International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Efficient Resource Allocation in 5G Communication Networks through Mobile Edge Computing: A Deep Learning Approach

KOUSHIK SAI POTHALA[1], MUDDAMSETTI SAI KRISHNA[1], DADAPEER C[1*]
[1]*Department of Computer Engineering,*
Indian Institute of Industry Interaction Education and Research, Chennai, Tamil Nadu 600066

---***---

Abstract - Techniques for computing offloading are challenging to design for the considerations such as the mobile edge networks, or base stations, which are deployed at random, limited capacities of servers, and the constant mobility of end users. In light of the capacity of DRL in dealing with intricate, dynamic challenges, this investigation formulates a procedure for determine the optimal method for compute offloading and resource management. First, the authors provide a comprehensive description of a specific case with a multi-user mobile edge network, which is MBS, SBS, and multiple terminal nodes. They next describe and elaborate on the Communication and Computation Overhead that is generated. In addition, the optimization goal, defined in this paper, involves taking the overall system energy consumption into full compte when harmonised with predictable task delays. The subsequent suggestion on decreasing the power consumption level will involve proposing a learning method, namely Deep Deterministic Policy Gradient (DDPG). Finally, simulation research proves that the introduced DDPG approach by the authors enhances the results of other correlation methods in the aspect of target value identification and the overall energy usage of the system, which is only 15. 6 J. Furthermore, it is possible to observe that the proposed algorithm predicts higher allowances of communication resources as compared to the existing method.

Key Words: Compute offloading, Mobile edge networks, Resource management, Deep Reinforcement Learning (DRL), Energy consumption, Deep Deterministic Policy Gradient (DDPG)

1.INTRODUCTION

With the development of 5G wireless network's access technology, and the extensive application of smart terminals, the previous IoT structure has gradually transferred to establish a new, far stronger new data-oriented intelligent society [1]. This is due to the availability of advanced technologies which make the user terminals offer more of multimedia options. This has led to the fact that several new mobile apps being used are accelerating and these are AR/VR apps, traffic avoidance apps, online gaming, voice interaction and many others [2]. Despite the beneficial changes in people's daily lives, these dynamic mobile applications also have been instrumental in fueling the unprecedented increases in data traffic carried by mobile communication networks. In the global mobility data centres study launched by International Data Corporation (IDC), it was found that it will grow to 162 Zettabytes (ZB) by year 2024. The edge of network means, the edge of the network will be charged with processing, analysing and storing of more than 76% of the data [3]. Increased requirements are also applied to Quality of Service (QoS) and processing power, two other aspects where such rapidly growing applications perform high calls. This implies that it is more demanding when it comes to power and other resources required for smartphone application than most applications. Furthermore, with the advancement of innovations such as Internet of Things [10], Artificial Intelligence, Fifth Generation mobile communication, the expectations from consumers are increasing pressure on data computational capability and reliable service quality [4].

MEC research begins with computation offload and content caching, two significant trends, that has received considerable attention from researchers across the globe in the last few years. Among the several characteristics of the next-generation wireless access network, the most critical one is that it has to deliver the necessary capacity increment, and this is where small cell networks are to be extensively distributed. In this case, a dynamic optimization model for handling resource allocation computation based on subgradient approach, and for solving the max-min fairness issue through the utilization of dual decomposition method coupled with continuous convex approximation (SCA) as claimed in reference. It implies that not only the criterion of fairness is satisfied, but also evaluation of different traffic models by comparing the obtained results of simulation shows a significant improvement in EE. Based on this strategy, a dynamic optimization model for 5G heterogeneous networks was introduced in reference [5], which aimed to minimize the overall power consumption while still delivering the required coverage and traffic capacity. The proposed scheme identifies when small cells should be on or off and follows the sharp variation in the QoC constraints of the users in order to identify to when allocate carriers and what power level should be utilised at specific intervals to enhance efficiency. The cooperative offloading options in the system is recommended to be fixed by following an iterative heuristic resource allocation model, coupled with partial computation offloading. Since most of these studies simply try to improve the efficiency of quasi-static systems, many of them neither consider the time-varying system environment in the time domain in their research, or have nothing to do with actual dynamic systems that exist in practice. Further there is a comprehensive dependency on contextual sensing for computing offloading in MEC paradigm, which arises due to varying responsiveness of computing jobs to time delay, limited resource availability and unpredictable resource demands.

To alleviate these problems, a lot of latest research have proposed the potential strategies to offload the computation tasks and to assign the resources in MEC by using deep reinforcement learning (DRL). More specifically, we employ Action Refinement in the framework of Deep Reinforcement Learning (DRL) and propose a new strategy on achieving the maximum of such values as compute offload and resources usage at the same time. In this paper, the unloading issue is formulated as a Parsely Observable Markov Decision Process (POMDP), and with the help of game theory we employ the strategy gradient approach based on Deep Reinforcement Learning (DRL). The authors proposed the

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

wireless recharging capability for MEC network and for the online decision making framework using a DRL-based Decision Tree to select the justifiable best actions.

Handling Huge Data Complexity through a Low Latency-Near optimal MEC Network A wireless rechargeable MEC should be capable enough to provide its users a near optimal solution for their trouble making decisions and the same has been suggested in the next heading. However, in practice, the Two traditional edge caching algorithm mentioned above are unable to perform well in dynamic environment. This is, however, has largely been due to the inability of conventional caching methods to incorporate the dynamic characteristics of the network. At the same time, it is crucial to critically assess the material's relevance and time-sensitive nature, the limitation on the period within which the content must be delivered, and the limited storage capacity of each access point in a CU.

In computing offloading, two essential factors that need to be determined involves are the time taken to execute the task as well as the energy consumed in the process. Thus, reducing delay, reducing energy consumption, and achieving the best compromises between these metrics are key goals of optimization. Secondly, The different allocation of resource might also have improvements with the help of Reinforcement Learning (RL) that maps out the hidden environment of resource. To lower the total content and time load, unpredictable caching selections utilizing RL algorithms Were examined in reference [6] Revolutionary concept for the edge caching of a distributed car network created by means of artificial intelligence. RL based solutions are applied to the particular algorithm to handle the dynamic allotment of compute and caching resources, and it enhances the worth of the system since it jointly optimizes the edge computing and content caching. The problem of resource sharing for communication, caching, and computing in the network was addressed by the use of an RL-based adaptive framework that had no need of prior information on the existence and type of data in the environment. Supposing there are many users that are connected to a single MEC server, the author of the paper suggests a NOMA-MEC system. They got the above-mentioned best state combination applied in RL to select users who offload at once without prior information about other users' activities, and hence achieved low system offloading latency. An energy-saving and low latency MEC offloading scheduling scheme is proposed for MEC system with decision making on task offloading, offloading scheduling, and power control. Lowering down the system latency and energy consumption was the overall aim of this approach. Explained how to mitigate the high-dimensional catastrophe problem of MDP raw states, and proposed a network resource configuration change management framework. The same was true for the related optimization issue though it was solved with slightly lesser complexity. Minimize the system costs (energy consumption and the delay in the time taken for the calculations) through offloading selection through the RL-based SARSA method and aimed at solving the problem of managing resources of the edge server. Similar to this, we can employ the same approach in studying another problem of placing and delivering content in the edge network. However, they also need to ensure that the content is delivered to the consumers without a significant delay from the production process while trying to keep the overall costs to a minimum. Besides, regarding the problem of content delivery delay and cache hit rate to solve the repetitive traffic discharge.

Reference [7] introduced a Federated DRL-based collaborative edge cache framework. This would help devices learn to optimize a certain environment and correct their data localization in anticipation of any changes. Co-optimization of computing, storage and network resources from the standpoint of Resource Co optimization in the edge setting for the relatively dynamic allocation of tasks and data generally requires the allocation of different computing, storage and network resources appropriately and equitably in dynamic edge networks with state factors such as equipment heterogeneity, system dynamics, resource constraints and task time delay sensitivity. Therefore, the focus of this paper is to present the computing offload and content caching through different RL methods so that the operating performance of the system can be enhanced and the effectiveness and cost of edge networks can be increased. This will be done by minimizing different costs related to the resources, which will include making reasonable decisions on multi-user offload and fairly distributing the resources, carrying out all these with MEC as the background. Therefore, this research provides an offloading and resource allocation method for a compute-intensive application that minimizes the aggregate energy consumption of the system through DDPG. It decreased the energy use by 'offering the opportunity to collect system-wide action and status data' and by basing better judgments on more global knowledge than the current approach, the suggested algorithm does the same.

2.ACTOR-CRITIC METHOD

Probably, the most unique advantage of Actor-Critic is that it combines the strengths of policy and value-based algorithms. An 'Actor' is a policy structure selecting an action, and 'Critic' is a function estimating the behaviour of an Actor. Critic then verifies the better or worse state generated from opting the chosen action. As far as learning goes, there is the necessity of gradient algorithms in both networks albeit at different times. In Figure 1 below, will depict the Actor-Critic architecture As kindly indicated by Seong et al. Here θ means parameters of the DNN and the policy objective function denoted by J (θ) = $E\sigma\theta$[r]. Locating local maxima in J (θ) is the essence of the policy gradient approach In general, the approach complements the event-based view that local minima occur in H (θ). DPG techniques representation of the actions in terms of parameterized functions (s|θ) is due to the fact that the optimization within the continuous action space may be costly and time taking. Here, $\theta\lambda$ is the set of parameters for the Actor network. DDPG is an Actor-Critic algorithm for an offline technique comprised of five models.
Building upon prior work, it expands DPGs to the high and large state-action domains and is also useful for approximating the policy spaces under continuous actions based on functions from deep neural networks. It is performed by adding some noise to the Actor policy every time an action is selected. Similar to DQN, SL uses replay buffers to minimize data dependences within specific constraints. Using it again is the unambiguous target network which is also specifically issued to actors. The texts are as follows: A: 'In hat, let Q-learning be represented as Q and let the ideal q-value and course of action be represented as Q* and a*, respectively.'

Diagram 1: If there is such a thing as an "architecture of the actor," then this architecture must embody both the dual capacities of Locke's spectator, that is, to 'observe both at once', as well as the 'between two times' structure that Zizck identifies as the 'formal reality' of the subject.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

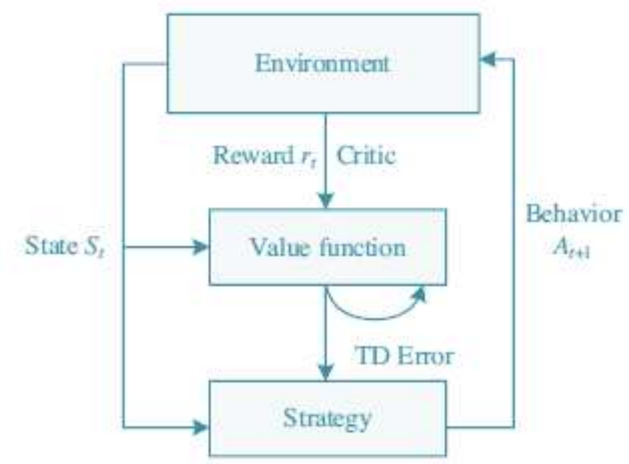


Fig 1: If there is such a thing as an "architecture of the actor," then this architecture must embody both the dual capacities of Locke's spectator, that is, to 'observe both at once', as well as the 'between two times' structure that Zizck identifies as the 'formal reality' of the subject.

Q-learning uses an alternative function called the Q-function for action selection. Acting according to the policy, the DDPG algorithm also incorporates an evaluation of the Q-value after action is selected. Besides, it modifies the policy to the optimal one and the chosen action adjusts the Q-value to the optimal level. Thus, Q* and π* are estimated using two different peer-to-peer networks.

3.ACTOR-CRITIC METHOD

3.1 System model

From Fig. 2, thus we find out that there are N terminal devices, K small-cell base stations and one macro-cell base station in a multi-user mobile edge network. Mobile terminal devices in the MBS coverage region are randomly distributed and there are N terminal devices. The sets of SBSs are represented as B = {b1, b2, … b j, … bK }, whereas, the sets of terminal devices are represented as U = {u1, u2, … ui … uN}. MEC server is deployed with each SBS and cloud server with each MBS for executing computational facilities on terminal devices. They seem to have a time-consuming and latency-sensitive job on the end device.

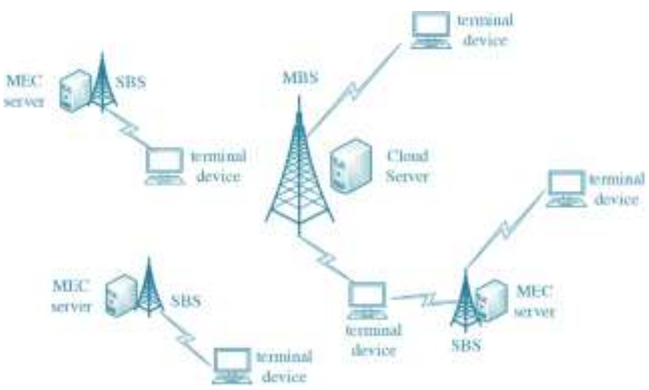


Fig 2: System framework

3.2 Communication model

These include connection power, bandwidth, and even the aspects such as interference on the channel if data must be transmitted between the devices and base stations. Devices and base stations use orthogonal frequency division multiple access OFDMA and SBS uses all the spectrum resources of MBS for optimum utilization of the spectrum. Of course, this model only considers the inter-SBS interference.

3.3 Computation modal

There are two phases involved in the computation offloading in communication. First, the terminal device transmits work orders to SBS. As the first step, if for a day, SBS pass the unprocessed tasks to MBS. Transmission energy is then defined as the utilized energy in the communication process for the two identified processes while the transmission latency is also described. If a job is offloaded from an SBS to an MBS there are costs involved in the communication which consists of the transmission time and energy of the SBS in uploading the task. Finally, the overhead in the information exchange process established during the return of the work result is also excluded for the sake of simplifying the calculations. There is the core network that integrates SBS and MBS. This is due to problems like multiple users sharing the same channel, complex routing that is used in selecting the path which data is to follow, and also due to the fact that the arrival of packets is random thus making it difficult to predict its delay accurately.

3.4 Problem modelling

With regards to impediments such as computational resources and limit of job delay in SBS and MBS, the goal is to minimise the energy consumption of the system. Therefore, computational offloading and resource management are shown as a joint numerosation problem. The optimization problem of minimising e(t) is NP-hard; therefore, it is not deterministic in finding a correct result. In addition, in the case of an extremely large number of terminal devices in the MEC scenario, the state information of the system will also become more complex. If there is a central point to gather all the information about the system and train it, the computationally complex will be highly increased significantly. Thus, the method incorporating the concept of RL is proposed.

3.ACTOR-CRITIC METHOD

This paper reveals the optimization problem as an NP-hard problem when under offloading decision constraints, delay constraints, and resource request constraints. This is to minimize total energy, however, computing resource constraints and task delay constraints are also taken into account. The described challenge can be addressed by proposing a DDPG-based resource allocation algorithm that will enable improving the user experience and minimizing the costs at the same time. Figure 3 illustrates the algorithmic framework.

In the deep RL approach to problem solving, there is state, action, and reward.

System Response. For the system action A, we have the offloading choice ai and the buffering decision xi, j, where x = [xi,1, xi,2, …, xi,n] and pi = [pui, psi] represents the uplink transmission power.

3. Automatic Bonus. This is the optimization objective of the total utility of the users in the compared MEC system. In each step, the agent will obtain a reward value R in the state S subsequent to the action A being executed. The reward function should ideally be related to the goal function. In this study, we minimize for least energy consumption while in deep learning, the objective is for maximum reward return provided the value of the award increases with the system's utility. R = Qi is defined as goal value, which implies that the reward is stated as it.

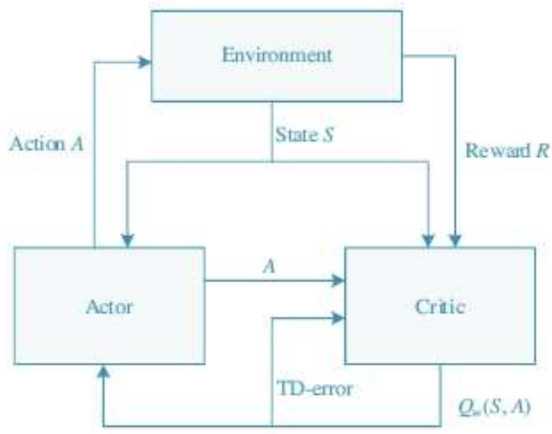


Fig 3: The architecture of DDPG is described. One of them is known as deep deterministic policy gradient, which is abbreviated as DDPG.

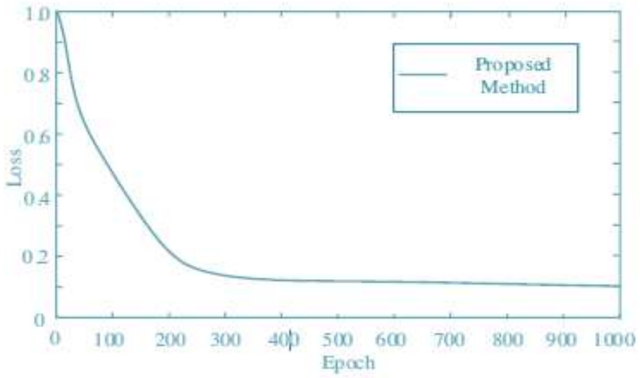
The component of DDPG architecture comprises of an actor and a critic, as illustrated in the above figure 3. With the help of the Actor component, it is possible to determine the parameterized policies and generate actions in accordance with the current state of the environment. The critical component, on the other hand, has to evaluate the current strategy and give feedback depending on the results worked out from the environment. It is crucial to understand that the Actor is responsible for the policy network as well as choices of actions, and the Critic is in charge of the value network. In his criticism, the critic employed the use of playback technology. Playback memory saves all the trajectories in the record with the help of tuples (St, At, Rt, St+1) and at every iteration it takes a small batch of these tuples to update the parameters. To optimize the loss function, you have to use the gradient descent to update the parameters of the network. Here, w′ is defined as the goal Q-network parameter, which changes less frequently than the current online Q-network. D is the playback memory, (St, At, Rt, St+1) ∈ D. A fixed-parameter target neural network calculates the loss function at each iteration.

3.ACTOR-CRITIC METHOD

5.1 Loss function

The L(w) loss function is used to calculate loss, and the Adam optimizer is used to (update) the parameters in an iterative manner. The training of the suggested DDPG-based resource allocation model was completed when the model has passed through 1000 episodes in the training set using the Keras framework with TensorFlow and GPU in all the training samples. The dynamics of the change to the loss function value throughout the process of training the network model is provided in Figure 4. The loss value of the suggested method is found to be declining steadily and the value function of Critic is converging to the actual value with the increase in training iterations. Thus, as per the suggested approach till 260 training iterations, the value of the loss function fluctuates relatively less and is about 0. 116. The rate of convergence of the suggested method is however very fast.

From the Figure 5, it is clear that the average energy consumption



per second of the system is related to different situations in the network.

5.2 System average power consumption per second under different network scenarios

When possible networks scenarios are considered, the average power consumption of the system per second is depicted in Figure 5. Nevertheless, in case of 'Only_local' option, it is possible to observe that the energy consumed by the system is the least. On the other hand, according to the values calculated for this network scenario, it has the lowest value of the task acceptance rate. For the analyses, let us compare its system energy use to the 'SBS_local' instance; the 'MBS_SBS_local' scenario reflects somewhat higher levels of system energy use. This is because, with the work transfer from SBS to MBS, the central cloud brought in 'MBS_SBS_local' results to increased energy consumption.

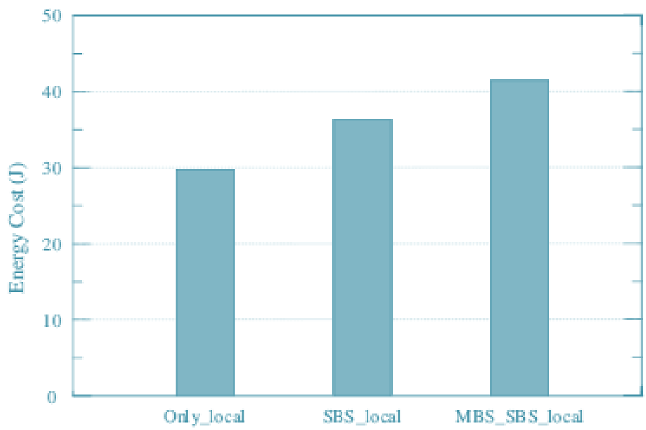


Fig 5: the average power used per second of the system depends on several network conditions.

5.3 Comparison with different algorithms

In order to show the effectiveness of the proposed approach for managing resources, figure 6 shows the results of the experiment where the suggested algorithm has been compared with the algorithms described in the papers [9] and [8]. Giving trainings, the power use of three kinds of algorithm systems is comparatively high in the beginning, and then gradually decreases and becomes steady. This indicates that the energy consumption of the algorithms presented in References [8] and [9] is 26. 6 J and 43. The drift decreases with time up to 4 J, whereas the influence of the suggested method remains at the level of 14. 7 J as the number of iterations increases. This is possible since the suggested algorithm consumes minimal power to hence be able to acquire information on the full system states and actions as well as make decisions based on the information. Energy increases because of the comparison algorithm that is not fully capable of analyzing the system's status and the actions taken, as well as due to the high energy demand required for the game that incorporates multiple actors. Figure 7 illustrates the trend in the amount of energy utilized by the system when terminal devices increase under different methods. They all have a positive linear trend as the number of devices increases, the energy consumption of the system also increases. As the number of device in a system increases in direct proportion, then the energy consumed by the system is equal to the energy consumed by each of the device that is connected in the system. Depending on the configuration, the amount of energy consumption of the system is minimalized. Nevertheless, gathering data on a global level is difficult, especially in a live, dynamic, and realistic edge network environment.

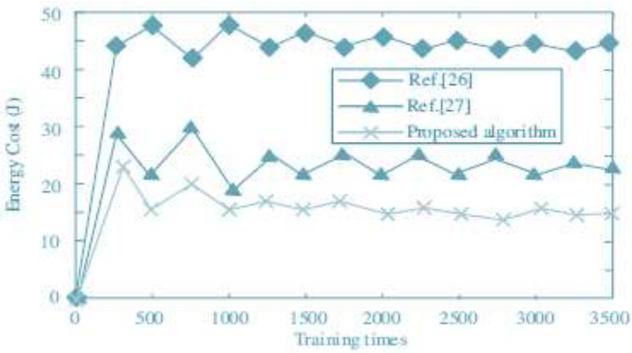
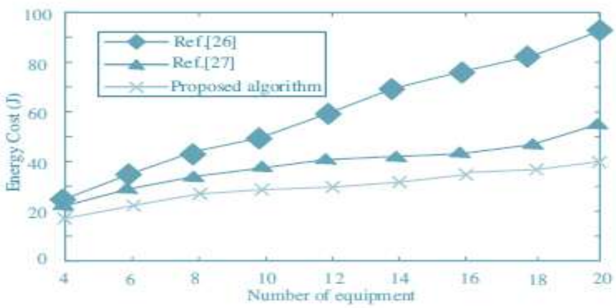


Fig 6: compares several approaches for calculating energy used by the system.

Since it is possible that a system of linked devices operate a number of devices which supposedly shares information and reproduces decisions based on such information. Hence, of the discussed, the multi-agent RL method proposed in this research is more feasible. As a result, the recommend approach is suitable for a better distribution of resources in the respective setting.



In Fig. 7, the impact of the increased number of devices on the system energy utilization under various conditions has been depicted.

5. CONCLUSION

In this paper, against the backdrop of pertaining RL approaches in connection with communication, caching, and computing in MEC, effective computing offloading and content caching schemes shall be designed. In essence, the goal is to enhance operation performance and cut better costs about various resources for edge networks, in officiating multi-user offloading decisions and subjects resources fairly. Thus, in this work, an RL-based solution is proposed to develop an improved approach to the distribution of communication resources of the 5G networks for MEC. Here, we propose the resource allocation strategy based on the DDPG which meets overall system energy, constrain on delay of tasks, computing abilities of MBS and SBS and the aim of minimizing energy utilization. Finally, the outcomes of the trials also depict that the recommended method is rather efficient to allocate the communication resources. Some specific features of real-world computing and ad-hoc edge nodes are that there are many different types of computing tasks and that some types of facilities are more suitable for some tasks than for others. As for the future direction of the research, one of the promising subjects might be a study of the computation offloading in various contexts. To take the algorithm one step further and address use cases that are different from each other, it is necessary to consider how the characteristics of some devices, the network parameters, and the interference in the cells affect the allocation of resources at the same time.

REFERENCES

1. Mao, Yuyi, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. "A survey on mobile edge computing: The communication perspective." IEEE communications surveys & tutorials 19, no. 4 (2017): 2322-2358.
2. Mao, Yuyi, Jun Zhang, and Khaled B. Letaief. "Dynamic computation offloading for mobile-edge computing with energy harvesting devices." IEEE Journal on Selected Areas in Communications 34, no. 12 (2016): 3590-3605.
3. Wang, Feng, Jie Xu, Xin Wang, and Shuguang Cui. "Joint offloading and computing optimization in wireless powered mobile-edge computing systems." IEEE transactions on wireless communications 17, no. 3 (2017): 1784-1797.
4. Li, Bin, Zesong Fei, and Yan Zhang. "UAV communications for 5G and beyond: Recent advances and future trends." IEEE Internet of Things Journal 6, no. 2 (2018): 2241-2263.
5. Mohajer, Amin, F. Sorouri, Abbas Mirzaei, Amir Ziaeddini, K. Jalali Rad, and Maryam Bavaghar. "Energy-aware hierarchical resource management and backhaul traffic optimization in heterogeneous cellular networks." IEEE Systems Journal 16, no. 4 (2022): 5188-5199.
6. Sadeghi, Alireza, Fatemeh Sheikholeslami, Antonio G. Marques, and Georgios B. Giannakis. "Reinforcement learning for adaptive caching with dynamic storage pricing." IEEE Journal on Selected Areas in Communications 37, no. 10 (2019): 2267-2281.
7. Wang, Xiaofei, Chenyang Wang, Xiuhua Li, Victor CM Leung, and Tarik Taleb. "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching." IEEE Internet of Things Journal 7, no. 10 (2020): 9441-9455.
8. Alfakih, Taha, Mohammad Mehedi Hassan, Abdu Gumaei, Claudio Savaglio, and Giancarlo Fortino. "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA." IEEE Access 8 (2020): 54074-54084.

9. Yang, Zhong, Yuanwei Liu, Yue Chen, and Gareth Tyson. "Deep reinforcement learning in cache-aided MEC networks." In ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2019.
10. Zhang, Ke, Yuming Mao, Supeng Leng, Quanxin Zhao, Longjiang Li, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks." IEEE access 4 (2016): 5896-5907.