Multi-Task Multi-User Offloading in Mobile Edge Computing

Nouhaila Moussammi[1], Mohamed El Ghmary[2], Abdellah Idrissi[3]
Department of Computer Sciences, Faculty of Sciences, Mohammed V University in Rabat, Morocco[1,3]
Department of Computer Sciences, FSDM Sidi Mohamed Ben Abdellah University, Fez, Morocco[2]

***Abstract*—Mobile Edge Computing (MEC) is a new method to overcome the resource limitations of mobile devices by enabling Computation Offloading (CO) with low latency. This paper proposes a multi-user multi-task effective system to offload computations for MEC that guarantees in terms of energy, latency for MEC. To begin, radio and computation resources are integrated to ensure the efficient utilization of shared resources when there are multiple users. The energy consumed is positively correlated with the power of transmission and the local CPU frequency. The values can be adjusted to accommodate multi-tasking in order to minimize the amount of energy consumed. The current methods for offloading aren't appropriate when multiple tasks and multiple users have high computing density. Additionally, this paper proposes a multi-user system that includes multiple tasks and high-density computing that is efficient. Simulations have confirmed the Multi-User Multi-Task Offloading Algorithm (MUMTOD). The results in terms of execution time and energy consumption are extremely positive. This improves the effectiveness of offloading as well as reducing energy consumption.**

Keywords*—*Time execution; energy consumption; computation offloading; mobile edge computing

I. Introduction

Mobile devices such as tablets and smart phones are becoming more and more popular. This has led to increasing the number of mobile apps, such as augmented Reality, natural language processing and interactive online gaming [1]. These kinds of mobile apps are typically resource-hungry and latency-sensitive, which means they require greater computing capacity and use more energy, but they have serious time constraints. Mobile devices are constrained in size and are limited in resources, such as CPU-cycle frequency and energy consumption, memory. Computing offloading is a viable solution. Mobile users are now able to perform their computing tasks to devices with the right computing resources or a computing server by using the computation offloading.

A new paradigm of computing called MEC was suggested to solve this issue. It is created to relieve mobile devices of massive computing workloads [2]. MEC offers CC capabilities at the edge of the cellular network, near mobile devices. MEC permits mobile apps to run directly on the device or on the MEC servers [3]. The MEC model offers high bandwidth, low latency and high computing speed when compared to traditional MCC. This is because of the smaller distance between mobile devices and the edge servers [4]. Recent MEC paradigm developments [5] shift computation-intensive tasks away from mobile devices and toward nearby MEC servers. Among these advancements is single-user computation. The majority of these technologies have a common objective which is to reduce energy use, allocate radios and increase computational resources, decrease costs and/or meet the delays required by mobile IoT networks. MEC is a rapidly developing computing model, extends cloud and the services, it provides to the edges of the network for applications that are resource-intensive and require the highest level of performance, MEC network computation offloading is a viable method to allow mobile apps that are resource-intensive. The SMD is a limited source of energy to execute tasks and this is a major issue. Additionally when offloading is utilized specific parts of applications that require computational power are divided into multiple, mutually exclusive offloadable tasks [6]. We believe that mobile users from multiple locations are able to offload computation tasks that they have duplicated to network edge servers and then share the results between them. Edge computing is the process of transferring processing and storage toward the edges of connected devices. Edge computing isn't located in devices.

Edge computing relies on the offloading of computation. It is first utilized in cloud computing and later in edge computing. To transfer computation tasks to servers devices may use computing offloading. In certain situations, all computation tasks are not able to transfer to servers because of limitations on network connectivity restrictions on network connectivity [7]. It is crucial to swiftly determine how many tasks need to be transferred to servers and which ones should remain local. Only when computation tasks are properly offloaded can achieve the QoS and enhance the QoE.

In this paper, we will discuss the allocation of resources, computation offloading to enhance the efficiency of multi-user multi-task offloading computation for MEC. This is the reason behind our research: The most significant factors that impact the effectiveness of multi-user MEC systems are the computational resources on the edge, wireless channel radio resources, and the offloading of computation of mobile device user's tasks. Then, it is essential to develop an approach to these problems. MEC system lets users transfer their application data to the MEC server through the wireless channel. This paper describes an integrated model for resource allocation. The model is based on multi-user, multi-task computing environment that permits MEC within mobile IoT networks. The goal is to decrease the consumption of energy within the computational latency constraints.

This paper is focused on the following aspects: A model that incorporates resource allocation can be described as an optimization issue to reduce the energy consumed under the constraints of computation latency in multi-user, multiple-task MEC systems that are used in mobile IoT networks.

The MUMTOD algorithm is developed to take the most efficient offloading choice for the tasks of computation of every mobile device user in the MEC system. The remainder of the paper is based on these steps: Section II discusses related research regarding computation offloading. Our system model multi-user multitask offloading of computations and problem formulation in section III. Our proposed solution in Section IV. Section V includes simulations to demonstrate our model for computation offloading. The paper concludes in Section VI.

II. Related Work

Task offloading is a crucial method to overcome the limitations of storage for edge computing as well as computing power within the IoT network. Edge devices can outsource some or all of its computing functions to an edge computing server. This can increase the speed of processing, conserve energy, and decrease the time to respond. The research continues to find the most efficient optimization strategy for various situations. MEC is a crucial element of devices. The MEC offers a wide range of storage and computation capabilities to mobile devices. The majority of these research address the problem of offloading computations to a single user such as [8]. Others address offloading of computation in multi-user environments[9].

The researchers [10] looked at three different types of offloading decisions including partial offloading, complete offloading, as well as local execution. The goal of this study is to optimize computational resources and decrease the amount of time needed to complete an independent task within the MEC system. A dynamic offline strategy for single-user computation offloading has been suggested using both deterministic and random methods[11]. This method takes into account offloading computation and resource scheduling to establish the most efficient schedule offloading policy. This method aims to decrease energy consumption while also satisfying the requirements for delay. A scenario where users receive the list of computation tasks to be offloaded. Each task must be handled by the MEC server within a fixed time frame[12]. The proposed optimization problems aim at the reduction of energy usage, the total processing delays as well as the insatiable processing workload.

The authors of [13] investigated multi-user edge computing scenarios that are based on orthogonal frequency division multiple access (OFDMA) and time division multiple access (TDMA). The limitation of computing delay implies that the optimal resource allocation algorithm should be convex. This resolves the issue by minimizing the weighting and power consumption of mobile devices. To simplify the process, a less efficient resource allocation algorithm is recommended for cloud systems with small capacity. The issue of offloading in multi-user situations was investigated by [14]. The authors proposed a simple search algorithm that consists of two segments to determine an optimal conditional time. A method of coordinate descent was designed to improve mode selection.

While the work [15] focused on one server system that fixed local computational resources, as well as the transmission capabilities, this is the first time investigating the real MEC system where some users are subscribers with the highest priority for services that offload computation. This research is an important expansion of the research [15]. Task offloading serves two primary objectives: to decrease the time required to execute applications that run on user equipment and to save energy . Efficiency of task offloading is a multi-fold issue. MEC servers might not be as efficient as cloud servers, therefore it is crucial to assign the tasks. Transferring offloading of user equipment to the MEC servers should be carried out in this way the MEC servers make use of their resources in a responsible manner.

The MEC Servers hosting on different Base station might not share the identical set of drivers Services, as well as the computational load on MEC servers could also be affected Variable as the passage of time. The optimal allocation of resources to relieve load Tasks, fair and steady load distribution among the MEC servers. The seamless coordination between edge and cloud is just one of the aspects that make up seamless integration. The issue is the difficulty in using of the MEC of the MEC services efficiently [16]. Another aspect The size of the defined remains a problem which is the Tasks that need to be delegated. The QoE requirements will determine the tasks that need to be delegated. The process of running applications can be difficult. It is also possible that the tasks be a matter of distinct priority, the impact of computing overhead, progress and other dependencies[17]. The scheduling, selection, location and the management of work are all a part of the process. A myriad of issues could arise from the task of offloading. This is why the key to optimal task delegation is to select the best correct task to offload to the correct MEC server at the correct time to allow for meeting. The most efficient performance of an app running when using the MEC utilizes resources efficiently. It is clearly an optimization that is multi-objective problem is considered to be NP hard [18]. In the context of this debate, task offloading could be defined Uploading the entire module to an application that contains the Calculation, data required and other libraries that are dependent on it can be delivered to remote locations server, and then receiving the result of the computation from the remote server. It is crucial to talk about the process of offloading, scheduling is crucial to offload tasks, which is a method for executing a list of tasks for a specific number of computer resources that can be utilized to achieve a goal [19].

In [20] an energy-saving, dynamic scheduling strategy and offloading method developed to cut down on energy use and speed up the completion of applications. The problem could be turned into a problem of energy efficiency by reducing task dependence and the time limit for completion. It is broken down into three sub-tasks: controlling the frequency of the clock, transmission power allocation, and calculation offload selection. Wang and coworkers addressed the MEC offloading problem. Wang et al. suggested a new framework for offloading based on deep reinforcement that can be used to automatically identify the most efficient way to load for different scenarios based on the specifics of each offloading job. This reduces the total delay in service [21], and the others concentrated on management of resources in a multi-user MEC system. The research did not offer any insights into the effect of channel-specific information on the devices' consumption of energy. To address these issues we will focus on task offloading process within multi-user MEC systems and provide an energy efficient stochastic framework. The optimization issue is solved with an effective algorithm that doesn't require prior knowledge of the time of task completion or any statistics about the channel state.

The authors of [22] looked at the possibility that the wireless energy transfer was the main driver of MCC , and developed an energy-efficient framework to increase the probability of computing while balancing the limitations of delay and energy. A majority of these researches relied on the assumption or prediction of the task's timing or channel status. The load-shedding of traffic from IoT devices and the quality of wireless channels are complex and are difficult to predict.

The work [23] focused on a partial computation offloading method, it optimizes computational and communication resources employing the dynamic voltage scaling, an algorithm design is included in the description of this method. The main goal of the method is to reduce energy consumption and delay in execution. This can be accomplished by taking into account computational time, transmission power and the offloading ratio.

An Markov chain theory-based search algorithm that is one-dimensional is described in [24]. The algorithm is presented in [24]. It determines the best strategy for offloading stochastic computations in a single-user MEC system. The perspective of scheduling flow shops is further developed by the work of [23] that has created an optimization issue for task-offloading scheduling and the allocation of power for transmission in an MEC system, which an energy harvesting technique is employed to cut down on the power consumption and energy consumption for data transmissions to enable offloading computation. It uses a simple algorithm to determine the best offloading option for every time slot [24].

III. System Model and Problem Formulation

This research aims to decrease the use of energy in multi-user edge computing devices that are multi-tasking by deploying the concept of offloading computation. This section is dedicated to a system model adopted during our research.

This paper investigates the concept of a multi-user as well as a multi-tasking system as illustrated in Fig. 1.

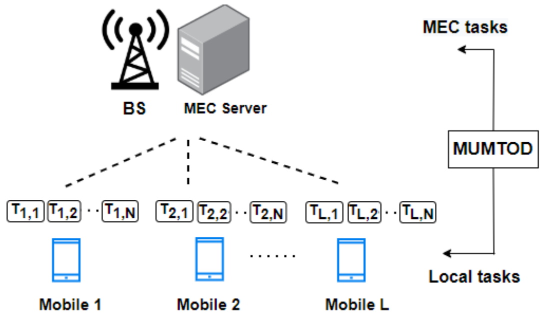

Fig. 1. System model and problem formulation

Take $U = \{1, 2, .., L\}$ as a set of mobile devices. Each mobile is assigned $T = \{1, 2, .., N\}$ which is a set of tasks that must be accomplished, which are linked with a base station. Let's start with the introduction of communication in a MEC, Our environment includes L users. Each user has to complete computation tasks. Let $x_{i,j} \in \{0, 1\}$ represents the number of integers that are used to compute offloading decisions for the task j of the user of a mobile device i. Particularly if $(x_{i,j} = 0)$, the task j of the user of a mobile device i will be performed locally. in contrast, $(x_{i,j} = 1)$ the task j of the user of mobile devices i is transferred to the MEC server through the wireless channel.

This is why we have chosen $X = \{x_{1,1}, x_{1,2}, .., x_{L,N}\}$, as the offloading profile to handle the computing tasks of all users of mobile devices. Frequency Division Multiple Access is (FDMA) is a multi-access technique with higher performance. Every user will be provided with just a portion of the bandwidth that the system provides. This is , the data uplink rate of every mobile user i is as follows:

$$r_i(p_i) = Blog(1 + p_i\frac{h^2}{\omega_0}) \tag{1}$$

which B is the bandwidth of the channel. p_i represents the mobile's i transmission power, h is the channel's gain, and ω the channel's noise power. If all mobile IoT devices decide to delegate their computing tasks to the wireless access channel at once in a computation offloading time, there is the limit on data rate R.

$$\sum_{i=1}^{L}\sum_{j=1}^{N} x_{i,j}r_i \leq R \tag{2}$$

Every smartphone user i is equipped with N computing tasks. The tasks can be completed local on the device, or remotely on the MEC server through wireless communication. We further define $u_{i,j}$ as the user of the mobile device i who is requesting the task of computation j to be executed.

Every computation task j is identified by $(C_{i,j}, D_{i,j})$ in which $C_{i,j}$ refers to the number of CPU cycles needed to complete the task of computation j and $D_{i,j}$ is the total size of data that is able to be offloaded. The time required to transfer the result from the MEC server to the mobile device users is not taken into account in this study. This is because the result is usually smaller than input data [25]. All computations are performed locally using a mobile device for local execution. the total time of the task i that is executed locally can be expressed as:

$$T_{i,j}^{loc} = \frac{C_{i,j}}{F_i^{loc}} \tag{3}$$

where F_i^{loc} represents the computing capacity of mobile i. Different mobile devices may have varying computational capabilities. The energy required to perform the task locally is determined as follows:

$$E_{i,j}^{loc} = y_i C_{i,j} \tag{4}$$

y_i is the value of the energy consumed by CPU cycles. It is determined by [24]. In MEC Server: A user of a mobile device i chooses to transmit computation task j to an edge server through the wireless channel. The time required for task execution when offloading is determined by time of transmission which is the amount of time required for users of mobile devices to offload the task of computation and also the amount of time needed for the computation task to run in the MEC servers which is task execution.

Furthermore, the energy usage for the offloading process is only measured by the cost of communication for offloading the task information to the MEC server. The total amount of offloading time which is the sum of the transmission time and execution time and energy used in the transmission are calculated using (1).

$$T_{i,j}^{up+exec} = \frac{D_{i,j}}{r_i} + \frac{C_{i,j}}{F_i^s} \tag{5}$$

$$E_{i,j}^{up} = p_i \frac{D_{i,j}}{r_i} \tag{6}$$

F_i^s represents the computing power, measured in the form of CPU cycles per (s) of the edge server which was assigned to the user i. We suppose that all users of mobile devices are part of the edge computing server's computation resources.

This section presents also an optimization problem to achieve efficacy in computation offloading in a multi-user multi-task MEC system. Every user of a mobile device is accountable for the execution time as well as the energy consumed. In addition, the allocation of available compute and radio resources on the edge server is also managed. The problem of offloading is described as a constrained optimization problem:

$$\begin{array}{lll}
\text{(P1)} & \min_x \sum_{i=1}^{L} \sum_{j=1}^{N} (E_{i,j}^{loc} + E_{i,j}^{up}) & \\
 & \quad \text{s.t} & \\
\text{(C1)} & \sum_{i=1}^{L} \sum_{j=1}^{N} x_{i,j}.r_{i,j} \leq R & \forall i,j \\
\text{(C2)} & \sum_{i=1}^{L} \sum_{j=1}^{N} x_{i,j}.F_i^s < F & \forall i,j \\
\text{(C3)} & x_{i,j} \in \{0,1\} & \forall i,j \\
\text{(C4)} & \sum_{i=1}^{L} \sum_{j=1}^{N} (E_{i,j}^{loc} + E_{i,j}^{up}) < E^{max} &
\end{array}$$

The goal of the optimization problem is to reduce the amount of energy consumed by mobile device users by using task offloading. The capacity for data rates is the constraint C1. The top limit of the CPU's frequency is shown by the constraint C2. Constraint C3 indicates that task-offloading decision variables are binary variables. Constraint (C4) means that both the Edge server and mobile device use less energy than the maximum energy consumption E^{max}.

IV. Proposed Solution

This section will introduce our solution (MUMTOD), a multi-user multi-task offloading decision algorithm. It addresses the optimization issue.

This section explains the multi-user multitask offloading decision algorithm. It provides the specifics of the procedure to get the most optimal MEC algorithm for computing offloading. Each computation task is initiated by using $x_{i,j} = 0$ that indicates local execution.

Every mobile device transmits the specifications of every computation task, including $\{C_{i,j}, D_{i,j}, y_i, p_i\}$ along with the computational capability F_i^{loc} to the MEC server. The MEC server then determines the mobile device's uplink data rate using the equation (1). MEC server then determines the most optimal solution for the offloading of computations decision by using the constraint (C2). In the end, MEC server sends the decision to each mobile device , which reducing the energy used in the entire system .The algorithm 1 describes the procedure of offloading multitask multiuser computation the decision.

Algorithm 1 Multi-User Multi-Task Offloading Decision Algorithm

```
1: **Initialization:** Each mobile device user $i$ initializes the
   offloading decision for each computation task with $x_{i,j} = 0$
2: **for all** each mobile user $i$ ($i \in U$) **do**
3:    **for all** each computation task $j$ ($j \in T$) **do**
4:       **Transmit** the computational capability of each mobile user $F_i^{loc}$ and the requirements of computation
         task $\{C_{i,j}, D_{i,j}, y_i, p_i\}$ to the edge server
5:    **end for**
6: **end for**
7: Based on equation (1): **Calculate** the uplink data rate $r_i$,
   for every device.
8: **Utilize** Equation (2) to determine the most efficient computation offloading decision value $x_{i,j}$ for all computation
   tasks. This will reduce energy consumption overall.
   **Send** the decision value $x_{i,j}$to each device
```

TABLE I. Simulation Parameters

Parameter	Value
Number of mobile device users	100
The background noise	10^{-9}
Task generation rate (seconds)	10
Computational load (millions of CPU cycles)	200
Number of computation tasks	100
Data entry size (MB)	10

To demonstrate and assess the model, a simulation of two various scenarios was conducted. They are:

Computation Offloading (CO): The CO policy permits all mobile device users to perform their computations locally on the device or via a MEC server. It is based on the model of optimization described in [26].

The Proposed Solution (PS): considers computation, communication, and the impact on the consumption of energy.

V. Simulation Results and Discussion

In this section in this section, we'll first outline the outcomes of the algorithm we propose and verify its effectiveness. Then, we'll demonstrate the results obtained by the parameters of the system.

The algorithm MUMTOD was implemented within Eclipse IDE version 2022- 03 (4.23.0) and written in the Java programming language. One application was identified as [27] for the experiment as shown in Table I.

The simulation parameters are described as follows, we are considering N=100 mobile devices and 100 independent computation tasks which can be run locally or remotely via

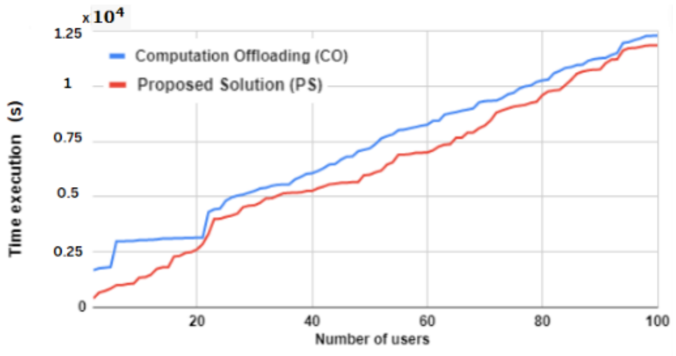

Fig. 2. Execution time for different mobile devices L=100

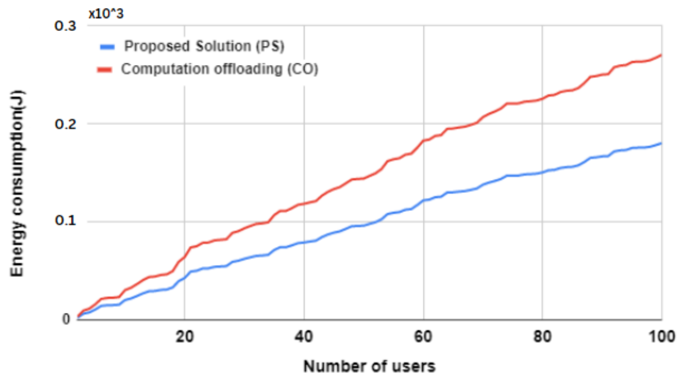

Fig. 3. Energy consumption for different mobile users L=100

the edge server. The following sections provide a summary of the results. The time of simulation was measured in (s) and the scenarios were constructed using a predetermined number 100 tasks.

In Fig. 2, computation tasks are measured in relation to the number of users on the mobile devices. The total time to execute for both scenarios is displayed in the figure in two curves. the execution of our model is nearly as fast or quicker than the time required by the computation offloading scenarios in the case of 24 mobile devices owners. Although as the number of mobile users is growing and our model is more efficient than the scenario that uses computation offloading. This is because the communication channels shared by all users are overloaded. The time required to improve communication speeds due to the growing number of mobile devices has also increased.

Fig. 3 illustrates the amount of energy used to complete the computation tasks in two scenarios. This is then compared to the number of users. It is observed from Fig. 3 that energy consumption significantly increases when the number of users of mobile devices increases the model proposed uses less energy than the alternatives. As the number of mobile devices grows the gap gets bigger. Our model can also help reduce the energy usage.

Offloading scenarios consume more energy than other model. Every mobile device user and information transmitted are competing for the limited resources of communication.

VI. Conclusion

This paper proposes a multi-user multi-task offloading algorithm to support MEC in the mobile user. A problem of optimization was designed to determine near-optimal offloading choices for every mobile user. This is done in order to reduce the energy consumed by mobile users of devices. A thorough process was employed to develop an efficient offloading algorithm that could be used to solve the optimization issue. Additionally,our model was more efficient than other computational offloading scenarios in terms of execution time as well as energy consumption which use different simulation settings and has an improved selection of tasks to offload. Our model is able to support multi-user, multiple-task computation offloading within MEC to mobile IoT networks.

For our future work, a secure computation offloading model that ensures the security of edge computing on mobile devices will be examined. This layer safeguards the data transmitted from cyberattacks.

References

[1] K. Panetta et al., " A comprehensive database for benchmarking imaging systems ", IEEE transactions on pattern analysis and machine intelligence, vol. 42, no 3, p. 509-520, 2018.

[2] L. Jiao, H. Yin, H. Huang, D. Guo, et Y. Lyu, " Computation offloading for multi-user mobile edge computing ", IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), p. 422-429, 2018.

[3] Y. Wen, W. Zhang, et H. Luo, " Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones ", proceedings IEEE Infocom, p. 2716-2720, 2012.

[4] J. Luo, X. Deng, H. Zhang, et H. Qi, " QoE-driven computation offloading for edge computing ", Journal of Systems Architecture, vol. 97, p. 34-39, 2019.

[5] P. Mach et Z. Becvar, " Mobile edge computing: A survey on architecture and computation offloading ", IEEE communications surveys & tutorials, vol. 19, no 3, p. 1628-1656, 2017.

[6] M. El Ghmary, et al. Efficient multi-task offloading with energy and computational resources optimization in a mobile edge computing node. International Journal of Electrical & Computer Engineering (2088-8708), 9(6), 2019.

[7] Y., Hmimz, et al.Bi-objective optimization for multi-task offloading in latency and radio resources constrained mobile edge computing networks. Multimedia Tools and Applications, 80(11), 17129-17166, 2021.

[8] M. El Ghmary, Y. Hmimz, T. Chanyour, A. Ouacha, et M. O. C. Malki, " Multi-task Offloading and Computational Resources Management in a Mobile Edge Computing Environment ", 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), p. 1-7, 2020.

[9] I. A. Elgendy, W. Zhang, Y.-C. Tian, et K. Li, " Resource allocation and computation offloading with data security for mobile edge computing ", Future Generation Computer Systems, vol. 100, p. 531-541, 2019.

[10] M. El Ghmary, et al.Time and resource constrained offloading with multi task in a mobile edge computing node. International Journal of Electrical & Computer Engineering (2088-8708), 10(4), 2020.

[11] Z. Ning, P. Dong, X. Kong, et F. Xia, " A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things", IEEE Internet of Things Journal, vol. 6, no 3, p. 4804-4814, 2018

[12] T., Chanyour, et al. Delay-aware and user-adaptive offloading of computation-intensive applications with per-task delay in mobile edge computing networks. International Journal of Advanced Computer Science and Applications, 11(1), 2020.

[13] C. You, K. Huang, H. Chae, et B.-H. Kim, " Energy-efficient resource allocation for mobile-edge computation offloading ", IEEE Transactions on Wireless Communications, vol. 16, no 3, p. 1397-1411, 2016.

[14] S. Bi et Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading", IEEE Transactions on Wireless Communications, vol. 17, no 6, p. 4177-4190, 2018.

[15] Y., Hmimz,et al. Energy efficient and devices priority aware computation offloading to a mobile edge computing server. In 2019 5th International Conference on Optimization and Applications (ICOA) pp. 1-6. IEEE, 2019.

[16] H. Guo, J. Liu, H. Qin, et H. Zhang, " Collaborative computation offloading for mobile-edge computing over fiber-wireless networks ", in GLOBECOM 2017-2017 IEEE Global Communications Conference, p. 1-,2017.

[17] Y. Mao, C. You, J. Zhang, K. Huang, et K. B. Letaief, " A survey on mobile edge computing: The communication perspective ", IEEE communications surveys & tutorials, vol. 19, no 4, p. 2322-2358, 2017.

[18] K. Wang, X. Wang, X. Liu, et A. Jolfaei, " Task offloading strategy based on reinforcement learning computing in edge computing architecture of internet of vehicles ", IEEE Access, vol. 8, p. 173779-173789, 2020.

[19] P. Mach et Z. Becvar, " Mobile edge computing: A survey on architecture and computation offloading ", IEEE communications surveys & tutorials, vol. 19, no 3, p. 1628-1656, 2017.

[20] S. Guo, B. Xiao, Y. Yang, et Y. Yang, " Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing ", in IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, p. 1-9, 2016.

[21] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, et N. Georgalas, " Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning ", IEEE Communications Magazine, vol. 57, no 5, p. 64-69, 2019.

[22] C. You, K. Huang, et H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer", IEEE Journal on Selected Areas in Communications, vol. 34, n 5, p. 1757-1771, 2016.

[23] J. Bi, H. Yuan, S. Duanmu, M. Zhou, et A. Abusorrah, " Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization ", IEEE Internet of Things Journal, vol. 8, no 5, p. 3774-3785, 2020.

[24] J. Li, H. Gao, T. Lv, et Y. Lu, " Deep reinforcement learning based computation offloading and resource allocation for MEC ", in 2018 IEEE Wireless communications and networking conference (WCNC), p. 1-6, 2018.

[25] X. Chen, " Decentralized computation offloading game for mobile cloud computing ", IEEE Transactions on Parallel and Distributed Systems, vol. 26, no 4, p. 974-983, 2014.

[26] K. Zhang et al., " Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks ", IEEE access, vol. 4, p. 5896-5907, 2016.

[27] L. Huang, X. Feng, L. Zhang, L. Qian, et Y. Wu, " Multi-server multi-user multi-task computation offloading for mobile edge computing networks ", Sensors, vol. 19, no 6, p. 1446, 2019.