter-node load differences, and system-level performance fluctuations. Therefore, task scheduling has evolved from a simple job assignment mechanism into a crucial foundational module connecting resource management, service assurance, and system optimization.

Against this backdrop, research on cloud computing task scheduling has gradually evolved from rule-driven to data-driven approaches, and from instantaneous response to sequential decision-making. Early methods often relied on fixed priorities, heuristic search, or threshold control, offering advantages such as simplicity and ease of deployment [7]. However, in complex scenarios, they often struggled to fully characterize the relationship between task behavior and environmental evolution. With the continuous development of monitoring data acquisition capabilities and intelligent modeling methods, scheduling systems are increasingly utilizing historical operational trajectories, node state sequences, and task requirement characteristics for modeling, thereby enhancing their ability to identify future load changes and resource competition trends. Especially under conditions of widespread heterogeneous nodes, rapidly changing business loads, and continuously refined service objectives, how to construct a modeling framework that can uniformly represent task requirements, node states, and scheduling benefits has become a key fundamental issue in cloud computing infrastructure task scheduling research.

3. Methodology

The proposed method is designed to couple future resource awareness, context-sensitive node representation, and long-horizon placement optimization within a unified scheduling framework. Unlike static heuristics that rely only on instantaneous residual capacity, the present formulation models each scheduling step as a state evolution process in which task demand, node condition, and expected placement utility interact over time. Let the cloud facility at decision step t contain a task set $\mathcal{Q}_t = \{q_1, q_2, \ldots, q_{N_t}\}$ and a node set $\mathcal{M}_t = \{m_1, m_2, \ldots, m_K\}$, where each task is characterized by requested CPU, memory, bandwidth, and expected duration, while each node maintains multi-dimensional operational statistics collected from the infrastructure monitor. This paper presents the overall model architecture, as shown in Figure 1.

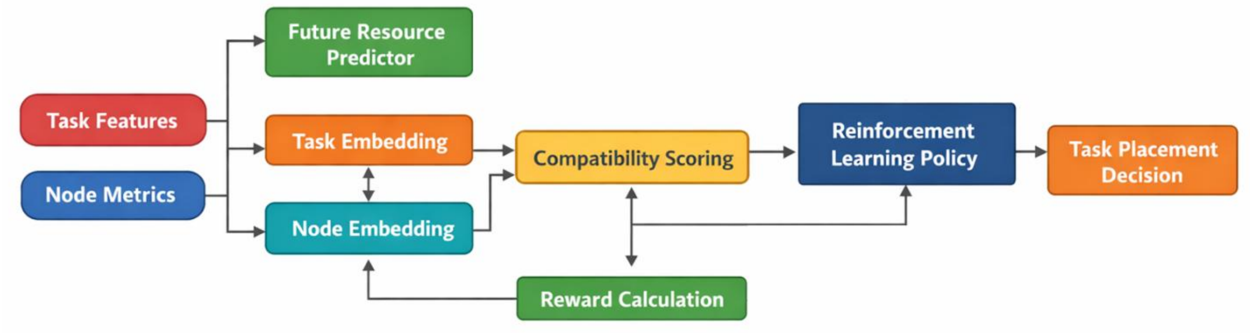

Figure 1. Overall model architecture

To preserve the temporal structure of resource pressure, the arriving task q_i is first represented as a demand vector $\boldsymbol{d}_i^t \in \mathbb{R}^F$ and then projected into a predictive demand space so that the scheduler can react to near-future contention rather than only to current load snapshots:

$$\widehat{\boldsymbol{d}}_i^{t+\tau} = f_{pred}\left(\boldsymbol{d}_i^t, \boldsymbol{d}_i^{t-1}, \dots, \boldsymbol{d}_i^{t-L+1}\right)$$

Given this prediction, the method estimates the occupancy pressure induced by task arrival over a short planning horizon, where $\boldsymbol{\rho}_k^t \in \mathbb{R}^F$ denotes the current resource occupancy of the node m_k and $\boldsymbol{c}_k \in \mathbb{R}^F$ denotes its total capacity:

$$\boldsymbol{o}_{i,k}^t = \frac{\boldsymbol{\rho}_k^t + \widehat{\boldsymbol{d}}_i^{t+\tau}}{\boldsymbol{c}_k}$$

Such a formulation is introduced to reduce myopic placement behavior, because an apparently feasible node may become highly congested shortly after assignment when temporally correlated requests continue to arrive. Rather than treating node status as an unordered feature list, the method further embeds historical utilization, queue length, migration count, and recent contention variation into a dynamic state encoder, which allows the scheduling policy to distinguish between transient idleness and structurally stable availability. Formally, the representation of the node m_k is generated by combining its recent observation sequence $\boldsymbol{z}_k^{t-H+1:t}$ with the predicted occupancy signal through a recurrent context function:

$$\boldsymbol{h}_k^t = f_{enc}\left(\boldsymbol{z}_k^{t-H+1:t}, \boldsymbol{o}_{i,k}^t\right)$$

Because resource competition in cloud facilities is shaped not only by remaining capacity but also by hidden operational tendencies, this dynamic encoding step improves the separability between nodes that are similarly loaded at the current moment yet behave differently in subsequent periods.

The scheduler then integrates task demand semantics and node context semantics into a placement compatibility score, aiming to describe whether a given task can be assigned to a given node with favorable long-term consequences. Meanwhile, the task-side embedding $\boldsymbol{u}_i^t$ is obtained from the predicted demand profile and task metadata, and the node-side embedding $\boldsymbol{h}_k^t$ is reused as the infrastructure context, so that both instantaneous matching and temporal adaptability are reflected in the same decision space. Specifically, the pairwise utility between task q_i and node m_k is defined as:

$$s_{i,k}^t = \boldsymbol{u}_i^{t\top} \boldsymbol{W}_s \boldsymbol{h}_k^t + b_s$$

By maximizing this compatibility before action selection, the method encourages assignments that remain beneficial beyond the current step, which is essential in environments where one poor placement may propagate into queue growth, hotspot concentration, and later migration overhead. Once all candidate utilities are obtained, infeasible nodes are masked according to hard constraints on capacity, isolation policy, and service requirements, thereby ensuring that subsequent optimization remains consistent with practical scheduling rules.

A reinforcement learning policy is adopted to transform compatibility estimation into sequential decision making, since the quality of a placement must be judged by its influence on future resource availability rather than by an isolated local score alone. During policy execution, the environment state $\boldsymbol{s}_t$ is formed by aggregating task-side information, encoded node states, and facility-level statistics, while the action a_t denotes the selected hosting node for the current task. The policy distribution is produced by applying a masked softmax over the candidate utilities:

$$\pi_\theta(a_t = k|\mathbf{s}_t) = \frac{\exp\left(s_{i,k}^t\right) \cdot \mathbb{I}_k^t}{\sum_{j=1}^{K} \exp\left(s_{i,j}^t\right) \cdot \mathbb{I}_j^t}$$

To align the scheduler with placement benefit maximization, the reward is constructed from resource efficiency, load balance, delay control, and fragmentation suppression, where ΔU_t measures utilization improvement, ΔB_t denotes balance enhancement, C_t is the scheduling cost, and F_t captures post-placement fragmentation:

$$r_t = \lambda_1 \Delta U_t + \lambda_2 \Delta B_t - \lambda_3 C_t - \lambda_4 F_t$$

This reward design is meaningful because cloud scheduling rarely pursues only feasibility; instead, the operational objective is to place tasks in a way that preserves future scheduling freedom and improves overall facility effectiveness. Beyond immediate gain, the learning objective is defined as the discounted return so that the agent explicitly values future consequences induced by current assignments:

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

Finally, policy optimization is carried out by iteratively updating the scheduler parameters according to trajectory feedback from the cloud environment, which enables the decision model to continuously refine its preference over heterogeneous nodes under changing load conditions. In practice, the state encoder and the policy network are trained jointly so that predictive resource signals, dynamic node context, and action utility can be adjusted toward the same scheduling objective rather than being optimized in isolation. For stable learning, the policy gradient is computed with an advantage term A_t that evaluates whether the selected placement performs better than the expected baseline under the current state:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|\mathbf{s}_t) A_t\right]$$

Consequently, the overall framework forms a closed loop in which resource occupancy prediction provides anticipatory demand awareness, dynamic node state encoding supplies context-rich infrastructure representation, and reinforcement learning converts placement evaluation into long-horizon optimization. From a methodological perspective, this design strengthens the connection between temporal perception and scheduling action, thereby making the decision process more suitable for cloud facilities with fluctuating arrivals, heterogeneous resources, and sustained pressure on placement benefit.

4. Experimental Results and Analysis

4.1 Dataset

This paper uses Microsoft Azure Traces as the foundation for its experimental data. This dataset, officially released in the Azure Public Dataset repository, represents real-world cloud platform virtual machine runtime trajectory data open to the research community, possessing a clear open-source acquisition channel and high engineering realism. Publicly available information indicates that this type of data covers virtual machine requests, lifecycles, resource specifications, and key information related to placement and packaging, effectively reflecting the actual characteristics of task arrival, resource usage changes, and scheduling pressure evolution in cloud computing facilities. Compared to static resource tables or manually constructed samples, this dataset is more suitable for supporting research settings focused on task scheduling, resource prediction, and placement optimization.

From the perspective of relevance to the paper's theme, this dataset shows strong consistency with research on task scheduling in cloud computing facilities. On one hand, the data includes virtual machine resource requirements, durations, and runtime trajectories of different types of requests, which can be used to build resource usage prediction modules and characterize dynamic features on the task side. On the other hand, the data support modeling node load states, resource stress levels, and changes in post-placement benefits, thus providing fundamental input for dynamic node state encoding and reinforcement learning decision-making. Due to its openness, representativeness, and strong adaptability to task scheduling, research based on this dataset can realistically depict the placement decision problem in the cloud environment and enhance the credibility of the paper in terms of data source and research scenario.

4.2 Experimental Results and Analysis

To highlight the differences between the proposed method and representative studies in the same field in cloud computing task scheduling, resource allocation, and virtual machine placement scenarios, a cross-sectional analysis of recent research with strong relevance was conducted. The selected literature focuses on cloud resource scheduling driven by reinforcement learning, deep reinforcement learning, or intelligent optimization, effectively covering similar research paths such as task scheduling, resource management, virtual machine scheduling, and joint placement optimization. Based on a unified comparative perspective, a comparison table was further constructed from four dimensions: resource utilization, service default, energy consumption, and response latency, to more clearly present the relative performance of different methods in terms of comprehensive scheduling capabilities. The experimental results are shown in Table 2.

Table 2. Experimental results compared with other models

Method	UR	SLAV	EC	RT
Peng et al. [8]	78.46	3.92	145.83	1.84
Zhang et al. [9]	80.15	3.51	139.27	1.72
Guo et al. [10]	82.64	3.08	133.45	1.61
Tong et al. [11]	83.27	2.96	131.88	1.57
Wang et al. [12]	84.51	2.73	126.94	1.49
Sheng et al. [13]	85.08	2.61	124.37	1.46
Liu et al. [14]	85.74	2.44	121.58	1.41
Ours	88.63	1.96	116.42	1.28

Table 2 lists the evaluation metrics that measure the comprehensive performance of the task scheduling method from four aspects: resource utilization efficiency, service quality assurance, energy consumption, and scheduling response capability. Specifically, UR reflects the degree to which resources in cloud computing facilities are effectively utilized; a higher value indicates that the scheduling strategy can reduce idle time and improve overall carrying capacity. SLAV characterizes service level agreement (SLA) breaches; a lower value indicates less performance fluctuation and resource shortage during task execution, and stronger service stability. EC measures the overall energy consumption level during scheduling; a lower value indicates better energy-saving capabilities in resource organization and node allocation. RT describes the time overhead of scheduling decisions; a lower value means the method can complete task placement more efficiently, making it more suitable for dynamically arriving and continuously changing cloud environments.

The overall results show that the proposed algorithm exhibits strong advantages in all four metrics, demonstrating that the method can improve resource utilization efficiency while simultaneously addressing service quality constraints, energy consumption control, and scheduling timeliness. The reason for this is that this method not only allocates tasks based on the current instantaneous resource availability, but also further combines resource occupancy prediction, dynamic encoding of node states, and a decision-making mechanism oriented towards long-term benefits to more fully model the relationship between task requirements and node evolution. This design enables the scheduling process to more accurately identify high-quality placement locations, reduce ineffective allocation, local congestion, and resource fragmentation accumulation, thereby achieving better overall scheduling performance under the complex operating conditions of cloud computing facilities. Furthermore, a visualization experiment demonstrating the correlation between resource usage prediction error and task placement quality is presented, as shown in Figure 2.

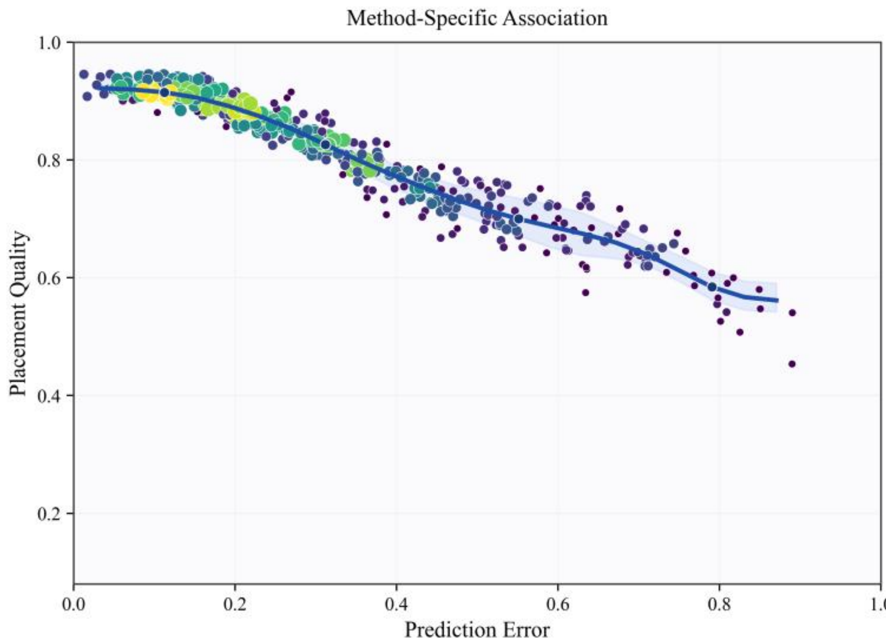

Figure 2. Visualization experiment on the correlation between resource usage prediction error and task placement quality

For task scheduling in cloud computing facilities, the accuracy of resource demand estimation directly impacts node matching, placement benefit judgment, and the stability of subsequent scheduling behavior. The results in the figure show that the proposed algorithm maintains good task placement quality even under varying prediction errors, indicating a strong synergy between the constructed resource occupancy prediction mechanism and the dynamic encoding process of node states. This also demonstrates the method's good state awareness and decision-making adaptability in complex operating environments.

From a methodological perspective, this result further illustrates that the proposed algorithm does not treat resource prediction as an isolated step, but rather embeds it into a unified framework of task representation, node representation, and policy decision-making, enabling prediction information to truly participate in the formation of placement quality. Because the reinforcement learning strategy comprehensively considers node context, future resource pressure, and long-term benefits during decision-making, the overall placement quality remains at a relatively high level even with certain prediction biases. This indicates that the proposed algorithm can effectively alleviate the scheduling imbalance problem caused by insufficient local information and demonstrates good comprehensive performance in terms of task allocation rationality, node utilization effectiveness, and scheduling process stability.

To verify the role of each component module in the overall scheduling framework, ablation analysis was further conducted, focusing on resource occupancy prediction, dynamic node state encoding, and strategy optimization mechanisms oriented towards long-term benefits. Table 3 presents the performance under different ablation settings, illustrating the impact of each key design on resource utilization efficiency, service stability, energy consumption control, and scheduling response capabilities.

Table 3. Ablation study results of different model settings

Ablation Setting	UR	SLAV	EC	RT
w/o Resource Occupancy Prediction	86.94	2.29	119.84	1.35
w/o Dynamic Node State Encoding	86.27	2.37	120.96	1.37
w/o RL-based Long-horizon Placement Policy	85.88	2.41	122.15	1.39
Ours	88.63	1.96	116.42	1.28

The ablation results in Table 3 demonstrate that the proposed algorithm achieves superior overall scheduling performance under the coordinated framework of the entire structure. Resource occupancy prediction, dynamic node state encoding, and the long-term placement strategy based on reinforcement learning all play crucial roles. Resource occupancy prediction enhances the scheduling process's ability to perceive future resource pressure, dynamic node state encoding improves the ability to express complex operational contexts, and the long-term placement strategy further ensures that task allocation not only meets current feasibility requirements but also considers subsequent scheduling benefits and system stability. It is precisely because these three parts work together effectively within a unified framework that the proposed method exhibits superior overall performance in terms of resource utilization, service assurance, energy consumption control, and scheduling efficiency.

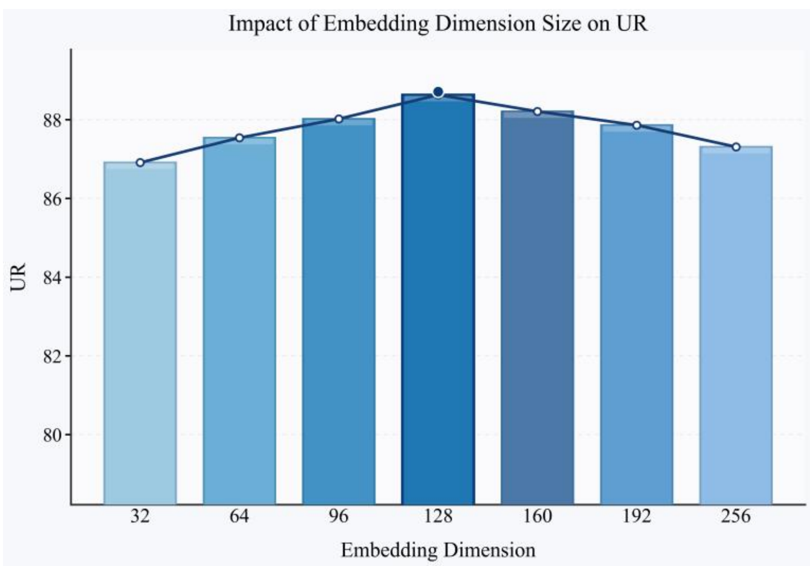

Figure 3. The impact of embedding dimension size hyperparameter on UR

This Figure 3 illustrates the direct impact of embedding dimension on resource utilization efficiency, demonstrating that the capacity configuration of the representation space affects the matching quality between task features, node states, and placement decisions. When the embedding dimension is too small, the model's ability to characterize complex resource relationships and dynamic contexts is limited, making it difficult to fully express the fine-grained differences between task requirements and node states. Conversely, when the dimension is increased to a suitable range, the feature representation capability is enhanced, thus more effectively supporting subsequent compatibility assessments and strategy decisions. Furthermore, the proposed method achieves good resource utilization results with a moderate embedding dimension setting, indicating that the current method achieves a reasonable balance between representation capability and scheduling stability. This also demonstrates the crucial role of embedding space design in improving the task scheduling quality of cloud computing facilities.

5. Conclusion

This paper addresses the key challenges in cloud computing infrastructure, including the dynamic nature of task arrival, the significant time-varying nature of node operating states, and the difficulty in balancing immediate and long-term placement benefits. It proposes a task scheduling method that combines resource occupancy prediction, dynamic node state encoding, and a reinforcement learning strategy oriented towards maximizing placement benefits. Starting from the real-world operating mechanisms of cloud computing infrastructure, this research integrates task demand evolution, node context changes, and scheduling decision feedback into a unified modeling framework. This overcomes the limitations of traditional static matching methods that rely solely on instantaneous resource availability for decision-making, enabling the scheduling process to simultaneously perceive future resource pressure, characterize dynamic node states, and optimize long-term placement effects. Overall, the proposed method not only strengthens the intrinsic connection between task representation, node representation, and sequential decision-making at the theoretical level but also provides a relatively complete and engineering-interpretable implementation path for intelligent scheduling research in complex cloud environments.

From a methodological perspective, the resource occupancy prediction module enhances the scheduling system's ability to anticipate future load changes, allowing task placement to move beyond local feasibility assessments at the current moment and further consider short-term resource pressure and potential congestion risks. The node state dynamic encoding module enhances the organization of multi-dimensional operational information of cloud computing facilities, enabling historical load, resource consumption, queue status, and operational fluctuations to be uniformly mapped into more discriminative state representations, thus providing more comprehensive contextual support for subsequent decisions. Building upon this, a reinforcement learning strategy oriented towards long-term gains is introduced, allowing the scheduling model to continuously adjust placement preferences in multi-step interactions, balancing multiple objectives such as resource utilization efficiency, service stability, energy consumption control, and response efficiency. This framework, comprised of prediction, representation, and decision-making collaboration, demonstrates strong systematicity and completeness, further illustrating that cloud computing task scheduling research is shifting from local rule optimization to intelligent collaborative optimization oriented towards global gains.

From an application perspective, this research has significant reference value for intelligent computing infrastructure, virtual machine resource management platforms, cloud-edge collaborative computing systems, and cloud deployment environments for large-scale model services. With the continuous growth of large-scale online businesses, real-time inference services, and high-density heterogeneous computing tasks, cloud computing facilities are placing higher demands on the refinement, real-time performance, and adaptive capabilities of task scheduling. This paper's method emphasizes the ability to characterize changes in task requirements, represent complex node states, and optimize long-term placement benefits. Therefore, it provides theoretical support for improving cloud platform resource carrying efficiency, alleviating local node hotspots, and reducing service fluctuation risks caused by unreasonable placement. Simultaneously, this research provides a methodological foundation for building a cloud resource management system with higher autonomy, helping to promote the development of cloud computing facilities from traditional resource orchestration models towards an integrated approach of intelligent perception, intelligent decision-making, and intelligent optimization.

Future research can be further deepened in several directions. First, it can further address more complex heterogeneous computing environments by studying a unified scheduling modeling mechanism that integrates computing, storage, network, and energy consumption characteristics to adapt to real-world scenarios where multiple types of hardware resources coexist. Second, it can combine more granular business semantics and service constraints to explore joint scheduling methods for service chains, containerized workloads, and multi-tenant isolation requirements, thereby enhancing the model's adaptability in practical platforms. Third, it can further study global resource orchestration problems under conditions of cross-data center, cross-edge node, and cloud-edge-device collaboration, extending task scheduling from single-facility internal optimization to broader collaborative optimization. Fourth, continuous progress can be made in terms of security, robustness, and interpretability, enabling scheduling strategies to not only possess strong performance but also maintain stability and reliability under abnormal loads, sudden business surges, and complex operating environments. Overall, research on task scheduling focusing on resource utilization prediction, dynamic state representation, and long-term benefit decision-making has a continuously expanding theoretical space and broad application prospects, and will play an increasingly important supporting role in the construction of future intelligent cloud infrastructure.

References

[1] C. Zhang, "Adaptive Multi-Tenant Resource Scheduling in Cloud Computing via Reinforcement Learning," 2024.

[2] A. Ghasemi, A. Toroghi Haghighat and A. Keshavarzi, "Enhancing Virtual Machine Placement Efficiency in Cloud Data Centers: A Hybrid Approach Using Multi-Objective Reinforcement Learning and Clustering Strategies," Computing, vol. 106, no. 9, pp. 2897-2922, 2024.

[3] M. T. Islam, S. Karunasekera and R. Buyya, "Performance and Cost-Efficient Spark Job Scheduling Based on Deep Reinforcement Learning in Cloud Computing Environments," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1695-1710, 2021.

[4] F. Cheng, Y. Huang, B. Tanpure et al., "Cost-Aware Job Scheduling for Cloud Instances Using Deep Reinforcement Learning," Cluster Computing, vol. 25, no. 1, pp. 619-631, 2022.

[5] J. Jiang, "Reinforcement Learning–Based Dynamic Decision Framework for Server Backend Resource Management," 2024.

[6] M. Arvindhan and D. R. Kumar, "Adaptive Resource Allocation in Cloud Data Centers Using Actor-Critical Deep Reinforcement Learning for Optimized Load Balancing," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 5S, pp. 310-318, 2023.

[7] S. Rahmani, V. Khajehvand and M. Torabian, "Burstiness-Aware Virtual Machine Placement in Cloud Computing Systems," The Journal of Supercomputing, vol. 76, no. 1, pp. 362-387, 2020.

[8] Z. Peng, D. Cui, J. Zuo et al., "Random Task Scheduling Scheme Based on Reinforcement Learning in Cloud Computing," Cluster Computing, vol. 18, no. 4, pp. 1595-1607, 2015.

[9] Y. Zhang, J. Yao and H. Guan, "Intelligent Cloud Resource Management With Deep Reinforcement Learning," IEEE Cloud Computing, vol. 4, no. 6, pp. 60-69, 2018.

[10] W. Guo, W. Tian, Y. Ye et al., "Cloud Resource Scheduling With Deep Reinforcement Learning and Imitation Learning," IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3576-3586, 2020.

[11] Z. Tong, H. Chen, X. Deng et al., "A Scheduling Scheme in the Cloud Computing Environment Using Deep Q-Learning," Information Sciences, vol. 512, pp. 1170-1191, 2020.

[12] B. Wang, F. Liu and W. Lin, "Energy-Efficient VM Scheduling Based on Deep Reinforcement Learning," Future Generation Computer Systems, vol. 125, pp. 616-628, 2021.

[13] J. Sheng, Y. Hu, W. Zhou et al., "Learning to Schedule Multi-NUMA Virtual Machines via Reinforcement Learning," Pattern Recognition, vol. 121, Art. no. 108254, 2022.

[14] H. Liu, X. Zhou, K. Gao et al., "An Integrated Optimization Method to Task Scheduling and VM Placement for Green Datacenters," Simulation Modelling Practice and Theory, vol. 135, Art. no. 102962, 2024.