# Combinatorial Client-Master Multiagent Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing

Extended Abstract

Tesfay Zemuy Gebrekidan
University of Southampton
Southampton, United Kingdom
tzg1e19@soton.ac.uk

Sebastian Stein
University of Southampton
Southampton, United Kingdom
ss2@ecs.soton.ac.uk

Timothy J.Norman
University of Southampton
Southampton, United Kingdom
t.j.norman@soton.ac.uk

## ABSTRACT

Deep reinforcement learning (DRL) is gaining popularity in task-offloading problems because it can adapt to dynamic changes and minimize online computational complexity. However, the various types of continuous and discrete resource constraints on user devices (UDs) and mobile edge computing (MEC) servers pose challenges to the design of an efficient DRL-based task-offloading strategy. Existing DRL-based task-offloading algorithms focus on the constraints of the UDs, assuming the availability of enough storage resources on the server. Moreover, existing multiagent DRL (MADRL)–based task-offloading algorithms are homogeneous agents and consider homogeneous constraints as a penalty in their reward function. In this work, we propose a novel combinatorial client-master MADRL (CCM_MADRL) algorithm for task offloading in mobile edge computing (CCM_MADRL_MEC) that allows UDs to decide their resource requirements and the server to make a combinatorial decision based on the UDs' requirements. CCM_MADRL_MEC is the first MADRL approach in task offloading to consider server storage capacity in addition to the constraints of the UDs. By taking advantage of the combinatorial action selection, CCM_MADRL_MEC has shown superior convergence over existing benchmark and heuristic algorithms.

## KEYWORDS

Multiagent Deep Reinforcement Learning; Combinatorial Action Selection; Mixed Constraints; Client-Master Multiagent Deep Reinforcement Learning; Distributed Solution

**ACM Reference Format:**
Tesfay Zemuy Gebrekidan, Sebastian Stein, and Timothy J.Norman. 2024. Combinatorial Client-Master Multiagent Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

Task offloading in MEC has become an attractive solution to meet the diverse computing needs of UDs [4] by distributing computational tasks between UDs and MEC servers. DRL has recently gained popularity in task offloading due to its advantage in reducing online computational complexity [11] and adapting to dynamic changes [7]. However, the existence of various types of resource constraints on UDs and MEC servers and the combination of discrete, continuous, and combinatorial action spaces pose challenges to the design of an efficient DRL-based task-offloading strategy. UDs have limitations such as finite battery life and limited computational capabilities [6, 12], as well as quality of service (QoS) requirements such as latency. Similarly, MEC servers come with storage constraints. DRL techniques, such as the deep Q network (DQN), have yielded encouraging results by modeling the task offloading problem as a Markov Decision Process (MDP) with a deep neural network (DNN) for the function approximation [8]. However, due to the curse of dimensionality, DQN is insufficient for learning with large discrete action spaces [1] and a combination of continuous and discrete action spaces [13]. Although multiagent deep deterministic policy gradient (MADDPG)-based task-offloading algorithms can handle continuous action spaces, the representation of discrete and continuous action spaces still poses a challenge [5, 13]. Despite the advances of MADRL in task offloading, such as cooperative offloading decisions [10] and mixed continuous and discrete action spaces [5, 13], most existing MADRL-based algorithms still formulate the constraints as a penalty in their reward function.

The main contributions of this work are fourfold.

- We propose a novel CCM_MADRL algorithm for task offloading in MEC with various types of constraints at the UDs, the wireless network, and the server. Client agents are deployed at the UDs to decide their resource allocation, and a master agent is deployed at the server to make combinatorial decisions based on the actions of the clients. The constraints of the UDs are considered as a penalty in the reward of the client agents whereas the channel and storage constraints are considered in the combinatorial decision of the master agent.
- We contribute to dimensionality reduction by avoiding the number of sub-channels from the state and action spaces and considering it as a constraint in the combinatorial action selection.
- This is the first DRL-based task offloading algorithm to consider combinations of continuous and discrete resource constraints on UDs, the communication channel, and the storage capacity of the server.
- We develop different heuristic benchmarking methodologies and perform a numerical analysis to determine the efficacy of the proposed algorithm.



*Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand*. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

The details of the system model and the formulation of the problem are provided in the supplementary material [1]

## 2 COMBINATORIAL CLIENT-MASTER MADRL ALGORITHM FOR TASK OFFLOADING IN MEC

Task offloading is a cost-minimization problem that includes time and energy consumption. We convert the cost minimization problem into a reward maximization problem and apply CCM_MADRL_MEC. The formulations of cost minimization and reward maximization are presented in the supplementary material. The states and actions of the client and the master agents are presented below.

### 2.1 State

The state $S(t)$ of the MEC environment at time $t$, which includes the set of the states of the UDs, is described as $S(t) = \{S_n(t)\}, \quad \forall n \in N$ where $n$ is a UD and $N$ is the set of UDs. The state of a UD, $S_n(t)$, is characterized by five components: task state $S_n^{\text{task}}(t)$, normalized channel gain state $S_n^{\text{gain}}(t)$, power transmission budget $S_n^{\text{pow}}(t)$, local resource allocation budget $S_n^{\text{res}}(t)$, and battery state $S_n^{\text{battery}}(t)$.

At the beginning of each time step, the UDs make decisions about their resource allocations using client agents. Then, the SDN controller collects information about the state and action of the UDs and performs one of the following three procedures using a master agent: 1) for the UDs that decide to process a task locally, the server does not interfere. 2) if the number of UDs that propose to offload their tasks is greater than the number of sub-channels or if the sum of the sizes of their tasks is greater than the capacity of the storage capacity of the server, the server makes a combinatorial decision on which of the requests of the UDs to approve and which of them to reject. The rejected tasks are processed by their UDs. 3) If the proposed requests are less than the constraints, the server accepts all of them. Finally, sub-channels are assigned to accepted UDs, and then the task offloading and processing process starts.

Existing DRL-based task offloading algorithms, such as [10] and [5], included the number of sub-channels in their state and action space. However, the sub-channels have equal transmission capacity from the perspective of a UD. By restricting the use of a channel to only one UD at a time, we excluded channel information from the state and action space and considered them as a constraint in the combinatorial action selection. Note that a channel can be reused by multiple UDs one after the other. The actions of the client agents and the master agents are described below.

*Client actions:* At each time step, each client agent produces three actions, which are continuous valued between [0, 1] inclusive, that decide the task offloading decision by the client $n$ (if $x_{c,n}(t) < 0.5$ then processes locally; otherwise the decision is proposed to be decided by the master agent) $x_{c,n}(t)$, the transmission power $p_{c,n}(t)$, and the resources allocation of local computation $f_{c,n}(t)$.

*Master action:* The master agent makes a combinatorial decision on the client agents whose $x_{c,n}(t) \geq 0.5$, about which of them should be accepted for processing by the MEC server and which of them should be processed locally.

[1] https://eprints.soton.ac.uk/486925

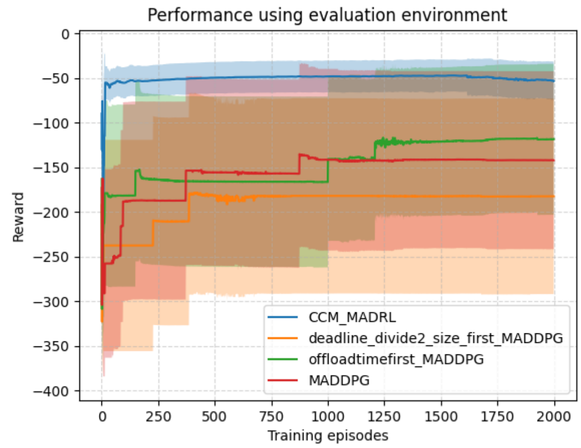


**Figure 1: Comparison of the CCM_MADRL with the heuristic and MADDPG algorithms**

In the classical MADDPG [9], the critic has a single Q-value for the combined state and action pair of all actors. The master agent in the CCM_MADRL_MEC applies the coalition action selection approach in [2] by modifying the critic part of the MADDPG algorithm to compute the relative Q value of each client, but uses per-action DQN [3] instead of a transformer neural network.

## 3 RESULTS

We compare our algorithm with MADDPG and heuristic algorithms. The heuristic algorithms differ from CCM_MADRL_MEC in that, instead of training a master agent to make combinatorial decisions about the decisions of the clients, they make decisions based on the order of shortest offloading time and deadline/size. A detailed description of the experimental setting and exhaustive experimental results are available in the extended supplementary material. As seen in Figure 1, the CCM_MADRL algorithm has performed better than the benchmarks, because once the client agents choose their action, the master agent makes a combinatorial decision on the action of the clients.

## 4 CONCLUSION

We propose a CCM_MADRL_MEC, that considers various constraints at the UDs, sub-channels, and server. By combining the advantages of both the policy gradient and value functions to output continuous and combinatorial actions, CCM_MADRL provides better convergence than existing homogeneous MADRL algorithms.

We plan to extend this work to multi-server CCM_MADRL_MEC where multiple servers cooperate to make combinatorial decisions.

## ACKNOWLEDGMENTS

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. This work was also supported by an EPSRC Turing AI Acceleration Fellowship (EP/V022067/1).

## REFERENCES


[1] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679* (2015).

[2] Tesfay Zemuy Gebrekidan, Sebastian Stein, and Timothy J.Norman. 2024. Deep Reinforcement Learning with Coalition Action Selection for Online Combinatorial Resource Allocation with Arbitrary Action Space. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)*. (In press).

[3] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2015. Deep reinforcement learning with a natural language action space. *arXiv preprint arXiv:1511.04636* (2015).

[4] Akhirul Islam, Arindam Debnath, Manojit Ghose, and Suchetana Chakraborty. 2021. A survey on task offloading in multi-access edge computing. *Journal of Systems Architecture* 118 (2021), 102225.

[5] Wei Jiang, Daquan Feng, Yao Sun, Gang Feng, Zhenzhong Wang, and Xiang-Gen Xia. 2023. Joint Computation Offloading and Resource Allocation for D2D-Assisted Mobile Edge Computing. *IEEE Transactions on Services Computing* 16, 3 (2023), 1949–1963.

[6] Te-Yi Kan, Yao Chiang, and Hung-Yu Wei. 2018. Task offloading and resource allocation in mobile-edge computing system. In *2018 27th Wireless and Optical Communication Conference (WOCC)*. 1–4.

[7] Xiaowei Liu, Shuwen Jiang, and Yi Wu. 2022. A novel deep reinforcement learning approach for task offloading in MEC systems. *Applied Sciences* 12, 21 (2022), 11260.

[8] Xiaowei Liu, Shuwen Jiang, and Yi Wu. 2022. A Novel Deep Reinforcement Learning Approach for Task Offloading in MEC Systems. *Applied Sciences* 12, 21 (2022).

[9] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6382–6393.

[10] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. 2023. Cooperative Task Offloading and Block Mining in Blockchain-Based Edge Computing With Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Mobile Computing* 22, 4 (2023), 2021–2037. https://doi.org/10.1109/TMC.2021.3120050

[11] Lili Nie, Huiqiang Wang, Guangsheng Feng, Jiayu Sun, Hongwu Lv, and Hang Cui. 2023. A deep reinforcement learning assisted task offloading and resource allocation approach towards self-driving object detection. *Journal of Cloud Computing* 12, 1 (2023), 131.

[12] Jia Yan, Suzhi Bi, and Ying-Jun Angela Zhang. 2018. Optimal Offloading and Resource Allocation in Mobile-Edge Computing with Inter-User Task Dependency. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–8.

[13] Jing Zhang, Jun Du, Yuan Shen, and Jian Wang. 2020. Dynamic Computation Offloading With Energy Harvesting Devices: A Hybrid-Decision-Based Deep Reinforcement Learning Approach. *IEEE Internet of Things Journal* 7, 10 (2020), 9303–9317.