Multi-Agent Deep Reinforcement Learning for Cooperative Task Offloading in Partially Observable Mobile Edge Computing Environment

Motivation: In MEC, each entity may need to make local decisions to improve network performance in dynamic and uncertain environments. Standard learning algorithms, such as single-agent RL or DRL [1], [2], have recently been used to enable each network entity to learn an optimal decision-making policy adaptively through interaction with the unknown environment. However, these algorithms fail to model cooperation or competition among network entities, treating other entities simply as part of the environment, which can lead to non-stationarity issues. MARL enables each network entity to learn its optimal policy by observing both the environment and the policies of other entities while interacting with a shared or separate environment to achieve specific objectives [3].

Problem Statement: Task offloading is a critical process to efficiently assign available resources to task requests, for high-performance, reliable, and cost-effective services. In MEC, the decision-making process of task offloading focuses on efficiently distributing tasks among edge servers, where resources refer to limited computation, storage, and communication resources of edge and cloud servers. Typically, the offloading process involves two layers of heterogeneous decisions making problems (**P1, P2**) as follows,

- **P1. Devise-edge task offloading.** Enables devices to independently make decisions on offloading resource-intensive tasks to nearby edge servers, fostering efficient utilization of resources.
- **P2. Edge-edge task offloading.** Leverages edge-edge collaborations, where tasks initially received by a local edge server can be offloaded to neighboring servers with underutilized resources.

Problem Model: The main problem can be formulated as the decomposition of sub-problems **P1** and **P2** as a **Decenteralized Partially Observable Markov Decision Processes (Dec-POMDP**) [4], where multiple devices and edge servers interacting with each other by its observation of the environment, which is a part of main overall state.

Research Methodology:

1. **Algorithm Design:** Developing a **MARL** algorithm using techniques such as **Deep Deterministic Policy Gradient (DDPG)** [5] or **Dueling Double Deep Q-Networks (D3QN)** [6], with a focus on communication and collaboration, coordination or competition between agents.
2. **Simulation Environment:** A simulated MEC environment will be developed using Python or a suitable simulation platform, where mobile devices can offload tasks to edge servers and edge servers can distribute their computation workloads, under different network conditions.
3. **Key Challenges:** (a) Coordination or competition between agents. (b) The non-stationary environment due to actions of other agents. (c) Scalability issues as the number of agents increases.

References

[1] L. Liao, Y. Lai, F. Yang, and W. Zeng, "Online computation offloading with reinforcement learning algorithm in mobile edge computing," *J Parallel Distrib Comput*, vol. 171, pp. 28–39, Jan 2023.

[2] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mob. Comput.*, vol. 19, no. 11, pp. 2581–2593, Jul 2019.

[3] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[4] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[5] T. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[6] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.