

5G Data Offloading Using Fuzzification with Grasshopper Optimization Technique

V. R. Balaji^{1,*}, T. Kalavathi², J. Vellingiri³, N. Rajkumar⁴ and Venkat Prasad Padhy⁵

¹Department of ECE, Sri Krishna College of Engineering and Technology, Coimbatore, 641008, India

²Department of EIE, Kongu Engineering College, Erode, 638060, India

³School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, 632014, India

⁴Department of CSE, School of Engineering, Presidency University, Bangalore, 560064, India

⁵School of Computing Science and Engineering, VIT Bhopal University, Bhopal, 466114, India

*Corresponding Author: V. R. Balaji. Email: balajivr@skcet.ac.in

Received: 17 June 2021; Accepted: 18 August 2021

Abstract: Data offloading at the network with less time and reduced energy consumption are highly important for every technology. Smart applications process the data very quickly with less power consumption. As technology grows towards 5G communication architecture, identifying a solution for QoS in 5G through energy-efficient computing is important. In this proposed model, we perform data offloading at 5G using the fuzzification concept. Mobile IoT devices create tasks in the network and are offloaded in the cloud or mobile edge nodes based on energy consumption. Two base stations, small (SB) and macro (MB) stations, are initialized and the first tasks randomly computed. Then, the tasks are processed using a fuzzification algorithm to select SB or MB in the central server. The optimization is performed using a grasshopper algorithm for improving the QoS of the 5G network. The result is compared with existing algorithms and indicates that the proposed system improves the performance of the system with a cost of 44.64 J for computing 250 benchmark tasks.

Keywords: 5G; energy consumption; task offloading; fuzzification; grasshopper optimization; QoS; mobile IoT

1 Introduction

Today, every application is computed via mobile devices, wearable devices in the network using the Internet of Things (IoT) [1,2]. Sensor technology uses wireless networks to transfer the data to the destination from the source and combines IoT with mobile devices. Health care applications are more dependent on timely data transfer [3,4]. Mobile devices support some high-level applications by supporting user's interests, creating big data every minute [5–8]. Today, cloud computing attempts to support a wide range of big data, but in reality, it causes more time with finite memory capacity [9,10]. This limitation cannot be accepted by 5G users when the users move towards high computation capacity. The performance of the 5G can be improved by offloading the data in the edge nodes to the cloud storage using IoT in the network.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The time required by applications to compute and energy required to compute the task in mobile devices and IoT must be reduced. Users are satisfied when applications are launched in the 5G network. Mobile IoT devices are deployed to the cloud in the network. Some devices may be very near the cloud and others may be located far away or even in remote areas. Remoteness of the location causes the remote cloud to process the data with high bandwidth. Meanwhile, a wide-area connectivity receives low bandwidth, resulting in high latency [11]. Advanced 5G networks must require enough bandwidth with a short offloading capacity to ensure fast uploading and downloading data. To satisfy 5G users the edge nodes are required to process the IoT devices. Edge nodes act as small data centers with required resources and base stations. The access point for the edge nodes is set in a radio access network with secured resources services [12].

The main contribution of the paper is as follows:

- The proposed work introduces the energy model by offloading the tasks generated by the mobile IoT devices using a fuzzyfication approach to the Edge cloud and 5G network macro stations.
- These energy model system parameters have been optimized further with the meta-heuristic algorithm called Grasshopper optimization to minimize the network energy. This optimization algorithm will offload the task with the optimization of system parameters that can enable the optimal solution to offloading the task to mobile devices, cloud edge, or macro stations.
- This heterogeneous network was designed specifically to improve the energy consumption of the 5G wireless communication networks using two parameters such as network and system.
- With the simulation experiments and comparative analysis, the proposed fuzzyfication-based meta-heuristic approach outperformed other approaches in terms of energy consumption, execution time, and resource utilization of 5G wireless communication networks.

The rest of the paper is organized into five sections as shown: The literature survey for the data offloading is studied and presented in Section 2. The implementation of the proposed methodology is described in Section 3. Section 4 evaluates the results of our proposed model by comparing it with the existing model. Section 5 concludes the work and provides suggestions for the future scope of research.

2 Related Work

Intelligent algorithms are widely used in the present research scenario to improve the performance of the 5G networks. In this section, existing mobile edge computing and IoT techniques are discussed.

An alternative iteration algorithm [13] is used for energy-efficient task offloading in the edge computing framework. The main problem is that it takes more time to search the resources in the mobile edge network. Optimization of the system energy uses the sub-optimal resources selection algorithm [14] for orthogonal frequency and time division multiple access offloading. In this research article, multiple users are processed in mobile edge computing to offload the data with less energy. Another model [15] for offloading data with less energy consumption uses a fish swarm optimization algorithm. This approach is an intelligent algorithm proposed for task communication and computation. It helps to solve energy-efficient task offloading in edge and cloud computing models. The performance of the edge computing model is increased through a new index technique [16]. In this paper, they combine an iterative algorithm with gradient descent to optimize the energy of the system. It reduces the consumption of energy and improves the performance of mobile edge networks. The algorithms only focused on energy saving through nearest device offloading. However, the density of the network remains a considerable problem that does not apply for 5G/6G networks.

Ultra-dense networks (UDN) such as 5G and beyond can be reviewed as follows. The UDN base station uses DEEP LSTM to predict the traffic in the network. This algorithm helps to avoid network congestion [17]. Power transmission and resource blocks allocation to save energy in UDN are improved using

non-dominant sorting genetic algorithm II [18]. This algorithm not only improves energy efficiency but also the spectral efficiency of the edge computing network.

The aim of minimizing delay in task offloading is addressed using a software-defined scheme [19]. This scheme helps UDN networks to save energy in managing delay. Lyapunov optimization algorithm [20] is used in a dynamic environment to offload the data with less computational cost. The execution cost of the task in this algorithm is effectively managed by a dynamic optimization role. The greedy algorithm [21] is designed for UDN to minimize the energy consumption in data offloading. It is a heuristic approach used to manage mobile devices for less energy consumption. Two convex function differences [22] are designed as an algorithm iteratively to search the energy-efficient nodes in multi-cell networks. It offers an optimal solution for data allocation and resources allocation in edge computing architecture.

3 Proposed Fuzzification with Grasshopper Optimization Methodology

The proposed architecture of the energy-efficient 5G network model has been shown in Fig. 1. Core cloud consists of the central server to store the data transmitted over the network. The Mobile IoT devices are the task creator. The mobile edge node has two base stations, namely, small (SB) and macro base stations (MB). While executing the tasks as a whole, the server energy is wasted and the resources are not utilized fully, which also increases the execution time that will affect the quality of service of the network. The proposed system offloads the task generated by the mobile IoT devices to the cloud server and mobile edge node to reduce the energy. The tasks are randomly offloaded to MD for execution first. Because of battery consumption, the remaining tasks are distributed to MB and SB of the mobile edge node. The choice of SB or MB is accomplished using the fuzzification method [23]. The tasks are distributed to a central server for execution. The system energy is reduced with the optimization technique called grasshopper optimizer. This optimization technique can improve the QoS of the 5G network system. The mobile devices are represented as $MD(i) = \{1, 2, 3, \dots, M\}$. The offloaded task to a central cloud, SB of Mobile edge node and MD is represented in Eq. (1).

$$D_{(n,m)} = \begin{cases} 1 & \text{if task } t_{n,i} \text{ is executed in MD} \\ 2 & \text{if task } t_{n,i} \text{ is executed in MB} \\ 3 & \text{if task } t_{n,i} \text{ is executed in central cloud} \end{cases} \quad (1)$$

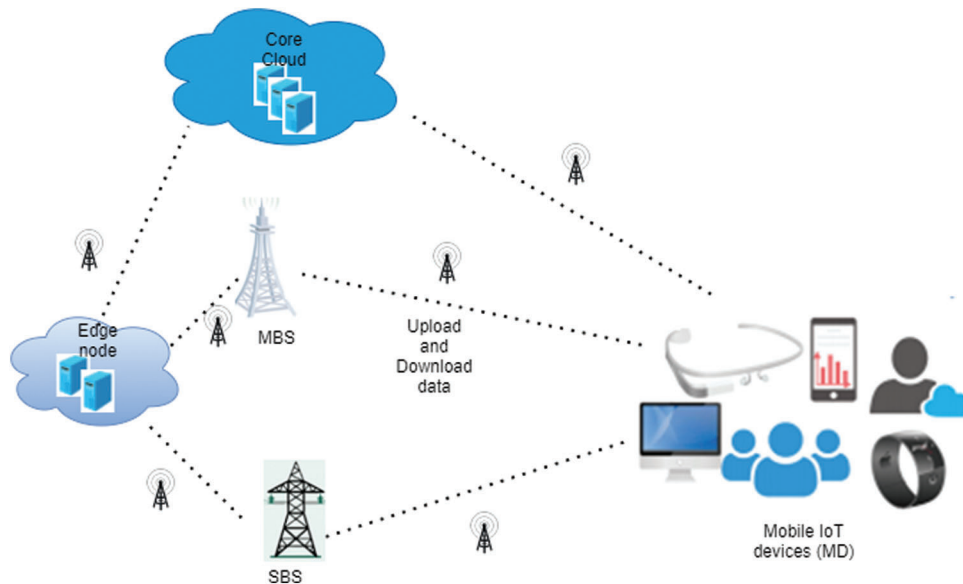


Figure 1: Proposed energy-efficient offloading system overview

where $n \in N = \{1, 2, 3, \dots, N\}$ refers to the number of MD and $m = \{p_1, p_2, p_3, \dots, p_n\}$ refers to the number of tasks of Nth device. The computation capacity of MB is represented as, $\rho_1^{mb}, \rho_2^{mb}, \rho_3^{mb}, \dots, \rho_n^{mb}$. In Eq. (1), the offloading decision 1 means that the task is offloaded to the mobile device itself, value 2 means the task is offloaded to the mobile edge node server of MB based on fuzzification, and value 3 means the task is offloaded to the central cloud. Executing the task in the MEC server through the MB and SB will reduce the energy consumption and reduce the time delay due to the utilization of the resources because the edge node is located near to the MD and the communication is passed faster as a wireless signal through the 5G network.

3.1 Energy Model

The process of the proposed offloading mechanism is based on the following conditions of ρ_1^{mb} and p_1

Condition 1: if $\text{capacity}(p_1 < \rho_1^{mb})$, then some tasks are offloaded to the mobile device itself and the remaining tasks will be offloaded to the macro station of the mobile edge node server.

Condition 2: if $\text{capacity}(p_1 > \rho_1^{mb})$ and no neighboring server to the MB exists, then some tasks are offloaded to the MB of the edge node server, and the remaining are offloaded to the central cloud.

Condition 3: if $\text{capacity}(p_1 > \rho_1^{mb})$ and a neighboring server to the SB exists, then some tasks are allocated to the SB and the remaining tasks are offloaded to MB.

Let $N = \{1, 2, 3, \dots, N\}$ be the number of mobile devices in the network, $m = \{1, 2, 3, \dots, M\}$ be the number of tasks generated by the MD, and $n = \{1, 2, 3, \dots, n\}$ be the number of SB servers in the mobile edge node. Assume the computational task c_i^t is generated from each MD user i , where $i \in N$ at each time slot t . In the case of condition 1, the energy used to execute the computation task at MD is represented in Eq. (2) and the execution time to complete that task at MD is shown in Eq. (3).

$$EC1_{(n,m)} = \theta_l \lambda(n, m) = \theta_l s(n, m) \quad (2)$$

$$ET1_{(n,m)} = \frac{s(n, m)}{f_l} \quad (3)$$

where θ_l is the energy capacity of local MD, λ is the total CPU cycles of executing the workflow, $s(n, m)$ refers to the total size of mth task at nth device, and f_l is the task processing rate. In the case of condition 2, the total arrival computation task of SBS at time t is shown in Eq. (4) while the total computation tasks to reach SBS is defined in Eq. (5).

$$c^t = \{c_n^t\}_{n \in N} \quad (4)$$

$$X^t = \{X_i^t\}_{i \in n} \quad (5)$$

Based on the fuzzy rules, the offloading decision will be made, and that value will be converted into crisp value later. The variables used in the fuzzification process for offloading the task to SB [23] are shown in Eq. (6).

$$\Phi = \{\alpha, \beta, \gamma, \delta, \theta\}, \quad \Phi \in X^t \quad (6)$$

where α refers to the number of input task size in the range of 0–50, β refers to the task delay in the range of 0–1, γ is the computation capacity of SB and VM utilization in the range of 20–100, δ is the network delay in the range of 2–40, and θ is the MB utilization near to the SBS. The membership function and its fuzzy set are shown Eqs. (7) and (8), respectively.

$$\psi_A(X) : X \rightarrow [0, 1] \forall x \in X \quad (7)$$

$$A = \{(X, \psi_A(X))\} : x \in X \quad (8)$$

This memberships function can be of different types such as triangular, Gaussian, piece-wise linear, singleton, etc., [24]. The proposed work uses the triangular membership function shown in Eq. (9). The memberships function is shown in Fig. 2.

$$\psi_A^{triangular}(X) = \begin{cases} 0 & \text{if } X \leq p \\ \frac{X-p}{n-p} & \text{if } p < X < n \\ \frac{q-X}{q-n} & \text{if } n < X < q \\ 1 & \text{if } X \geq q \end{cases} \quad (9)$$

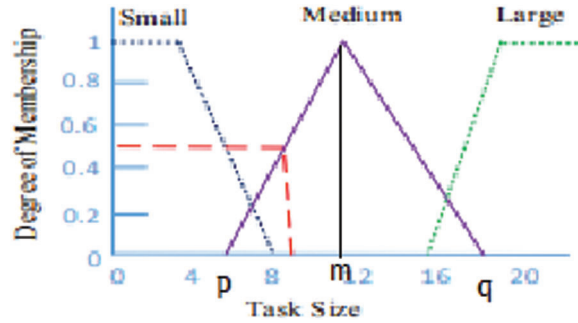


Figure 2: Triangular memberships function of the proposed work

The transmission times of the uplink and downlink the data is denoted as d_{tu} and d_{td} , respectively and the data rate between the SB and MB are denoted as r_0 and r_1 . The transmission times of the uplink and downlink the component on the offloading site use Eqs. (10) and (11) and the execution time of this MEC server is shown in Eq. (12).

$$EMEC_{t(m,i)}^s = \frac{d_{tu}}{r_i} \forall x \in X \text{ and } \forall i \in [0, N] \quad (10)$$

$$EMEC_{t(m,i)}^r = \frac{d_{td}}{r_i} \forall x \in X \text{ and } \forall i \in [0, N] \quad (11)$$

$$TMEC_{t(m,i)} = EMEC_{t(m,i)}^s + EMEC_{t(m,i)}^r \quad (12)$$

As per condition 3, the energy consumption at the central cloud of the offloaded task is calculated using Eq. (13) and execution time is calculated using Eq. (14)

$$EC_{(n,m)} = \sigma s(n, m) \quad (13)$$

$$TC_{(n,m)} = \frac{s(n, m)}{b_n} + \frac{\lambda}{f_c} \quad (14)$$

where σ is the positive coefficient, b is the bandwidth of MD, and f_c is the task processing of central cloud. The overall energy efficiency of the system is optimized with the energy optimization algorithm called Grasshopper Optimizer.

3.2 Optimization Using Grasshopper Optimizer

The selected cluster head using the blockchain approach has been optimized with a grasshopper optimizer to select the optimal solution to reduce the energy. The clustering is based on the grasshopper swarm behavior. Each agent represents the route that contains the vehicle ID selected by the cluster head. All the parameters, such as vehicle's place on the highway, vehicle direction, speed and velocity of the vehicle, inter-vehicle distance, and grasshopper search space are initialized. The random position initialized is used to start a node in a random direction. The search space is created and the fitness function calculated to form the cluster. For each iteration, the position of the vehicle and the weight are updated. The output of the process is the optimal clusters near the RSU. The operators used in GO is shown as follows:

$$\varphi a = Sf_a + Gf_a + Ad_k \quad (15)$$

where φa is the position of the a^{th} agent, Sf_a represents the social attraction and repulsion force, Gf_a refers to the gravity force, and Ad_k -ari drift variable.

These equations are based on the random behavior of the swarm with the random numbers $r1$, $r2$, and $r3$ with the interval of $[0,1]$. Hence the equation is rewritten as follows:

$$\varphi a = r1 \times Sf_a + r2 \times Gf_a + r3 \times Ad_k \quad (16)$$

The operators are shown as follows:

$$Sf_a = \sum_{p=1, l \neq a}^n s(dpj)D \quad (17)$$

$$\text{social force } s(\gamma) = fi \times e^{-p} - e^{-\gamma} \quad (18)$$

$$\text{gravitational force } Gf_a = -G \times E_g \quad (19)$$

$$\text{air drift } Ad_a = Dc \times E_w \quad (20)$$

where is the dpj is the distance between p^{th} and j^{th} agent, l refers to the length scale, fi represents the attraction intensity, E_g is the earth vector, G is the gravitational constant, Dc refers to the drift constant, and E_w is the wind direction.

Initially, the agents have no wings and they move in the direction of the wind. For each agent i , the operators are replaced in Eq. (10) as shown in Eq. (16). The stages of the growth of the grasshopper and its social behavior and the workflow of the proposed energy-efficient IoV method are shown in Fig. 3.

$$\varphi a = \sum_{p=1, l \neq a}^n s(|\varphi p - \varphi l|)(\varphi p - \varphi l)/dlj - G \times E_g + Dc \times E_w \quad (21)$$

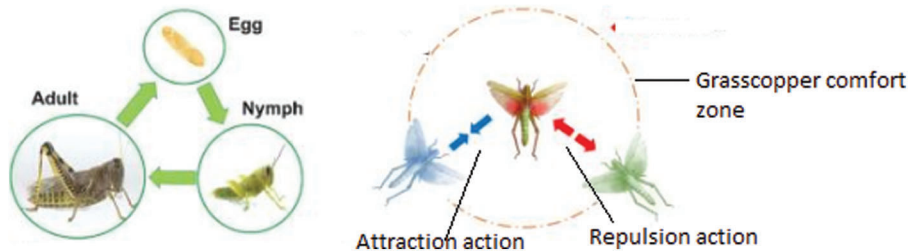


Figure 3: Stages of the growth of the grasshopper and its social behavior

Algorithm 1: Proposed energy efficient 5G system

Input: Workflow of MD, initial population X , max number of iteration t_{max} , size S , control parameter a , and swing factor C and E .

Output: offloading decision with energy and time and optimal solution ω

Step 1: while stopping condition not met

Step 2: for $n = 1$ to N

Step 3: for $m = 1$ to M

Step 4: $C = 0, \Phi = \{\alpha, \beta, \gamma, \delta, \theta\}, \Phi \in X'$

Step 4: If $(C < 50)$ then

Step 5: compute Energy consumption of MD using Eq. (2)

Step 6: compute Execution cost of MD using Eq. (3)

Step 7: Else

Step 8: //offload the task at SB/MB of edge node

Step 9: compute energy consumption using (10) and (11)

Step 10: compute execution time using Eq. (12)

Step 11: End if

Step 12: //offload task into central cloud

Step 13: compute energy using Eq. (13)

Step 14: compute execution time using Eq. (14)

Step 15: End for

Step 16: End for

Step 17: // Grasshopper Optimization

Step 18: for iteration = 1 to 15 (stall iteration)

Step 4: while (nodes != empty)

Step 5: node cluster = all nodes (vehicle)

Step 6: end while

Step 7: while $1 \leq \text{iterations}$

Step 8: for $i = 1$ to population

Step 9: update C using Eq. (19)

Step 9: distance normalization, position update using Eq. (21)

Step 10: bring all the agents into lower and upper bound of the cluster

Step 11: end for

Step 12: update best route cost

Step 13: iteration = iteration + 1

Step 14: end while

Step 15: return best route cost

Step 16: end for

The energy optimization with Grasshopper optimization will reduce the network energy further and improve the quality of the 5G wireless communication services efficiently. The offloading task will effectively utilize the network resources. Hence, the proposed offloading scheme with an evolutionary algorithm outperforms the other approaches in terms of energy consumption, execution time, and resource utilization.

4 Simulation Results and Discussions

The proposed offloading mechanism on the 5G network was examined using the MECHREVO-Ti2 experimental setup. The simulation parameters are listed in [Tab. 1](#).

Table 1: Simulation parameters

Parameter	Value
Distance between MD and MB	1200 m
Distance between MD and SB	120 m
Channel bandwidth	6 MHz
Gaussian noise	σ^2 -128dbm
Transmission power of MBS and SBS	[0.2,.03] w
Transmission power of MD	0.2 W
Scale of tasks	[50, 100, 150, 200, 250, 300]
Edge services size	[0.5, 0.8]
Computing power of edge-cloudlet	2300 MHz
Computing power of central cloud	45000 MHz
Scale of VMs	[1, 5]

Mobile IoT devices are the source of task generators. The generated number of tasks are listed in column 1 of [Tab. 2](#). The energy consumption of executing the multiple tasks using the proposed offloading method is shown in [Tab. 2](#).

Table 2: Energy and execution cost of the proposed offloading method

Number of Tasks	Energy Consumption (J)				Execution Time (s)			
	MD	Central cloud	MB-edge node	Total	MD	Central cloud	MB-Edge node	Total
50	8.7	2.31	1.34	12.35	10.02	4.13	3.61	17.76
100	10.72	5.91	3.23	19.86	15.82	10.27	8.6	34.69
150	13.03	7.6	5.94	26.57	18.63	12.4	10.29	41.32
200	18.62	10.31	7.52	36.45	20.82	19.3	16.82	56.94
250	22.34	13.26	9.04	44.64	23.61	20.3	18.52	62.43

[Tab. 2](#) shows that the number of tasks executed in different offloading sites such as Central cloud, Mobile Device, and mobile edge node-MB's energy and execution time is evaluated. The number of tasks

executed on mobile devices is less because of the limited number of resources and battery consumption. The remaining tasks are allotted to the central cloud and macro base station of the edge node server for execution. Because of the use of the 5G network, more tasks are allocated to the edge node server located nearer the MD for the fastest execution. It will save on the execution time and the energy. The evaluated results are illustrated in Figs. 4 and 5. Comparatively, the tasks allocated to the edge server through the wireless signals are executed faster than the other two offloading sites. The distributive nature of the tasks to the offloading sites will consequently decrease the energy and time with efficient resource utilization.

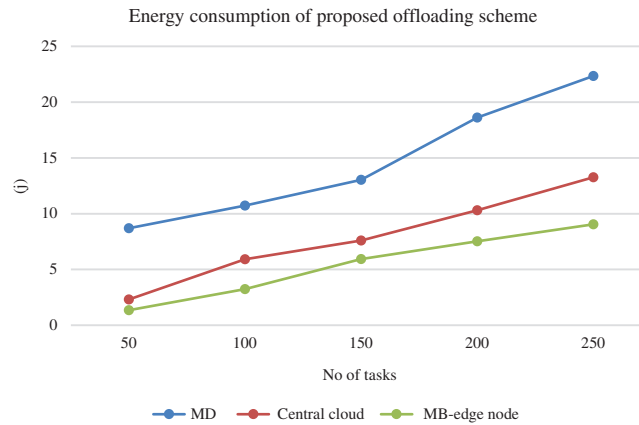


Figure 4: Energy consumption of proposed offloading

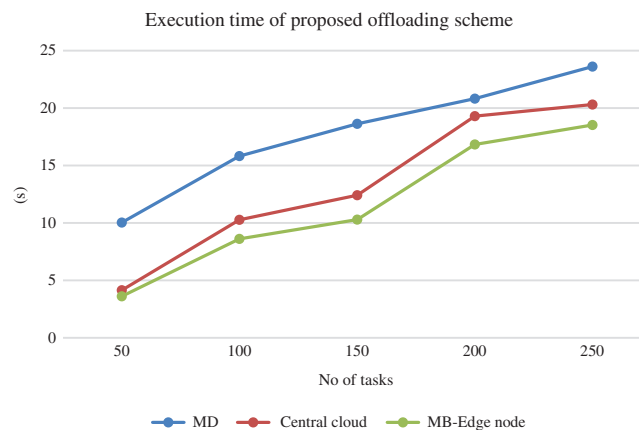


Figure 5: Execution time of the proposed offloading scheme

Comparative Analysis

The efficiency and effectiveness of the proposed offloading method with energy optimization based on GO on 5G network are proved by analyzing the comparative analysis with basic existing task offloading methods such as Benchmark, First Fit decreasing based task offloading with time-saving method (FFD) [25] and Fuzzy logic based collaborative task offloading (FCTO). In the benchmark, the initial node has been allocated to the execution of the task. FFD allows the required number of VMs to execute the tasks that are ranked in order. In FCTO, the edge nodes are allocated to execute the tasks to reduce the time and complexity of the system using the fuzzification technique. The existing offloading methods are based on single-site offloading, while our proposed approach executes the tasks in multi-sites and

optimizes with GWO to reduce the energy further and save on the execution time. The comparative analysis in terms of energy consumption of these algorithms are shown in [Tab. 3](#) and illustrated in [Fig. 6](#).

Table 3: Energy cost comparison of proposed vs. existing offloading methods

No of Tasks	Energy Cost (J)			
	Benchmark	FFD	FCTO	Proposed Energy Efficient offloading
50	150.82	130.02	180.52	12.35
100	190.42	210.92	240.93	19.86
150	210.32	240.41	280.81	26.57
200	284.81	313.02	360.1	36.45
250	320.51	380.07	390.81	44.64

From the observation of [Tab. 3](#) and [Fig. 6](#), the proposed multi-level offloading with GO can ensure less energy compared to the traditional algorithms. In the benchmark, MD is the main source of execution. Due to this, the energy and time are increased on executing the tasks which are higher than other algorithms. Comparatively the proposed algorithm reduces the energy for executing the tasks with the major difference to the existing approaches. For 250 tasks, the proposed algorithm consumes only 44.64 J for execution. Various other algorithms such as benchmark, FFD, FCTO consumes 320.51, 380.07 J, and 390.81 J respectively. Hence, our proposed multi-level fuzzification offloading with energy optimization using grasshopper optimizer reduces the energy at the maximum on 5G network. The comparison of the execution cost is shown in [Tab. 4](#).

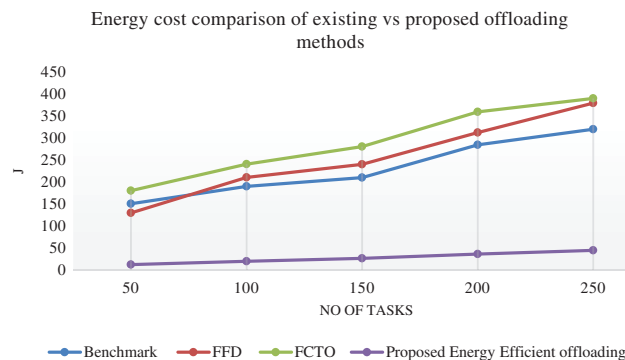
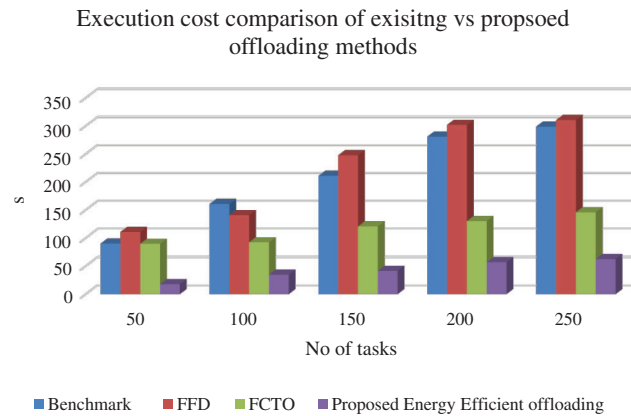


Figure 6: Energy comparison of existing vs. proposed offloading schemes

The comparative analysis in terms of the execution time of various offloading methods is illustrated in [Fig. 7](#). The resource limitation of existing algorithms results in the consumption of more time to execute the allocated number of tasks. Our proposed algorithm was able to manage the resources because of its distributive nature. A low number of tasks are allocated to MD and the remaining tasks are divided and offloaded to the edge server and central cloud for execution. Comparatively, our proposed algorithm executes the task with less time than other existing algorithms. For 250 tasks, the proposed offloading method obtains 62.43 s only. Various algorithms such as benchmark, FFD, and FCTO consume 298.62, 310.45, and 145.67 s, respectively. Hence, our proposed algorithm is efficient and effective in obtaining less energy with minimum execution time on completing the various tasks generated by the MD through the 5G network.

Table 4: Execution time comparison of existing vs. proposed methods

Number of Tasks	Execution Cost (s)			
	Benchmark	FFD	FCTO	Proposed Energy Efficient offloading
50	90.01	110.73	89.62	17.76
100	160.62	140.52	92.32	34.69
150	210.72	247.53	120.62	41.32
200	280.67	301.72	130.02	56.94
250	298.62	310.45	145.67	62.43

**Figure 7:** Execution cost comparison of existing vs. proposed offloading schemes

5 Conclusion

Data offloading in a 5G environment is a challenging problem for future applications. This research article focused on efficient data offloading in the 5G computing network. In our work, we processed the tasks at two base stations as small base station and macro base station. When the incoming task is executed fully, energy consumption is very high resulting in a waste of resources and bad QoS in the 5G model. This problem is overcome by offloading the data at the mobile devices themselves, while the battery and memory requirements are acquired by the small base station or macro base station located nearby. Our proposed approach helps in addressing the offloading of data very efficiently and achieving resources efficiency with SB and MB. The cost of energy consumed in our model is at a minimal when compared to existing offloading schemes. The task does not require waiting for resources because of the availability of an alternative base station. It results in less computation time for the proposed 5G data offloading. This 5G data offloading model outperforms the other model in energy, cost, QoS, etc. In the future, the energy-efficient offloading can be implemented with artificial intelligence technologies to improve the performance of 5G networks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare no conflict of interest on the publication of this paper.

References

- [1] M. Bahrami, A. Khan and M. Singhal, "An energy-efficient data privacy scheme for IoT devices in mobile cloud computing," in *Proc. MS*, IEEE, San Francisco, CA, USA, pp. 190–195, 2016.
- [2] R. H. Jhaveri, N. M. Patel, Y. Zhong and A. K. Sangaiah, "Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial IoT," *IEEE Access*, vol. 6, pp. 20085–20103, 2018.
- [3] M. A. Razzaque, M. Milojevic-Jevric, A. Palade and S. Clarke, "Middleware for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, 2015.
- [4] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. Leung *et al.*, "Wireless energy harvesting for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.
- [5] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin *et al.*, "A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 481–492, 2018.
- [6] S. Sathesh and S. Maheswaran, "Agricultural harvest with incarcerate image using single board computer," *International Journal of Advanced Research Trends in Engineering and Technology*, vol. 2, pp. 11–15, 2015.
- [7] C. Wu, E. Zapevalova, Y. Chen and F. Li, "Time optimization of multiple knowledge transfers in the big data environment," *Computers, Materials & Continua*, vol. 54, no. 3, pp. 269–285, 2018.
- [8] S. Maheswaran, B. Vivek, P. Sivaranjani, S. Sathesh and K. Pon Vignesh, "Development of machine learning based grain classification and sorting with machine vision approach for eco-friendly environment," *Journal of Green Engineering*, vol. 10, no. 3, pp. 526–543, 2020.
- [9] X. Chen, Q. Shi, L. Yang and J. Xu, "Thrifty edge: Resource-efficient edge computing for intelligent IoT applications," *IEEE Network*, vol. 32, no. 1, pp. 61–65, 2018.
- [10] X. Wang, L. T. Yang, H. Liu and M. J. Deen, "A big data-as-a-service framework: State-of-the-art and perspectives," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 325–340, 2017.
- [11] X. Wang, L. T. Yang, X. Xie, J. Jin and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 80–85, 2017.
- [12] N. Ansari and X. Sun, "Mobile edge computing empowers internet of things," *IEICE Transactions on Communications*, vol. 101, no. 3, pp. 604–619, 2018.
- [13] Y. Hao, M. Chen, L. Hu, M. S. Hossain and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11365–11373, 2018.
- [14] C. You, K. Huang, H. Chae and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2016.
- [15] L. Yang, H. Zhang, M. Li, J. Guo and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, 2018.
- [16] H. Sun, F. Zhou and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3052–3056, 2019.
- [17] Y. Zhou, Z. M. Fadlullah, B. Mao and N. Kato, "A deep-learning-based radio resource assignment technique for 5G ultra dense networks," *IEEE Network*, vol. 32, no. 6, pp. 28–34, 2018.
- [18] S. Xu, R. Li and Q. Yang, "Improved genetic algorithm based intelligent resource allocation in 5G ultra dense networks," in *Proc. WCNC*, IEEE, Barcelona, Spain, pp. 1–6, 2018.
- [19] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.
- [20] Y. Mao, J. Zhang and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [21] H. Guo, J. Liu and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 14–19, 2018.
- [22] J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou *et al.*, "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.

- [23] M. D. Hossain, T. Sultana, V. Nguyen, T. D. Nguyen, L. N. Huynh *et al.*, “Fuzzy based collaborative task offloading scheme in the densely deployed small-cell networks with multi-access edge computing,” *Applied Sciences*, vol. 10, no. 9, pp. 1–23, 2020.
- [24] J. M. Mendel, “Fuzzy logic systems for engineering: A tutorial,” in *Proc. of the IEEE*, vol. 83, no. 3, pp. 345–377, Los Angeles, USA, 1995.
- [25] J. Jiang, X. Zhang and S. Li, “A task offloading method with edge for 5G-envisioned cyber-physical-social systems,” *Security and Communication Networks*, vol. 2020, pp. 1–9, 2020.