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

Motivation: Meta DRL focuses on training agents that can quickly adapt to new tasks or environments with minimal additional learning [1]. 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, 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** [2]. The cloud caches all services with sufficient storage space. Considering the 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** [3]. The task offloading decision-making process focuses on efficiently distributing tasks among edge servers. Edge-edge collaborations enable edge servers to offload their computation workload to neighboring servers, ensuring better resource utilization.
- **P3. Intra-edge resource allocation** [4]. 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 problem can be formulated as an individual MDP model. The MDP learning process should 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 **Meta-Actor and Meta-Critic Networks** [5], 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 cloud and edge servers be able to cache services and distribute 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] J. Beck, R. Vuorio, E. Z. Liu, Z. Xiong, L. Zintgraf, C. Finn, and S. Whiteson, "A survey of meta-reinforcement learning," *arXiv preprint arXiv:2301.08028*, 2023.

[2] V. Farhadi, F. Mehmeti, T. He, T. F. La Porta, H. Khamfroush, S. Wang, K. S. Chan, and K. Poularakis, "Service placement and request scheduling for data-intensive applications in edge clouds," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 779–792, 2021.

[3] R. Han, S. Wen, C. H. Liu, Y. Yuan, G. Wang, and L. Y. Chen, "Edgetuner: Fast scheduling algorithm tuning for dynamic edge-cloud workloads and resources," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 880–889.

[4] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in iot edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[5] W. Ding, F. Luo, C. Gu, Z. Dai, and H. Lu, "A multiagent meta-based task offloading strategy for mobile-edge computing," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 1, pp. 100–114, 2023.