

Reinforcement Learning for Dynamic Resource Allocation in Intelligent Systems

Dr. Javier Berrocal

Associate Professor, University of Salamanca, Spain

Abstract

Dynamic resource allocation is a fundamental challenge in intelligent computing systems in which computational demand, communication traffic, energy availability, queue conditions, and service priorities vary continuously. Conventional static allocation policies are simple to implement but may perform poorly when operating conditions change rapidly, while rule-based adaptive methods can become difficult to tune across heterogeneous environments. Reinforcement learning offers an alternative by allowing an agent to improve sequential allocation decisions through interaction with an environment and feedback from a reward function. This study examines the suitability of reinforcement learning for dynamic resource allocation through a conceptual review and a reproducible synthetic queueing simulation.

The experimental environment models changing workload demand, finite resource capacity, queue accumulation, allocation cost, and service-level constraints. A tabular Q-learning policy is compared with static allocation and an adaptive heuristic under four levels of demand volatility. The simulation uses a fixed random seed, explicit state discretization, five resource-allocation actions, 1,800 training episodes, and a 12,000-step evaluation horizon for each condition. Results show that differences between policies become substantially more pronounced as workload volatility increases. At the highest simulated volatility, Q-learning maintains a service-level satisfaction rate of 94.46%, compared with 62.68% for static allocation and 60.58% for the adaptive heuristic.

The findings demonstrate the potential of reinforcement learning to maintain adaptive performance under nonstationary resource demand while also highlighting the importance of reward design, safe exploration, reproducibility, scalability, and deployment-specific validation. The paper concludes that reinforcement learning is most valuable when resource management is sequential, feedback-driven, and sufficiently dynamic to justify adaptive policy learning.

Keywords: reinforcement learning, dynamic resource allocation, intelligent systems, Q-learning, adaptive computing, resource scheduling, workload management, service-level optimization

1. Introduction

Intelligent systems increasingly operate in environments where computational resources must be distributed among multiple competing tasks under continuously changing demand. Cloud platforms, edge-computing infrastructures, wireless networks, autonomous systems, cyber-physical platforms, and Internet of Things environments routinely make decisions concerning processor capacity, communication bandwidth, memory, energy, storage, and task scheduling. The central difficulty is not

merely determining how much resource is available, but deciding how limited resources should be distributed over time while maintaining acceptable latency, throughput, energy use, reliability, and service quality. Resource allocation is consequently a sequential decision problem in which present actions may influence both immediate performance and future system conditions.

Traditional allocation mechanisms generally rely on fixed thresholds, predefined rules, optimization routines, or heuristics developed from assumed workload characteristics. Such approaches remain valuable because they can be computationally inexpensive, interpretable, and reliable when operating conditions are relatively stable. Their limitations become more visible when workload arrival patterns change rapidly or when the relationship among demand, resource availability, queue formation, and quality-of-service requirements becomes nonlinear. A fixed allocation that performs efficiently under moderate load may create substantial resource waste during low demand and serious service degradation during demand surges. Similarly, a heuristic optimized for one workload distribution may respond inadequately when system conditions depart from its original assumptions.

Reinforcement learning provides a fundamentally different approach to resource management. Instead of requiring a complete mathematical model of system dynamics, an RL agent can learn a decision policy through repeated interactions with an environment. Sutton and Barto describe reinforcement learning as a framework in which an agent observes its condition, selects an action, receives feedback in the form of a reward, and progressively learns a policy that maximizes expected long-term return. Developments in deep reinforcement learning have further demonstrated that value functions and policies can be approximated in complex, high-dimensional environments. Mnih and colleagues established the effectiveness of deep Q-learning for sequential control, while later research extended related principles to computing and networking applications.

Resource-management research has provided particularly relevant evidence for this approach. Mao and colleagues introduced DeepRM, framing resource scheduling as a learning problem in which a system learns allocation behavior through experience. Their work demonstrated that reinforcement learning could produce resource-management strategies comparable with established scheduling heuristics while adapting to workload conditions. Subsequent studies have used reinforcement-learning principles for wireless communication, vehicle-to-vehicle resource allocation, dynamic channel access, and mobile-edge computation offloading. These applications show that RL is particularly attractive when allocation decisions are repeated, system states change over time, and closed-form optimization becomes computationally difficult.

The present study investigates reinforcement learning for dynamic resource allocation through a simulation-based methodological framework designed to remain transparent about the distinction between synthetic experimentation and real operational evidence. A reproducible queueing environment is developed in which workload arrivals vary stochastically and resource decisions influence queue accumulation and service-level satisfaction. Static allocation, an adaptive heuristic, and tabular Q-learning are compared under progressively increasing demand volatility. The study aims not to claim universal superiority of one algorithm but to identify the environmental conditions under which reinforcement learning can provide meaningful adaptive value and to examine the methodological and governance considerations required before such techniques are transferred into operational intelligent systems.

2. Review of Literature

2.1 Reinforcement Learning as a Sequential Decision Framework

Reinforcement learning is commonly formulated through a Markov decision process in which an agent repeatedly observes a state, performs an action, receives a reward, and encounters a new state. Unlike supervised learning, which generally learns from labeled examples, reinforcement learning learns from the consequences of actions. This structure is particularly appropriate for resource allocation because allocation decisions are sequential and the quality of an action can depend on how it influences subsequent system states.

Sutton and Barto provide the foundational formulation of value-based learning, temporal-difference methods, policy learning, exploration, and long-term return optimization. Q-learning is among the most widely recognized model-free techniques. It estimates the value of taking an action within a given state and updates that estimate from observed reward and future value.

The underlying principle is expressed as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

where $Q(s_t, a_t)$ represents the estimated action value, α is the learning rate, r_t is the immediate reward, and γ determines the relative importance of future rewards.

Mnih and colleagues substantially extended value-based reinforcement learning by combining Q-learning with neural-network function approximation. Their Deep Q-Network demonstrated that an agent could learn control policies from high-dimensional observations. Although the present study deliberately uses tabular Q-learning for methodological transparency, deep RL becomes relevant when realistic resource-management systems contain state spaces that cannot be efficiently enumerated.

2.2 Reinforcement Learning in Computing and Network Resource Management

Resource management is well suited to reinforcement learning because allocation decisions are highly repetitive and generate continuous performance feedback. Mao and colleagues' DeepRM work represented an important application of deep reinforcement learning to computer-system resource management. The study translated job scheduling with multidimensional resource requirements into a learning formulation and reported that the learned policy could adapt to workload conditions and achieve performance comparable with sophisticated heuristics.

Wireless and distributed-computing research has subsequently demonstrated related benefits. Ye, Li, and Juang developed a decentralized deep-reinforcement-learning approach for resource allocation in vehicle-to-vehicle communications. Individual communication agents selected sub-bands and transmission power using locally available information, demonstrating how distributed learning can support resource decisions without continuously requiring centralized global coordination.

Huang, Bi, and Zhang examined deep reinforcement learning for online computation offloading in wireless-powered mobile-edge computing. Their work addressed an environment where offloading and wireless-resource decisions had to adapt to time-varying channel conditions. The reported framework substantially reduced online computational complexity while approaching the performance of computationally intensive optimization approaches. These findings are particularly relevant to intelligent systems requiring decisions within tight operational time windows.

2.3 Adaptation, Reproducibility, and Multi-Agent Resource Allocation

Dynamic resource allocation frequently involves several interacting agents rather than a single centralized decision-maker. Buşoniu, Babuška, and De Schutter's survey of multi-agent reinforcement learning emphasizes that learning becomes more complicated when the environment is influenced by the simultaneous decisions of other agents. In distributed systems, telecommunications, robotics, and autonomous infrastructures, resource-allocation policies may therefore need to account for strategic interaction, cooperation, competition, and changing policies among neighboring agents.

Reproducibility represents another important methodological concern. Henderson and colleagues demonstrated that reinforcement-learning results may be sensitive to implementation choices, stochasticity, evaluation protocols, and experimental settings. Performance claims based on a single training run can therefore be unreliable. Reporting random seeds, hyperparameters, baselines, evaluation horizons, and reward definitions is necessary for meaningful interpretation.

The literature collectively indicates that reinforcement learning has substantial potential for adaptive resource allocation, but its effectiveness is conditional. The approach is most defensible where state changes are frequent, decisions are repeated, meaningful feedback can be expressed through a reward function, and experimentation can be performed without unacceptable operational risk.

3. Objectives of the Study

3.1 Primary Research Objective

The primary objective of this study is to evaluate whether reinforcement learning can provide more robust resource-allocation behavior than fixed and heuristic allocation strategies when workload demand becomes increasingly volatile. The focus is not solely on average resource utilization but on the ability of a policy to maintain service quality as operating conditions become less predictable.

A second component of the objective is methodological. The study seeks to demonstrate how an intelligent resource-allocation problem can be converted into a transparent Markov decision framework using observable workload conditions, discrete allocation actions, and an explicit multi-component reward function.

3.2 Specific Objectives

The study pursues the following objectives:

- To formulate dynamic resource allocation as a sequential reinforcement-learning problem involving system states, allocation actions, rewards, and changing workload conditions.
- To develop a reproducible synthetic queueing environment for comparing adaptive resource-allocation policies.
- To compare Q-learning with static allocation and a rule-based adaptive heuristic under progressively increasing workload volatility.
- To examine the robustness of each allocation strategy using service-level satisfaction as the principal comparative measure.
- To identify methodological and operational challenges affecting the deployment of reinforcement learning in real intelligent systems.

3.3 Research Questions

The investigation is guided by whether reinforcement learning becomes increasingly advantageous as workload variability grows and whether an RL policy can preserve service-level performance without relying on a fixed resource rule.

A further research question concerns the limits of this advantage. The analysis therefore considers whether the benefit arises from reinforcement learning itself or from particular choices concerning state representation, action resolution, reward design, benchmark configuration, and training conditions.

4. Research Methodology

4.1 System Model and Markov Decision Formulation

The research uses a synthetic discrete-time queueing environment representing an intelligent system receiving computational tasks over time. Incoming workload is stochastic. If allocated service capacity is lower than current workload, unprocessed demand accumulates in a queue and is transferred to the subsequent time step.

The resource-management problem is formulated as:

$$M=(S,A,P,R,\gamma)M=(S,A,P,R,\gamma)$$

where SS denotes the state space, AA the allocation-action space, PP the environmental transition mechanism, RR the reward function, and γ the future-reward discount factor.

The state contains two operational signals: current queue length and workload observed during the preceding decision interval. Continuous values are discretized into state bins.

Five allocation actions are available:

$$A=\{20,40,60,80,100\}A=\{20,40,60,80,100\}$$

The actions represent abstract resource-capacity units rather than physical CPU cores, gigabytes, or radio channels. This abstraction allows the model to represent several types of intelligent-resource systems without falsely implying empirical validation in a specific infrastructure.

4.2 Simulation and Learning Configuration

Synthetic demand is generated using a normal distribution centered on 50 workload units and clipped at zero to prevent negative arrivals. Four volatility levels are evaluated using standard deviations of 5, 15, 25, and 35 units. Increasing volatility therefore changes the unpredictability of demand without changing the nominal average arrival level.

Q-learning is trained for 1,800 episodes with 80 allocation steps in each episode. The learning rate is 0.15 and the discount factor is 0.92. Exploration begins with $\epsilon=1.00$ and gradually declines to a minimum value of 0.03. A fixed random seed of 42 is used.

Table 1: Synthetic Resource-Allocation Simulation Configuration

Component	Configuration
Mean workload arrival	50 abstract units
Demand volatility	5, 15, 25, and 35 standard-deviation units
Allocation actions	20, 40, 60, 80, and 100 units
RL algorithm	Tabular Q-learning

Training episodes	1,800
Steps per training episode	80
Learning rate	0.15
Discount factor	0.92
Minimum exploration rate	0.03
Evaluation horizon	12,000 steps per condition
Random seed	42
SLA condition	Post-allocation queue $\leq$ 40 units

The reward function balances service completion, resource cost, queue accumulation, and excessive-backlog penalties:

$$R_t = \text{Served}_t - 0.25 \text{Allocation}_t - 0.35 \text{Queue}_t + 1 - 10I(\text{Queue}_t + 1 > 80)$$

$$R_t = \text{Served}_t - 0.25 \text{Allocation}_t - 0.35 \text{Queue}_{\{t+1\}} - 10I(\text{Queue}_{\{t+1\}} > 80)$$

The indicator term $I(\cdot)$ adds a penalty whenever the remaining queue exceeds 80 units. These coefficients are experimental design parameters and are not claimed to represent universally optimal operational weights.

4.3 Benchmark Policies and Evaluation Procedure

The Q-learning policy is compared with two benchmarks. Static allocation assigns 60 resource units at every step regardless of current conditions.

The adaptive heuristic estimates resource need from the preceding workload and current queue:

$$\text{Target}_t = 0.75D_{t-1} + 0.45Q_t$$

The available allocation action nearest to the calculated target is selected.

The principal evaluation metric is service-level satisfaction.

A time step is counted as satisfying the simulated SLA when the post-allocation queue remains at or below 40 units:

$$\text{SLA Satisfaction} = \frac{\text{Number of compliant time steps}}{\text{Total evaluation time steps}} \times 100$$

Each policy is evaluated for 12,000 time steps at every workload-volatility level. The simulation uses synthetic data only. No human participants, proprietary infrastructure logs, customer information, or confidential operational records are involved.

5. Analysis

5.1 Comparative Performance Across Demand Conditions

At the lowest volatility level, static allocation and Q-learning both achieve complete SLA satisfaction in the simulation. Static allocation records 100.00%, while Q-learning also records 100.00%. The adaptive heuristic achieves 91.46%. This result is significant because it demonstrates that reinforcement learning does not necessarily create meaningful operational advantage when the environment is already sufficiently predictable for a simple policy to perform well.

At the second volatility level, Q-learning maintains 99.87% SLA satisfaction compared with 98.52% for static allocation and 76.63% for the heuristic. The difference between Q-learning and static

allocation remains relatively small. The important pattern therefore concerns not the superiority of RL under easy conditions but the changing performance relationship as volatility increases.

At a volatility level of 25 units, static performance falls to 82.94%, while Q-learning remains at 96.78%. At the highest tested volatility, the distinction becomes substantial: Q-learning retains 94.46% SLA satisfaction, whereas static allocation decreases to 62.68% and the adaptive heuristic reaches 60.58%.

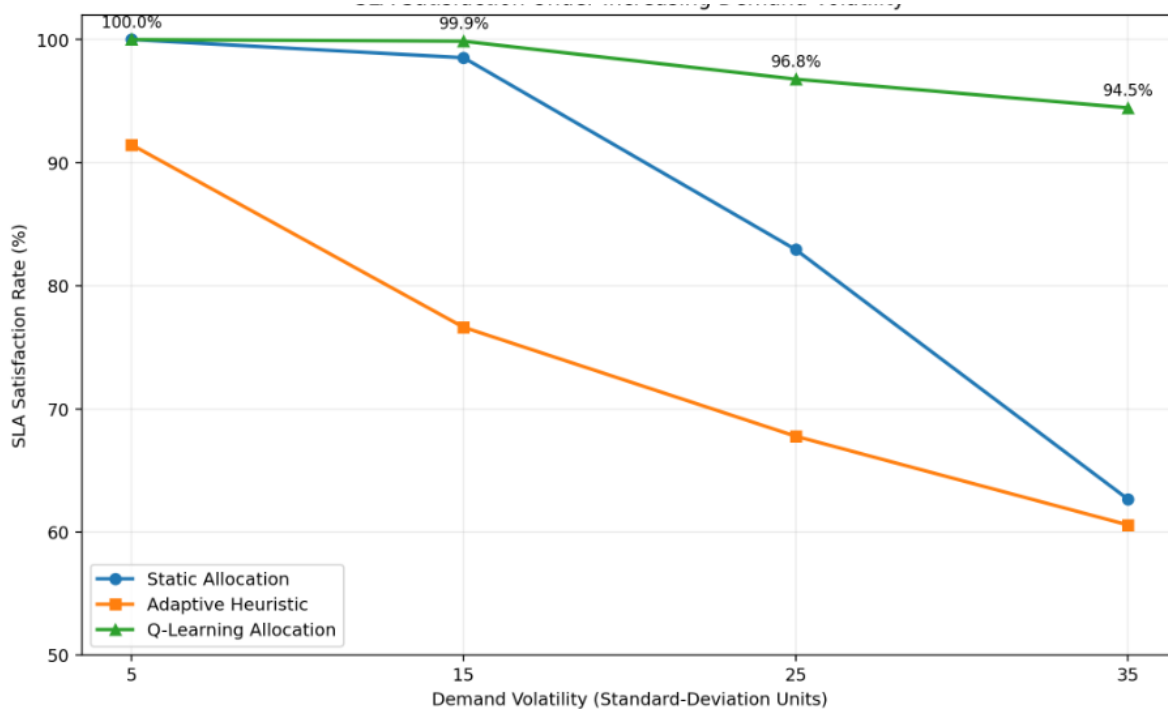
Table 2: SLA Satisfaction Under Increasing Workload Volatility

Demand Volatility	Static Allocation	Adaptive Heuristic	Q-Learning	Q-Learning Advantage over Static
5	100.00%	91.46%	100.00%	0.00 percentage points
15	98.52%	76.63%	99.87%	1.35 percentage points
25	82.94%	67.78%	96.78%	13.84 percentage points
35	62.68%	60.58%	94.46%	31.78 percentage points

Note: These values are computational simulation outputs and are not measurements from a production intelligent system.

5.2 Graphical Analysis of Dynamic Robustness

Graph 1: SLA Satisfaction Under Increasing Demand Volatility



The graph associated with Table 2 shows markedly different performance trajectories. Static allocation remains highly effective while demand remains close to its average level, but its performance deteriorates rapidly once fluctuations become large. This occurs because a constant allocation cannot increase capacity in response to accumulated queues or unusually large workload arrivals.

The adaptive heuristic responds dynamically but performs less effectively in the particular experimental environment. Its decision rule depends partly on previous workload observations and therefore reacts with a degree of temporal lag. This result should not be generalized to all heuristic allocation algorithms. A differently designed predictive or optimization-based heuristic could achieve substantially stronger performance.

Q-learning demonstrates the most stable trajectory. Its service-level satisfaction decreases from 100.00% to 94.46% across the full volatility range. The principal analytical observation is therefore robustness under changing conditions, rather than simple superiority at every operating point.

5.3 Interpretation of Adaptive Resource Allocation

The widening performance gap at higher volatility illustrates one of the principal reasons reinforcement learning is attractive for resource management. Static policies rely upon assumptions about normal operating conditions. Once those assumptions become unreliable, the policy has no mechanism for learning from accumulated queue behavior.

Q-learning can instead associate combinations of backlog and recent workload with allocation actions that produce stronger long-term reward. Its policy therefore incorporates both immediate resource cost and the future consequences of allowing queues to accumulate.

The highest-volatility condition is particularly informative. Q-learning exceeds static allocation by 31.78 percentage points in SLA satisfaction. It also exceeds the evaluated heuristic by 33.88 percentage points. These differences are meaningful within the simulated environment, but they should not be interpreted as universal performance estimates for cloud computing, telecommunications, robotics, or other real infrastructures. External validation would be required before operational claims could be made.

6. Challenges

6.1 Reward Design and Safe Exploration

A reinforcement-learning agent optimizes the reward structure provided to it. Poorly designed rewards can therefore produce behavior that is mathematically optimal but operationally undesirable. A model heavily rewarding throughput may over-allocate resources and increase energy consumption. Excessive resource-cost penalties may encourage under-provisioning and increase latency.

The challenge becomes especially important in complex systems where objectives conflict. Cloud infrastructure may simultaneously seek low latency, high utilization, low cost, reduced energy consumption, fairness among users, and contractual SLA compliance. Combining these objectives into one scalar reward requires governance and domain expertise.

Exploration creates a second challenge. Reinforcement-learning agents normally need to test unfamiliar actions to discover improved policies. Unrestricted exploration may be acceptable in simulation but dangerous in safety-critical or production environments. Offline training, digital twins, constrained RL, safety filters, and controlled shadow deployment are therefore preferable before autonomous control is permitted.

6.2 Nonstationarity and Generalization

Resource environments may change after training. New application types, changing user behavior, hardware upgrades, network failures, seasonal demand, and software modifications can alter state

transitions. An RL policy trained under one workload distribution may therefore degrade when the operational environment changes.

Continuous adaptation can help but introduces additional risks. If a deployed agent continues learning, system behavior itself becomes nonstationary. A poorly controlled update can replace a stable policy with a degraded one.

Generalization should consequently be tested across workload distributions rather than assessed using a single simulation. Stress testing, scenario perturbation, out-of-distribution evaluation, and policy rollback mechanisms become important deployment safeguards.

6.3 Scalability, Coordination, and Interpretability

Tabular Q-learning becomes impractical when the number of states and actions becomes very large. Real intelligent infrastructures may contain thousands of servers, communication channels, users, queues, or autonomous agents. Deep reinforcement learning and function approximation can address dimensionality, but they introduce additional training complexity and interpretability problems.

Distributed resource allocation also creates multi-agent challenges. One agent's allocation decision may alter the resources available to another. Independent learning can therefore make each agent perceive the environment as continuously changing.

Interpretability remains important because resource managers need to understand why capacity was increased, reduced, or redistributed. Purely opaque policies can complicate operational debugging and accountability. Logging, policy visualization, counterfactual evaluation, and human override mechanisms should therefore accompany practical deployment.

7. Discussion

7.1 Reinforcement Learning Becomes More Valuable as the Environment Becomes Dynamic

The simulated results support the proposition that reinforcement learning provides its strongest advantage when resource conditions are sufficiently dynamic to undermine fixed assumptions. Under low workload volatility, static allocation performs as well as Q-learning on the selected SLA metric. A sophisticated adaptive system would therefore offer little justification if its additional implementation and computational costs produced no meaningful operational improvement.

The situation changes as demand becomes less predictable. At intermediate and high volatility, queue accumulation increasingly penalizes the static policy. Q-learning responds more effectively because its allocation policy has learned associations among current backlog, recent workload observations, resource cost, and future queue consequences.

This interpretation is consistent with prior resource-management research. DeepRM demonstrated that resource management can be converted into a learning problem, while reinforcement-learning studies in wireless systems have shown that learned policies can adapt decisions to dynamic channel and workload conditions. The broader contribution of RL therefore lies in adaptive sequential control rather than automation for its own sake.

7.2 Reward Engineering Determines What the Agent Learns

The study also illustrates an important limitation: RL performance is inseparable from reward design. The simulated Q-learning agent does not possess an intrinsic understanding of efficiency or service quality. It optimizes the mathematical incentives supplied by the experimental environment.

Changing the allocation-cost coefficient, queue penalty, backlog threshold, state bins, or action set could change the learned policy and potentially alter the ranking of the evaluated methods. The reported results must therefore be interpreted as conditional on the stated design.

Operational implementations should define reward functions through collaboration among system engineers, domain specialists, financial decision-makers, safety experts, and service owners. Multi-objective optimization or constrained reinforcement learning may be preferable when important requirements should not be traded against one another freely.

7.3 From Synthetic Performance to Operational Deployment

The present simulation provides proof of methodological plausibility rather than proof of production superiority. Real systems contain characteristics absent from the simplified environment, including heterogeneous tasks, correlated demand, hardware failures, migration overhead, network delays, distributed agents, delayed feedback, energy constraints, and changing service priorities.

A robust deployment pathway should therefore move gradually from synthetic simulation to trace-driven evaluation and subsequently to controlled operational testing. Comparison should include strong domain-specific baselines rather than weak benchmarks chosen only to make the learning algorithm appear favorable.

Reproducibility is equally important. Reporting seeds, hyperparameters, training budgets, evaluation horizons, variance across multiple runs, and benchmark definitions is essential when more complex RL architectures are used. The fixed-seed design of the present study improves transparency but remains only an initial validation. Future research should use repeated independent seeds and confidence intervals before drawing stronger statistical conclusions.

8. Conclusion

Reinforcement learning provides a promising framework for dynamic resource allocation because it transforms resource management from a static optimization exercise into a sequential learning problem. Instead of applying the same allocation repeatedly, an RL agent can associate changing system states with actions that improve cumulative performance.

The synthetic experiment demonstrates that the benefit of reinforcement learning depends strongly on environmental volatility. Static allocation performs extremely well when workload demand is predictable, indicating that reinforcement learning should not be introduced merely because it is technologically sophisticated. As volatility increases, however, the Q-learning policy maintains substantially higher service-level satisfaction than the tested static and heuristic policies. At the highest simulated volatility, Q-learning achieves 94.46% SLA satisfaction compared with 62.68% for static allocation.

These results support the use of reinforcement learning where intelligent systems encounter repeated decisions, dynamic resource demand, observable feedback, and long-term trade-offs among capacity, delay, utilization, and operational cost. Nevertheless, the method introduces important challenges related to reward specification, safe exploration, scalability, nonstationarity, reproducibility, and interpretability.

The principal conclusion is therefore conditional rather than universal. Reinforcement learning should be treated as an adaptive decision-control mechanism whose value increases with environmental complexity and uncertainty, not as an automatic replacement for conventional optimization or heuristic

scheduling. Future research should extend the experiment to multi-resource environments, multiple agents, continuous action spaces, deep reinforcement learning, workload traces, constrained safety objectives, repeated random seeds, and real-system validation before strong deployment claims are made.

References

1. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237–285. <https://doi.org/10.1613/jair.301>
2. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
3. Tesauro, G. (1995). Temporal difference learning and applications in backgammon. *Communications of the ACM*, 38(3), 58–68. <https://doi.org/10.1145/203330.203343>
4. Tesauro, G. (2003). Extending Q-learning to general adaptive multi-agent systems. In *Proceedings of the 2003 International Conference on Autonomous Agents and Multiagent Systems* (pp. 871–878). ACM.
5. Kumar, A., & Singh, S. (2006). A reinforcement learning approach to dynamic resource allocation. *Engineering Applications of Artificial Intelligence*, 19(8), 893–903. <https://doi.org/10.1016/j.engappai.2006.04.001>
6. Hussin, M., Abdul Hamid, N. A. W., & Kasmiran, K. A. (2015). Improving reliability in resource management through adaptive reinforcement learning for distributed systems. *Journal of Parallel and Distributed Computing*, 75, 93–100. <https://doi.org/10.1016/j.jpdc.2014.10.001>
7. Peng, Z., Cui, D., Zuo, J., & Lin, W. (2015). Research on cloud computing resources provisioning based on reinforcement learning. *Mathematical Problems in Engineering*, 2015, 916418. <https://doi.org/10.1155/2015/916418>
8. Asheralieva, A., & Miyanaga, Y. (2016). Dynamic resource allocation with integrated reinforcement learning for a D2D-enabled LTE-A network with access to unlicensed band. *Mobile Information Systems*, 2016, 4565203. <https://doi.org/10.1155/2016/4565203>
9. Liu, Y. Y., & Yoo, S. J. (2017). Dynamic resource allocation using reinforcement learning for LTE-U and WiFi in the unlicensed spectrum. In *2017 International Conference on Ubiquitous and Future Networks* (pp. 471–475). IEEE. <https://doi.org/10.1109/ICUFN.2017.7993829>
10. Swain, S. N., Thakur, R., & Murthy, C. S. R. (2017). Design and stochastic geometric analysis of an efficient Q-learning based physical resource block allocation scheme to maximize the spectral efficiency of device-to-device overlaid cellular networks. *Computer Networks*, 119, 71–85. <https://doi.org/10.1016/j.comnet.2017.03.014>
11. Wilhelmi, F., Bellalta, B., Cano, C., & Jonsson, A. (2017). Implications of decentralized Q-learning resource allocation in wireless networks. *arXiv preprint arXiv:1705.10508*.
12. Santos, E. C. (2017). A simple reinforcement learning mechanism for resource allocation in LTE-A networks with Markov decision process and Q-learning. *arXiv preprint arXiv:1709.09312*.
13. Li, J., Gao, H., Lv, T., & Lu, Y. (2018). Deep reinforcement learning based computation offloading and resource allocation for MEC. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1–6). IEEE. <https://doi.org/10.1109/WCNC.2018.8377343>
14. Huang, L., Feng, X., Qian, L., & Wu, Y. (2018). Deep reinforcement learning-based task offloading and resource allocation for mobile edge computing. In *Machine Learning and Intelligent Communications* (pp. 33–42). Springer. https://doi.org/10.1007/978-3-030-00557-3_4

15. Hu, X., Liu, S., Chen, R., Wang, W., & Wang, C. (2018). A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems. *IEEE Communications Letters*, 22(8), 1612–1615. <https://doi.org/10.1109/LCOMM.2018.2844243>
16. Yang, T., Hu, Y., Gursoy, M. C., Schmeink, A., & Mathar, R. (2018). Deep reinforcement learning based resource allocation in low latency edge computing networks. In *2018 International Symposium on Wireless Communication Systems* (pp. 1–5). IEEE. <https://doi.org/10.1109/ISWCS.2018.8491089>
17. Li, R., Zhao, Z., Sun, Q., I, C.-L., Yang, C., Chen, X., Zhao, M., & Zhang, H. (2018). Deep reinforcement learning for resource management in network slicing. *IEEE Access*, 6, 74429–74441. <https://doi.org/10.1109/ACCESS.2018.2881964>
18. Ye, H., Li, G. Y., & Juang, B.-H. F. (2018). Deep reinforcement learning based resource allocation for V2V communications. *arXiv preprint arXiv:1805.07222*.
19. Ye, H., & Li, G. Y. (2018). Deep reinforcement learning based distributed resource allocation for V2V broadcasting. In *2018 International Wireless Communications and Mobile Computing Conference* (pp. 214–219). IEEE. <https://doi.org/10.1109/IWCMC.2018.8450518>
20. Shah-Mohammadi, F., & Kwasinski, A. (2018). Deep reinforcement learning approach to QoE-driven resource allocation for spectrum underlay in cognitive radio networks. In *2018 IEEE International Conference on Communications Workshops* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICCW.2018.8403658>