Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.4, 2024

Learning Collaborative and Robust Scheduling Policies for Microservice Backends under Uncertainty

Jiarong Qiu[1]*

[1] *University of Southern California, Los Angeles, USA*

**Corresponding author: Jiarong. qiujiaro@usc.edu*

Abstract: With the continuous development of cloud computing, containerized deployment, and service-oriented architecture, microservice server backends face higher demands in handling high-concurrency requests, coordinating complex service dependencies, and managing dynamic resources. Traditional scheduling methods relying on static rules or local optimization are no longer suitable for operating environments with multiple coupled services, rapid state changes, and abnormal disturbances. To address this issue, this paper proposes a reinforcement learning-based collaborative scheduling and robust optimization method for microservice server backends. Starting from the overall system operation process, it unifies information such as queue state, resource consumption, dependency strength, and latency fluctuations into a scheduling state representation. Based on this, a joint decision-making mechanism is constructed, incorporating request allocation, instance scaling, and resource reallocation into a unified action space to improve overall coordination capabilities under complex service chain conditions. Furthermore, to address uncertainties in the backend environment, such as traffic surges, node degradation, and resource jitter, this paper introduces robust optimization concepts into the policy learning process. Through disturbance modeling and constraint control, it enhances the stability and adaptability of the scheduling strategy in complex operating scenarios. To ensure the engineering usability of the method, this paper simultaneously considers service quality constraints, resource capacity limitations, and system operating costs, enabling the proposed model to balance deployment rationality and control feasibility while pursuing improved throughput and response efficiency. Related research results show that the proposed method can achieve good overall scheduling performance in microservice server backend scenarios, demonstrating strong advantages in service processing capacity, resource utilization, and operational stability. This research provides a modeling approach for intelligent scheduling in microservice environments that considers global collaboration, dynamic decision-making, and robust control, and also offers methodological references for research on cloud platform backend resource management and service governance.

Keywords: Service orchestration; joint decision-making; uncertainty modeling; backend resource governance

1. Introduction

With the widespread adoption of cloud computing, containerized deployment, and service-oriented architecture, microservices have become a crucial organizational method for modern server backend systems. Compared to traditional monolithic architectures, microservices significantly improve system flexibility, maintainability, and continuous delivery capabilities by breaking down complex business logic into multiple independently deployable and scalable service units. However, while microservice architecture brings modularity advantages, it also introduces greater heterogeneity, dynamism, and coupling complexity to the backend operating environment. The number of service instances fluctuates with load, task requests are frequently forwarded between multiple services, and issues such as resource contention, call chain blocking, and local congestion propagation become more prominent. This makes server backend scheduling and optimization no longer a static configuration problem at the single-node level, but rather a complex optimization task involving global state awareness, real-time decision-making, and collaborative control[1,2].

In microservice server backend scenarios, scheduling problems not only involve the rational allocation of basic resources such as computing, storage, and network, but are also closely related to many other factors such as request routing, service dependencies, task priorities, load balancing, and fault recovery. Because services typically have significant upstream and downstream dependencies, resource constraints or scheduling imbalances in one local node often propagate outwards along the call chain, further impacting the overall system's throughput, response latency, and service stability. Especially under high concurrency, sudden traffic surges, and complex business collaboration conditions, traditional optimization methods based on dependency rule settings or static threshold controls often struggle to adapt to environmental changes, easily leading to problems such as response lag, uneven resource utilization, and insufficient global optimality[3]. Therefore, how to achieve collaborative scheduling among multiple services, nodes, and tasks under dynamic loads and complex dependency constraints has become a key research direction in the field of intelligent operation and performance optimization of microservice backends.

Reinforcement learning provides a new theoretical foundation and technical path for solving such dynamic decision-making problems. This method can autonomously learn the long-term reward relationship between states and actions through continuous interaction with the environment, thereby forming optimization strategies oriented towards global goals in complex, uncertain systems with significant feedback delays[4,5]. Compared to traditional methods that rely on human experience to construct rules, reinforcement learning is more suitable for handling the characteristics of high-dimensional state spaces, long decision chains, diverse resource constraints, and continuously evolving systems in microservice backends. Building upon this foundation, introducing the concept of collaborative scheduling into a reinforcement learning optimization framework helps overcome the limitations of single-point and localized decision-making, enabling multiple service instances, scheduling units, and resource management modules to achieve coordinated optimization under a unified goal. Simultaneously, considering the risks of traffic disturbances, node anomalies, resource jitter, and service degradation in real-world backend environments, integrating robust optimization

mechanisms into the reinforcement learning scheduling process also helps improve the stability, generalization, and anti-interference capabilities of the strategy in complex environments[6].

Researching reinforcement learning, collaborative scheduling, and robust optimization for microservice server backends has significant theoretical and practical value. Theoretically, this research helps promote the deep integration of reinforcement learning in the field of distributed system scheduling, enriches the research system of intelligent resource management and dynamic control in microservice scenarios, and provides new ideas for joint decision-making modeling in complex backend environments. From an application perspective, this research can provide more adaptive scheduling optimization methods for modern data centers, cloud platforms, and enterprise-level service systems, promoting improved server resource utilization efficiency, enhanced service quality assurance capabilities, and reduced system operational risks. With the continuous development of intelligent computing infrastructure, the requirements for real-time performance, stability, and elastic regulation capabilities of backend systems are constantly increasing[7]. Research on reinforcement learning, collaborative scheduling, and robust optimization in microservice scenarios not only aligns with the development trend of intelligent operation and maintenance and autonomous decision-making, but also provides important support for building efficient, stable, and sustainably evolving server backend systems.

2. Methodology

To address the highly coupled decision process in microservice backends, the scheduling problem is formulated as a collaborative Markov decision process in which service instances, computing nodes, and incoming requests are optimized under a shared control objective. Here, the overall model architecture is also presented, as shown in Figure 1.

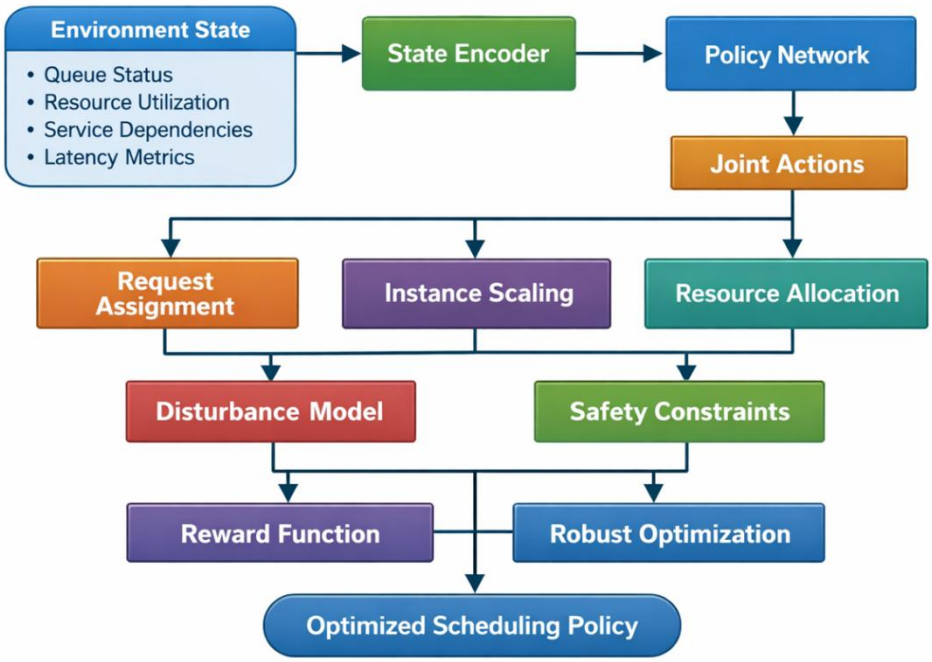

Figure 1. Overall model architecture

At time step t, the global state integrates queue pressure, resource occupancy, service dependency intensity, and latency variation so that the policy can react not only to local congestion but also to cross-

service interference that propagates along invocation paths. This design is necessary because a backend scheduler that focuses on isolated utilization signals often misses the structural origin of cascading delay, whereas a unified state representation makes it possible to align local dispatch decisions with system-level stability. The environmental state is defined as:

$$s_t = [q_t; u_t; d_t; l_t] \tag{1}$$

where q_t denotes the queue backlog vector, u_t represents multi-resource utilization, d_t encodes service dependency intensity, and l_t characterizes end-to-end latency fluctuation. Based on this state, the scheduler selects a joint action that simultaneously determines request assignment, instance scaling, and execution redistribution across available nodes:

$$a_t = \pi_\theta(s_t) \tag{2}$$

with π_θ denoting a parameterized policy that maps the observed backend condition to a coordinated control response. Such a formulation is meaningful because it transforms fragmented operational decisions into a closed-loop optimization process, thereby enabling the scheduler to learn how present actions influence future congestion, resource contention, and service continuity over a long horizon.

Rather than optimizing only immediate throughput, the proposed framework introduces a collaborative utility that explicitly balances responsiveness, load consistency, and scheduling overhead, which is crucial in microservice systems because aggressive short-term acceleration can easily induce instability in downstream services. The instantaneous reward is therefore constructed as:

$$r_t = -\alpha D_t - \beta L_t - \gamma C_t + \delta T_t \tag{3}$$

where D_t is the aggregated response delay, L_t measures load imbalance across service instances and nodes, C_t denotes the control cost caused by migration or scaling, and T_t reflects completed service throughput under the current decision. Since backends usually operate under long-chain dependency and delayed feedback, the policy is trained to maximize cumulative return rather than an isolated one-step gain:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H} \lambda^t r_t\right] \tag{4}$$

in which λ is the temporal discount factor, and H denotes the decision horizon. By organizing the objective in this way, the scheduler is encouraged to prefer actions that preserve long-term service quality, which strengthens the practical relevance of the method in continuously evolving backend environments where myopic decisions often amplify future recovery costs.

Beyond standard collaborative control, robustness is incorporated to reduce the vulnerability of the learned strategy to workload bursts, transient node degradation, and stochastic resource disturbances. This step is important because a policy trained only on nominal operating conditions may achieve acceptable behavior in stable periods while failing abruptly when service graphs experience demand shocks or partial infrastructure degradation. To account for such uncertainty, the transition model is described with an adversarial disturbance term:

$$s_{t+1} = f(s_t, a_t) + \epsilon_t \tag{5}$$

where $f(\cdot)$ denotes the nominal backend evolution function and ϵ_t captures perturbations induced by workload noise and runtime instability. Under this setting, the learning target is converted into a worst-case robust objective:

$$J_{rob}(\theta) = \min_{\epsilon_t \in \Omega} \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{H} \lambda^t r_t(s_t, a_t, \epsilon_t) \right] \tag{6}$$

with Ω defining the admissible disturbance set. Incorporating this criterion improves the resilience of the scheduling policy because optimization is no longer driven by average-case performance alone, but instead by the ability to maintain acceptable coordination behavior when the backend departs from its nominal operating regime.

Finally, practical deployment requires the decision process to remain compliant with hard system constraints, since backend scheduling quality cannot be judged independently of resource safety and service-level guarantees. For that reason, the collaborative robust policy is constrained by capacity and quality-of-service conditions so that learning remains aligned with deployable operating rules:

$$\min_{\theta} \; -J_{rob}(\theta) s.t. \, g(s_t, a_t) \leq 0 \tag{7}$$

where $g(\cdot)$ jointly represents resource limits, queue capacity bounds, and latency-related service constraints. Embedding constraints directly into the optimization process is meaningful because it prevents the learned controller from pursuing apparent performance gains through unsafe or infeasible actions, while also preserving the interpretability of decision boundaries in complex service orchestration scenarios. In essence, the overall method unifies state-aware collaboration, long-horizon reinforcement learning, and uncertainty-driven robust optimization within a single backend scheduling framework, thereby providing a principled foundation for improving adaptability, stability, and control reliability in microservice server environments.

3. Experimental Results and Analysis

3.1 Dataset

This paper selects the open-source dataset Anomalies in Microservice Architecture, built upon the Train Ticket microservice system, as its data foundation. This dataset, publicly released on Zenodo, provides monitoring data for microservice architecture scenarios. Its core content includes logs, Jaeger call chain tracing information, and Prometheus metrics, comprehensively reflecting the dynamic characteristics of the backend system during service calls, resource fluctuations, and changes in operational status. Since Train Ticket is a typical ticketing business system comprising 41 microservices, covering various backend interactions such as querying, booking, payment, and order processing, its data structure is highly consistent with the collaborative scheduling problem of microservice server backends, providing reliable support for state modeling, dependency characterization, and scheduling decision learning.

Compared to data sources containing only single performance metrics or static resource records, this dataset simultaneously preserves service-level observation information, link-level propagation information, and anomaly perturbation information, making it more suitable for reinforcement learning, collaborative scheduling, and robust optimization research. On the one hand, multi-source monitoring

data helps construct a comprehensive state representation of queue pressure, resource consumption, service dependency strength, and latency changes, enabling scheduling strategies to perceive the operational status of microservice backends from a global perspective. On the other hand, the anomaly and version change information contained in the dataset can provide a basis for modeling uncertain disturbances, thereby enhancing the adaptability of scheduling strategies to sudden loads, local degradation, and chain-like performance fluctuations. Therefore, this open-source dataset maintains a strong match with the theme of reinforcement learning, collaborative scheduling, and robust optimization for microservice server backends in terms of task scenarios, system architecture, and data modality.

3.2 Experimental setup

To verify the effectiveness of the proposed reinforcement learning-based collaborative scheduling and robust optimization method for microservice server backends, experiments were conducted on open-source backend runtime data based on a microservice architecture. A unified configuration was implemented for state modeling, joint action decision-making, robust perturbation injection, and constraint optimization processes. The overall experimental environment employed a collaborative training approach using a single-policy network and a single-value evaluation network. Service queue state, resource utilization, dependency strength, and latency fluctuation information were used as state inputs, while request allocation, instance scaling, and resource reallocation were used as joint action outputs. Perturbation simulation was introduced during training to enhance the policy's adaptability to dynamic loads and local anomalies. The optimizer Adam was chosen, with a discount factor characterizing long-term returns. Batch size and the number of training epochs were used to balance policy convergence stability and training efficiency. Specific experimental settings are shown in Table 1.

Table 1. Experimental setup

Parameter Category	Parameter Name	Setting Value
Policy Network	Number of Hidden Layers	2
Policy Network	Hidden Units	128, 64
Value Network	Number of Hidden Layers	2
Value Network	Hidden Units	128, 64
Optimizer	Optimization Algorithm	Adam
Optimizer	Learning Rate	0.001
Reinforcement Learning Parameters	Discount Factor	0.99
Reinforcement Learning Parameters	Batch Size	64
Reinforcement Learning Parameters	Training Episodes	200
Reinforcement Learning Parameters	Maximum Decision Steps	100

3.3 Experimental Results and Analysis

To more comprehensively characterize the position of reinforcement learning-based collaborative scheduling and robust optimization methods for microservice server backends within the relevant research spectrum, this paper selects several representative works closely related to microservice resource management, elastic scaling, concurrency control, and cloud scheduling optimization for horizontal comparison. These methods all revolve around dynamic load awareness, service-level resource allocation, performance constraint maintenance, and system adaptive control in their problem settings, showing strong consistency with the collaborative decision-making scenario of the microservice backend that this paper focuses on. Based on this, the table presents a unified comparison of the related methods and the method in this paper from dimensions such as latency, throughput, resource utilization, scheduling success rate, service level achievement, CPU overhead, memory overhead, and overall cost, to conduct subsequent analysis under the same evaluation framework. The experimental results are shown in Table 2.

Table 2. Experimental results compared with other models

Method	Lat	Thr	RU	SR	SLA	CPU	Mem	Cost
Qiu et al.[8]	24.38	91.72	78.46	95.21	97.34	63.58	58.42	18.76
Liu et al.[9]	23.91	92.35	79.18	95.84	97.62	62.94	57.81	18.23
Abdel Khaleq et al.[10]	22.67	93.48	80.74	96.35	98.01	61.83	56.92	17.85
Hossen et al.[11]	21.94	94.12	81.56	96.89	98.26	60.97	55.88	17.42
Choi et al.[12]	21.36	94.85	82.41	97.14	98.53	60.15	55.07	17.18
Zhang et al.[13]	20.88	95.27	83.02	97.46	98.71	59.63	54.51	16.94
Bai et al.[14]	20.41	95.83	83.76	97.81	98.95	58.92	53.86	16.52
Ours	19.76	96.58	85.14	98.27	99.18	57.84	52.73	15.89

The evaluation metrics in the table characterize the overall performance of the scheduling method from three aspects: service performance, resource utilization, and operational overhead. Specifically, Lat represents the latency level of system requests during backend processing, reflecting the impact of the scheduling strategy on response efficiency; Thr represents the task throughput capacity that the system can complete per unit time, reflecting the overall service capacity; RU represents resource utilization, measuring whether computing resources are fully allocated; SR represents scheduling success rate, reflecting the ability of tasks to be effectively executed under complex service dependencies; SLA represents service level agreement satisfaction rate, evaluating the system's stable service capability under quality constraints; CPU and Mem represent processor and memory usage, respectively, characterizing resource consumption intensity; and Cost represents the overall cost of the system during scheduling and operation, reflecting the actual deployment cost of the method beyond performance improvement.

Overall, the algorithm proposed in this paper demonstrates good comprehensive optimization capabilities across multiple key metrics, indicating that this method can achieve more balanced and effective collaborative scheduling in microservice server backend scenarios. On the one hand, this method demonstrates outstanding performance in response efficiency, task processing capacity, and service stability, indicating its ability to effectively coordinate the relationship between request allocation, instance scaling, and resource reallocation, and to alleviate local congestion and performance fluctuations under complex call chain conditions. On the other hand, resource consumption and operating cost control are also maintained at a good level, suggesting that the constructed reinforcement learning collaborative scheduling and robust optimization mechanism not only focuses on improving service quality but also takes into account resource constraints and deployment feasibility during the backend system's operation. Therefore, it can provide valuable support for efficient, stable, and low-cost scheduling in a microservice environment.

As shown in Figure 2, the proposed algorithm maintains good throughput performance under the given hyperparameter settings, indicating that the constructed reinforcement learning cooperative scheduling mechanism can maintain strong task processing and policy adaptability under different learning rates. This demonstrates that the current method establishes a relatively stable cooperative relationship among state representation, joint action decision-making, and robust optimization constraints, enabling the model to effectively learn key scheduling information in microservice server backend scenarios and further ensuring overall service processing efficiency and operational quality.

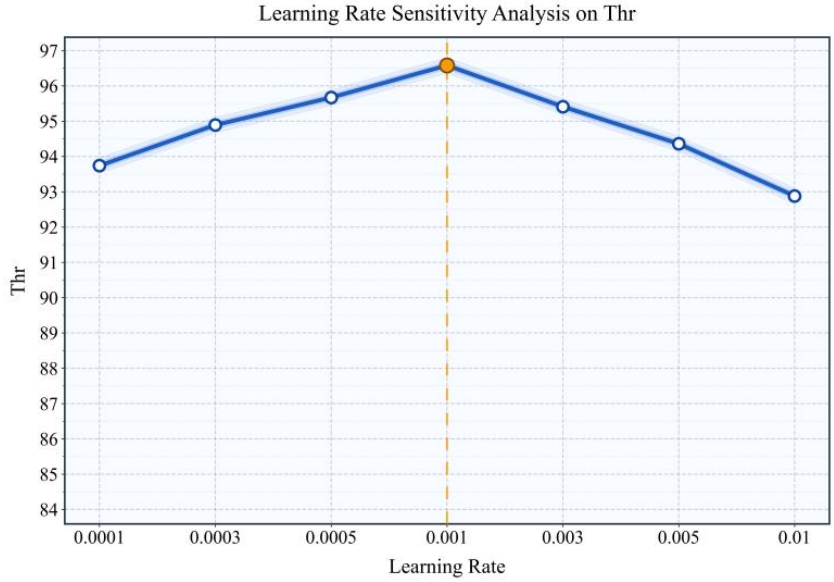

Figure 2. Experimental results of the Learning Rate hyperparameter on Thr

Figure 3 shows that the proposed algorithm exhibits good recovery capabilities under abnormal disturbance scenarios, quickly returning to a stable scheduling range after the backend running state is impacted. This indicates that the constructed reinforcement learning collaborative decision-making mechanism can effectively perceive state changes in the microservice environment and adjust scheduling actions in a timely manner under complex service dependencies and resource constraints, thereby maintaining the overall system operating quality and service continuity. Simultaneously, the robust optimization process also demonstrates a significant suppression effect on abnormal disturbances, enabling the scheduling strategy to maintain strong adaptability and controllability in the face of local fluctuations.

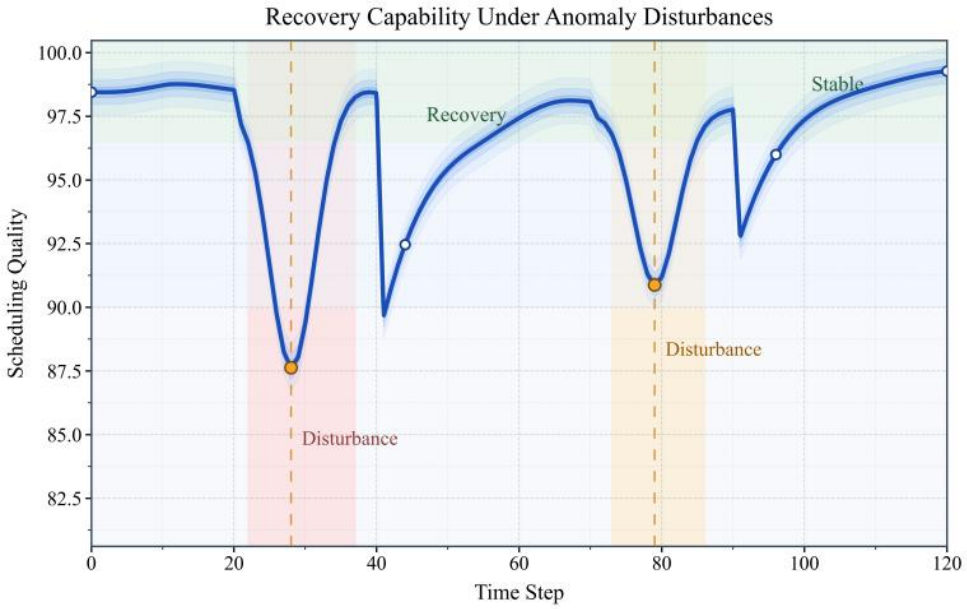

Figure 3. Visualization Experiment on the Recovery Capability of Scheduling Strategies under Abnormal Disturbance Scenarios

The proposed method not only demonstrates good abnormal scenario handling capabilities but also reflects its stable scheduling advantages in microservice server backend task organization. By incorporating request allocation, instance scaling, and resource reallocation into a unified decision framework, the system can maintain high scheduling quality even after external disturbances, indicating that the method establishes a relatively effective collaborative relationship between state representation, policy learning, and constraint control. Therefore, the proposed algorithm can provide reliable support for microservice backend scheduling under abnormal disturbance conditions and has good practical application value.

The abnormal perturbation strength coefficient determines the degree of influence of external uncertainties on scheduling learning during robust optimization. Its setting directly affects the adaptability and control stability of the policy in complex microservice backend environments. Finally, this paper conducts a sensitivity analysis around this parameter, which helps to characterize the throughput maintenance capability of the proposed method under different perturbation modeling conditions and further demonstrates the coupling relationship between the reinforcement learning cooperative scheduling mechanism and robust constraints. The experimental results are shown in Figure 4.

The proposed algorithm maintains good throughput even under varying perturbation intensity coefficients, indicating that the constructed reinforcement learning-based collaborative scheduling and robust optimization mechanism can effectively adapt to state changes caused by abnormal perturbations and maintain a high level of task processing in complex microservice backend environments. This demonstrates that the proposed method establishes a relatively stable joint decision-making relationship among request allocation, instance scaling, and resource reallocation. It also reflects the positive role of robust modeling in policy learning, thus providing reliable support for efficient scheduling and stable operation in microservice server backend scenarios.

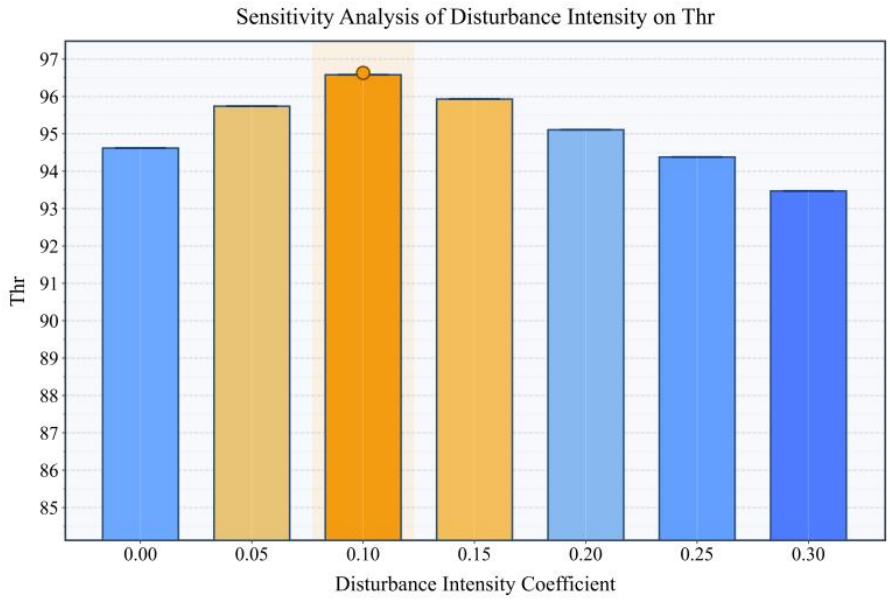

Figure 4. Sensitivity analysis of the perturbation intensity coefficient to Thr

4. Conclusion

This paper addresses the collaborative scheduling challenges faced by microservice server backends under high dynamic loads, complex service dependencies, and uncertain operating environments. It proposes a unified modeling method integrating reinforcement learning-based collaborative decision-making and robust optimization. Starting from the operational characteristics of microservice backend systems, the research incorporates key information such as queue state, resource utilization, dependency strength, and latency fluctuations into a unified state representation space. Based on this, a joint action decision-making mechanism is constructed, integrating request allocation, instance scaling, and resource reallocation into the same scheduling framework, thereby improving the overall collaborative capability of the backend system under complex service chain conditions. Simultaneously, addressing the common issues of traffic surges, local node degradation, and resource disturbances in real-world deployment environments, robust optimization is further introduced. This ensures that the learned strategies not only improve efficiency under normal operating conditions but also better adapt to the stability control requirements under abnormal interference. Overall, this work provides an intelligent solution to the microservice server backend scheduling problem that balances globality, dynamism, and reliability.

From a research perspective, the value of this paper lies not only in expanding the microservice scheduling modeling approach but also in enriching the backend intelligent control methodology. Traditional backend optimization methods often emphasize static rules, local resource allocation, or single-objective orientation. In contrast, the method proposed in this paper focuses on the synergistic effects of multiple factors in complex operational scenarios, incorporating long-term benefits, service quality constraints, and system robustness into a unified optimization perspective. This research approach helps to shift the scheduling problem in microservice scenarios from experience-driven to learning-driven, and provides clearer theoretical support for the application of reinforcement learning methods in distributed systems, autonomous operation and maintenance, and service governance. By organically combining a system state-aware decision-making mechanism with an optimization strategy oriented towards perturbation constraints, this paper further enhances the scheduling model's ability to express the complexity of real-world backend environments, making intelligent scheduling research more closely aligned with actual engineering needs.

At the application level, this research has strong potential for widespread adoption. As cloud computing platforms, enterprise-level service systems, online trading platforms, intelligent manufacturing backends, and edge service infrastructure continue to evolve towards service-oriented, containerized, and high-concurrency architectures, the scheduling efficiency and operational stability of backend systems have become crucial factors affecting overall service capabilities. The proposed method provides more flexible and robust scheduling support for multi-service collaborative operation scenarios, which is of practical significance for improving resource utilization efficiency, enhancing service continuity, reducing the risk of anomaly propagation, and optimizing the overall system operating cost. Especially in latency-sensitive application areas that have complex business chains and exhibit significant service load fluctuations, the combination of reinforcement learning-based collaborative scheduling and robust optimization is expected to become an important technical path supporting the intelligent upgrade of backend systems. Therefore, this paper not only promotes research on microservice server backends but also provides theoretical reference and methodological guidance for related industries in building highly available, highly elastic, and highly reliable digital infrastructure.

Future research still has room for further expansion. First, existing methods can be further refined for larger-scale and more heterogeneous microservice cluster environments, enabling scheduling strategies to have stronger cross-platform adaptability and multi-layer resource collaboration capabilities. Second, research focusing on real-time online learning, adaptive constraint adjustment, and multi-objective joint optimization will help further enhance the model's continuous evolution and autonomous decision-making capabilities in open environments. Furthermore, by combining digital twins, explainable intelligent operation and maintenance, and security mechanisms to more meticulously model the decision-making basis, anomaly propagation paths, and risk control boundaries in the backend scheduling process, it is hoped that microservice intelligent scheduling can move from performance optimization to trusted control and autonomous management. Overall, research on reinforcement learning-based collaborative scheduling and robust optimization for microservice server backends still has broad development prospects, and related achievements have value for continuous deepening and expansion in cloud service governance, data center management, enterprise application support, and the construction of next-generation intelligent infrastructure.

References

[1] M. Xu, C. Song, S. Ilager, S. S. Gill, J. Zhao, K. Ye and C. Xu, "CoScal: Multifaceted scaling of microservices with reinforcement learning", IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 3995-4009, 2022.

[2] Z. Jian, 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.

[3] S. Agarwal, M. A. Rodriguez and R. Buyya, "A deep recurrent-reinforcement learning method for intelligent autoscaling of serverless functions", IEEE Transactions on Services Computing, vol. 17, no. 5, pp. 1899-1910, 2024.

[4] I. Pintye, J. Kovács and R. Lovas, "Enhancing machine learning-based autoscaling for cloud resource orchestration", Journal of Grid Computing, vol. 22, no. 4, p. 68, 2024.

[5] J. Santos, T. Wauters, B. Volckaert and F. De Turck, "gym-hpa: Efficient auto-scaling via reinforcement learning for complex microservice-based applications in kubernetes", NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, pp. 1-9, 2023.

[6] J. P. K. Santos Nunes, S. Nejati, M. Sabetzadeh and E. Y. Nakagawa, "Self-Adaptive Requirements-Driven Autoscaling of Microservices", arXiv preprint arXiv:2403.08798, 2024.

[7] K. Hu, M. Xu, K. Ye and C. Xu, "MSARS: Meta-Learning Assisted Reinforcement Scaling for Adaptive Microservice Resource Allocation", arXiv preprint arXiv:2409.14953, 2024.

[8] H. Qiu, et al., "AWARE: Automate workload autoscaling with reinforcement learning in production cloud systems", Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC 2023), 2023.

[9] J. Liu, S. Zhang and Q. Wang, " μ ConAdapter: Reinforcement learning-based fast concurrency adaptation for microservices in cloud", Proceedings of the 2023 ACM Symposium on Cloud Computing, 2023.

[10] A. Abdel Khaleq and I. Ra, "Intelligent microservices autoscaling module using reinforcement learning", Cluster Computing, vol. 26, no. 5, pp. 2789-2800, 2023.

[11] M. R. Hossen, M. A. Islam and K. Ahmed, "Practical efficient microservice autoscaling with QoS assurance", Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing, 2022.

[12] B. Choi, et al., "pHPA: A proactive autoscaling framework for microservice chain", Proceedings of the 5th Asia-Pacific Workshop on Networking, 2021.

[13] Y. Zhang, et al., "Sinan: ML-based and QoS-aware resource management for cloud microservices", Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2021.

[14] H. Bai, et al., "Drpc: Distributed reinforcement learning approach for scalable resource provisioning in container-based clusters", IEEE Transactions on Services Computing, vol. 17, no. 6, pp. 3473-3484, 2024.