**Dynamic Task Offloading in Edge Computing for Computer Access Point Selection based on Adaptive Deep Reinforcement Learning with Meta-Heuristic Optimization**

**Introduction**

With the rapid development of the Internet of Things and wireless technologies, highly advanced mobile applications, such as augmented reality face recognition and speech recognition, are emerging and attracting [9]. These applications typically result in high energy consumption and require powerful computation capacity. However, mobile devices (MDs) are usually limited in computation capabilities and battery power [10]. The limited computing resources and battery power of MDs have become a significant constraint for future mobile applications development [11]. To solve above problems, the MEC builds an open platform for data collection, data processing and data analyzing at the edge of the network, so that mobile devices can actively offload computing tasks to edge servers, thereby reducing service response time, improving device battery life, ensuring data security and user privacy [12]. In a heterogeneous network with embedded computing, decentralized technology is used to provide the best mechanism for unloading calculations and radio capital allocation [13]. With limited connectivity or compute capabilities, joint enhancements to the compute offload approach, radio, and compute resources were offered [14]. MEC consists of edge servers deployed at the edge of the network and implemented either at the cellular Base Stations (BSs) or at the local wireless Access Points (APs) [15]. As the MEC computation resources are more limited than those available in MCC, MEC servers can at any time offload their demanding tasks to MCC servers through the Internet whenever they are overloaded [16]. Further, multiple MEC servers can collaborate and offload their tasks to each other’s (e.g., via a backhaul network) to provide better services for the mobile users through balancing their workloads and sharing their resources.

The formulated optimization problem should take into account the tradeoff between communications delay and computing time, plus the inherent heterogeneity in terms of mobile devices’ computing capabilities, computation jobs’ QoS requirements, and the computing resources available at the MES [17]. the input parameters to the optimization problem may change from time to time due to the time-varying network environment and users’ applications [18]. This leads to frequent re-computation of the optimal offloading decision and computational resource allocation in near-real-time [19]. However, conventional mathematical optimization techniques usually converge slowly and have forbidden complexity for real-time implementations [20]. In MEC, there are two main questions related to task offloading. The first question is whether a mobile device should offload its task to an edge node or not [21. The second question is that if a mobile device decides to perform offloading, then which edge node should the device off load its task to [22]. In practice, however, edge nodes may have limited processing capacities, so the processing capacity that an edge node allocated to a mobile device depends on the load level at the edge node (i.e., number of concurrent tasks offloaded to the edge node) [23]. When a large number of mobile devices offload their tasks to the same edge node, the load at that edge node can be high, and hence those offloaded tasks may experience large processing delay.

Some existing works have addressed the load levels at the edge nodes and proposed centralized task offloading algorithms. Here considered a software-defined ultra-dense network, and designed a centralized algorithm to minimize the task processing delay [24]. Other works have proposed distributed task offloading algorithms considering the load levels at the edge nodes, where each mobile device makes its offloading decision in a decentralized manner. The existing methods are more challenging to deal with, because for the task of a mobile device arrived in a time slot, its delay can be affected by the decisions of the tasks of other devices arrived in the previous time slots [25]. In order to address these challenges, a novel computation offloading framework is proposed in this work based on Deep Reinforcement Learning algorithm.

**Related works**

In 2022, Kumaran *et al.* [1] have proposed a technique for the optimization problem of NP-hard that solved by the separates from sub-problems of resource allocation and offloading technique. An algorithm called Dynamic Weighed Quantum Arithmetic Optimization Algorithm (DWQAOA) was proposed for this offloading problem NP-hard. To achieve the ideal solution, a transition probability introduced the optimal Pareto connection during the optimization process. Numerous simulation results showed that the edge task latency could be reduced from 3 to 25% relative to competing solutions.

In 2021, Yang *et al.* [2] have proposed a novel offloading framework for the multi-server MEC network where each AP is equipped with an MES assisting mobile users (MUs) in executing computation-intensive jobs via offloading. Specifically, we formulate the offloading decision problem as a multiclass classification problem and formulate the MES computational resource allocation problem as a regression problem. Then a multi-task learning based feedforward neural network (MTFNN) model was designed and trained to jointly optimize the offloading decision and computational resource allocation. Numerical results showed that the proposed MTFNN outperforms the conventional optimization method in terms of inference accuracy and computational complexity.

In 2018, Bi *et al.* [3] have considered a decoupled optimization, where we assume that the mode selection was given and proposed a simple bi-section search algorithm to obtain the conditional optimal time allocation. On top of that, a coordinate descent method was devised to optimize the mode selection. The method was simple in implementation but may suffer from high computational complexity in a large-size network. To address this problem, we further proposed a joint optimization method based on the alternating direction method of multipliers (ADMM) decomposition technique, which enjoys a much slower increase of computational complexity as the networks size increases. Extensive simulations showed that both the proposed methods could efficiently achieve a near-optimal performance under various network setups, and significantly outperform the other representative benchmark methods considered.

In 2019, Li *et al.* [4] have proposed a computation offloading strategy and resource allocation optimisation scheme in a multiple wireless access points network with MEC, which aims to minimise the system cost by providing the optimal computation offloading strategy, transmission power allocation, bandwidth assignment, and computation resource scheduling. The proposed scheme decouples the optimisation problem into subproblems of offloading strategy and resource allocation since the problem was NP-hard. The offloading strategy involves the optimal access point selection, which was analysed by the potential game. The resource allocation was obtained using Lagrange multiplier. The authors' analysis and simulation results verify the convergence performance of the proposed scheme, and the proposed scheme outperforms the simple resource allocation scheme and the offloading strategy optimisation scheme in terms of the system cost.

In 2019, Alameddine *et al.* [5] have designed a novel thoughtful decomposition based on the technique of the Logic-Based Benders Decomposition. This technique solves a relaxed master, with fewer constraints, and a subproblem, whose resolution allows the generation of cuts which will, iteratively, guide the master to tighten its search space. Ultimately, both the master and the sub-problem will converge to yield the optimal solution. We showed that this technique offers several order of magnitude (more than 140 times) improvements in the run time for the studied instances. One other advantage of this method was its capability of providing solutions with performance guarantees. Finally, we use this method to highlight the insightful performance trends for different vertical industries as a function of multiple system parameters with a focus on the delay-sensitive use cases.

In 2022, Tang *et al.* [6] have established a multi-user and multi-task MEC model and design an offloading indicator, through which we analyze what the current environment belongs to. In the cases where the computational resources of devices were sufficient or partially sufficient, we utilize the relationship between the offloading indicator and the cost incurred by the tasks that were executed in the current workflow to find the optimal offloading decision. In the cases where the computation on local and edge were both insufficient, we proposed a novel Offloading Algorithm based on K-means clustering and Genetic algorithm for solving Multiple knapsack problem (OAKGM), aiming not only to jointly optimize the time and energy incurred by the tasks that were executed in the current workflow, but also to penalize the overflowed computations so that the task pressure in the next workflow could be greatly reduced. In addition, a simplified Offloading Algorithm based on Multiple Knapsack Problem (OAMKP) was proposed to further cope with the environments with a large number of users or tasks. Experimental results demonstrate the effectiveness and superiority of the proposed algorithms when compared with several benchmark offloading algorithms, which could better exploit the computing capacities of IoT devices and the edge server, greatly avoid resource occupation in edge nodes and make sustainable MEC possible.

In 2020, Lu et al. [7] have designed the mobile edge computing (MEC) model of mobile devices with random mobility and hybrid access point (HAP) with data transmission and energy transmission. On this basis, the selection of target server and the amount of data offloading were taken as the learning objectives, and the task offloading strategy based on multi-agent deep reinforcement learning was constructed. Then combined with MADDPG algorithm and SAC algorithm, the problems of multi-agent environment instability and the difficulty of convergence were solved. The final experimental results showed that the improved algorithm based on MADDPG and SAC has good stability and convergence. Compared with other algorithms, it has achieved good results in energy consumption, delay and task failure rate.

In 2022, Tang *et al.* [8] have considered non-divisible and delay-sensitive tasks as well as edge load dynamics, and formulate a task offloading problem to minimize the expected long-term cost. We proposed a model-free deep reinforcement learning-based distributed algorithm, where each device could determine its offloading decision without knowing the task models and offloading decision of other devices. To improve the estimation of the long-term cost in the algorithm, we incorporate the long short-term memory (LSTM), dueling deep Q-network (DQN), and double-DQN techniques. Simulation results showed that our proposed algorithm could better exploit the processing capacities of the edge nodes and significantly reduce the ratio of dropped tasks and average delay when compared with several existing algorithms.

**Problem Statement**

With the sudden improvement in the successful of IoT, cloud computing become more demand in the business sector, where the data gathering is too large. Continuous and immediate integration is needed for augmented reality. To promote the effective processing of task, MEC and multi access edge computing are designed. The mobile devices promoted by the MEC facilitates to offload their computationally intensive to the neighbor edge nodes for managing and minimizing the task processing delay. It also minimizes the dropped task ratio for those delay sensitive task. Many existing works have developed the technique based on task offloading to evaluate the offloading decision to enhance the network revenue. But still there have some challenges in it, which are defined in the following.

* In the existing work, the mobile devices consider the offloading decision by means of MEC server, which computed the mixed integer nonlinear programming issue that cannot be optimized with minimum cost. Therefore, the new technique is introduced based on the deep learning model to minimize the cost.
* It is impractical to use some of the earlier model, because it is more challenging to obtain the optimal solution by considering the offloading optimization problem. Hence this work utilizes the reinforcement learning approach to manage the offloading effectively.
* If all the user utilizes the same access point for the task offloading, then the computation can bring the maximum system cost. Hence this study develops the computation offloading strategy based on deep learning method to minimize the system cost and enhance the offloading performance.
* However, the investigation of present MEC task offloading is based on the machine learning models only focus on learning target like the selection of edge server and data offloading. Therefore, this article implements the advanced deep learning method for the effective outcome of task offloading.
* Moreover, other conventional optimization model uses slow convergence and have further complexity for the real time implementation. In order to solve this issue, a novel computation offloading architecture is developed in this work based on deep learning model.

**Table 1:** Features and challenges of task offloading in edge computing using deep learning model

|  |  |  |  |
| --- | --- | --- | --- |
| **Author [citation]** | **Methodology** | **Features** | **Challenges** |
| Kumaran *et al.* [1] | DWQAOA | * It enhances the performance under different circumstances. * It attains maximum specified value. | * It requires more computation than the other models. |
| Yang *et al.* [2] | MTFNN | * It sort out the MINLP issues in real time execution. * It can train multiple tasks in same time | * It has more time complexity to process the scale of the network |
| Bi *et al.* [3] | ADMM | * It helps to optimize the computing mode selection and time allocation. * It solves the dual optimal problem. | * It has computational complexity |
| Li *et al.* [4] | CSAO | * It decouples the computation offloading problem. * It minimizes the system cost. | * It has computation complexity |
| Alameddine *et al.* [5] | LBBD | * It determines the quality of other trial solution to find the optimal solution. * It is more scalable in nature. | * It requires master to control the variable sharing |
| Tang *et al.* [6] | OAMKP | * It has more flexibility and performs effectively. * It provides better computational performance | * It generates communication delay, that can’t able to satisfy the need of application |
| Lu *et al.* [7] | MADDPG | * It sorts out the issue of instability in multi agent environment by distributed execution | * It only has partial observability and have computational difficulty. |
| Tang *et al.* [8] | LSTM | * It captures difficult temporal pattern * It extends the period for remembering the task information | * It has overfitting problem in the training and validation process. * It has complicated structure that make difficult to analyse. |

**Research Methodology**

Nowadays, mobile devices are responsible for processing more and more computationally intensive tasks, such as data processing, artificial intelligence, and virtual reality. Despite the development of mobile devices, these devices may not be able to process all their tasks locally with a low latency due to their limited computational resources. Multi-access edge computing (MEC) has already shown great potential in enabling mobile devices to bear the computation-intensive applications by offloading some computing jobs to a nearby access point (AP) integrated with a MEC server (MES). MEC is an innovative computing paradigm to enhance the computing capacity of mobile devices (MDs) by offloading computation-intensive tasks to MEC servers. With the widespread deployment of wireless local area networks, each MD can offload computation task to server via multiple wireless access points (WAPs). In mobile edge computing systems, an edge node may have a high load when a large number of mobile devices offload their tasks to it. Those offloaded tasks may experience large processing delay or even be dropped when their deadlines expire. However, computation offloading can bring a higher system cost if all users select the same access points to offload their tasks. In this proposal, the task offloading problem is determined by considering the delay sensitive task along with edge load dynamics to reduce the expected long-term cost. The distributed algorithm based on Adaptive Deep Reinforcement Learning (ADRL) based will be proposed, where every device will be analyzed for offloading decision without knowing the task model of other devices. To enhance the performance, the parameters in the model will be optimized using the Improved Piranha Foraging Optimization Algorithm (IPFOA) [26]. The simulation outcome with a greater number of mobile devices and corresponding edge nodes shows the developed optimization minimizes the ration of dropped task and average task delay respectively. The result of the designed model outperformed better than other available models.

Mobile edge computing

Task offloading

Parameter optimized by developed PFOA

**Figure 1:** Diagrammatic representation of proposed model for task offloading in edge computing.

**Expected Outcome**

The proposed model was evaluated in MATLAB 2020a and the performance analysis is carried out to validate the effectiveness of the developed model. From the findings, the proposed model achieves the impressive results to satisfy the requirement.

**References**

1. K. Kumaran, E. Sasikala, "Computational access point selection based on resource allocation optimization to reduce the edge computing latency", Measurement: Sensors, Vol. 24, pp. 100444, December 2022.
2. B. Yang, X. Cao, J. Bassey, X. Li and L. Qian, "Computation Offloading in Multi-Access Edge Computing: A Multi-Task Learning Approach," in IEEE Transactions on Mobile Computing, vol. 20, no. 9, pp. 2745-2762, 1 Sept. 2021.
3. S. Bi and Y. J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading," in IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 4177-4190, June 2018.
4. Qiuping Li, Junhui Zhao, Yi Gong, "Computation offloading and resource allocation for mobile edge computing with multiple access points", vol.13, Issue.17, pp. 2668-2677, October 2019.
5. H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi and C. Assi, "Dynamic Task Offloading and Scheduling for Low-Latency IoT Services in Multi-Access Edge Computing," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 3, pp. 668-682, March 2019.
6. H. Tang, H. Wu, Y. Zhao and R. Li, "Joint Computation Offloading and Resource Allocation Under Task-Overflowed Situations in Mobile-Edge Computing," in IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 1539-1553, June 2022.
7. H. Lu, C. Gu, F. Luo, W. Ding, S. Zheng and Y. Shen, "Optimization of Task Offloading Strategy for Mobile Edge Computing Based on Multi-Agent Deep Reinforcement Learning," in IEEE Access, vol. 8, pp. 202573-202584, 2020.
8. M. Tang and V. W. S. Wong, "Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems," in IEEE Transactions on Mobile Computing, vol. 21, no. 6, pp. 1985-1997, 1 June 2022.
9. J. Xia et al., "Opportunistic Access Point Selection for Mobile Edge Computing Networks," in IEEE Transactions on Wireless Communications, vol. 20, no. 1, pp. 695-709, Jan. 2021.
10. X. Lai, J. Xia, L. Fan, T. Q. Duong and A. Nallanathan, "Outdated Access Point Selection for Mobile Edge Computing With Cochannel Interference," in IEEE Transactions on Vehicular Technology, vol. 71, no. 7, pp. 7445-7455, July 2022.
11. J. Wang, J. Hu, G. Min, A. Y. Zomaya and N. Georgalas, "Fast Adaptive Task Offloading in Edge Computing Based on Meta Reinforcement Learning," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 1, pp. 242-253, 1 Jan. 2021.
12. Liang Huang, Xu Feng, Luxin Zhang, Liping Qi and Yuan Wu, "Multi-Server Multi-User Multi-Task Computation Offloading for Mobile Edge Computing Networks", Sensors, vol. 19(6), 1446, 2019.
13. T. Alfakih, M. M. Hassan, A. Gumaei, C. Savaglio and G. Fortino, "Task Offloading and Resource Allocation for Mobile Edge Computing by Deep Reinforcement Learning Based on SARSA," in IEEE Access, vol. 8, pp. 54074-54084, 2020.
14. J. Wang, J. Hu, G. Min, W. Zhan, A. Y. Zomaya and N. Georgalas, "Dependent Task Offloading for Edge Computing based on Deep Reinforcement Learning," in IEEE Transactions on Computers, vol. 71, no. 10, pp. 2449-2461, 1 Oct. 2022.
15. Haifeng Lu, Chunhua Gu, Fei Luo, Weichao Ding, Xinping Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning", Future Generation Computer Systems, Vol.102, pp. 847-861, January 2020.
16. W. Chu, P. Yu, Z. Yu, J. C. S. Lui and Y. Lin, "Online Optimal Service Selection, Resource Allocation and Task Offloading for Multi-Access Edge Computing: A Utility-Based Approach," in IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 4150-4167, 1 July 2023.
17. H. Huang, Q. Ye and Y. Zhou, "Deadline-Aware Task Offloading With Partially-Observable Deep Reinforcement Learning for Multi-Access Edge Computing," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 6, pp. 3870-3885, 1 Nov.-Dec. 2022.
18. Z. Sun and M. R. Nakhai, "An Online Learning Algorithm for Distributed Task Offloading in Multi-Access Edge Computing," in IEEE Transactions on Signal Processing, vol. 68, pp. 3090-3102, 2020.
19. Z. Zhang and F. Zeng, "Efficient Task Allocation for Computation Offloading in Vehicular Edge Computing," in IEEE Internet of Things Journal, vol. 10, no. 6, pp. 5595-5606, 15 March15, 2023.
20. X. Qi, H. Xu, Z. Ma and S. Chen, "Joint Network Selection and Task Offloading in Mobile Edge Computing," 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Melbourne, Australia, pp. 475-482, 2021.
21. H. Wang, H. Xu, H. Huang, M. Chen and S. Chen, "Robust Task Offloading in Dynamic Edge Computing," in IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 500-514, 1 Jan. 2023.
22. X. -Q. Pham, T. -D. Nguyen, V. Nguyen and E. -N. Huh, "Joint Service Caching and Task Offloading in Multi-Access Edge Computing: A QoE-Based Utility Optimization Approach," in IEEE Communications Letters, vol. 25, no. 3, pp. 965-969, March 2021.
23. Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu and F. Tian, "Dynamic Task Offloading and Resource Allocation for Mobile-Edge Computing in Dense Cloud RAN," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3282-3299, April 2020.
24. Z. Sun, Y. Mo and C. Yu, "Graph-Reinforcement-Learning-Based Task Offloading for Multiaccess Edge Computing," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3138-3150, 15 Feb.15, 2023.
25. F. Wei, S. Chen and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," in China Communications, vol. 15, no. 11, pp. 149-157, Nov. 2018.
26. S. Cao, Q. Qian, Y. Cao, W. Li, W. Huang and J. Liang, "A Novel Meta-Heuristic Algorithm for Numerical and Engineering Optimization Problems: Piranha Foraging Optimization Algorithm (PFOA)," in IEEE Access, vol. 11, pp. 92505-92522, 2023.