Energy and Latency Minimization in a Mobile Edge Computing System

Mohamed El Ghmary[1][a], Youssef Hmimz[1][b], Tarik Chanyour[1][c]
and Mohammed Ouçamah Cherkaoui Malki[1][d]
[1]*Department of Computer Sciences, SidiMohamed Ben Abdellah University, FSDM, Fez, Morocco*

Keywords: Mobile edge computing, computation offloading, energy, processing time, bi-objective optimization, heuristic, multi-task.

Abstract: Mobile Edge Computing (MEC) extends the Cloud Computing paradigm to the edge of the network, thus enabling a new breed of applications and services; hence it is considered as a promising technique to meet the demands of computing intensive and delay sensitive applications. This article describes the value for a trade off between processing time and energy in a MEC environment. The optimization problem formulated takes into account both the processing time and the dedicated energy capacity. To solve this problem we proposed a heuristic solution scheme. To evaluate our solution, we performed a comparative study with a brute force search solution. The results obtained are entirely very satisfactory.

1 INTRODUCTION

Cloud computing (CC) invaded and promised a better way to improve server efficiency, shrink infrastructure footprint and allowed the use of centralized storage and made computing units available and on demand for the end user. However, with the rapid growing amount of data caused by the massive diffusion of Internet of Things (IoT), which refers to the billions of physical devices with limited computing capacities that are connected to the internet and need a real-time data processing (Bilal, 2018), CC is no longer considered as a suitable solution. Therefore, Mobile Cloud Computing (MCC) was introduced as a potential solution to these limitations (Dinh et al., 2013) at first, but the data processing and storage were still in the Clouds which leads us to a new concept called Fog computing where the data remains distributed among the Fog nodes (Al-Qamash et al., 2018) hence the performance is still limited because of its dependence on the Cloud.

Motivated by the above challenges and the need to make a move towards the next cellular network generation 5G that aspires to achieve substantial improvement on quality of service (Hassan et al., 2019) for IoT devices users, Mobile Edge Computing appeared with capabilities such as storage and computational capacities that offer low latency, proximity since they are located at the Edge of the network which means within the Radio Access Network (RAN) and base stations (Hassan et al., 2019), location awareness and higher bandwidth (Sabella et al., 2016). However, the main challenge is that the IoT will connect a large number of heterogeneous devices with limited local computing resources, which will require a strategy that enables these devices to offload their intensive tasks to other servers. Therefore, mobile edge server should be able to make an effective offloading decision.

The crucial part in computation offloading is deciding whether to offload or not, according to (Hmimz et al., 2019),(El Ghmary et al., 2020) we can find three possibilities: local execution, partial offloading and full offloading. In this regard, many studies have proposed a variety of solutions in order to reduce total energy consumption in a single device multi-task environment. The authors (Li, 2018) compared an algorithm that selects tasks with large data size and heavy calculation burden to another algorithm that relies on a probabilistic technique. Meanwhile in a single server, multi-user environment, (You et al., 2016) investigated the energy efficiency using a centralized method that makes a binary offloading decision for each mobile where users will perform the offloading with high or low priorities according to a given threshold. However,(Rahati-Quchani et al., 2019) consider

[a] https://orcid.org/0000-0001-5970-481X
[b] https://orcid.org/0000-0001-7625-2106
[c] https://orcid.org/0000-0002-7676-0749
[d] https://orcid.org/0000-0001-8811-9820

optimizing energy consumption by offloading mobile users with only a better utility. Finally, considering a multi-server, multi-user environment, (Huang et al., 2019) and (Lyu et al., 2016) used a decentralized assignment mechanism where the MEC aims to select mobile devices with better benefits to minimize energy consumption, and (Xu et al., 2019) employed the enumerating algorithm and branch-and-bound to get the optimal decision that minimizes energy consumption, while (Chen, 2014) proposed a decentralized computation offloading game theory that allows users to make the offloading decision assuming that they know how to keep the system balanced.

Many studies (Li, 2018), (Lyu et al., 2016), (Rahati-Quchani et al., 2019), (Xu et al., 2019) and (Chen, 2014) neglected the response time while sending the output from the MEC to the Mobile device assuming that the output is much smaller than the input, hence the return time has little effect on the total time consumption. In a single device multi-task environment, we are going to consider the local execution time, the required time to send the input data to the MEC, the execution time on the server and finally the response time to send the output to the end user in order to reduce the global time of the offloading process. In this paper, we consider a single smart mobile device multi-task offloading environment, besides, we introduce the available energy of the smart mobile device (SMD) as a constraint. Moreover, we introduced the MEC Server's frequency and the SMD frequency as a decision variable in our optimization problem. Therefore, we can extend the battery lifetime of the SMD and reduce the processing time of its tasks.

The rest of this paper is structured as follows: Section 2 gives the system model. The optimization problem formulation and its solution are discussed in Sections 3, 4 and 5. Then, the simulation results are discussed in Section 6. And finally, we conclude this paper in section 7.

2 SYSTEM MODEL

In this system model, we will consider a single user scenario where a SMD has N independent tasks that are executed locally or offloaded to the MEC Server for computation.

Figure 1 shows a general network topology that connects a SMD to a MEC Server. This SMD intends to offload a list of independent tasks by the means of an Edge Access Point (EAP). In this paper, we aim to examine the behavior of the offloading process in a MEC environment, while we optimize computing resources disposable at the SMD as well as at the MEC Server. Especially, the energy which can be taken by the SMD for tasks execution is restricted. Moreover, in the context of offloading, some parts of a computationally intensive application are splitted into several mutually independent offloadable tasks (Chun et al., 2011). Consequently, depending on the free computational and radio resources, some tasks are selected from the tasks list to be offloaded to the MEC Server for processing. The remaining tasks are handled locally by the MDS itself. The completion of all tasks must take place within the application deadline. In addition, we assume that the SMD can simultaneously perform computations and wireless transmission.

Let note: $\tau \triangleq \{\tau_1, \tau_2, ..., \tau_N\}$a list of N independent tasks, these tasks are assumed heavy and delay sensitive. In addition, these tasks can be executed by the SMD or by the MEC Server. Besides, the processing time of the task τ_i cannot exceed a required maximum latency T_i^{sup} and the overall energy for the execution of all the tasks locally must be less than the quantity of energy tolorated E^{sup}. Also, Every task represents an atomic input data that cannot be divided into sub-tasks and it is mainly characterized by three parameters $\tau_i \triangleq \langle \Gamma_i, d_i, T_i^{sup} \rangle$ where $\gamma_i[cycles]$ identifies the workload needed to accomplish the execution of this task, $d_i[bits]$ identifies the input data size and program codes to deliver from the SMD to the MEC Server. and T_i^{sup} refers to the maximum latency required for this task. In addition, In line with Shannon equation, the transmission rate($bits/s$) can be expressed in the following formula as equation (1).

$$r = B\log\left(1 + \frac{p^T g}{BN_0}\right) \tag{1}$$

where B stands for upstream bandwidth, p^T is the transmission power required by SMD to offload the input data to MEC Server, g is its channel gain, and N_0 is the noise power spectral density. The execution decision for a task τ_i either locally by SMD or by offloading to the MEC Server is denoted y_i where $y_i \in \{0;1\}$. $y_i = 1$ indicates that the SMD has to offload τ_i to the MEC Server, and $y_i = 0$ indicates that

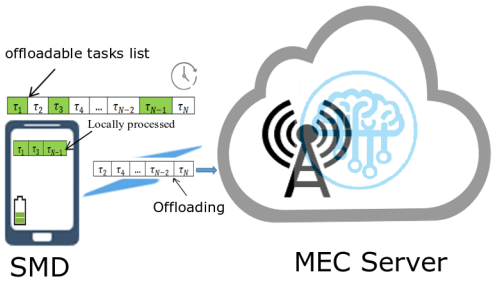

Figure 1: System model illustration

τ_i is locally executed.
If the task τ_i executed locally by the SMD, the time of its local execution is: $t_i^{smd} = \frac{\gamma_i}{f_{smd}}[seconds]$. And for all tasks we have:

$$t^{smd} = \sum_{i=1}^{N} (1-y_i) \frac{\gamma_i}{f_{smd}}[seconds], \quad (2)$$

where f_{smd} stands for CPU local frequency.
The energy consumption of its local execution is given by: $e_i^{smd} = k_{smd} f_{smd}^2 \gamma_i [J]$. Consequently, the total energy consumption for all tasks locally executed by the SMD is given by:

$$e^{smd} = \sum_{i=1}^{N} e_i^{smd} (1-y_i) = k_{smd} f_{smd}^2 \sum_{i=1}^{N} \gamma_i (1-y_i)\,[J]. \quad (3)$$

If task τ_i is offloaded to the MEC Server, The total computing time is: $t_i^{mec} = t_i^C + t_i^S + t_i^R$, where t_i^C is the time to offload the task to the MEC Server, and it is given by: $t_i^C = \frac{d_i}{r}[seconds]$.$t_i^S$ is the time to process the task τ_i at the MEC Server, and it is given by: $t_i^S = \frac{\gamma_i}{f_{mec}} + t_i^R[seconds]$. t_i^R is the time to send the result out from the MEC Server. We ignore this response time and its energy consumption as adopted by (Zhang et al., 2016) since the output data size is usually ignored compared to the input data size. Hence, for the τ_i task: $t_i^{mec} = y_i \left(\frac{d_i}{r} + \frac{\gamma_i}{f_{mec}} \right)[seconds]$. And for all tasks, we have:

$$t^{mec} = \sum_{i=1}^{N} y_i \left(\frac{d_i}{r} + \frac{\gamma_i}{f_{mec}} \right)[seconds], \quad (4)$$

where f_{mec} stands for MEC Server frequency.

Thus, the communication energy can be formulated as: $e_i^C = \frac{p^T d_i}{r}[J]$. And for all tasks, we have $e^C = \frac{p^T \sum_{i=1}^{N} y_i d_i}{r}[J]$. Similarly, energy consumption (Chen et al., 2015) at the MEC Server while executing τ_i is given by: $e_i^{mec} = k_{mec} f_{mec}^2 \gamma_i [J]$. The total energy consumption for all tasks executed by the MEC Server is given by:

$$e^{mec} = k_{mec} f_{mec}^2 \sum_{i=1}^{N} \gamma_i y_i [J]. \quad (5)$$

Finally, given the offloading decision set $\mathbb{Y} \triangleq \{x_1, x_2, ..., x_N\}$ for all tasks, the SMD execution frequency f_{smd} and the MEC Server execution frequency f_{mec}, the total processing time for all tasks can be formulated as:

$$T(\mathbb{Y}, f_{smd}, f_{mec}) = \frac{\sum_{i=1}^{N} \gamma_i - \sum_{i=1}^{N} \gamma_i y_i}{f_{smd}} + \frac{\sum_{i=1}^{N} d_i y_i}{r} + \frac{\sum_{i=1}^{N} \gamma_i y_i}{f_{mec}}. \quad (6)$$

Similarly,the total energy consumption for all tasks can be formulated as:

$$E(\mathbb{Y}, f_{smd}, f_{mec}) = K \sum_{i=1}^{N} \gamma_i y_i + k_{smd} f_{smd}^2 \sum_{i=1}^{N} \gamma_i + \frac{p^T \sum_{i=1}^{N} y_i d_i}{r}, \quad (7)$$

where $K = k_{mec} f_{mec}^2 - k_{smd} f_{smd}^2$.

3 PROBLEM FORMULATION

Now, we present in this section our optimization problem formulation that plans to minimize the total energy consumption and total processing time in the offloading process, while maintaining the battery lifetime. The obtained problem is written as follows:

$$CTE(\mathbb{Y}, f_{smd}, f_{mec}) = \frac{\alpha}{T^{sup}} T(\mathbb{Y}, f_{smd}, f_{mec}) + \frac{\beta}{E^{sup}} E(\mathbb{Y}, f_{smd}, f_{mec}) \quad (8)$$

Where α and β are the weights given to the two objectives, respectively, with $\alpha + \beta = 1$. The role of E^{sup} and $T^{sup} = max_i \left\{ T_i^{sup} \right\}$, is to eliminate the units of energy and of processing time and to normalize them in the objective function.

$$\mathcal{P}_1 : \begin{cases} \min \{CTE(\mathbb{Y}, f_{smd}, f_{mec})\} \\ s.t. \\ \quad (C_{1.1})\ y_i \in \{0;1\}; \qquad\qquad i \in [\![1;N]\!] \\ \quad (C_{1.2})\ F_{smd}^{inf} \le f_{smd} \le F_{smd}^{sup} \\ \quad (C_{1.3})\ 0 < f_{mec} \le F_S \\ \quad (C_{1.4})\ \frac{(1-y_i)\sum_{k=1}^{i} \gamma_k (1-x_k)}{f_{smd}} \le T_i^{sup}; \quad i \in [\![1;N]\!] \\ \quad (C_{1.5})\ y_i \sum_{k=1}^{i} x_k \left(\frac{d_k}{r} + \frac{\gamma_k}{f_s} \right) \le T_i^{sup}; \quad i \in [\![1;N]\!] \\ \quad (C_{1.6})\ k_{smd} f_{smd}^2 \sum_{i=1}^{N} \gamma_i (1-y_i) + \frac{p^T}{r} \sum_{i=1}^{N} y_i d_i \le E^{sup} \end{cases}$$

In this paper, each task can be either executed locally by the SMD or offloaded to the MEC Server. So, every feasible offloading decision solution has to satisfy the constraints below:
The first constraint $(C_{1.1})$ refers to the offloading decision variable y_i for task τ_i which equals 0 or 1. The second constraint $(C_{1.2})$ indicates that the allocated variable local frequency f_{smd} belongs to a priori fix interval given by $[F_{smd}^{inf}, F_{smd}^{sup}]$. Similarly, the allocated variable remote MEC Server frequency f_{mec} belongs to the interval $]0, F_S]$ in the third constraint $(C_{1.3})$. The fourth constraint $(C_{1.4})$ shows that the execution time of all decided local tasks must be less than the given latency requirement T_i^{sup}. In the same way, in the fifth constraint $(C_{1.5})$, the offloading time of all decided remote tasks must satisfy the same latency requirement T_i^{sup}. The final constraint $(C_{1.6})$ is important especially if the SMD's battery power is critical. It imposes that the local execution energy must not exceed the specified amount E^{sup}.

4 PROBLEM RESOLUTION

In this section, we present how we could resolve the obtained optimization problem.

4.1 Problem Decomposition

In our suggested model, the offloading decision set for all the tasks is denoted $\mathbb{Y}$. Let declare the set that contains the offloadable tasks' identifiers:
$\mathbb{Y}_0 = \{i \in \mathbb{Y} / y_i = 0\}$ and $\mathbb{Y}_1 = \{i \in \mathbb{Y} / y_i = 1\}$.

We define: $\Gamma_i = \sum_{k=1}^{i} \gamma_k$, $\Gamma_i^1 = \sum_{k=1}^{i} x_k \gamma_k$, $\Gamma = \Gamma_N - \Gamma_1^N$, $D_i = \sum_{k=1}^{i} d_k$ and $D_i^1 = \sum_{k=1}^{i} x_k d_k$.
Given the decision set $\mathbb{Y}_0$, constraint $(C_{1.4})$ for a local task can be reformulated as $\frac{\Gamma_i - \Gamma_i^1}{T_i^{sup}} \leq f_{smd}$; $\forall i \in [\![1;N]\!]$, that is $\max_i \left\{ \frac{\Gamma_i - \Gamma_i^1}{T_i^{sup}} \right\} \leq f_{smd}$. The constraint $(C_{1.5})$ for an offloadable task implies $\frac{D_i^1}{r} + \frac{\Gamma_i^1}{f_{mec}} \leq T_i^{sup}$; $\forall i \in [\![1;N]\!]$. Then $\frac{D_i^1}{r}$ and $\frac{\Gamma_i^1}{f_{mec}}$ must be strictly less than T_i^{sup}; $\forall i \in [\![1;N]\!]$; in particular $min_i \left\{ T_i^{sup} - \frac{D_i^1}{r} \right\} > 0$. In that case, the constraint $(C_{1.5})$ can be reformulated as: $\frac{\Gamma_i^1}{T_i^{sup}} - \frac{D_i^1}{r} \leq f_{mec}; i \in [\![i;N]\!]$. Eventually, it sums to a constraint: $\max_i \left\{ \frac{\Gamma_i^1}{T_i^{sup}} - \frac{D_i^1}{r} \right\} \leq f_{mec}$. The constraint $(C_{1.6})$ can be written as: $f_{smd} \leq \sqrt{\frac{E^{sup}}{k_{smd}\Gamma_N - \Gamma_N^1}}$. For more facilitation of use, we note: $f_{smd}^- = max_i \left\{ \frac{\Gamma_i - \Gamma_i^1}{T_i^{sup}} \right\}$, $f_{smd}^+ = \sqrt{\frac{E^{sup}}{k_{smd}(\Gamma_N - \Gamma_N^1)}}$ and $f_{mec}^- = max_i \left\{ \frac{\Gamma_i^1}{T_i^{sup}} - \frac{D_i^1}{r} \right\}$. Hence, for a given offloading decision set $\mathbb{Y}$, Considering the continuous variables f_{smd} and f_{mec}, $P1$ is an optimization problem with continuous multivariables. The objective function $CTE(\mathbb{Y}, f_{smd}, f_{mec})$ can be divided into two independent functions: $CTE_1(f_{smd})$ and $CTE_2(f_{mec})$ where:

$$CTE_1(f_{smd}) = \Gamma \left(\frac{\alpha}{T^{sup} f_{smd}} + \frac{\beta k_{smd} f_{smd}^2}{E^{sup}} \right) \tag{9}$$

$$\begin{aligned} CTE_2(f_{mec}) &= \Gamma_1^N \left(\frac{\alpha}{T^{sup} f_{mec}} + \frac{\beta k_{mec} f_{mec}^2}{E^{sup}} \right) + \\ &\quad \frac{D_1^N}{r} \left(\frac{\alpha}{T^{sup}} + \frac{\beta p^T}{E^{sup}} \right). \end{aligned} \tag{10}$$

$$\mathcal{P}_{2.1}(\mathbb{Y}) : \begin{cases} \min\{CTE_1(f_{smd})\} \\ s.t. \\ \quad (C_{2.1.1})\ F_{smd}^{inf} \leq f_{smd} \leq F_{smd}^{sup} \\ \quad (C_{2.1.2})\ f_{smd}^- \leq f_{smd} \leq f_{smd}^+ \end{cases}$$

$$\mathcal{P}_{2.2}(\mathbb{Y}) : \begin{cases} \min\{CTE_2(f_{mec})\} \\ s.t. \\ \quad (C_{2.2.1})\ f_{mec}^- \leq f_{mec} \leq F_S \end{cases}$$

4.2 Problems Resolution

For the $P2.1$ problem, variation study of the objective function $CTE_1(f_{smd})$ shows that it has an optimal minimum value at the point $\sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{smd} T_i^{sup}}}$ without considering constraints $(C_{2.1.1})$ and $(C_{2.1.2})$. Then, the optimum f_{smd}^* of the $CTE_1(f_{smd})$ function given by:

$$f_{smd}^* = \begin{cases} 0 & \textbf{if} \quad \mathbb{Y} = \mathbb{Y}_1 \\ \emptyset & \textbf{if} \quad E^{sup} \leq \frac{p^T D_1^N}{r} \textbf{ or } f_{smd}^- > F_{smd}^{sup} \\ & \textbf{or } f_{smd}^+ < F_{smd}^{inf} \textbf{ or } f_{smd}^- > f_{smd}^+ \\ f_{smd}^- & \textbf{if} \quad \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{smd} T^{sup}}} < f_{smd}^- \\ f_{smd}^+ & \textbf{if} \quad \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{smd} T^{sup}}} > f_{smd}^+ \\ \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{smd} T^{sup}}} & \textbf{otherwise} \end{cases} \tag{11}$$

For the $P2.2$ problem, variation study of the objective function $CTE_2(f_{mec})$ shows that it has an optimal minimum value at the point $\sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{smd} T_i^{sup}}}$ without considering constraint $(C_{2.2.1})$. Therefore, the optimum f_{mec}^* of the $CTE_2(f_{mec})$ function given by:

$$f_{mec}^* = \begin{cases} 0 & \textbf{if} \quad \mathbb{Y} = \mathbb{Y}_0 \\ \emptyset & \textbf{if} \quad f_{mec}^- > F_S \textbf{ or } \frac{D_1^N}{r} > T^{sup} \\ f_{mec}^- & \textbf{if} \quad \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{mec} T^{sup}}} < f_{mec}^- \\ F_S & \textbf{if} \quad \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{mec} T^{sup}}} > F_S \\ \sqrt[3]{\frac{\alpha E^{sup}}{2\beta k_{mec} T^{sup}}} & \textbf{otherwise} \end{cases} \tag{12}$$

5 PROPOSED SOLUTIONS

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

5.1 Exact Solution

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

5.2 Heuristic Solution

In this section, we present our proposed solution, which we denote Simulated Annealing Offloading (SAO), we utilize a Simulated Annealing based heuristic (Fan et al., 2013) and (Chen et al., 2017). We begin by a random offloading decision state $\mathbb{Y}$. Then, for every step, some neighboring state $\mathbb{Y}^*$ of the

current state $\mathbb{Y}$ and probabilistically decides between moving the system to state $\mathbb{Y}^*$ or staying in state $\mathbb{Y}$. In practice, varying a state consists of changing the decision to offload certain tasks to the MEC Server. These probabilistic transitions eventually cause the system to switch to lower energy states. Usually, this step is repeated until a good compromise between energy and processing time is reached, or until a given number of iterations is reached.

6 EVALUATION AND RESULTS

6.1 Simulation Setup

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. The reached results in this paper are averaged for 100 time executions. Moreover, the basic parameters of the simulation experiments are listed in Table 1.

Table 1: Simulations' parameters.

Parameter	values
F_{smd}^{inf}	$1MHz$
F_{smd}^{sup}	$60MHz$
F_S	$6GHz$
p^T	$0.1Watt$
k_{smd}	10^{-26}
k_{mec}	10^{-29}
T_i^{sup}	$[0.5,2]$
E^{sup}	$[0.6,0.8]\Gamma k_{smd}(F_{smd}^{sup})^2$
d_i	$[30,300]KB$
γ_i	$[60;600]MCycles$
r	$100KB/s$
$Temp_0$	100
α	0.5

6.2 Evaluation

The proposed solutions solve the general problem in two phases. The first phase consists in finding the optimal local frequency f_{smd} as well the remote MEC Server's frequency f_{mec} according to (11) and (12). The second phase consists in finding an optimal compromise between the total energy and the overall processing time.

For the evaluation purpose, we performed an experiment to compare the performance of the optimal ESO solution and the heuristic SAO solution that solve the formulated optimization problem. We vary the number of tasks (N) between 2 and 26. Next, we study two metrics: total energy consumption and total processing time.

Figure 2 shows the average obtained results. It shows a small difference between the optimal ESO solution and the heuristic SAO solution. This difference ranges from 0.00% to 2.00%.

Figure 3 shows the average execution time of both solutions ESO and SAO while we vary N between 2 and 26. In fact, with $N = 26$ SAO solution reaches only $0,31ms$; whereas ESO solution attains $107702,35ms$.

Indeed, this experiment shows that the higher the number of tasks N is, the longer the execution time is. Additionally, figure 3 shows an exponential variation of the execution time for the optimal ESO solution. It shows stable execution time for SAO with comparable output in terms of the trade-off between energy efficiency and processing time for both solutions.

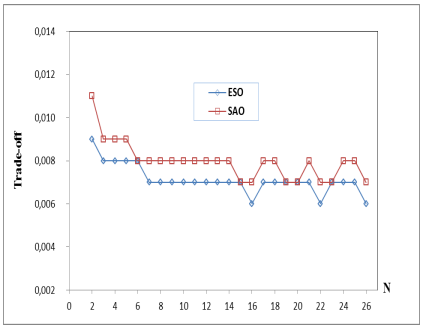

Figure 2: Trade-off energy-time w.r.t. N.

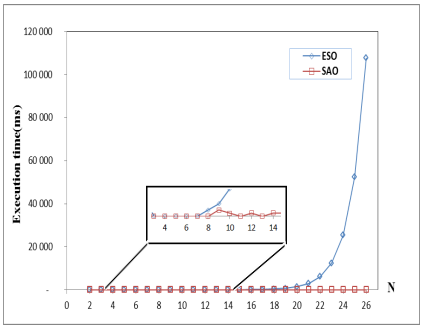

Figure 3: Decision time w.r.t. N.

7 CONCLUSION

In this paper we have proposed a heuristic solution that jointly optimizes two metrics, namely the minimization of the total processing time and the total energy consumption of all the tasks. We have dealt with a two-objective optimization problem. For that purpose, we based ourselves on an approach of aggregation of the weights which consider well spec-

ified weights for each of the two metrics in order to achieve a compromise between them. Our solution has been validated by comparing the results obtained with an accurate solution based on an exhaustive search method.

REFERENCES

Al-Qamash, A., Soliman, I., Abulibdeh, R., and Saleh, M. (2018). Cloud, fog, and edge computing: A software engineering perspectivee. In *2018 International Conference on Computer and Applications (ICCA)*, pages 276–284. IEEE.

Bilal, Kashif & Khalid, O. . E. A. . K. S. U. (2018). Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks*, 130:94–120.

Chen, L., Wu, J., Long, X., and Zhang, Z. (2017). Engine: Cost effective offloading in mobile edge computing with fog-cloud cooperation. *arXiv preprint arXiv:1711.01683*.

Chen, X. (2014). Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):974–983.

Chen, X., Jiao, L., Li, W., and Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808.

Chun, B.-G., Ihm, S., Maniatis, P., Naik, M., and Patti, A. (2011). Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314.

Dinh, H. T., Lee, C., Niyato, D., and Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611.

El Ghmary, M., Chanyour, T., Hmimz, Y., and Malki, M. O. C. (2020). Processing time and computing resources optimization in a mobile edge computing node. In *Embedded Systems and Artificial Intelligence*, pages 99–108. Springer.

Fan, Z., Shen, H., Wu, Y., and Li, Y. (2013). Simulated-annealing load balancing for resource allocation in cloud environments. In *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 1–6. IEEE.

Hassan, N., Yau, K.-L. A., and Wu, C. (2019). Edge computing in 5g: A review. *IEEE Access*, 7:127276–127289.

Hmimz, Y., Chanyour, T., El Ghmary, M., and Cherkaoui Malki, M. O. (2019). Energy efficient and devices priority aware computation offloading to a mobile edge computing server. In *2019 5th International Conference on Optimization and Applications (ICOA)*, pages 1–6. IEEE.

Huang, L., Feng, X., Zhang, L., Qian, L., and Wu, Y. (2019). Multi-server multi-user multi-task computation offloading for mobile edge computing networks. *Sensors*, 19(6):1446.

Li, H. (2018). Multi-task offloading and resource allocation for energy-efficiency in mobile edge computing. *Int. J. Comput. Tech*, 5:5–13.

Lyu, X., Tian, H., Sengul, C., and Zhang, P. (2016). Multiuser joint task offloading and resource optimization in proximate clouds. *IEEE Transactions on Vehicular Technology*, 66(4):3435–3447.

Rahati-Quchani, M., Abrishami, S., and Feizi, M. (2019). An efficient mechanism for computation offloading in mobile-edge computing. *arXiv preprint arXiv:1909.06849*.

Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., and Giust, F. (2016). Mobile-edge computing architecture: The role of mec in the internet of things. *IEEE Consumer Electronics MagazinEE*, 5(4):84–91.

Xu, J., Hao, Z., and Sun, X. (2019). Optimal offloading decision strategies and their influence analysis of mobile edge computing. *Sensors*, 19(14):3231.

You, C., Huang, K., Chae, H., and Kim, B.-H. (2016). Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411.

Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S., and Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks. *IEEE access*, 4:5896–5907.