

# Task Offloading of Edge Computing Based on Reinforcement Learning: A Survey

Pu Yangyizhen, Xie Yijie, Zheng Tongzhou, Zhou Xufei

**Abstract**—abstract here

**Index Terms**—Edge Computing, Task Offloading, Reinforcement Learning, Multi-Agent Reinforcement Learning.

## I. INTRODUCTION

**E**DGE computing refers to a distributed and open platform that integrates the core capabilities of networking, computing, storage, and applications on the network edge side close to the data source. It provides edge intelligence services nearby to meet the key requirements of industry digitalization in aspects such as agile connection, real-time services, data optimization, application intelligence, security, and privacy protection [1]. Traditional cloud computing generally centralizes data in data centers for processing, while edge computing assigns some computing tasks to the network edge for processing, handling data where it is generated.

The importance of edge computing is becoming increasingly prominent. Taking the booming autonomous driving in recent years as an example, autonomous driving requires extremely high real-time performance. If traditional cloud computing is used, the delay caused by vehicles transmitting data to the data center for processing and then getting the results back is unacceptable and may even lead to traffic accidents. However, edge computing enables vehicles to process data locally and make timely decisions, ensuring real-time performance.

Meanwhile, edge computing has also become important in terms of data privacy protection. For instance, many sensitive data (such as medical and health data, financial transaction data, etc.) are not suitable for being transmitted to remote cloud for processing. Edge computing allows data processing and analysis to be carried out on local devices or edge servers, reducing the risk of data leakage.

Although edge computing is gradually becoming a key technology to support various emerging applications and business models, with the development of mobile devices and the Internet of Things, the amount of data that devices need to handle and computing tasks are becoming increasingly complex. To further improve the processing efficiency, task offloading has emerged.

Task offloading means transferring the computing tasks of a device to other devices or servers with stronger computing capabilities for processing. It is common to offload the tasks of mobile devices to edge servers or the cloud [2].

However, in actual dynamic environments, task offloading faces numerous challenges:

- **Latency Challenge:** In dynamic environments, the network state can change at any time. Situations such as

network congestion and unstable signals may lead to a significant increase in data transmission latency. For example, in the Internet of Vehicles scenario, vehicles are in a high-speed moving state and network connections are constantly switching. During the process of offloading tasks from vehicles to edge servers or the cloud, it is difficult to predict and control the data transmission latency [3].

- **Energy Consumption Challenge:** Although offloading tasks to external computing resources can utilize stronger computing capabilities, the energy consumption issue during the task offloading process also needs to be considered. Packaging and transmitting data as well as interacting with external resources all consume energy. Especially for mobile devices with limited battery power, an unreasonable task offloading strategy may cause a sharp increase in device energy consumption and shorten the device's battery life [4].
- **Device Heterogeneity Challenge:** The development of mobile devices and the Internet of Things has led to an increasingly diverse range of devices, with huge differences in hardware architectures, computing capabilities, storage capacities, and so on. This heterogeneity has caused two main problems.
  - On the one hand, the software and hardware environments of different devices are different. If the offloaded tasks are to run smoothly on the target devices, complex compatibility adaptation work is required.
  - On the other hand, the performance and computing capabilities of different devices also vary. If the same tasks are assigned to each device, it will lead to a decline in performance.

Therefore, tasks need to be reasonably allocated according to the actual situation of the devices.

Faced with a series of complex challenges such as latency, energy consumption, and device heterogeneity that task offloading encounters in dynamic environments, traditional methods based on rules or static optimization are gradually proving to be inadequate. These traditional methods often struggle to cope with the dynamic changes of the environment and complex system constraints, and thus cannot achieve optimal performance.

Against this background, reinforcement learning, as a powerful machine learning technique, has provided highly promising ideas and methods for solving the problem of task offloading.

The core of reinforcement learning lies in the fact that an agent interacts with the environment, continuously tries different actions, and learns the optimal strategy based on the obtained reward feedback, so as to maximize the cumulative reward in the long term [5]. This learning mode is inherently suitable for the dynamically changing task offloading scenarios because it can dynamically adjust task offloading decisions according to the real-time environmental states (such as network conditions, device loads, etc.), achieving self-adaptation to complex environments.

For example, when dealing with the latency issue, the reinforcement learning algorithm can, through continuous trial-and-error learning, choose to offload tasks to local devices, edge servers, or the cloud according to the real-time latency situation of the current network, so as to ensure that the overall task execution latency is minimized [6].

Through continuous learning and optimization, reinforcement learning can help agents find near-optimal task offloading strategies, providing a powerful technical support for solving the difficulties faced by task offloading in dynamic environments. It is expected to become a key tool for promoting the development of edge computing task offloading technology.

## II. RESEARCH STATUS OF REINFORCEMENT LEARNING IN TASK OFFLOADING

### A. The Foundation of Reinforcement Learning

### B. Applications of Classical Reinforcement Learning

### C. Applications of Deep Reinforcement Learning

In the research on task offloading, enhanced reinforcement learning methods have also been continuously developed and innovated. Here, we mainly introduce two methods. One is the Double Dueling DQN (D3QN), which demonstrates unique advantages in solving offloading problems in highly dynamic environments. The other is Multi-Agent Reinforcement Learning (MARL), which plays an important role in edge collaboration scenarios.

Edge computing networks face difficulties such as limited device resources and dynamic environmental changes. Zhang et al. [7] modeled the online offloading problem as a Markov Decision Process (MDP) and, on this basis, proposed the C - D3QN algorithm [7]. This algorithm combines the D3QN algorithm with a clustering algorithm and effectively improves the decision-making performance in highly dynamic environments by introducing a double dueling mechanism.

In terms of algorithm design, the centralized agent collects information on edge servers, mobile terminal devices, and channel states to form the state space. The action space covers the task execution decisions of terminal devices, and the reward function combines incentives such as task completion latency, terminal energy consumption, and task success rate to guide the agent's decisions.

When it comes to exploration and action selection, the agent adopts the  $\epsilon$ -greedy method. In the early stage, it randomly explores with a high probability, and in the later stage, it selects the action with the maximum Q value. The C - D3QN model introduces a fixed target mechanism, initializing the

online learning and target networks. The online learning network updates parameters in real time, and the target network is periodically updated to the parameter values of the online learning network to stabilize training and avoid overestimation of Q values. Meanwhile, the dueling mechanism changes the network output from a single Q value to a value function V value and an advantage function A value, enabling a more accurate estimation of Q values and enhancing the adaptability to highly dynamic environments.

Experiments show that the C - D3QN algorithm has obvious advantages over DQN in complex dynamic scenarios. It has a faster convergence speed and can quickly adapt to environmental changes for decision-making. In terms of task completion latency, it selects appropriate execution methods according to tasks and resources to reduce latency. Regarding terminal energy consumption, it makes reasonable offloading decisions to avoid excessive energy consumption and extend the battery life.

As the scale of edge computing expands, the collaboration among edge nodes becomes increasingly important. Multi-Agent Reinforcement Learning (MARL) can better cope with complex edge collaboration scenarios through the interaction and collaboration among multiple agents.

In edge computing task offloading, different edge servers or devices can be regarded as different agents. They work together to complete the offloading tasks while also taking into account factors such as the communication costs and differences in computing capabilities among different devices and servers [8]. The MARL approach allows each agent to make decisions based on its own observations and local information, and at the same time communicate and collaborate with other agents to gradually achieve the globally optimal collaboration strategy.

However, MARL also faces challenges. Conflicts of interest among agents may lead to difficulties in collaboration, and an increase in the number of agents causes the state and action spaces to grow exponentially, resulting in the "curse of dimensionality" and increasing the computational complexity and convergence difficulty. To address these challenges, researchers have proposed various solutions. A reasonable reward mechanism can adjust the reward function, rewarding agents that actively collaborate and punishing selfish behaviors, prompting them to pursue their own interests while achieving the goals of the system [9]. The hierarchical MARL architecture divides agents into different layers, with higher-level agents coordinating lower-level ones, reducing complexity and improving collaboration efficiency [10].

In practical applications, MARL algorithms can support edge devices to coordinate the allocation of task shares and resource utilization in real time. They can also assist vehicles and Roadside Units (RSUs) in sharing traffic information and facilitating task offloading. In the future, MARL algorithms are also expected to play a significant role in the Internet of Things.

In conclusion, the applications of Double Dueling DQN and Multi-Agent Reinforcement Learning in the field of edge computing task offloading provide powerful means to solve the problems in highly dynamic environments and edge col-

laboration. However, they also face numerous challenges at the same time. Future research needs to further optimize the algorithms to adapt to the continuously developing demands of edge computing and promote the application and development of edge computing technology in a broader range of fields.

### III. OPTIMIZATION AND INNOVATION OF REINFORCEMENT LEARNING METHODS

#### A. Method optimization

#### B. Compression of action space and state space

### IV. CURRENT CHALLENGES AND FUTURE PROSPECTS

See for resources on formatting math into text and additional help in working with  $\text{\LaTeX}$ .

#### A. Challenges

#### B. Future Prospects

### V. CONCLUSION

The conclusion goes here.

### REFERENCES

- [1] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
- [2] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.
- [3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [4] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 843–10 856, 2021.
- [5] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, p. 1054, 1998.
- [6] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181–198, 2020.
- [7] Z. Zhang, H. Li, Z. Tang, D. Gu, and J. Zhang, "A clustering offloading decision method for edge computing tasks based on deep reinforcement learning," *New Generation Computing*, vol. 41, pp. 85–108, 2023. [Online]. Available: <https://doi.org/10.1007/s00354-022-00199-7>
- [8] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9282–9293, 2021.
- [9] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Deep reinforcement learning (drl)-based device-to-device (d2d) caching with blockchain and mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6469–6485, 2020.
- [10] W. Hou, H. Wen, H. Song, W. Lei, and W. Zhang, "Multiagent deep reinforcement learning for task offloading and resource allocation in cybertwin-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 256–16 268, 2021.