Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 17 No. 20 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijim.v17i20.40999

PAPER

An Energy and Latency Trade-off for Resources Allocation in a MEC System

Mohamed El Ghmary[1,2](✉), Said Ziani[3], Tarik Chanyour[4], Youssef Hmimz[1]

[1]Departement of Computer Science, Faculty of Sciences Dhar El Mahraz (FSDM), Sidi Mohamed Ben Abdellah University (USMBA), Fez, Morocco

[2]Departement of Computer Science, Intelligent Processing and Security of Systems (IPSS) Team, Faculty of Sciences, Mohammed V University in Rabat, Morocco

[3]Department of Health Technologies Engineering, Research Group in Biomedical Engineering and Pharmaceutical Sciences, ENSAM, Mohammed V University in Rabat, Morocco

[4]LIS Lab, Sciences Faculty of Ain-Chock, Hassan II University, Casablanca, Morocco

mohamed.elghmary@usmba.ac.ma

ABSTRACT

This paper addresses the issue of efficient resource allocation in a Mobile Edge Computing (MEC) system, taking into account the trade-off between energy consumption and operation latency. The increasing deployment of connected devices and data-intensive services in the Internet of Things (IoT) poses significant challenges in terms of managing computational resources. In this study, we propose a MEC system model that considers energy constraints and the need to minimize latency to ensure optimal performance. We formulate the resource allocation problem in terms of a trade-off between energy consumption and latency, and explore solutions based on heuristic task offloading techniques. Our experiments demonstrate that our approach achieves improved latency performance while reducing energy consumption. We also evaluate the impact of various parameters, such as workload and resource availability, on the energy-latency trade-off.

KEYWORDS

mobile edge computing, multitask, processing time, energy, computation offloading, optimization

1 INTRODUCTION

Mobile Edge Computing (MEC) is an approach to distributed computing that aims to bring computation and storage resources in close proximity to the network edge. By locating these resources near devices such as smartphones, sensors, and IoT devices, MEC enables faster response times and increased data transfer capacity. This is particularly beneficial for applications that demand real-time processing or involve the transfer of large volumes of data.

In a context where connected devices and data-intensive applications are experiencing exponential growth, MEC systems offer a promising solution by bringing computation and data storage closer to the network edge. This reduces latency, improves application performance, and provides a better user experience. However, the efficient allocation of resources in these heterogeneous and dynamic environments is

El Ghmary, M., Ziani, S., Chanyour, T., Hmimz, Y. (2023). An Energy and Latency Trade-off for Resources Allocation in a MEC System. *International Journal of Interactive Mobile Technologies (iJIM)*, 17(20), pp. 133–147. https://doi.org/10.3991/ijim.v17i20.40999

Article submitted 2023-05-01. Revision uploaded 2023-08-20. Final acceptance 2023-08-26.

a complex challenge. This paper proposes an approach that considers the trade-off between energy consumption and latency introduced during resource allocation. The objective is to optimize this allocation to minimize energy consumption while ensuring acceptable latency times. The results of this study can provide valuable guidelines for the design and optimization of MEC systems, contributing to better resource utilization and enhanced performance in next-generation networks.

There is a significant amount of research that has been conducted in the area of MEC. The authors in [1] propose an algorithm based on a linear programming approach to simultaneously optimize energy consumption and latency in MEC systems. The algorithm aims to optimally allocate tasks to MEC servers, taking into account capacity constraints and latency requirements. The authors in [2][3] developed an optimization algorithm based on a heuristic approach to minimize energy consumption in MEC systems. The algorithm considers the energy characteristics of edge resources such as MEC servers, antennas, and IoT devices to optimize their utilization and reduce overall energy consumption. The authors in [4] proposed a latency prediction method based on machine learning for MEC resource allocation. By using historical data on workload and network performance, the predictive model can estimate the expected latency for different resource allocation configurations, enabling informed decision-making. The authors in [5][6] proposed an adaptive energy management approach for MEC servers, focusing on optimizing energy consumption based on workload variations. The algorithm dynamically adjusts energy management parameters, such as processor frequency and sleep mode, according to processing demand and latency constraints. The authors in [7][8] proposed an energy-efficient caching mechanism to reduce latency and energy consumption in MEC systems. The mechanism leverages spatial and temporal locality characteristics of data requests to determine which content should be cached at the MEC servers, minimizing response times and bandwidth usage. The authors in [9] have explored different architectural approaches for implementing MEC systems, including centralized, decentralized, and hybrid architectures. The authors in [10] have studied how to effectively allocate resources, such as computation, storage, and bandwidth, in MEC systems to optimize performance and reduce costs. The authors in [8] have investigated how to ensure that MEC systems can deliver the necessary level of Quality-of-Service (QoS) to applications, including low latency and high reliability. The authors in [11] have examined how to protect MEC systems from various types of attacks and how to ensure the privacy of user data. The authors in [12] have studied how MEC can be used to improve the performance of specific applications, such as augmented and virtual reality, gaming, telemedicine, and autonomous vehicles. The authors in [13] have investigated various networking and communication protocols and technologies that can be used in MEC systems to improve performance and reliability. Overall, the field of MEC is a rapidly evolving area of research that has the potential to have a significant impact on a wide range of applications and industries.

We live in an era where everything is connected; IoTdevices have invaded the world of computing. Due to their limited capacities, the needs in terms of storage, communication, and especially computing resources have increased. The traditional solution consists of adopting Cloud Computing in order to relieve this equipment of heavy calculations. But real-time data processing has become more important. This is where the role of edge computing lies, bringing compute-intensive processing closer to the mobile device. Thus, several restrictions have been overcome. This feature of edge computing is called calculation offloading.

The MEC paradigm aims to extend the cloud resources to the network edge [14][15]. MEC is a set of small servers located at access points or base stations [16].

In addition, recent works enrich the MEC with a lot of features and functionalities, such as mobile web browser acceleration [17], healthcare [18], connected vehicles [19][20], and video streaming and analysis [21]. In recent times, several works have studied computation offloading for the MEC environment [22][23][24]. Inspired by previous work on computational offloading and mobile cloud computing [25][26], we distinguish two categories of computational offloading applications. Computational offloading and WPT to the edge servers reduce the energy consumption of smart devices [27][28]. Efficient computation offloading in MEC has become an effective technique to bring down the latency constraints that are usually associated with geo-dispersed cloud computing [29][30]. In order to reduce transmission latency, a MEC server is placed near the terminals [31][32]. Edge computing can effectively improve the security and reliability of cloud systems [33][34][35].

In this paper, we focus on the computational offloading decision process for intelligent multitasking optimization in an MEC system. The latter is composed of an Edge server and a Smart Device Mobile (SMD) loaded with a list of offloadable tasks. In other words, our objective is to explore new strategies for the decision of effective offloading in order to improve the performance of mobile devices and applications from the point of view of calculation times as well as energy consumption, while guaranteeing the constraints of application execution latency and the available amount of local energy (the critical battery of mobile devices).

This paper is structured as follows: Section 2 presents the methodology. Section 3 shows the results and discussion while Section 4 concludes the paper.

2 METHODOLOGY

Figure 1 illustrates our hypothetical scenario of a single smart mobile device with limited energy and a list of offloadable tasks $\tau = (\tau_1, \tau_2, \ldots, \tau_N)$ that can be offloaded. Additionally, there is a time limit T_i^d set by the application that must be followed for the execution of the complete list. We formulate the appropriate optimization problem as a result, and we provide a heuristic solution. Additionally, the SMD has a finite amount of energy (E^{max}) to complete this set of tasks.

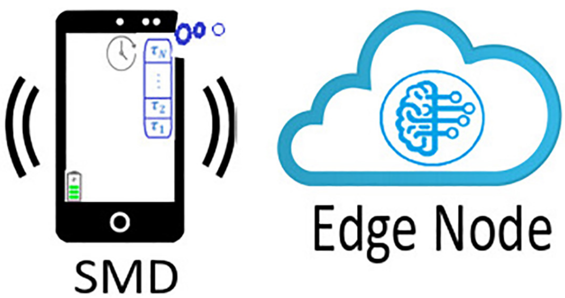

Fig. 1. System process

Depending on the available computation and radio resources, there are three types of offloading decisions:

- **Local execution** is carried out locally by the mobile device itself.
- **Complete offloading** consists of sending the entire set of tasks to the MEC server.
- **Partial offloading** only tasks involves heavy calculations and they are sent to the MEC server.

Execution of all these tasks must occur within the time specified by the application.

The task is also defined by the following four parameters: The quantity of computing required to complete this task's processing is referred to as the workload λ_i[cycles], d_i[bits] the volume of program code and input parameters that must be transferred from the SMD to the edge server, T_d^i reference to the maximum latency required for this task, and x_i the execution nature decision for a task τ_i either by SMD or by offloading to the edge server. If $x_i = 1$ then task τ_i execution is performed locally by the mobile device itself. If $x_i = 0$ then consists of sending the task τ_i to the MEC server. All used formulas and expressions are presented in Table 1.

Table 1. Formulas

Formula	Signification
$t_{li} = \frac{\lambda_i}{f_L}$	Execution time of task τ_i locally by the SMD itself, where f_L refers to the frequency of the SMD.
$t_{ci} = \frac{\lambda_i}{r}$	Time to transmit the task τ_i to the edge server, where r is the transmission rate (bits/s) in line with Shannon equation.
$t_{si} = \frac{\lambda_i}{f_S}$	Time to execute the task τ_i at the edge server, where f_s is the execution frequency of the edge.
$el = (el1, el2, \ldots, elN) = klf_l^2\Lambda$	Vector containing the energy consumption of task τ_i locally by the SMD itself, where kl is the energy efficiency coefficient.
$e_{ci} = \frac{p_t d_i}{r}$	Energy consumption of the communication process of task τ_i where p_t signifies the transmitting power.
$e_{si} = K_S f_S^2 \lambda_i$	Energy consumption at the edge server of task τ_i, where K_s is the energy efficiency coefficient.
$X = (x_1, x_2, \ldots x_N)$	Vector containing the x_i of N tasks.
$X_0 = (x_1, x_2, \ldots x_N)$	Vector containing the x_i of N tasks where $x_i = 0$.
$U = (1, 1, \ldots, 1)$	Vector filled with ones.
$X_1 = U - X$	Vector containing the x_i of N tasks where $x_i = 1$.
$\Lambda = (\lambda_1, \lambda_2, \ldots \lambda_N)$	Vector filled with λ_i.
$\Lambda_0 = U\,\Lambda^T$	Scalar product between the two vectors U and transposed of Λ.
$\Lambda_1 = X\,\Lambda^T$	Scalar product between the two vectors X and transposed of Λ.
$D = (d_1, d_2, \ldots, d_N)$	Vector filled with d_i.
$D_0 = UD^T$	Scalar product between the two vectors U and transposed of D.
$D_1 = XD^T$	Scalar product between the two vectors X and transposed of D.
$t_l = (t_{l1}, t_{l2}, \ldots, t_{lN}) = \frac{1}{f_L}\Lambda$	Vector containing the t_{li} of N tasks

(Continued)

Table 1. Formulas *(Continued)*

Formula	Signification
$T_L = X_l t_l^T$	Scalar product between the two vectors X and transposed of t_l.
$t_c = (t_{c1}, t_{c2}, \ldots, t_{cN}) = \frac{1}{r} D$	Vector containing the t_{ci} of N tasks
$T_c = X_0 t_c^T$	Scalar product between the two vectors X_0 and transposed of t_c.
$t_s = (t_{s1}, t_{s2}, \ldots, t_{sN}) = \frac{1}{f_S} \Lambda$	Vector containing the t_{si} of N tasks
$T_s = X_0 t_s^T$	Scalar product between the two vectors X_0 and transposed of t_s.
$e_l = (e_{l1}, e_{l2}, \ldots, e_{lN}) = k_l f_l^2 \Lambda$	Vector containing the e_{li} of N tasks
$E_L = X_l e_l^T = k_l f_l^2 X_l \Lambda^T$	Scalar product between the two vectors X_l and transposed of e_l.
$e_c = (e_{c1}, e_{c2}, \ldots, e_{cN}) = \frac{p_t}{r} D$	Vector containing the e_{ci} of N tasks
$E_C = X_O e_C^T = p_t X_l D^T$	Scalar product between the two vectors X_0 and transposed of e_c.
$e_S = (e_{s1}, e_{s2}, \ldots, e_{sN}) = k_S f_S^2 \Lambda$	Vector containing the e_{si} of N tasks
$E_S = X_0 e_S^T = k_S f_S^2 X_0 \Lambda^T$	Scalar product between the two vectors X_0 and transposed of e_s.

2.1 Problem formulation

To formulate our optimization problem, our aim is to minimize the trade-off between energy consumption and processing time during the offloading process while also ensuring that the minimum amount of energy (the battery lifetime) and processing time are both met. We first calculate the energy consumption and the total processing time of the SMD.

$$T(X, f_S, f_L) = \frac{\Lambda - \Lambda_1}{f_L} + \frac{D_1}{r} + \frac{\Lambda_1}{f_s} \tag{1}$$

$$E\,(X, f_S, f_L) = \Lambda_1 (k_S f_S^2 - k_L f_L^2) + \frac{p_t D_1}{r} + \Lambda k_L f_L^2 \tag{2}$$

Then we utilize to the two objective functions the weights α1 and α2 such that $\alpha_1 + \alpha_2 = 1$and we use E^{max} and $max(T_d^i)$ to normalize the energy and the global objective function processing time and remove their units. The resulting problem is formulated as follows:

$$C_E^T(X, f_S, f_L) = \frac{\alpha_1}{\max(T_d^i)}\ T(X, f_S, f_L) + \frac{\alpha_2}{E^{max}} E(X, f_S, f_L) \tag{3}$$

$$\mathcal{P}_1:\begin{cases}\min\left\{C_E^T(\mathbb{X},f_S,f_L)\right\}\\ s.t.\\ \quad (C_{1.1})\,x_i\in\{0;1\};i\in[\![1;N]\!]\\ \quad (C_{1.2})\,F_L^{min}\le f_L\le F_L^{max}\\ \quad (C_{1.3})\,0<f_s\le F_S\\ \quad (C_{1.4})\,T_L^i\le T_d^i\\ \quad (C_{1.5})\,T_C^i+T_L^i\le T_d^i\\ \quad (C_{1.6})\,\mathrm{E}_L+E_C\le E^{max}\end{cases}$$

In this problem, each potential offloading decision solution must adhere to the constraints ($C_{1,1}$) to ($C_{1,6}$) mentioned above.

We can observe that problem P1 is a multivariate optimization problem. To solve it, we will break it down into the following two sub-problems:

$$\mathcal{P}_{2.1}(\mathbb{X}):\begin{cases}\min\left\{C_{1E}^{\,T}(f_S)\right\}\\ s.t.\\ \quad (C_{2.1.1})\;F_L^{\min}\le f_L\le F_L^{\max}\\ \quad (C_{2.2.1})\;\max_i\left(\dfrac{\Lambda_i-\Lambda_i^1}{T_d^i}\right)\le f_L\le\sqrt{\dfrac{E^{\max}-\dfrac{p_tD_1}{r}}{k_L(\Lambda_N-\Lambda_N^1)}}\end{cases}$$

And

$$\mathcal{P}_{2.2}(\mathbb{X}):\begin{cases}\min\left\{C_{2E}^{\,T}(f_S)\right\}\\ s.t.\\ \quad (C_{2.2.1})\max_i\left(\dfrac{\Lambda_1^i}{T_d^i-\dfrac{D_1}{r}}\right)\le f_S\le F_S\end{cases}$$

With:

$$C_{1E}^{\,T}(f_L)=\Lambda_0\left(\frac{\alpha_1}{T_d^if_L}+\frac{\alpha_2k_Lf_L^2}{E^{max}}\right)\tag{4}$$

And

$$C_{2E}^{\,T}(f_S)=\Lambda_1\left(\frac{\alpha_1}{T_d^if_S}+\frac{\alpha_2k_Sf_S^2}{E^{max}}\right)+\frac{D_1}{r}\left(\frac{\alpha_1}{T_d^i}+\frac{\alpha_2pt}{E^{max}}\right)\tag{5}$$

According to equations 4 and 5 obtained above in the two sub-problems **P2.1** and **P2.2**, and with a given offloading decision vector X, the first function is written in the following form

$C_{1E}^{\,T}(f_L)=\dfrac{A_1}{f_L}{}_L+A_2f_L^2$

and the second function is written in the following form

$C_{2E}^{\,T}(f_S)=\dfrac{A_3}{f_S}+A_4f_S^2+A_5$ with $A_1,A_2,A_3,A_4,A_5,$ are variables independent constants.

The algorithm 1 gives the two optimal values, the frequency f_L^* of the SMD, as well as the frequency f_S^* of the Edge server.

Algorithm 1 local optimum frequency and edge optimum frequency for a given $\mathbb{X}$

Require: The offloading policy $\mathbb{X}$.
Ensure: f_S^* , f_L^*

Determinate $\mathbb{X}_1$;
if $\mathbb{X} = \mathbb{X}_1$ **then**
  $f_L^* = 0$;
else
  if $\max_i\left(\frac{\Lambda_i - \Lambda_i^1}{T_d^i}\right) < F_L^{max}$ **or** $F_L^{min} < \sqrt{\frac{E^{max} - \frac{p_t D_1}{r}}{k_L(\Lambda_N - \Lambda_N^1)}}$ **then**
    $f_L^* = \emptyset$;
  else
    if $\min\left\{F_L^{max}, \sqrt{\frac{E^{max} - \frac{p_t D_1}{r}}{k_L(\Lambda_N - \Lambda_N^1)}}\right\} \leq \sqrt[3]{\frac{\alpha_1 E^{max}}{2\alpha_2 k_L T_d^i}}$ **then**
      $f_L^* = \min\left\{F_L^{max}, \sqrt{\frac{E^{max} - \frac{p_t D_1}{r}}{k_L(\Lambda_N - \Lambda_N^1)}}\right\}$;
    else
      $A = \max\left\{F_L^{min}, \max_i\left(\frac{\Lambda_i - \Lambda_i^1}{T_d^i}\right)\right\}$
      $B = \sqrt[3]{\frac{\alpha_1 E^{max}}{2\alpha_2 k_L T_d^i}}$
      $f_L^* = \max\{A, B\}$;
    end if
  end if
end if
Determinate $\mathbb{X}_0$;
if $\mathbb{X} = \mathbb{X}_0$ **then**
  $f_S^* = 0$;
else
  if $\max_i\left(\frac{\Lambda_1^i}{T_d^i - \frac{D_1}{r}}\right) > F_S$ **then**
    $f_S^* = \emptyset$;
  else
    if $F_S \leq \sqrt[3]{\frac{\alpha_1 E^{max}}{2\alpha_2 k_S T_d^i}}$ **then**
      $f_S^* = F_S$;
    else
      $f_S^* = \max\left\{\max_i\left(\frac{\Lambda_i - \Lambda_i^1}{T_d^i}\right), \sqrt[3]{\frac{\alpha_1 E^{max}}{2\alpha_2 k_S T_d^i}}\right\}$;
    end if
  end if
end if
return (f_S^*, f_L^*)

2.2 Method based on brute force search

Brute Force Search (BFS) is an analysis method that involves testing all possibilities to find a solution to a problem. This method is often used in the fields of cryptography and computer security.

Algorithm 2 BFS Offloading.

Require: The list τ of N tasks, $\mathbb{X}$.
Ensure: $\mathbb{X}^*$

$C_{min}{}_E^T \leftarrow 1$;
$i \leftarrow 0$;
repeat
  Use the N bits representation of integer i to build the policy $\mathbb{X}$;
  Call equation 3 to calculate $C_{new}{}_E^T$ using τ and $\mathbb{X}$;
  if $C_{new}{}_E^T \neq C_{min}{}_E^T$ **then**
    $C_{min}{}_E^T \leftarrow C_{new}{}_E^T$
    $\mathbb{X}^* \leftarrow \mathbb{X}$
  end if
  $i \leftarrow i + 1$
until $i = 2^N$
return $\mathbb{X}^*$

By using BFS, all possible combinations are tested until the optimal solution is found. This method can be very effective for problems with a small number of possibilities. However, for more complex problems, the time required to test all possibilities can be very long or even impossible in practice.

2.3 Heuristic offloading method based on simulated annealing

Simulated Annealing(SA) [36–38] is a stochastic optimization algorithm that is often used to find a near-global minimum or maximum of a complex function, particularly in the context of combinatorial optimization problems. It is a probabilistic technique that is based on the physical process of annealing, in which a material is heated and then slowly cooled to increase its structural order.

In SA, the optimization problem is viewed as a system with a certain energy level, and the goal is to find the state with the lowest energy (i.e., the global minimum). The algorithm starts with an initial solution and then iteratively generates new solutions by making random changes to the current solution. These new solutions are evaluated and accepted or rejected based on a probability function that depends on the difference in energy between the current and the new solution, as well as a temperature parameter that is gradually decreased over time.

At the beginning of the algorithm, the temperature is set high, allowing the algorithm to make large jumps in the search space. As the temperature decreases, the algorithm becomes more selective, only accepting moves that decrease the energy of the system. This allows the algorithm to escape local minima and search for the global minimum.

Simulated Annealing is a powerful optimization technique that can be used to solve a wide range of problems, including problems with many local minima or maxima, problems with discontinuous or non-differentiable objective functions, and problems where the search space is large and complex. For our solution, we denote HOMBSA.

Algorithm 3 HOMBSA

Require: The set τ of N sub-tasks and $Temp_0$.

Initialize: $\mathbb{X}$;
Calculate C_E^T using τ and $\mathbb{X}$;
for $k = 0$ to kmax **do**
  $Temp \leftarrow Temp_0(1 - x\frac{k}{kmax})$;
  Pick a random neighbour, $\mathbb{X}_{new} \leftarrow$ neighbour($\mathbb{X}$);
  Calculate ${C_{new}}_E^T$ using τ and $\mathbb{X}_{new}$;
  if ${C_{new}}_E^T \neq \infty$ **then**
    if $P(C_E^T, {C_{new}}_E^T, Temp) \geq$random(0,1) **then**
      $\mathbb{X} \leftarrow \mathbb{X}_{new}$;
      $C_E^T \leftarrow {C_{new}}_E^T$;
    end if
  end if
end for
$\mathbb{X}^* \leftarrow \mathbb{X}$;

In this algorithm, a random number in the range of [0, 1] is generated using the function random (0, 1). The neighbor (*X*) function produces a set of decision states that are close to the input *X*. The acceptance probability, denoted as $P(C_E^T, {C_{new}}_E^T, Temp)$, determines whether a new state, X_{new}, is accepted based on the processing times C_E^T and C_{newE}^T of the current state *X* and the new state X_{new}, respectively. Additionally, the acceptance probability also depends on a global parameter called *Temp*, which represents the temperature and varies across iterations.

2.4 Simulation setup

The results obtained were averaged over 100-time executions. The programs were developed using C++ on a PC with 8GB of RAM and an Intel Core processor i7 – 2620M 2.7GHz. The simulation experiments were conducted with the basic parameters listed in Table 2.

Table 2. Simulations' parameters

Parameter	Value	Parameter	Value	Parameter	Value
f^{min}	1MHz	K_s	10^{-29}	r	100 KB/s
f^{max}	60MHz	K_L	10^{-26}	E^{max}	$[0.6, 0.8] \Lambda K_L (F^{max})^2$
F_s	6GHz	T_d	[0.5, 2]	d_i	[30, 300]KB
p_t	0.1Watt	T_{emp0}	100	λ	[60, 600]$MCycles$
α	0.5	CF	0.85	ϵ	0.3

3 RESULTS AND DISCUSSION

The energy consumption of an application is determined by the quantity of energy required to run its services. The difference between the energy stored in the battery and the energy utilized by the primary components has grown with successive iterations. The consumption of energy is a result of the software's demand for ever-increasing quantities of its components, resulting in the consumption of energy.

It is consequently essential to measure and comprehend the energy consumption of the SMD. In order to enhance the user experience, this will motivate researchers to develop various strategies for reducing energy use. Its energy consumption has a substantial effect on its autonomy but a small effect on its total energy consumption. This work focuses on the relationship between energy usage and execution time by constructing a mathematical model that satisfies all industrial requirements. The acquired simulation results support the efficiency of the proposed model, with the expected outcome being that the time-energy tradeoff in the BFS scenario is optimal, as depicted in Figure 4, and significantly lower than in the ALL LOCAL and ALL OFFLOAD situations, as depicted in Figures 2 and 3, respectively. Optimizing this trade-off is thus of crucial importance. In order to comprehend and debug the energy consumption of mobile applications, it is important to be able to measure the energy consumption of mobile devices.

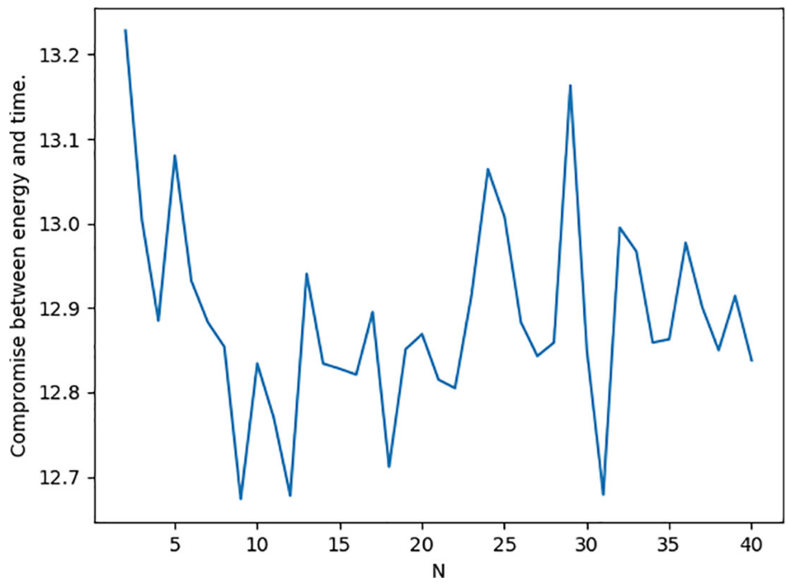

Fig. 2. ALL OFFLOAD: Energy- time trade-off

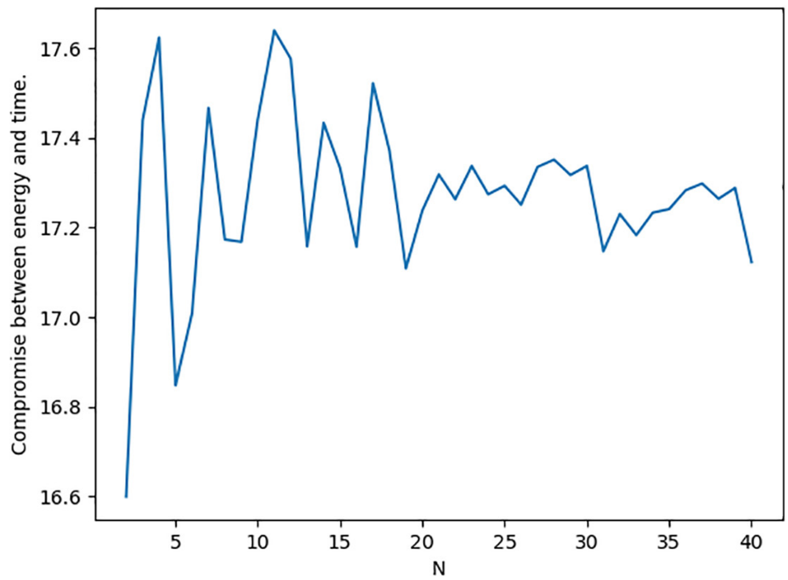

Fig. 3. ALL LOCAL: Energy-time trade-off

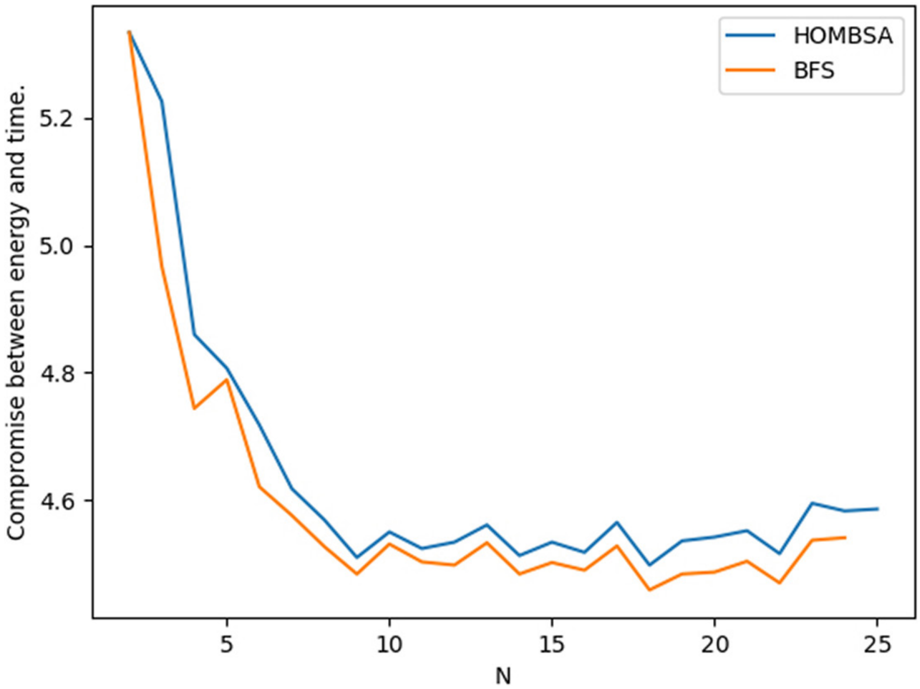

Fig. 4. Trade-off energy-time

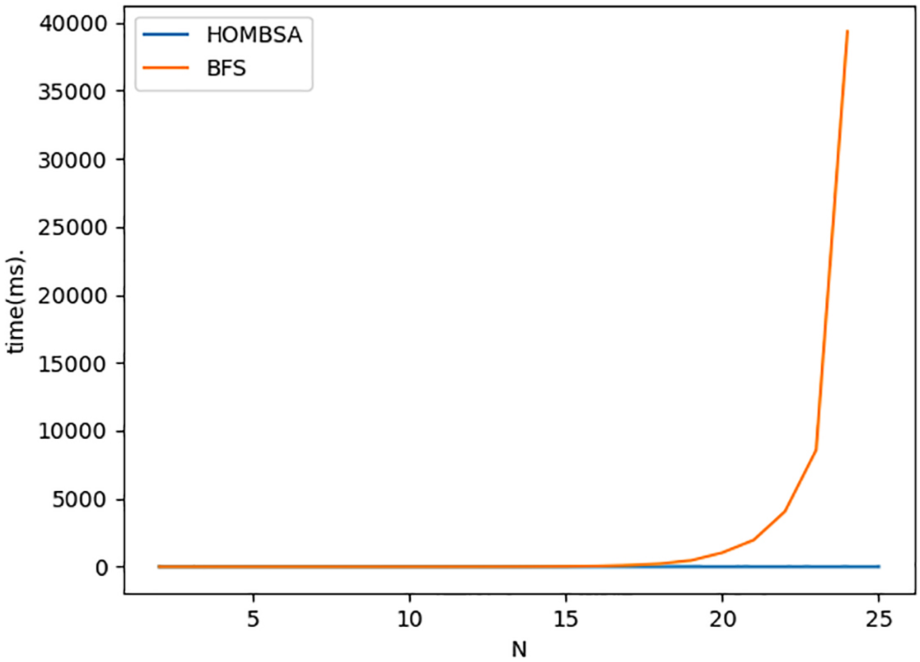

Fig. 5. Execution time

The energy cost required for hardware components to perform their functions defines energy consumption. This activity is driven by the software operations associated with the user action. In fact, the software generates a certain amount of work in a specific amount of time, which causes hardware actions.

An experiment involved varying the parameter for the number of tasks, ranging from 2 to 24. The results of this experiment are presented in two figures. Figure 4 illustrates the outcomes obtained for both the exhaustive search solution and the simulated annealing-based solution (HOMBSA). It demonstrates a minimal gap between the curves, representing the average processing time of the processed tasks. As a result, the disparity between the exhaustive search (BFS) approach and the HOMBSA method fluctuates between 0.00% and 0.25%.

The average execution time in milliseconds for obtaining offloading decisions is illustrated in Figure 5, showcasing both solutions, while the number of tasks N is varied between 2 and 24. The figure distinctly portrays the exponential increase in execution time in relation to N for the exhaustive search method. In contrast, the HOMBSA solution maintains a consistent execution time, where it only reaches 1.00 ms when N = 24.

4 CONCLUSION

In this study, we have examined the trade-off between energy consumption and latency in resource allocation within a MEC system. The increasing deployment of connected devices and data-intensive services in the IoT has presented significant challenges in managing computational resources effectively. Accordingly, we proposed a MEC system model that takes into account energy constraints and the need to minimize latency in order to achieve optimal performance. By formulating the resource allocation problem as a trade-off between energy consumption and latency, we explored an exact solution based on the BSF methods and an approximate solution based on heuristic task offloading techniques.

Our study highlights the importance of considering the energy-latency trade-off in resource allocation for MEC systems. The obtained results provide practical insights for the design and efficient management of such systems, opening perspectives for advanced applications based on the Internet of Things.

5 REFERENCES

[1] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2017. https://doi.org/10.1109/JIOT.2017.2786343

[2] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2450–2463, 2018. https://doi.org/10.1109/TCOMM.2018.2881725

[3] S. Maftah, M. El Ghmary, H. El Bouabidi, M. Amnai, and A. Ouacha, "Optimal task processing and energy consumption using intelligent offloading in mobile edge computing," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 20, 2022. https://doi.org/10.3991/ijim.v16i20.34373

[4] G. Liu, Y. Xu, Z. He, Y. Rao, J. Xia, and L. Fan, "Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles," *IEEE Access*, vol. 7, pp. 114487–114495, 2019. https://doi.org/10.1109/ACCESS.2019.2935463

[5] A. Ouacha and M. El Ghmary, "Virtual machine migration in mec based artificial intelligence technique," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, p. 244, 2021. https://doi.org/10.11591/ijai.v10.i1.pp244-252

[6] Y. Zhang, B. Zhang, and S. Zhang, "An adaptive energy-aware relay mechanism for IEEE 802.15.6 wireless body area networks," *Wireless Personal Communications*, vol. 115, pp. 2363–2389, 2020. https://doi.org/10.1007/s11277-020-07686-4

[7] J. Albert Mayan, S. V. Manikanthan, A. Hussain, S. Nithyaselvakumari, and A. Vinnarasi, "Clustering technique for mobile edge computing to detect clumps in transportation-related problems," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 4, pp. 47–63, 2023. https://doi.org/10.3991/ijim.v17i04.37801

[8] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1517–1530, 2021. https://doi.org/10.1109/JIOT.2021.3091142

[9] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D.J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018. https://doi.org/10.1109/JSAC.2018.2874142

[10] L. Li, Q. Cheng, X. Tang, T. Bai, W. Chen, Z. Ding, and Z. Han, "Resource allocation for noma-mec systems in ultra-dense networks: A learning aided mean-field game approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1487–1500, 2020. https://doi.org/10.1109/TWC.2020.3033843

[11] S. R. Chaudhry, A. Palade, A. Kazmi, and S. Clarke, "Improved qos at the edge using serverless computing to deploy virtual network functions," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10673–10683, 2020. https://doi.org/10.1109/JIOT.2020.3011057

[12] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, 2018. https://doi.org/10.1109/MWC.2018.1700291

[13] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile internet and its applications in 5g era: A comprehensive review," *International Journal of Communication Systems*, vol. 32, no. 14, p. e3981, 2019. https://doi.org/10.1002/dac.3981

[14] G. Zheng, C. Xu, M. Wen, and X. Zhao, "Service caching based aerial cooperative computing and resource allocation in multi-uav enabled mec systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10934–10947, 2022. https://doi.org/10.1109/TVT.2022.3183577

[15] M. Satyanarayanan, "Edge computing for situational awareness," in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2017, pp. 1–6. https://doi.org/10.1109/LANMAN.2017.7972129

[16] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017. https://doi.org/10.1109/MC.2017.9

[17] G. A. Lewis, *Software architecture strategies for cyber-foraging systems*. PhD thesis, Vrije Universiteit Amsterdam, 2016.

[18] N. Takahashi, H. Tanaka, and R. Kawamura, "Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated web browsing," in *3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, 2015, pp. 233–234. https://doi.org/10.1109/MobileCloud.2015.23

[19] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, 2015, pp. 1–5. https://doi.org/10.1109/Ubi-HealthTech.2015.7203325

[20] M. Li, P. Si, and Y. Zhang, "Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9073–9086, 2018. https://doi.org/10.1109/TVT.2018.2865211

[21] O. Zhdanenko, J. Liu, R. Torre, S. Mudriievskiy, H. Salah, G. T. Nguyen, and H. F. Fitzek, "Demonstration of mobile edge cloud for 5g connected cars," in *16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2019, pp. 1–2. https://doi.org/10.1109/CCNC.2019.8651783

[22] L. Van Ma, V. Q. Nguyen, J. Park, and J. Kim, "Nfv-based mobile edge computing for lowering latency of 4k video streaming," in *Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 670–673. https://doi.org/10.1109/ICUFN.2018.8436725

[23] H. Yao, C. Bai, M. Xiong, D. Zeng, and Z. Fu, "Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 16, p. e3975, 2017. https://doi.org/10.1002/cpe.3975

[24] L. Ma, J. Wu, and L. Chen, "Dota: Delay bounded optimal cloudlet deployment and user association in wmans," in *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 196–203. https://doi.org/10.1109/CCGRID.2017.34

[25] D. G. Roy, D. De, A. Mukherjee, and R. Buyya, "Application-aware cloudlet selection for computation offloading in multi-cloudlet environment," *The Journal of Supercomputing*, vol. 73, no. 4, pp. 1672–1690, 2017. https://doi.org/10.1007/s11227-016-1872-y

[26] G. A. Lewis, Software architecture strategies for cyber-foraging systems. PhD thesis, Ph. D. dissertation, Vrije Universiteit Amsterdam, 2016.

[27] Y. Gao, W. Hu, K. Ha, B. Amos, P. Pillai, and M. Satyanarayanan, "Are cloudlets necessary?" School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-15–139, p. 8, 2015.

[28] M. El Ghmary, M. O. Cherkaoui Malki, Y. Hmimz, and T. Chanyour, "Energy and computational resources optimization in a mobile edge computing node," in *9th International Symposium on Signal, Image, Video and Communications (ISIVC)*, 2018, pp. 323–328. https://doi.org/10.1109/ISIVC.2018.8709200

[29] M. El Ghmary, Y. Hmimz, T. Chanyour, and M. O. C. Malki, "Energy and processing time efficiency for an optimal offloading in a mobile edge computing node," *International Journal of Communication Networks and Information Security*, vol. 12, no. 3, pp. 389–393, 2020. https://doi.org/10.17762/ijcnis.v12i3.4750

[30] K. Peng, V. Leung, X. Xu, L. Zheng, J. Wang, and Q. Huang, "A survey on mobile edge computing: Focusing on service adoption and provision," *Wireless Communications and Mobile Computing*, vol. 2018, 2018. https://doi.org/10.1155/2018/8267838

[31] O. Egwuche, M. Ganiyu, and M. Ibiyomi, "A survey of mobile edge computing in developing countries: Challenges and prospects," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2034, p. 012004, 2021. https://doi.org/10.1088/1742-6596/2034/1/012004

[32] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017. https://doi.org/10.1109/COMST.2017.2682318

[33] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017. https://doi.org/10.1016/j.future.2016.09.015

[34] Y. Jararweh, "Enabling efficient and secure energy cloud using edge computing and 5g," *Journal of Parallel and Distributed Computing*, vol. 145, pp. 42–49, 2020. https://doi.org/10.1016/j.jpdc.2020.06.014

[35] Y. Hmimz, M. El Ghmary, T. Chanyour, and M. O. C Malki, "Computation offloading to a mobile edge computing server with delay and energy constraints," in *International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1–6. https://doi.org/10.1109/WITS.2019.8723733

[36] M. Yaakob, A. A. Salameh, O. Mohamed, and M. A. H. Ibrahim, "Enabling edge computing in 5G for mobile augmented reality," *Int J Interact Mob Technol*, vol. 16, no. 14, pp. 23–30, 2023. https://doi.org/10.3991/ijim.v16i14.32623

[37] I. Al Ridhawi, M. Aloqaily, A. Boukerche, and Y. Jararweh, "Enabling intelligent iocv services at the edge for 5g networks and beyond," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5190–5200, 2021. https://doi.org/10.1109/TITS.2021.3053095

[38] Z. Fan, H. Shen, Y. Wu, and Y. Li, "Simulated-annealing load balancing for resource allocation in cloud environments," in *International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2013, pp. 1–6. https://doi.org/10.1109/PDCAT.2013.7

6 AUTHORS

Mohamed El Ghmary is a Professor of Computer Science at the Faculty of Sciences Dhar El Mahraz (FSDM), Sidi Mohamed Ben Abdellah University (USMBA), Fez, Morocco. He is an associate member of the Intelligent Processing and Security of Systems (IPSS) team of Computer Science Department at the Faculty of Sciences, Mohammed V University (UM5), Rabat, Morocco. He is also an associate member of Research Laboratory in Computer Science and Telecommunications (LARIT), Team Networks and Telecommunications, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests includes mobile edge computing (MEC), cloud computing, machine learning, deep learning, and intelligent systems and optimization (E-mail: mohamed.elghmary@usmba.ac.ma).

Said Ziani is a Professor in the Department of Electrical Engineering at Hassan II University (ESTC) in Casablanca, Morocco. He is also associated with Health Technologies Engineering Department, Research Group in Biomedical Engineering and Pharmaceutical Sciences, ENSAM, Mohammed V University, Agdal, Morocco. His research interests include digital design, industrial applications, industrial electronics, industrial informatics, power electronics, motor drives, renewable energy, FPGA and DSP applications, embedded systems, adaptive control, neural network control, automatic robot control, motion control, and artificial intelligence. He is a senior Member of IEEE with Member/Customer Number: 98711129 (E-mail: said.ziani@univh2c.ma).

Tarik Chanyour is an Assistant Professor in the Sciences Faculty of Ain Chock, Hassan II University of Casablanca, Morocco. He is a member of Computer Science and Systems Laboratory within the Computer Science and Mathematics department. He holds his PhD in Computer Science, a Specialized Master in Computer Science Engineering from the Sciences Faculty, USMBA, Fez, Morocco. His research focuses include mobile edge computing, cloud computing, mobile computing, Internet of Things, blockchain and optimization (E-mail: tarik.chanyour@univh2c.ma).

Youssef Hmimz received PhD degree from USMBA, Fez, Morocco in 2022. He received the Master's degrees in computer engineering and networks from USMBA in 2007 and Master's degrees in algebra - number theory and applications to information sciences from the same University in 2009. He is a member of the LPAIS Laboratory at the same University. His current research interests include multi-objective optimization, mobile computing, and Markov decision processes (E-mail: youssef.hmimz@usmba.ac.ma).