# Multi-Agent Deep Reinforcement Learning for Cooperative Resource Management in Partially Observable Mobile Edge Computing Environment

**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:** 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 Reinforcement Learning (RL) or Deep Reinforcement Learning (DRL), 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. Multi-Agent Reinforcement Learning (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.

**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 the MEC, the task offloading decision-making process 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 proceses involves two layers of heterogeneous decisions making prolems (**P1, P2**) as follow,

- **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, ensuring better resource utilization.

**Problem Model:** The main problem can be formoulated as decomposation of sub-problems **P1** and **P2** as a **Decenteralized partially observable markove decision procsses (Dec-POMDP**), where multibe devices and edge servers interacting with each other by its own observation of environment, which is a part of main overall state.

## Research Methodology:

1. **Algorithm Design:** Developing a Multi-Agent Deep Reinforcement Learning algorithm using techniques such as **Deep Deterministic Policy Gradiant (DDPG)** or **Dueling Deep Q-Networks (DDQN)**, with a focus on communication and collaboration, cordination or competitation 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 thier computation workloads, under different network conditions.
3. **Key Challenges:** (a) Coordination or competition between agents. (b) Non-stationary environment due to actions of other agents. (c) Scalability issues as the number of agents increases.

# 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.