Meta-Reinforcement Learning for Optimized Task Scheduling in Heterogeneous Edge Computing Systems

Abstract

Mobile edge computing often suffers from the dynamic and unknown nature of the environment such as time-varying conditions, heterogeneous devices, and frequent communication requests, imposing significant challenges on improving system performance. To meet the rapidly growing demands of computation-intensive and time-sensitive applications, Reinforcement learning [1] has been proposed as an effective tool to establish low-latency and energy-efficient networks. RL enables network entities to interact with the environment and learn an optimal decision-making policy, usually modeled as a Markov decision process [2].

Introduction

Mobile Edge Computing is emerging as a promising paradigm to enhance the computational capacity of mobile devices by offloading tasks to nearby edge servers. This paradigm aims to reduce latency, energy consumption, and improve Quality of Experience (QoE) for end-users. However, one of the major challenges in MEC is the efficient decision-making process for computation offloading, considering the dynamic nature of the network, user demands, and limited resources. Traditional offloading strategies, which often rely on heuristic or single-agent models, fail to capture the complexity and stochastic nature of modern MEC systems.

Motivation: Meta DRL focuses on training agents that can quickly adapt to new tasks or environments with minimal additional learning. It is designed for scenarios where agents face a wide variety of tasks, and the aim is to learn a policy that generalizes well across different tasks. The primary objective is to equip the agent with meta-knowledge, allowing it to efficiently adapt to new tasks by leveraging past learning experiences. In MEC, a meta-trained agent could adapt its offloading strategy efficiently when moving between different environments (e.g., from urban to rural networks), quickly optimizing its offloading decisions in unfamiliar settings.

Problem Statement: Efficient task offloading is crucial to ensure seamless resource distribution in MEC. Typically, the overall Resource Management process involves three layers of heterogeneous Resource scheduling decisions (**P1**, **P2**, **P3**), each of which performs in a specific collaboration manner.

- **P1. Edge-cloud service placement.** The cloud caches all services with sufficient storage spaces. Considering storage limits of edge servers, only a subset of services can be placed in each edge server. Services can be migrated from a cloud to an edge or between edge servers, which requires efficient collaboration.
- **P2. Edge-edge computation offloading.** The task offloading decision-making process focuses on efficiently distributing tasks among edge servers. Edge-edge collaborations enables edge servers to offloade their computation workload to neighboring servers, ensuring better resource utilization.
- **P3. Intra-edge resource allocation.** On edge servers, there may be several tasks competing for resources among offloaded tasks on the same server. Intra edge there is a resource competition among offloaded tasks on the same server. Intra-edge resource allocation aims to determine how resources should be allocated to each offloaded task.

Problem Model: To apply Meta-RL for address combination of sub-problems **P1**, **P2**, and **P3**, each problems can be formoulated as an individual MDP models. The MDP learning process shoud be decomposed into two parts: **learning a meta policy efficiently across all MDPs** and **learning a specific strategy for an MDP quickly based on the learned meta policy**.

Research Methodology:

1. **Algorithm Design:** Developing a Multi-Agent Meta-Reinforcement Learning algorithm using techniques such as **Multi-Agent Meta-Actor and Meta-Critic Networks**, with a focus on global optimization in MEC.
2. **Simulation Environment:** A simulated MEC environment will be developed using Python or a suitable simulation platform, where cload and edge servers be able to cache services and distribut tasks in whole resources, under different network conditions.

3. **Key Challenges:** (a) The meta-learned policy should work well across different, unseen tasks. (b) Balancing between exploration (learning new tasks) and exploitation (using learned knowledge).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.