

# Energy and Processing Time Efficiency for an Optimal Offloading in a Mobile Edge Computing Node

Mohamed El Ghmary<sup>1\*</sup>, Youssef Hmimz<sup>1</sup>, Tarik Chanyour<sup>1</sup> and Mohammed Ouçamah Cherkaoui Malki<sup>1</sup>

<sup>1</sup>Sidi Mohamed Ben Abdellah University, LIIAN Labo, FSDM, Fez, Morocco.

**Abstract:** This article describes a processing time, energy and computing resources optimization in a Mobile Edge Computing (MEC). We consider a mobile user MEC system, where a smart mobile device (SMD) demands computation offloading to a MEC server. For that, we consider a SMD contains a set of heavy tasks that can be offloadable. The formulated optimization problem takes into account both the dedicated energy capacity and the processing times. We proposed a heuristic solution schema. To evaluate our solution, we realized a range of simulation experiments. The results obtained in terms of treatment time and energy consumption are very.

**Keywords:** Computation Offloading; Mobile Edge Computing; Energy; Processing time; Bi-objective Optimization; Heuristic; Multi-task.

## 1. Introduction

MEC represents a key technological and architectural concept for enabling 5G evolution as it advances the mobile broadband network transformation into a programmable world and helps meet the high demands of 5G, terms of latency, scalability and automation [1]. It provides services to consumers, businesses, mobile operators, and adjacent industries that can now deploy critical applications over the wireless network. The MEC environment is characterized by low latency, proximity, high bandwidth, and real-time visibility of radio network information and location. It enables massive, low-power devices to handle of computation-extensive tasks in real-time, which has therefore attracted growing research interests in both academia and industry.

MEC allows mobile terminals to access abundant computing and storage resources in the Edge server. This allows a mobile terminal to offload resource intensive tasks for processing and execution in the Edge, when the local capacity is not sufficient. In the literature, several "frameworks" which allow offloading in the cloud have been proposed such as [2-4]. Among the conditions, which favored the offloading approach, we cite: the available resources of devices such as the CPU, RAM and the state of the network. Partition tasks for execution between servers and devices, in order to reduce the combination of network bandwidth and CPU consumption [5]. The focus of [6] is to review the architectures, infrastructure, and algorithms that underpin resource management in fog/edge computing. Fog computing is considered a local extension of the cloud, it is a complementary technology to cloud computing [7]. Fog nodes have been used to provide computing services close to endpoint equipment and to minimize the response time of these nodes under an energy efficiency constraint [8, 9]. The authors [10] have proposed a heuristic solution to solve a

hard decision problem that jointly optimized the overall energy of the system and maximized the satisfaction of SMDs while maintaining their priority. The authors [11] have built a distributed application that manages communication and processing by distributing the load between Cloud Computing and peripherals in order to speed up processing compared to the Internet of Things (IoT) application entirely hosted in the Cloud.

In this work, we consider a multitasking offloading environment with a single user, in order to optimize the communication resources, the local frequency of the SMD and the frequency of the Edge Node (EN), by introducing the available energy of SMD as a constraint. Moreover, we introduced the Edge server's frequency as a decision variable in our optimization problem. Therefore, we can extend the battery life-time of the SMD and reduce the processing time of its tasks. The authors of [12, 13] also proposed multitasking offloading by optimizing communication resources and local frequency without taking into account the amount of local energy available. In addition, they considered the frequency of the Edge server constant.

In the following, we present the system model and the optimization problem formulation in sections 2 and 3. Then, we present the solution of the proposed problem in sections 4 and 5. Evaluation and result are presented in section 6. Finally, section 7 concludes the paper.

## 2. System Model

As shown in Figure 1, a SMD containing an offloadable multi-task set. This SMD is connected to an EN that is equipped with a resource-rich server. Its intends to offload a set of independent tasks by the mean of an Edge Access Point (EAP). In this paper, we plan to study the behavior of the offloading process in an Edge environment, while we optimize computing resources available at the SMD as well as at the EN. Especially, the available energy at the SMD for tasks execution is limited. Besides, in the context of offloading, some pieces of a computationally intensive application are divided into multiple mutually independent offloadable tasks [3]. Therefore, according to the available computational and radio resources, some tasks are pick-up from the resulting tasks set to be offloaded to the EN for computing. The others are performed locally on the SMD itself. The execution of the completely set must happen within the time limit of the application. Additionally, it is assumed that the SMD concurrently performs computation and wireless transmission.

Let note  $\tau \triangleq \{\tau_1, \tau_2, \dots, \tau_N\}$  a set of N independent tasks, these tasks are assumed to be computationally intensive and

delay sensitive and have to be executed by the SMD or at the EN. In addition, the processing time of the whole tasks set cannot exceed a required maximum latency  $T^{max}$  and the total local execution energy must not exceed the tolerated given amount  $E^{max}$ . Every task is mainly characterized by two parameters  $\tau_i \triangleq (\lambda_i, d_i)$ . Also, it represents an atomic input data that cannot be divided into sub-tasks. The first one denoted  $\lambda_i$  [cycles] specifies the workload referring to the computation amount needed to accomplish the processing of this task. The second one denoted  $d_i$  [bits] identifies the amount of the input parameters and program codes to transfer from the user's local device to the Edge server. In addition, In line with Shannon equation, the transmission rate (bits/s) can be expressed in the following formula as equation (1).

$$r = W \log \left( 1 + \frac{p^T g}{W N_0} \right) \quad (1)$$

Where  $W$  stands for upstream bandwidth,  $p^T$  is the transmit power of transmission rate required by SMD to offload the input data to Edge server,  $g$  is its channel gain, and  $N_0$  is the noise power spectral density.

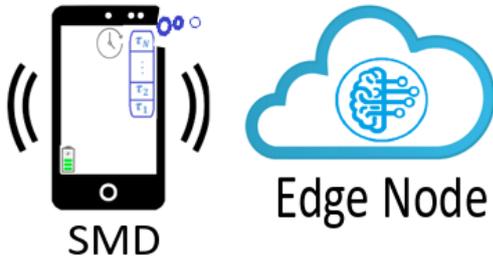


Figure 1. System model illustration

The execution nature decision for a task  $\tau_i$  either by SMD or by offloading to the EN is denoted  $x_i$  where  $x_i \in \{0; 1\}$ .  $x_i = 1$  indicates that the SMD has to offload  $\tau_i$  to the EN, and  $x_i = 0$  indicates that  $\tau_i$  is locally processed. If the SMD locally executes task  $\tau_i$ , the completion time of its local execution is:

$$t_i^L = \frac{\lambda_i}{f_L} \quad (2)$$

And for all tasks, we have:

$$t^L = \sum_{i=1}^N \frac{(1-x_i)\lambda_i}{f_L} \quad (3)$$

Additionally, the corresponding energy consumption is given by:

$$e_i^L = k_L f_L^2 \lambda_i \quad (4)$$

Hence, the total energy consumption while executing all tasks that were decided to be locally executed in the SMD is given by:

$$e^L = \sum_{i=1}^N e_i^L (1 - x_i) = k_L f_L^2 \sum_{i=1}^N \lambda_i (1 - x_i) \quad (5)$$

If task  $\tau_i$  is offloaded to the Edge Node, the offloading process completion time is:

$$t_i^0 = t_i^{Com} + t_i^{Exec} + t_i^{Res} \quad (6)$$

Where  $t_i^{Com}$  is the time to transmit the task to the EAP, and it is given by:

$$t_i^{Com} = \frac{d_i}{r} \quad (7)$$

$t_i^{Exec}$  is the time to execute the task  $\tau_i$  at the EN, and it can be formulated as:

$$t_i^{Exec} = \frac{\lambda_i}{f_S} + t_i^{Res} \quad (8)$$

$t_i^{Res}$  is the time to receive the result out from the Edge Node. Because the data size of the result is usually ignored compared to the input data size, we ignore this relay time and its energy consumption as adopted by [14]. Hence, for the  $\tau_i$  task:

$$t_i^0 = x_i \left( \frac{d_i}{r} + \frac{\lambda_i}{f_S} \right) \quad (9)$$

And for all tasks, we have:

$$t^0 = \sum_{i=1}^N x_i \left( \frac{d_i}{r} + \frac{\lambda_i}{f_S} \right) \quad (10)$$

So, the energy consumption of the communication process can be obtained by multiplying the resulting transmission period by the transmission undertaken power  $p^T$ . Thus, the energy is:

$$e^C = \frac{p^T \sum_{i=1}^N x_i d_i}{r} \quad (11)$$

Similarly, energy consumption at the Edge server [9] while executing  $\tau_i$  is given by:

$$e_i^S = k_S f_S^2 \lambda_i \quad (12)$$

The execution energy for all the offloaded tasks is:

$$e^S = k_S f_S^2 \sum_{i=1}^N \lambda_i x_i \quad (13)$$

Finally, given the offloading decision vector  $\mathbb{X} \triangleq \{x_1, x_2, \dots, x_N\}$  for all tasks, the local execution frequency  $f_L$  of the SMD, and the server execution frequency  $f_S$  at the Edge, the total execution time for the SMD is composed of its local execution time, the communication time as well as the execution time at the EN, and it is composed as:

$$T(\mathbb{X}, f_L, f_S) = t^L + t^0 \quad (14)$$

Then, according to equations (3) and (10), the total execution time can be formulated as:

$$T(\mathbb{X}, f_L, f_S) = \left\{ \frac{\sum_{i=1}^N \lambda_i (1-x_i)}{f_L} + \frac{\sum_{i=1}^N d_i x_i}{r} + \frac{\sum_{i=1}^N \lambda_i x_i}{f_S} \right\} \quad (15)$$

Similarly, the total energy consumption for the SMD is composed of its local energy consumption, the communication energy as well as the execution energy at the EN, and it is composed as:

$$E(\mathbb{X}, f_L, f_S) = e^L + e^C + e^S \quad (16)$$

Then, according to equations (5), (11) and (13), the total execution time can be formulated as:

$$E(\mathbb{X}, f_L, f_S) = (k_S f_S^2 - k_L f_L^2) \sum_{i=1}^N \lambda_i x_i + \frac{p^T}{r} \sum_{i=1}^N d_i x_i + k_L f_L^2 \sum_{i=1}^N \lambda_i \quad (17)$$

### 3. Problem Formulation

In this section, we present our optimization problem formulation that aims to minimize the overall energy consumption and overall processing time in the offloading

process, while maintaining the battery lifetime. The obtained problem is formulated as:

$$CTE(\mathbb{X}, f_L, f_S) = \frac{\alpha}{T^{\max}} T(\mathbb{X}, f_L, f_S) + \frac{\beta}{E^{\max}} E(\mathbb{X}, f_L, f_S) \quad (18)$$

Where  $\alpha$  and  $\beta$  are the weights given to the two objectives, respectively, with  $\alpha + \beta = 1$ . The role of  $E^{\max}$  and  $T^{\max}$  is to normalize the energy and processing time for the objective function, and to eliminate their units.

$$\mathcal{P}1: \min_{\{x, f_L, f_S\}} \{CTE(\mathbb{X}, f_L, f_S)\}$$

$$s.t. (C_{1.1}) x_i \in \{0; 1\}; \quad i \in \llbracket 1; N \rrbracket;$$

$$(C_{1.2}) F_L^{\min} \leq f_L \leq F_L^{\max};$$

$$(C_{1.3}) 0 < f_S \leq F_S;$$

$$(C_{1.4}) t^L = \sum_{i=1}^N \frac{\lambda_i}{f_L} (1 - x_i) \leq T^{\max};$$

$$(C_{1.5}) t_i^O = x_i \sum_{k=1}^i x_k \left( \frac{d_i}{r} + \frac{\lambda_i}{f_S} \right) \leq T^{\max};$$

$$(C_{1.6}) e^L + e^C = \frac{k_L f_L^2}{r} \sum_{i=1}^N \lambda_i (1 - x_i) + \frac{p^T}{r} \sum_{i=1}^N d_i x_i \leq E^{\max}.$$

In this work, each one of the available tasks can be either executed locally or offloaded to the Edge Node. Thus, every feasible offloading decision solution has to satisfy the above constraints:

**Table 1. Problem constraints**

Constraints	Signification
(C <sub>1.1</sub> )	Refers to the offloading decision variable $x_i$ for task $\tau_i$ which equals 0 or 1
(C <sub>1.2</sub> )	Indicates that the allocated variable local frequency $f_L$ belongs to a priori fix interval given by $[F_L^{\min}, F_L^{\max}]$
(C <sub>1.3</sub> )	Indicates that the allocated variable remote Edge server frequency $f_S$ belongs to the interval $]0, F_S^{\max}]$
(C <sub>1.4</sub> )	Shows that the execution time of all decided local tasks must be less than the given latency requirement $T^{\max}$
(C <sub>1.5</sub> )	The offloading time of all decided remote tasks must satisfy the same latency requirement $T^{\max}$
(C <sub>1.6</sub> )	Is important especially if the SMD's battery power is critical. It imposes that the total local execution energy must not exceed the tolerated given amount $E^{\max}$

## 4. Problem Resolution

In this section, we will introduce how we derive our solution from the obtained optimization problem.

### 4.1. Problem Decomposition

In our proposed model, the offloading decision vector for all the tasks is denoted  $\mathbb{X}$ . Let define the vector that contains the offloadable tasks' identifiers:

$$\mathbb{X}_1 = \{i \in \mathbb{X} / x_i = 1\} \quad (19)$$

$$\mathbb{X}_0 = \{i \in \mathbb{X} / x_i = 0\} \quad (20)$$

For ease of use, let note:

$$\Lambda_i = \sum_{k=1}^i \lambda_k,$$

$$\Lambda_1 = \sum_{i=1}^N x_i \lambda_i, \quad (21)$$

$$D_1 = \sum_{i=1}^N x_i d_i, \quad (22)$$

Where  $\Lambda_1$  is the total CPU cycles of all offloadable tasks and  $D_1$  is the total data of all offloadable tasks.

$$\Lambda_0 = \Lambda - \Lambda_1 \quad (23)$$

$$f_L^- = \frac{\Lambda_0}{T^{\max}} \quad (24)$$

$$f_L^+ = \sqrt{\frac{E^{\max} - \frac{p^T D_1}{r}}{k_L \Lambda_0}} \quad (25)$$

$$f_S^- = \frac{\Lambda_1}{T^{\max} - \frac{D_1}{r}} \quad (26)$$

In addition, constraint (C<sub>1.4</sub>) can be reformulated as  $\frac{\Lambda_0}{T^{\max}} \leq f_L$  and constraint (C<sub>1.5</sub>) can be similarly reformulated as:

$\frac{\Lambda_1}{T^{\max} - \frac{D_1}{r}} \leq f_S$ . Thus, for a given offloading decision vector

$\mathbb{X}$ , Considering the continuous variables  $f_L$  and  $f_S$ , problem P1 is a continuous multi-variable optimization problem. The objective function  $CTE(\mathbb{X}, f_L, f_S)$  can be decomposed into the following two independent functions  $CTE_1(f_L)$  and  $CTE_2(f_S)$  where:

$$CTE_1(f_L) = \Lambda_0 \left( \frac{\alpha}{T^{\max} f_L} + \frac{\beta k_L f_L^2}{E^{\max}} \right) \quad (27)$$

$$CTE_2(f_S) = \Lambda_1 \left( \frac{\alpha}{T^{\max} f_S} + \frac{\beta k_S f_S^2}{E^{\max}} \right) + \frac{D_1}{r} \left( \frac{\alpha}{T^{\max}} + \frac{\beta p^T}{E^{\max}} \right) \quad (28)$$

This last can be equivalently decomposed into the following two independent optimization sub-problems.

$$\mathcal{P}1.1(\mathbb{X}): \min_{\{f_L\}} \{CTE_1(f_L)\}$$

$$s.t. (C_{1.1.1}) F_L^{\min} \leq f_L \leq F_L^{\max};$$

$$(C_{1.1.2}) f_L^- \leq f_L \leq f_L^+.$$

$$\mathcal{P}1.2(\mathbb{X}): \min_{\{f_S\}} \{CTE_2(f_S)\}$$

$$s.t. (C_{1.2.1}) f_S^- \leq f_S \leq F_S.$$

### 4.2. Problems Resolution

For the  $\mathcal{P}1.1$  problem, the objective function  $CTE_1(f_L)$  is a continuous function according to its variable  $f_L$  with a first order derivate:  $\frac{\partial CTE_1(f_L)}{\partial f_L} = \Lambda_0 \left( \frac{2\beta k_L f_L}{E^{\max}} - \frac{\alpha}{T^{\max} f_L^2} \right)$ .

Consequently,  $CTE_1(f_L)$  decreases on  $\left] 0, \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}} \right]$  and increases on  $\left[ \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}}, +\infty \right[$ . Then,  $CTE_1$  has an optimal minimum value at the point  $\sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}}$  without considering constraint (C<sub>1.2.1</sub>). Therefore, with (C<sub>1.2.1</sub>), we can derive the following function's optimum  $f_L^*$  given by:

$$f_L^* = \begin{cases} 0 & \text{if } \mathbb{X} = \mathbb{X}_1 \\ \emptyset & \text{if } E^{\max} \leq \frac{p^t D_1}{r} \text{ or } f_L^- > F_L^{\max} \\ & \text{or } f_L^+ < F_L^{\min} \text{ or } f_L^- > f_L^+ \\ f_L^- & \text{if } \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}} < f_L^- \\ f_L^+ & \text{if } \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}} > f_L^+ \\ \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_L T^{\max}}} & \text{otherwise} \end{cases} \quad (29)$$

For the  $\mathcal{P}1.2$  problem, the objective function  $CTE_2(f_S)$  is a continuous function according to its variable  $f_S$  with a first order derivate:

$$\frac{\partial CTE_2(f_S)}{\partial f_S} = \Lambda_0 \left( \frac{2\beta k_S f_S}{E^{\max}} - \frac{\alpha}{T^{\max} f_S^2} \right).$$

Consequently,  $ECTE_2(f_S)$  decreases on  $\left] 0, \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}} \right]$  and

increases on  $\left[ \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}}, +\infty \right[$ . Then,  $CTE_2$  has an optimal

minimum value at the point  $\sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}}$  without considering

constraint (C<sub>3.2.1</sub>). Therefore, with (C<sub>3.2.1</sub>), we can derive the following function's optimum  $f_S^*$  given by:

$$f_S^* = \begin{cases} 0 & \text{if } \mathbb{X} = \mathbb{X}_0 \\ \emptyset & \text{if } f_S^- > F_S \text{ or } \frac{D_1}{r} > T^{\max} \\ f_S^- & \text{if } \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}} < f_S^- \\ F_S & \text{if } \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}} > F_S \\ \sqrt[3]{\frac{\alpha E^{\max}}{2\beta k_S T^{\max}}} & \text{otherwise} \end{cases} \quad (30)$$

## 5. Proposed Solutions

Next, the problem relies on determining the optimal offloading decision vector  $\mathbb{X}$  that gives the optimal energy consumption and the optimal processing time. However, to iterate over all possible combinations of a set of  $N$  binary variables, the time complexity is exponential. That is not practical for large values of  $N$ . In the following, we propose a low complexity approximate algorithm to solve this question.

### 5.1 Exhaustive Search Solution

For comparison purpose, we introduce the Exhaustive Search method for feasible small values of  $N$ . This method explores all cases of offloading decisions and saves the one with the minimum trade-off the energy and processing time as well as its completion time.

### 5.2 Simulated Annealing based Solution

For our proposed solution, we use a Simulated Annealing based method [15, 16]. We start by a random offloading decision state  $\mathbb{X}$ . Then, at every step, some neighboring state  $\mathbb{X}^*$  of the current state  $\mathbb{X}$  and probabilistically decides between moving the system to state  $\mathbb{X}^*$  or staying in state  $\mathbb{X}$ . Practical, a state's variation consists of changing the offloading decision of some tasks among the set. These probabilistic transitions ultimately lead the system to move to states of lower energy. Generally, this step is repeated

until getting a good trade-off for energy and processing time is reached, or until a given number of iterations is reached.

## 6. Results and Discussion

In this section, we carried out a serie of experiments to evaluate the performance of our proposed solution. First, we present simulation setup parameters. Then, several performance analysis are detailed to prove the efficiency of our approach.

### 6.1. Simulation Setup

The presented results in this work are averaged for 100 time executions. All developed C++ simulation programs were built with GCC version 6.4.0 and run using a 2.7GHz Intel Core i7-2620M processor in a PC with a maximum 8GB of RAM. Moreover, the basic parameters of the simulation experiments are listed in Table 2.

**Table 2.** Simulations' parameters

Parameters	Values
$F_L^{\min}$	1
$F_L^{\max}$	60MHz
$F_S$	6GHz
$F_L$	$10^{-26}$
$K_L$	$10^{-29}$
$T^{\max}$	[0.5, 2]
$E^{\max}$	$[0.6, 0.8] \wedge K_L (F_L^{\max})^2$
$p^t$	0.1Watt
$r$	100Kb/s
$d_i$	[30, 300]Kb
$\lambda_i$	[60, 600]MCycles
$\alpha$	0.5

### 6.2. Evaluation

We start by studying the Trade-off between Energy Efficiency and Processing Time for each method. Thus, we carried an experiment where we vary the number of tasks parameter between 3 and 26. The experiment's results are depicted in the following tow figures. Figure 2 represents the obtained results for both Exhaustive Search Offloading based solution (ESO) and Simulated Annealing Offloading based solution (SAO). It shows a small distance between the curves representing the realized averaged tasks' energy consumption and processing time. Accordingly, the differences between the optimal ESO method and the SAO method vary from 0.00% to 0.63%.

Now, Figure 3 depicts the average of the execution time in *ms* to get the offloading decisions for both schemes. While the tasks count  $N$  is between 2 and 26, it clearly shows the exponential variation of the ESO execution time w.r.t.  $N$ . Additionally, The SAO solution gives a stable execution time that reached only 0.05*ms* for  $N=26$ .

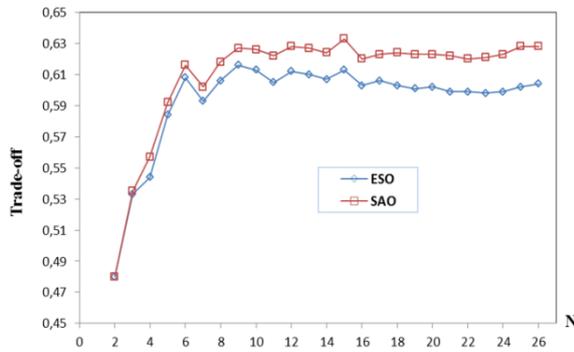


Figure 2. Trade-off energy-time w.r.t. N.

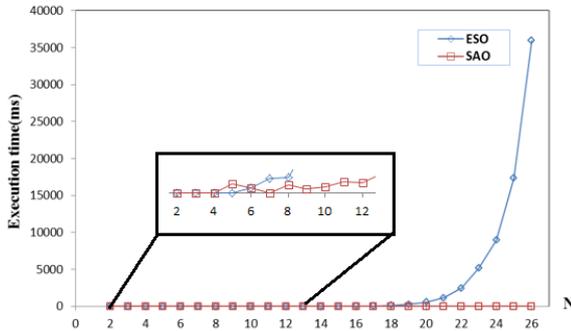


Figure 3. Decision time w.r.t. N.

This experiment shows that our proposed heuristic scheme achieves a good trade-off between the solution's execution time and the accomplished processing delays of the offloaded tasks within the EN.

## 7. Conclusion

In this paper, we propose a heuristic solution to solve a hard decision problem that jointly optimizes the computing resources, as well as trade-off between both the energy consumption and the processing time in a MEC node. A calculation task is authorized to be offloaded when the offloading consumes less time and energy than the local execution. The obtained results in terms of processing time and energy consumption are very encouraging. In addition, the proposed solution performs the offloading decisions within an acceptable and feasible timeframes.

## References

- [1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI white paper, vol. 11, no. 11, pp. 1-16, 2015.
- [2] E. Cuervo et al., "MAUI: making smartphones last longer with code offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services, pp. 49-62, 2010.
- [3] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," presented at the Proceedings of the sixth conference on Computer systems, pp. 301-314, 2011.
- [4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in 2012 Proceedings IEEE Infocom, pp. 945-953: IEEE, 2012.
- [5] R. Newton, S. Toledo, L. Girod, H. Balakrishnan, and S. Madden, "Wishbone: Profile-based Partitioning for Sensornet Applications," in NSDI, vol. 9, pp. 395-408, 2009.
- [6] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures,

infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1-37, 2019.

- [7] T. A. Ahanger, U. Tariq, and M. Nusir, "Mobility of Internet of Things and Fog Computing: Concerns and Future Directions," *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, pp. 534, 2018.
- [8] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, IEEE, pp. 1-9, 2017.
- [9] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795-2808, 2016.
- [10] Y. Hmimiz, T. Chanyour, M. El Ghmary, and M. O. Cherkaoui. Malki, "Energy Efficient and Devices Priority Computation Offloading to a Mobile Edge Computing Server," in *2019 5th International Conference on Optimization and Applications (ICOA)*, IEEE, pp. 1-6, 2019.
- [11] O. Carvalho, M. Garcia, E. Roloff, E. D. Carreño, and P. O. Navaux, "IoT workload distribution impact between edge and cloud computing in a smart grid application," in *Latin American High Performance Computing Conference*, Springer, pp. 203-217, 2017.
- [12] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud," presented at the 2016 IEEE International Conference on Communications (ICC), pp. 1-6, 2016.
- [13] L. Cui et al., "Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4791-4803, 2018.
- [14] K. Zhang et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE access*, vol. 4, pp. 5896-5907, 2016.
- [15] Z. Fan, H. Shen, Y. Wu, and Y. Li, "Simulated-Annealing Load Balancing for Resource Allocation in Cloud Environments," presented at the 2013 International Conference on Parallel and Distributed Computing, Applications and Technologies, pp.1-6, 2013.
- [16] L. Chen, J. Wu, X. Long, and Z. Zhang, "ENGINE: Cost Effective Offloading in Mobile Edge Computing with Fog-Cloud Cooperation," *arXiv preprint arXiv:1711.01683*, pp. 1-11, 2017.