Efficient Power Management in Mobile Computing with Edge Server Offloading Using Multi-Objective Optimization

P.Anusha[1,*], R.V.Siva Balan[1]

[1]Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Thucklay, Kanyakumari (Dt), Tamil Nadu, India-629 180

Abstract

INTRODUCTION: The internet of mobile things is subjected to execute on data centers such as cloudlet, cloud servers and also on devices; it solves the problem of multi-objective optimization and tries to discover active scheduling with low energy consumption, execution time and cost.
OBJECTIVES: To alleviate the conflicts between the support constraint of 'smart phones and customers' requests of diminishing idleness as well as extending battery life, it spikes a well-known wave of offloading portable application for execution to brought together server farms, for example, haze hubs and cloud workers.
METHODS: The test to develop the methodology for mobile phones, with enhanced IoT execution in cloud-edge registering. Then, to assess the feasibility of our proposed process, tests and simulations are carried out.
RESULTS: The simulator is used to test the algorithm, and the outcomes show that our calculations can lesser over 18% energy utilization.
CONCLUSION: The optimization approaches using PSO and GA based on simulation data, with the standard genetic algorithm providing the highest overall value for mission offloading in fog nodes using multi-objectives. With the assumption of various workflow models as single and multi-objective in data centers as cloud servers, fog nodes, and within computers, we extracted the analytic results of energy usage, delay efficiency, and cost. Then formulated the multi-objective problem with different constraints and solved it using various scheduling algorithms based on the obtained data.

Keywords: Cloud-edge computing, Cloudlets, Fog nodes, Optimization

Received on 01 April 2021, accepted on 07 June 2021, published on 08 July 2021

doi: 10.4108/eai.8-7-2021.170288

*Corresponding author. Email: anushajournals@gmail.com

1. Introduction

(A) Background

Despite better efficiency of mobile devices in fog nodes, cloud servers, inside devices, offloading mobile apps, resource distribution, and optimization of VMs in formalized edge servers presents a major challenge in maximizing execution time, energy usage for mobile devices (MD), and cost estimation. To address this challenge, offloading of mobile devices is to domicile the multi target improvement issue of undertaking transfer both in haze hubs and cloud servers. Certainly, real experiment and simulations are conducted to substantiate the efficiency of the offloading strategies and better resource allocation using suitable scheduling algorithm.

The multi-objective optimization problem is formulated based on the scheduling strategies, with the joint aims of minimizing energy consumption, expense, and execution delay. The formulated problem is addressed using scheduling algorithm methods. The efficiency of offloading methods and best optimization can be proven using comprehensive simulations. Since the cloud sent indirectly from the portable gadgets, transfer the portable devices to the far-off cloud possesses considerable speed of the center organization, causing network clog to a serious degree. The cloud through WAN and the data transfer capacity of offloading the versatile applications is low, which prompts highest inertness as we think about the preparing in edge hubs. Thus, much time is exhausted during the time spent offloading the portable applications to the cloud, causing huge offloading delay, particularly for the information escalated registering errands. The cloudlets associate with mobile phones through LAN, which is described by high BW capacity and low inertness [3]. When compared with processing time of mobile application with the cloud, the pressure of center organization is diminished, subsequently computing utilizing edges reduces offloading latency and makes network more proficient with the goal that it gives a time saving computing paradigm.

(B) Motive

To enhance the pursuance of the portable applications, cloudlets make offloading measures. Portable applications are regularly established as work techniques that include some registering tasks with the restraint as montage, cyber shake, epigenomics, sight and in spiral. Edge computing and mobile in the internet of things are enabled to offload computational tasks to the virtual machine or to the cloud in order to reduce the preparation dormancy of cell phones. Nonetheless, offloading the computing errands nearly enhances the ET, cost and the energy utilization of the mobile gadget, because of the nearest assets of the cloudlet. Therefore, it says tough assignment optimization with the multi- objective parameters of the cell gadgets within the cloud- part computing surroundings. To label the project, two operating strategies are choosed as offloading techniques and scheduling algorithms for cloud-edge computing.

Table 1. Key notation & Description

Symbol	Descriptions
MD	The number of mobile devices
V_{mwf}	The computing task set of the mth workflow
ED_m	The dependency set of the mth workflow
d_m	The input data of the computing task vm,i receives
w_m	The computation workload vm,i
TT	The time consumption in executing the mth Workflow
ET	The energy consumption of the mth workflow.
$E^e(Xm)$.	The energy consumption in executing the mthorkflow
$C^m(Xm)$	The cost consumption for the mth mobile device
P_A	The parameter mentioned during the workflow
C_{ll}	The length of the cloudlet
Max^p	The maximum power consumed by the host
Avg^m	The average allocated MIPS for VM
Sta^p	The total power consumption
Tot^m	The total allocated MIPS for VM
J_i	The total input size of data transmission
J_o	The total output size of data transmission
J_b	The both chromosome job

(C) Contribution

The major contribution of this paper is given as follows

- Time, cost and energy utilization of the mobile gadgets are examined, and also errand offloading measure in cloud-edge (CE) figuring is characterized as a multi-objective and with different enhancement technique.

- Affirm the unique schedules of the simultaneous work processes in CE registering to choose the ideal schedule procedure by utilizing additive weighting.
- Conduct extensive tests and assessments to verify the efficiency and effectiveness of cloud-edge computing in a particular environment.

The remaining section of this paper is coordinated as follows. Section II proposed, System model and problem definition which contains system resource model, energy consumption model, ET model, execution price cost and algorithm design. Section III elaborates on the process of offloading over mobile data cloud-edge computers, examines the technique, and discusses related work. And finally, this paper concludes in section 4 and some future scope also added in this section.

The remaining section of this paper is coordinated as follows. Section II proposed, System model and problem definition which contains system resource model, energy consumption model, ET model, execution price cost and algorithm design. Section III elaborates on the process of offloading over mobile data cloud-edge computers, examines the technique, and discusses related work. And finally, this paper concludes in section 4 and some future scope also added in this section.

2. System Model and Problem Definition

A Cloud-Edge computing architecture approach in this section is designed to calculate the time complexity, cost, and energy consumption of mobile devices. Its goal is to reduce delays, increase network and service delivery performance, and provide a better user experience. Table 1 contains a list of main notations and definitions.

2.1. System Resource Model

$\geq$

In this scheme, imagine a case in which a cloudlet uses ED mobile devices to link to a cloud that is located elsewhere. Using the different forms of workflow settings, each mobile framework is formalized as a workflow. Several multi-constrained programming functions make up a workflow. Let wf (m_{wf}, ED_m, X_m) = {1, 2, . . . ,M} be the mth mobile device, where Vm = { vm,i|1 i ≤ |Vm|} indicate the set of figuring errands in the mth workflow.

$$ED_m = \{(V_{M,p}, V_{M,q}) / V_{M,p}, V_{M,q} \in V_M \wedge p \neq q \tag{1}$$

Equation (1) depicts the relay between the processing errands V_M,p and V_M,q. Let the required multi-obliged information for handling of each registering task be a tuple, signified as (d_m,p,w_m,q), where d_m,p and w_m,q mirror the info information the figuring task V_M,p gets from its designated processing assignments and the preparing separately. pre (V_M,p) addresses the antecedent processing errands of $V_{M,p}$. Just all the registering errands in pre ($V_{M,p}$) finish executions, can $V_{M,p}$ be executed.

Through computation offloading, the computational activities in a workflow may be performed by a portable gadget such as cell phones, a cloudlet, or cloud servers in cloud-edge computing. The mth workflow DAGm's hybrid computing offloading strategies are represented by Ym, a |Vm|- sequence. The computation offloading strategy of the computing task $V_{M,p}$ is represented by the element $Y_{m,p}$, which is measured as

$$Y_{m,p} = \begin{cases} 0, & \text{if } V_{M,p} \text{ is executed in mobile device,} \\ 1, & \text{if } V_{M,p} \text{ is offloaded to the cloudlet,} \\ 2, & \text{if } V_{M,p} \text{ is offloaded to the cloud.} \end{cases} \tag{2}$$

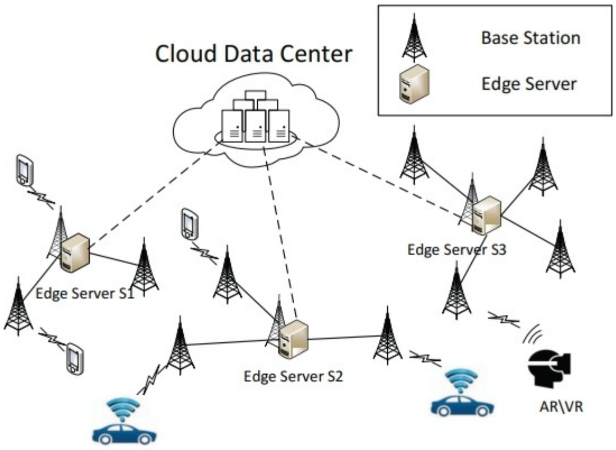

Figure 1. Three level architecture of MEC

Figure (1) shows the three level architecture of mobile edge computing, But this paper mostly interested in the edge server level, which is made up of n (n = 1, 2, 3,...) BS and m (m = 1, 2, 3,...) edge servers. This level's network can be thought of an undirected graph G = (U, Y), wherein U = B s. The set of BS is denoted by B, and the set of prospective venues for edge servers is denoted by s. Taking into account that setting edge workers at BS is the essential suspicion in edge registering, B approaches S here. Accordingly, |U | = |B| = |S| = m. E addresses the arrangement of connections between BS and edge workers. Any of these associations reflects not just the actual link between a BS and an edge worker, but also the distinct interaction that the BS has with the edge worker.

Thusly, the execution season of DAG_m is isolated into three classifications, i.e., the offloading inertness T^L, the figuring time T^e, furthermore, the transmission time T^t.

The offloading latency $T^L(X_{m,i})$ is calculated by

$$T^L(Y_{m,p}) = \begin{cases} 0, & Y_{m,p} = 0, \\ L_{LAN}, & Y_{m,p} = 1, \\ L_{WAN}, & Y_{m,p} = 2. \end{cases} \tag{3}$$

Where L_{LAN} denotes the latency (delay) of local area network, L_{WAN} denotes the latency of wide area network. Thus, the offloading idleness of all the processing errands in the mth work process, i.e., $T^L(Y_{m,p})$ is determined by

$$T^L(Y_{m,p}) = \sum_{Y_{m,p} \in Y_m} T^L(Y_{m,p}) \tag{4}$$

The performance of figuring task and the registering time is dictated by processing errands and registering force of performance stage. For the figuring errands $V_{M,i}$, the computing time $T^e(X_{m,i})$ is calculated by

$$T^e(Y_{m,p}) = \begin{cases} \frac{w_{m,i}}{f_{local}}, & Y_{m,p} = 0, \\ \frac{w_{m,i}}{f_{cl}}, & Y_{m,p} = 1, \\ \frac{w_{m,i}}{f_c}, & Y_{m,p} = 2, \end{cases} \tag{5}$$

Where f_{local} means computing power of mobile gadgets, f_{cl} means computing power of cloudlet, and f_c means cloud separately. Consequently, the processing time for the execution of the mth work process is determined by

$$T^e(Y_{m,p}) = \sum_{Y_{m,p} \in Y_m} T^e(Y_{m,p}) \tag{6}$$

The offloading methodologies of these two registering tasks determine the transmitting time between two figuring projects with dependence. Let A_p (p = 1, 2, 3) be the estimated transmission system of the two figuring undertakings.

$$A_{p=} \begin{cases} \{(0,1),(1,0)\}, & p = 1 \\ \{(0,0),(1,1),(2,2)\} & p = 2 \\ \{(0,2),(2,0),(1,2),(2,1)\} & p = 3 \end{cases} \tag{7}$$

Where ($Y_{m,p}, Y_{m,q}$)$\in A_1$ denotes that data get sent from the MD to the f_{cl} via LAN or vice versa. If ($Y_{m,p}, Y_{m,q}$) $\in A_2$, the information is communicated in a similar processing circumstance. If ($Y_{m,p}, Y_{m,q}$) $\in A_3$, refers to data transmission via WAN between the MD and the cloud or between the cloudlet and the cloud. The among T^t $V_{M,p}$ and $V_{M,q}$ signified as $T^t(Y_{m,p}Y_{m,j})$ is determined by

$$T^t(Y_{m,p}Y_{m,j}) = \begin{cases} \frac{d_{m,j}}{B_L}, & (Y_{m,p}, Y_{m,q}) \in A_1 \\ 0, & (Y_{m,p}, Y_{m,q}) \in A_2 \\ \frac{d_{m,i}}{B_W}, & (Y_{m,p}, Y_{m,q}) \in A_3 \end{cases} \tag{8}$$

Where, BW and BL address the transfer speed of WAN and LAN separately. The T^t controlled by information and the data transfer capacity of the organization. When the two computational errands are executed in similar circumstances, the transmission time is disregarded. The transmission time of the mth work process Tt (X_m) is determined by

$$T^t(Y_m) = \sum_{(V_{m,p}, V_{M,q}) \in ED_m} T^t(Y_{m,p}, Y_{m,q}) \tag{9}$$

Let $T^t(Y_m)$ be the ET of the *m*th workflow, which is determined by

$$T^t(Y_m) = T^L(Y_m) + T^e(Y_m) + T^t(Y_m) \tag{10}$$

2.2. Energy Consumption Model

In WAN the gross energy time consumption is calculated by the generated energy during processing Pi and generated energy by the transmission Pj . In the case of implementing the mobile devices for energy computation when the mission is performed locally, there is no offloading energy consumptions implemented in the mobile device [5]. The state of a server's CPU, memory, hard drive, network card, and other components all have an impact on its energy consumption. Among these components, the CPU consumes the most energy [8]. Furthermore, agreeing on a server's power consumption can be precisely represented by a liner connection between the power utilization and the CPU use. Thus, the CPU usage is commonly used to address the use of worker resources, and we can obtain the worker's energy utilization in a roundabout manner based on the CPU usage. A notable reality that cannot be overlooked in the investigation of server energy utilization is that the necessary energy utilization of a worker out of gear state represents more than 62 percent of the energy utilization of a worker running in full state. In this manner, the figuring energy utilization of the mth cell phone in IoT is determined by

$$ET\ local\ (Y_m) = M_p * C_1/AM_p$$

$$ET_{cl}\ (Y_m) = Stap * C_1\ /\ AM_p + P_s * J_s/\ PA/LAN$$

$$ET_c(Y_m) = M_p * C_1/AM_p + P_s * J_i + J_o/PAN/WAN \tag{11}$$

2.3. Execution Time Model

For the registering task vm, I, come about the calculation offloading procedure xm,i, where Vm,i is the mth work process in the performance of figuring errand, the registering time is dictated the responsibility of the registering task and the figuring power of the performance stage. The Cloudlet offloading chance is rejected by the VMs. If there is a time requirement for access to assets of calculating undertakings sent on the cloudlet, the cell phones decide whether to perform these errands or offload them to the cloud. The Total time utilization for the mth work process is determined as

$$TT_{local}(X_m) = Cll/Avg_m$$

$$TT_{cl}\ (X_m) = Cll/Avg_m + J_{size}/PA/LAN$$

$$TT_c(X_m) = Cll/Avg_m + J_{size/\mathrm{PA/WAN}} \tag{12}$$

Where, TTcl and TTc signify the figuring force of the cell phones, the cloudlet and the cloud separately. CT local, CTcl, CTc implies the absolute execution season of the work process figuring is done on the neighborhood execution on the mobiles, sent from the cell phone to the cloudlet, or then again through LAN or to the cloud separately, where BW and BL address the transfer speed of WAN and LAN individually. The T^t controlled by the responsibility of the transmission information and transfer speed of the organization.

2.4. Execution Price cost

According to the power consumption, we can compute the average cost of the MDs as follows C_{local}=0 where C_{local} define to be the offloading process of the devices within the same devices means, it doesn't cost anything, where as in cloud servers and fog nodes the cost is calculated by input data transfer +processing cost +output transfer , thus the cost calculation is done.

2.5. Algorithm Design

In this segment, at first validate the complex agenda of concurrent workflows in cloud-edge computing using different scheduling approaches, followed by an assessment using Fog Work Flow sim Simulator to select the optimum results for the computational errands in the different scheduling algorithm such as FCFS, PSO, Round Robin, Genetic , MaxMin, MinMin, Data Aware, Static thereby comparing the results the traditional genetic algorithm optimization in the fog nodes resulted best for the multiobjectives optimization and the PSO algorithm resulted only for the time and c ost objectives in cloud offloading, with respect to edge computing. By comparing the result obtained, we have got different strategy regarding designing the workflows and address the formulated problem. To get the good solution with respect to consideration of optimal strategy, the three objective functions Cost, Time, Energy values are normalized and is denoted as w_1(C), w_2 (T) and w_3(E). The utility value is chosen as the most optimal strategy, which is 1. As a result, the fitness functions for cost, time, and energy are set to 0.4, 0.3, and 0.3, respectively, to measure the utility values of the solutions. The b-the chromosome's utility value is denoted by

$$w\ (Jb) = w_1(C) + w_2(T) + w_3\ (E)$$

$$w_1 + w_2 + w_{3=1}$$

Therefore, the optimal solution is the output of multi-objective optimization.

3. Result and Discussion

From these assessments we can unmistakably comprehend the hereditary calculation execution in Cloudlet being the awesome energy utilization. To evaluate the proficient ideal arrangement is energy and are utilized. The techniques are carried out under the far reaching utilized FogWorkFlowSim system on a PC with

Intel Core i3-5005U CPU@ 2.00 GHz processors and 4 Giga Byte Random Access Memory. The primary key goal of these contrastive technique is given as follows,

Table 2. Parameter s and their values

Parameters	Values
The latency of MD	0.1 W
The total number of ECD	5
The BW of LAN	100Mbps
The BW of WAN	40 Mbps
The number of task per job	20
The total number of VM's	100
Data size of figuring errands	[0,300] MB

EC, ET, CT as the combining two objectives to get optimal solution on offloading strategies. ECT as the combination of three objectives on offloading with respect to implementation of migration techniques.

Table 3. Analysis based on CLOUD time, the PSO value is low.

Offloading Strategy	EC_T	ET_T	CT_T	ECT_T
GA - CLOUD	474	474.92	474.92	474.95
PSO - CLOUD	421.42	262.58	243.58	460.89

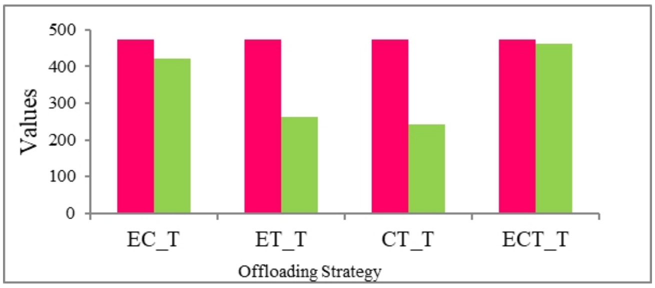

Figure 2. MDs offloaded in Cloud with respect to Time

Table 4. Analysis based on CLOUD time, the PSO value is low.

Offloading Strategy	EC_E	ET_E	CT_E	ECT_E
GA - CLOUD	14.24	14.24	14.24	14.24
PSO - CLOUD	39.64	18.79	163.82	59.97

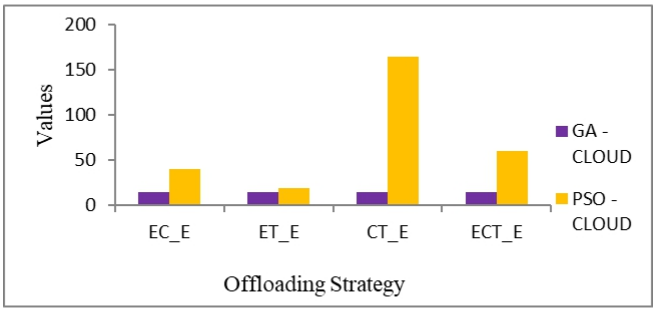

Figure 3. MDs offloaded in Cloud with respect to Energy

In Figure 3, the average execution time obtained with respect to energy is comparatively lesser for GA scheduling when compared to PSO as the MDs are offloaded to the cloud. As a result, we can compute the CPU utilization indirectly and obtain the total energy consumption of MDs in the cloud server. As basic PSO not suits direct implementation it should be used in edge nodes and cloud servers.

Table 5. Analysis based CLOUD cost, the PSO value is low

Offloading Strategy	EC_C	ET_C	CT_C	ECT_C
GA - CLOUD	455.83	455.8	455.83	455.8
PSO - CLOUD	220.59	279.1	70.45	281.5

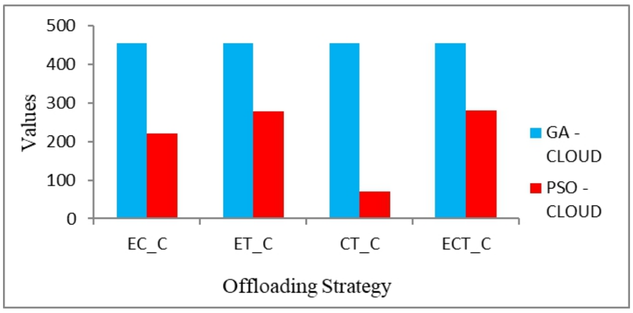

Figure 4. MDs offloaded in Cloud with respect to Cost

In Figure 4, the average execution time obtained with respect to cost is lesser for PSO scheduling when compared to GA as the MDs are offloaded to the cloud.

Table 6. Analysis based on FOG time, the PSO value is low.

Offloading Strategy	EC_T	ET_T	CT_T	ECT_T
GA - FOG	278.16	278.16	278.16	278
PSO - FOG	282.93	278.86	517.79	381.77

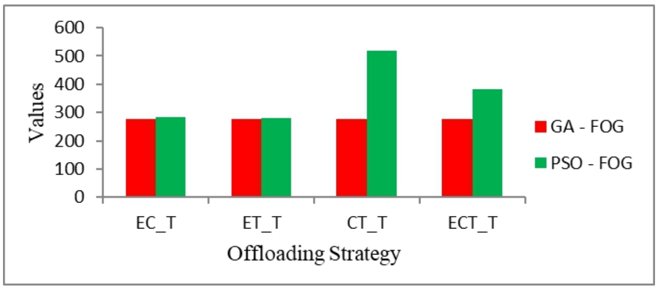

Figure 5. MDs offloaded to Fog with respect to Time

In Figure 5, the average execution time obtained with respect to time is less for GA scheduling when compared to PSO as the MDs are offloaded to the cloudlet.

Table 7. Analysis based on FOG energy, the GA value is low.

Offloading Strategy	EC_E	ET_E	CT_E	ECT_E
GA - FOG	8.34	8.34	8.34	8.34
PSO - FOG	24.98	12.4	359.09	47.54

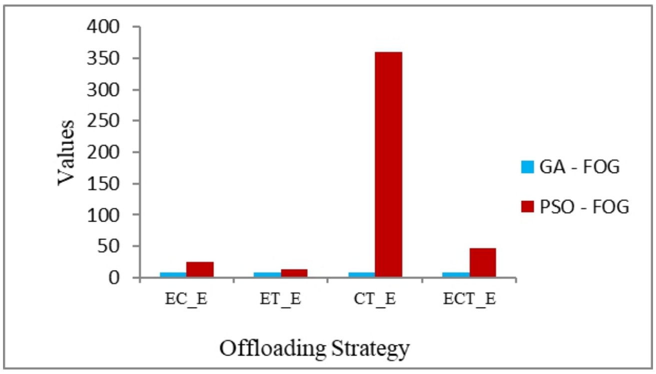

Figure 6. MDs offloaded in Fog with respect to Energy

Table 8. Analysis based on FOG cost; the GA value is low.

Offloading Strategy	EC_C	ET_C	CT_C	ECT_C
GA - FOG	133.47	133.47	133.47	133.5
PSO - FOG	155.26	301.9	92.23	239

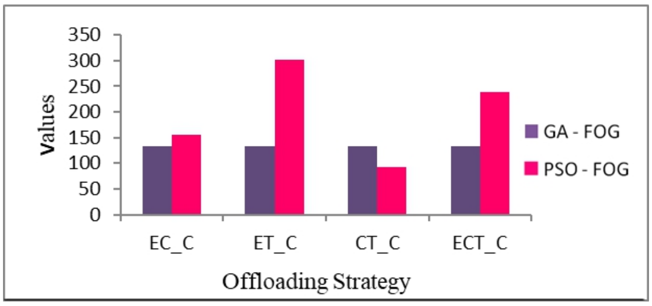

Figure 7. MDs offloaded Fog with respect to Cost

In Figure 7, the average execution time obtained with respect to cost is less for GA scheduling when compared to PSO as the MDs are offloaded to the cloudlet. For better offloading strategies, the MDs are offloaded in the cloudlets which are the fog nodes where the tasks computing are done.

The scheduling algorithms and contrasts with single objective and multi-objective optimization for fog nodes can be analyzed in cloud servers and inside devices in this segment. The execution time, energy usage, and cost analysis of MD are three main metrics used to evaluate the efficiency of computing offloading methods. We analyze the distribution of computing tasks in different methods using various optimization techniques. The corresponding results are illustrated in Fig. 1, 2, 3, 4, 5 and 6. Based on the simulated results, the PSO technique gives better

better performance for time consumption and cost benefit in offloading of cloud servers whereas the GA gives overall best performance in all the three objectives during the task offloading in fog nodes.

4. Conclusion and Future work

In this article, we looked at how task offloading affects energy usage, delay efficiency, and cost in mobile devices, edge nodes, and the cloud. For each mobile unit, we tailored the offloading probability and scheduling strategies, and normalized weighting factors were placed on the goals to find the best solution. This paper introduced a optimization approaches using PSO and GA based on simulation data, with the standard genetic algorithm providing the highest overall value for mission offloading in fog nodes using multi-objectives. With the assumption of various workflow models as single and multi-objective in data centers as cloud servers, fog nodes, and within computers, we extracted the analytic results of energy usage, delay efficiency, and cost. Then formulated the multi-objective problem with different constraints and solved it using various scheduling algorithms based on the obtained data. Experimental analysis of the method, the cloud data time is 11% lower than the GA, the cloud cost of the method is 51.6% lesser than the GA, method and the fog time method of the proposal method is16% higher than the GA. In future work, it will adapt and build on the suggested solution in a real-world IoT scenario. This Paper also expand our research into improved overall task offloading efficiency in edge computing. This strategy will be the cornerstone of our future efforts.

References

[1] Xiaolong Xu,Qingxiang Lie, Yun Lup, Peng,Xuyung Zhang,Shunmei Meng, Lianyong Qi, “A computation offloading method over big data for IoT-enabled cloud- edge computing,” Vol. 95, pp. 522-533, 2019.

[2] Xu Chen,Qian Shi,Lei Yang, Jie Xu, “ThriftyEdge: Resource- Efficient Edge Computing for Intelligent IoT Applications,” IEEE Network. Vol: 32, Issue: 1, pp.61-65, Jan.-Feb, 2018.

[3] Yuanzhe Li, Shangguang Wang, “An energy-aware Edge Server Placement Algorithm in Mobile Edge Computing,” IEEE (2018).

[4] Abderrezak Rachedi,Abderrahim Benslimane, “Multi-objective optimization for Security and QoS adaptation in Wireless Sensor Networks,” IEEE ICC 2016 Ad-hoc and Sensor Networking Symposium.

[5] Xiaolong Xu, Xuyun Zhang, Honghao Gao, Yuan Xue, Lianyong Qi, Wanchun Dou, “BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing”, IEEE Transaction on Industrial Informatics, Vol. 16, Issue: 6, pp. 4187-415 June 2020.

[6] Sergio Barbarossa, Stefania Sardelliii, Paolo Di Lorenzo, “Computation offloading for mobile cloud computing based on wide cross-layer optimization , “ Future Network & Mobile Summit, IEEE 2013.

[7] Xu Chen, Lei Jiao, Wenzhong Li, Xiaoming Fu, “Efficient Multi-user Computation Offloading for Mobile-Edge Cloud Computing,” IEEE/ACM Transactions on Networking, vol. 24, pp. 2795-2808, 2015.

[8] Changsheng You, Kaibin Huang, Hyukijim Chae, Byoung-Hoon Kim,” Energy-Efficient Resource Allocation Mobile-Edge Computing,” IEEE/ACM Transaction on Wireless communications, vol. 16, pp. 1397-1411,2016.

[9] Liqing Liu, Zheng Chang, Xijuan Guo, Shiwen Mao, Tapani Ristaniemi, “Multiobjective Optimization for Computation Offloading in Fog Computing,” IEEE Internet of Things, vol.5, pp. 283-294, 2017.

[10] Hong Quy Le; “Efficient resource allocation in mobile- edge computation offloading: Completion time minimization”, IEEE International Symposium in Information Theory(ISIT), 2017.

[11] Ke Zhang, Yuming Mao, Supeng Leng, Quaxin Zhao et. Al ,”Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks”, IEEE Access, vol. 4, pp. 5896-507, 2016.

[12] Amal Ellouze, Maurice Gagnaire, Ahmed Haddad, “A Mobile Application Offloading Algorithm for Mobile Cloud Computing”, IEEE, 2015.

[13] Wang, J., Wu, W., Liao, Z., Sangaiah, A.K. and Sherratt, R.S., 2019. An energy-efficient off-loading scheme for low latency in collaborative edge computing. IEEE Access, 7, pp.149182-149190.

[14] Yang, X., Yu, X., Huang, H. and Zhu, H., 2019. Energy efficiency based joint computation offloading and resource allocation in multi-access MEC systems. IEEE Access, 7, pp.117054-117062.

[15] Sun, H., Zhou, F. and Hu, R.Q., 2019. Joint offloading and computation energy efficiency maximization in a mobile edge computing system. IEEE Transactions on Vehicular Technology, 68(3), pp.3052-3056.

[16] Peng, K., Huang, H., Wan, S. and Leung, V.C., 2020. End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment. Wireless Networks, pp.1-12.