Journal of Autonomous Intelligence (2024) Volume 7 Issue 3
doi: 10.32629/jai.v7i3.1203

ORIGINAL RESEARCH ARTICLE

ETOSP: Energy-efficient task offloading strategy based on partial offloading in mobile edge computing framework for efficient resource management

Chander Diwaker, Aarti Sharma*

Department of CSE, UIET, Kurukshetra University, Haryana 136119, India

*** Corresponding author:** Aarti Sharma, uietphd2124aartisharma@kuk.ac.in

ABSTRACT

Mobile edge computing (MEC) is a very promising paradigm that facilitates efficient processing and analysis of Internet of Things (IoT) data at the network edge. MEC is a cloud-based platform that offers a range of online resources and sophisticated mobile apps to users of mobile devices. The continuing trend among mobile users is the growing need for accessing current apps and cloud-based services on their mobile computing devices (MCDs) with high data transfer rates and low latency. MCD's frequently experience situations where they are either overwhelmed with excessive resource demands or insufficiently used due to imbalanced requests for resources. Offloading strategies play a crucial role in optimizing the efficiency of real-time data processing and analysis. This study proposes an ETOSP: Energy-efficient Task Offloading Strategy based on Partial Offloading in Mobile Edge Computing framework for efficient resource management as a solution to address the aforementioned difficulty. The application of the genetic algorithm is employed to produce strategies that provide balanced resource allocation for the purpose of identifying the most optimum offloading approach. The performance evaluation of ETOSP is shown through simulated experiments.

Keywords: cloud computing; offloading types; mobile devices; delay; energy consumption; mobile edge computing

ARTICLE INFO

Received: 31 August 2023
Accepted: 3 November 2023
Available online: 27 December 2023

1. Introduction

The growing need for apps and services for mobile consumers is made possible by the exponential rise in the technological appraisal of mobile devices. Every day, more and more people are using their mobile devices, and with that comes a greater demand for high data rates and high-quality services[1]. The rapid development of mobile computing devices (MCDs) such as smartphones, laptops, and tablets has led to the development of an abundance of in-demand services and applications. In spite of the improvements in mobile devices' CPUs, they may not be able to run apps that require a great deal of processing power in a short time. Furthermore, consumers still face a substantial barrier to the complete delight of demanding applications due to high battery usage. Due to this, the idea of cloud computing (CC) has been developed for mobile phone users. It is the on-demand availability of data storage and computational power in a computer system that does not require the user to actively manage these resources. EC involves storing and processing data on internet-based servers, not local servers or personal computers, as a next step after cloud computing. For faster responses, lower latency rates, and easier maintenance in computing, CC must be upgraded to take advantage of 5G wireless technology.

The concept of EC involves the processing of data in close proximity to its source, hence facilitating faster and more extensive data processing capabilities. This, in turn, enables the generation of real-time outcomes that are driven by immediate actions. The integration of centralized and distributed architectures is facilitated by Edge. CC and EC collaborate synergistically to provide novel and enhanced user experiences. Data is created or gathered from many sources and afterwards sent to the cloud, where computing is consolidated. This centralization facilitates the processing of data in a unified and cost-effective manner, enabling the aggregation of data at a large scale. EC leverages data created at the local level to provide instantaneous response, hence facilitating the development of novel user experiences. The utilization of EC technology results in a reduction in latency, hence diminishing reaction time. As a result, computation is done close to the data source, rather than sending data to a distant cloud infrastructure and waiting for it to respond.

The goal of EC is to do real-time, low-latency analysis on data close to where it is stored rather than sending it to a centralised data centre. By allowing mobile users to utilise resources and applications locally, EC provides a distributed framework that diminishes the burden on the central server while also improving reliability and reducing latency. EC has a few drawbacks, such as the fact that information is important to the success of any enterprise. Data and information gathered at the edge server must undergo stringent compliance and regulatory checks. EC's networked architecture makes it more vulnerable to previously identified threats. A system with certain weaknesses can easily be infiltrated by malware. The integration of blockchain and EC is intended to mitigate the risk of data loss, leveraging the promising decentralised nature of blockchain technology. More computers are needed as more data is protected at the edge. Therefore, more bandwidth is required. EC requires balanced network bandwidth in order to function properly. A newly formed industry specification group (ISG) within the European Telecommunications Standards Institute (ETSI) is developing a proposal to eliminate these drawbacks in the EC by incorporating the EC into the mobile network design. ETSI proposes a solution they call MEC. The ISG MEC group's primary objective is to facilitate the smooth and effective incorporation of cloud computing features into mobile networks, as well as to support the creation of beneficial conditions for all parties involved (vendors, service providers, mobile operators, and end users) as shown in **Figure 1**[2–4].

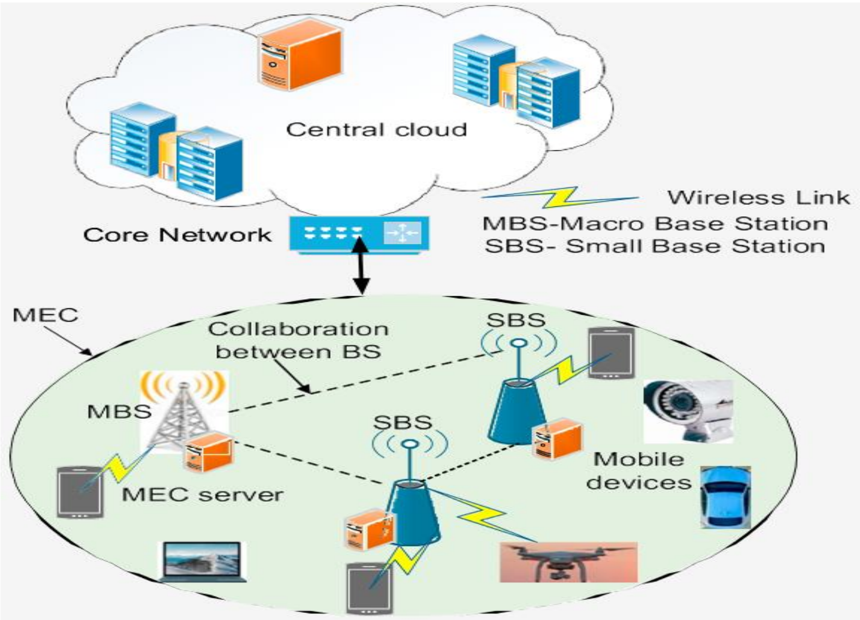

Figure 1. Overview of MEC architecture.

Next-generation wireless and computing networks will likely implement MEC[5]. Mainframe approaches suffer from latency and network congestion due to big data and long transmission durations, which are exacerbated by 5G/6G networks and the Internet of Things. By locating processing, transmission, and data storage closer to end users, MEC improves performance with lower latency and overhead costs while also enhancing the user's skills. Offloading strategies are utilised in MEC locations to increase mobile app reading by shifting the load to contiguous servers. Offloading mechanisms typically involve running resource functionality scripts locally in MEC. The purpose of offloading is to decrease the workload and cut down on computational overhead. Both MDs and MECs are utilised to manage the offloading architecture of computations[6].

The paper is systematized into six sections. In Section 1, the role of offloading in MEC has been discussed. Section 2 presents a review of the literature. Further, in Section 3, discussed about problem formulation and, in Section 4, working mechanism of proposed algorithm has been presented. In Section 5, empirical analysis has been depicted; at last, in Section 6, conclusions and future research directions have been presented.

The major contributions of the paper are as follows:

1) A MEC framework built on optimization technique that ensures data integrity during computation offloading is the implementation of Energy- efficient Task Offloading Strategy based on Partial Offloading (ETOSP) in 5G networks. An optimized resource utilization strategy is determined by the Genetic Algorithm (GA).
2) The proposed resource management method ETOSP has been tested extensively through simulations and experiments.

2. Related work

The future iteration of wireless communications, known as 5G, is anticipated to offer enhanced data transfer speeds, reduced latency, and improved quality of service for users[7]. Several emerging technologies are being considered and studied for integration into the 5G architecture with the aim of enhancing user Quality of Service (QoS)[8,9]. To enhance the overall performance of a network, Cho et al. developed a novel cross-layer architecture incorporating software defined radio and software defined networking capabilities[10]. With the projected commercial availability of 5G technology in 2020, it is anticipated that we will soon have the opportunity to witness and benefit from the convenience that 5G offers[11]. The integration of EC with 5G networks is employed to enhance the quality of service (QoS) experienced by consumers, as the computational capabilities of mobile devices are constrained by restricted resources[12]. Several significant topics are addressed by Mao et al.[13]. These include EC nodes, cache-enabled MEC, management of dynamic MEC systems, and privacy protection mechanisms. Rimal and colleagues[14] conducted a study to explore the potential of enabling combined wireless access networks for the purpose of provisioning MEC computing capability. Fajardo et al. have developed a method that addresses the issue of restricted network connectivity in cells, by supplying processing and storage capabilities in closer proximity[15]. This approach aims to enable the provision of services in cells that are now unable to offer them. The optimization of the offloading procedure is currently being investigated. Zhang et al.[16] devised a framework that integrates the utilization of different network accesses within the context of 5G networks. The use of this strategy results in a significant reduction in energy usage during the offloading process. The decentralized computation migrating method described by Chen et al.[17] seeks to achieve a Nash equilibrium in order to pick the most optimum solutions for decentralized computation migration. Indeed, the Mobile Edge Computing (MEC) paradigm does include several limitations, including the potential for data loss or privacy breaches during the offloading procedure. The blockchain technology, renowned for its robust security features, is being proposed as a prospective remedy to tackle prevailing security concerns. Joshi and colleagues[18] did a comprehensive assessment on blockchain technology, focusing specifically on several consensus methods. In this, Zheng et al.[19] provided a comprehensive overview of many conventional blockchain consensus-achieving methods and presented an extensive array of applications pertaining to blockchain technology. The development of blockchain technology has been progressing fast, and significant advancements have also been made in integrating blockchain with MEC technology. Xiong et al.[20] conducted a study on that investigate an effective resource management approach to reduce costs associated with handling mobile blockchain applications for EC. A proposed solution is introduced in this study, which utilizes blockchain technology to effectively manage computer resources while ensuring privacy protection. IoT has given rise to a novel and substantial technological domain that is confronted with increasing consumer expectations for internet connectivity technologies[21,22]. Roy et al. studied about data routing technique to address the issue of energy consumption in IoT. Their proposed approach involves transmitting data to base stations using a mobile data collector, with the aim of minimizing power consumption and reducing the time taken up in the sensing area[23]. Salameh et al. conducted a study on a unique probabilistic-based channel assignment method in order to mitigate the adverse effects caused by transmission delay limitations, as discussed in reference[24]. However, both methodologies failed to provide a comprehensive analysis of both energy usage and transmission delay. Dolui and Datta[25] conducted an in-depth review and examination of various features related to emerging paradigms such as fog computing, cloudlet computing, and MEC in the domain of cloudlet and MEC. In order to mitigate the drawbacks of overloaded MEC system, Satria introduced two distinct recovery techniques[26]. One proposed approach involves the redistribution of tasks from an excessively burdened MCD to nearby devices located within its transmission range. An alternative approach involves utilizing surrounding user devices of the MCD that are connected to the overloaded MCD. Yu et al.[27] proposed a scalable and dynamic load balancer (SDLB) that utilizes minimum hashing as the main technique for task migration destination determination. The rising prevalence of resource-intensive apps in MEC presents significant safety concerns, as the differentiation between legitimate users and potential attackers becomes increasingly difficult. The utilization of blockchain technology serves to enhance data security and mitigate the risk of unwanted alterations. Furthermore, the consideration of vehicle reputations is crucial to the preservation of data sharing quality. In order to ensure the confidentiality of users, use of blockchain technology is employed to establish a decentralized and secure video system with peer-to-peer access[28].

However, there are only a few studies that provide a thorough examination of several factors such as data security, offloading process time cost, MCD power consumption, and load balancing. Developing an efficient offloading approach that addresses the aforementioned objectives, particularly data security and privacy protection, remains a substantial undertaking.

3. System model

We apply a GA-based computation offloading model to the MEC architecture. In our proposal, we suggest minimizing parameters such as time, energy, delay, and payment cost. It is well known that task-offloading optimization is a challenging problem. Various objectives are taken into account when choosing whether to process a task locally or at the edge in order to minimize the overall offloading costs. We use ETOSP to formulate the task offloading problem in order to optimize time, energy, delay, and cost.

3.1. Energy consumption model

In order to get precise estimations of time consumption in the context of integrated blockchain-based MEC system, a time consumption model has provided that contains two distinct components. The first component of the model is the offloading time model, whereas the second component is the task-based queuing model. The improvement in offloading time has resulted in an enhanced Quality of Experience (QoE) for users of mobile devices. Hence, it is important to take into account the duration required for transferring data from smart devices to edge computing devices (ECDs).

Consider the binary variable B_m, which finds whether computing work is transferred to cloud platform.

$$B_m(t) = \begin{Bmatrix} 1, if\, j_m is\ migrated\ to\ ECD \\ 0, \quad otherwise \end{Bmatrix} \tag{1}$$

The Energy consumption of offloading, j_m, is calculated by

$$E_m^{off}(t) = B_m(t).\frac{S_m}{\beta_z}.ZD_m^z \tag{2}$$

where S_m is size of data for j_m and β_z is the rate of data transmission between two data transmitters in ec_z. Furthermore, ZD_m^z denotes the shortest path The offloading time of j_m to the cloud is calculated by:

$$E_m^{off'}(t) = \left(1 - B_m(t)\right).\frac{D_m}{\omega} \tag{3}$$

where ω represents the speed at which data is transmitted. All tasks are offloaded at the same time using the following formula:

$$E_m^{off}(t) = \sum_{m=1}^{M} T_m^{off}(t) + \sum_{m=1}^{M} T_m^{off'}(t) \tag{4}$$

3.2. Execution delay model

In the context of offloading computational duties to edge nodes (ENs), it is essential to evaluate the transmission latency involved in various stages. UEs, DUs, and ENs are all considered here, as well as the latency between ENs and DUs. Once the migration method has been selected, it is necessary to take into account the delay associated with block generation and verification. However, many other factors must be considered, such as the amount of data to be sent, the bandwidth available on the wireless channel, and how many CPU cycles each byte of received data will take. Additionally, the processing capabilities of edge servers (ESs) and mobile devices (MDs) must be considered. Mobile n's requests are sent or received using bandwidth allocated for BD_n^S and BD_n^R, respectively. D_{uv}^S and D_{uv}^R can be defined as:

$$D_{uv}^S = \frac{d_{uv}^S}{BD_n^S} \tag{5}$$

$$D_{uv}^R = \frac{d_{uv}^R}{BD_n^R} \tag{6}$$

Assuming d_{uv}^S and d_{uv}^R represent the vth job of mobile n sends and receives data of size ds and dv. As an alternative way of expressing latency at the edge, the following can be used:

$$D_{uv}^{server} = \frac{d_{uv}^S N_{uv}^c}{f^{server}} \tag{7}$$

Each bit of received data is processed at ES at a rate of f^{server} and each cycle is processed at N. As a result, the total latency of execution offloading can be calculated using the following equation:

$$D_{uv}^{offload} = D_{uv}^S + D_{uv}^R + D_{uv}^{server} + D^{(wait)} \tag{8}$$

where D_{wait} is the total waiting time for the request. For estimating delay for local execution, use the following equation:

$$D_{uv}^{local} = \frac{d_{uv}^S N_{uv}^c}{f^{local}} + D^{(wait)} \tag{9}$$

where f^{local} is the processing rate of local MD.

4. Proposed algorithm

The flow diagram depicted in the **Figure 2**, that illustrates the procedural steps of the proposed algorithm, namely ETOSP. Firstly, the identification of spare space in the EN system is conducted. The method takes as input the set of offloading transactions, O_t, is current time of offloading. It then outputs the identification of the spare space, EN. Initially, the acquisition of the offloading transaction is performed for each block. Subsequently, the initiation and duration of activities are identified and revised in order to assess the continued occupancy of the virtual machine by these tasks. Based on the current task processing condition, the EN spare space S has been achieved. The challenge of computation migration is formally stated in our methodology as a means to achieve a balance between the transmission delay and load distribution of ENs.

To begin, the offloading approach for each job is encoded in order to facilitate subsequent selection. In our system, we encode each offloading strategy as a gene, and each job is associated with just one offloading strategy. Every gene is represented by an array of numbers, which consists of the DUs (Device Units) and ENs (Edge Nodes) to which the job is assigned for offloading. At a given time point t, the collective set of genes constitutes the chromosome, which serves as a representation of the migratory techniques employed by computing jobs at that specific time point t. In order to assess the optimality of an offloading approach within our methodology, we use fitness functions that incorporate both the transmission delay of offloaded tasks and the degree of load balancing among all deployed ENs.

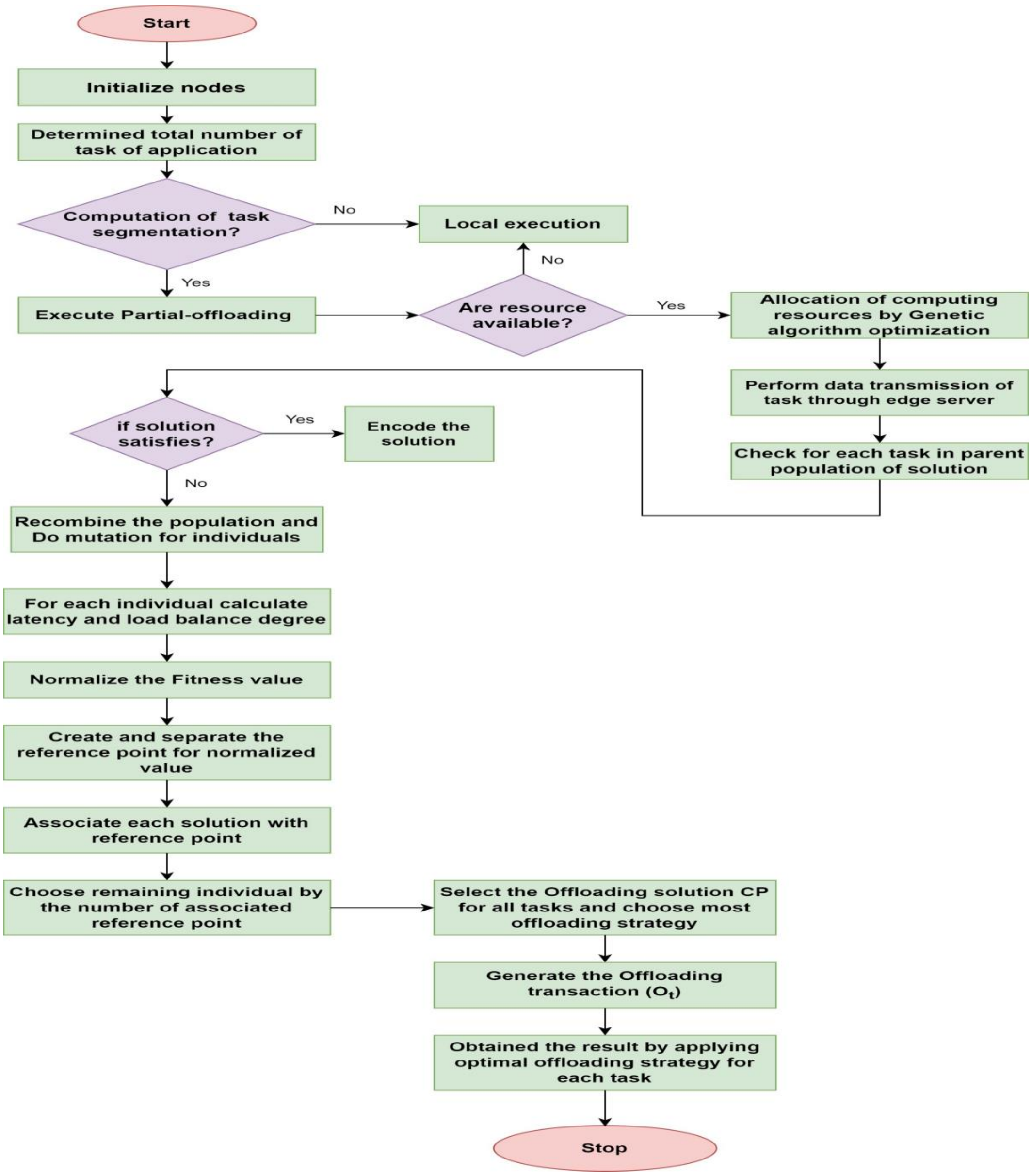

Figure 2. Workflow of ETOSP algorithm.

The fitness values are computed in order to conduct sorting among the genes within the initial population. The parent population is composed of solutions with the greatest fitness values after undergoing the sorting process. Subsequently, the recombination and mutation techniques are implemented to facilitate subsequent selections.

The single-point recombination procedure involves the generation of new chromosome pairs by using the original chromosome pairs from the parent population. Subsequently, a pair of chromosomes is formed in the child. Furthermore, the mutation process is executed with the objective of attaining optimal fitness values. The mutation procedure involves the random modification of a single gene inside the chromosome, resulting in the generation of a new person.

Until the population size reaches H, an individual is chosen sequentially from the highest nondominant front. The persons in the last front may not be entirely picked, and the solutions in the final front are afterwards chosen.

The first step in facilitating additional selection processes is to normalise fitness values within the population. We want to find the 2H people with the lowest latency and load balance degree in the parent population. Individuals whose fitness levels have been normalised are linked to reference sites.

The solutions in the final nondominant front are ranked based on the number of reference points associated with them. The solution exhibiting the highest number of interconnected reference points is chosen and incorporated into the created population. The method remains active until the population size of the offspring reaches the value of H.

This research presents a novel approach for enhancing transmission delay reduction and load balancing efficiency through the utilization of an optimization-based resource management mechanism. Initially, when a collection of activities necessitates offloading, the offloading transaction is primarily retrieved.

Based on the current state of ENs, it is anticipated that the redistribution and offloading of jobs to new ENs would be implemented in order to achieve load balance. Following the completion of the offloading procedure, the block information undergoes another change in preparation for the subsequent offloading of tasks. The resulting output consists of the most optimum offloading method, denoted as U_{max}, for each individual job. Whenever the tasks need offloading, the available capacity in the execution nodes (ENs) is initially determined.

Experimental setup

This section presents the experimental data that evaluate the performance of the suggested technique, ETOSP. The experimental setting is first detailed with great precision. In order to assess the efficacy of our suggested strategy, we provide numerous comparing methodologies. Lastly, the performance assessment and comparative analyses are given. In our simulation, there exists a multitude of User Equipment (UEs) that delegate duties to Edge Nodes (ENs). In this part, we apply numerous comparison methodologies to assess and demonstrate the superiority of our suggested method, ETOSP. The concept of combinatorial ordering refers to the arrangement or sequencing of elements from a given set in a certain order. It involves non-cooperative game model based on sub-gradient, also known as NCGG. The transmitting power of each mobile user in U_n is initially set to its maximum level at the beginning. The game process will repeat until the users' transmitting power reaches equilibrium, and the game model will reach equilibrium. During each iteration of the process, mobile users update the transmitting power of their devices to minimize their energy consumption. Users of mobile phones need to offload delay-sensitive requests to macro-BSs or BSs after allocating their transmitting power. A technique for dealing with this type of problem is called joint request offloading and computing resource scheduling (JRORS). As a result, this problem can be divided into two parts: request offloading (RO) and computing resource scheduling (RS). Particle swarm optimization (PSO) is used to effectively solve this problem. Inspired by the social behaviour of birds and fish, PSO is a technique of computational optimization. A natural path or location can be found by imitation of how these organisms cooperate and communicate. PSO is a population-based optimization algorithm. It starts with a population of potential solutions, called particles, which move through the search space. Each particle adjusts its position based on its own experience and the experiences of its neighbours. But it may converge to a local optimum and can be sensitive to parameter settings. The ETOSP based on GA is an effective way to solve the local optimum problem of JRORS based on PSO.

5. Performance evaluation of (ETOSP)

We used Matlab R-2023b to implement ETOSP in order to evaluate the effectiveness of the proposed algorithms. The simulations were run on an Intel i5, 3.7GHz PC with 8 GB RAM. We consider a multi-zone edge computing system with mobile users, multiple BSs, and a macro-BS. Each BS is outfitted with an edge server and covers a zone. The parameters of the simulation are shown in **Table 1**.

Table 1. Parameter settings.

Parameter	Value
Number of mobile users U	{12, 20, 32, 40, 52, 60, 72, 80, 92, 100}
Number of BSs N	{3, 5, 8, 10, 13, 15, 18, 20, 23, 25}
The number of VM instances in an EN	100
The transmission rate from UE to DU	1000 Mbps
The transmission rate from DU to EN	500 Mbps
The bandwidth between ENs	2000 Mbps
The number of ENs	60
The number of tasks G	200, 400, 600, 800, 1000 and 2000

The present study addresses a multi-objective optimization issue that focuses on minimizing delay and achieving a balanced load distribution. The offloading technique that is considered to have the highest utility value is deemed to be the most optimum option. This section presents a comparative analysis of the performances of JRORS and ETOSP.

5.1. Comparative analysis on energy consumption

We assess energy consumption under various numbers of mobile users. It is discovered that energy consumption is proportional to maximum power p_{max}, i.e., the higher the p_{max}, the higher the energy consumption. This is because the average transmitting power of mobile users is higher when the maximum power p_{max} is larger, resulting in more energy consumption. **Figure 3** depicts the JRORS algo based on PSO performing an energy usage analysis.

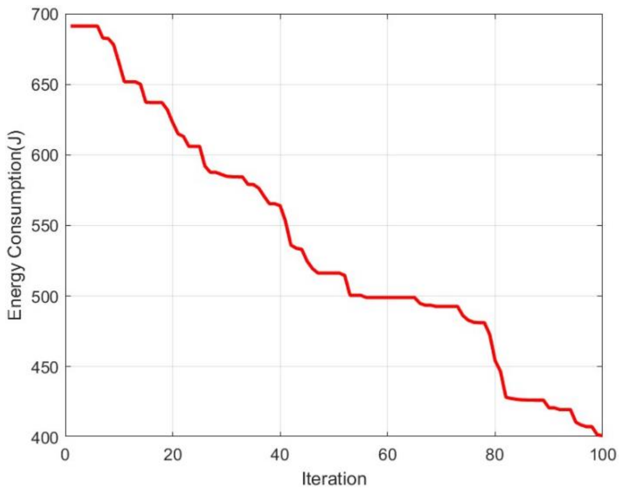

Figure 3. Computation of energy consumption using JRORS algo based on PSO.

As shown in **Figure 4**, our proposed approach i.e., ETOSP algo based on GA approach. The data indicates that ETOSP generates a lower amount of energy compared to other approaches. This may be attributed to ETOSP's utilization of a smaller number of ECDs and its reduced wastage of idle VM instances. These factors contribute significantly to the overall energy consumption in ECDs. Moreover, when the work scale expands, there is an accompanying rise in energy consumption. This suggests that ETOSP may be able to mitigate its limitations when it comes to handling huge computational job sizes.

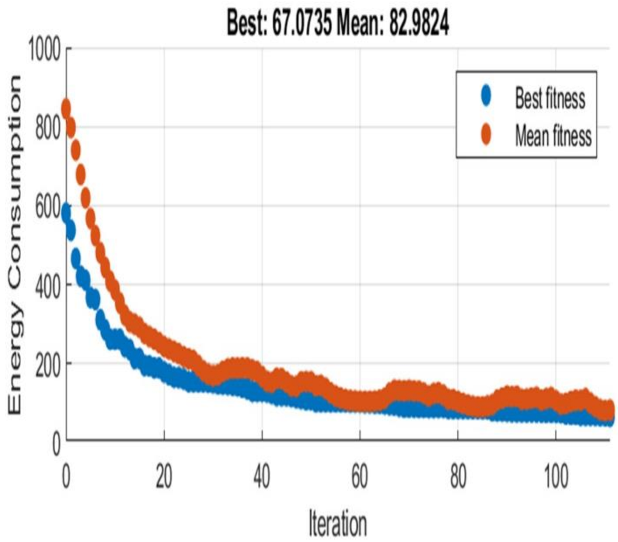

Figure 4. Computation of Energy Consumption using ETOSP algo based on GA approach.

Figure 5 shows a comparison of the existing approach and our suggested strategy. It can be seen that as the number of mobile users increases, so does energy consumption. **Figure 5** shows how ETOSP outperforms the other compared approach when the number of mobile users is varied. It should be noted that ETOSP can achieve lower energy consumption even when a large number of mobile users are present, demonstrating ETOSP's extensibility.

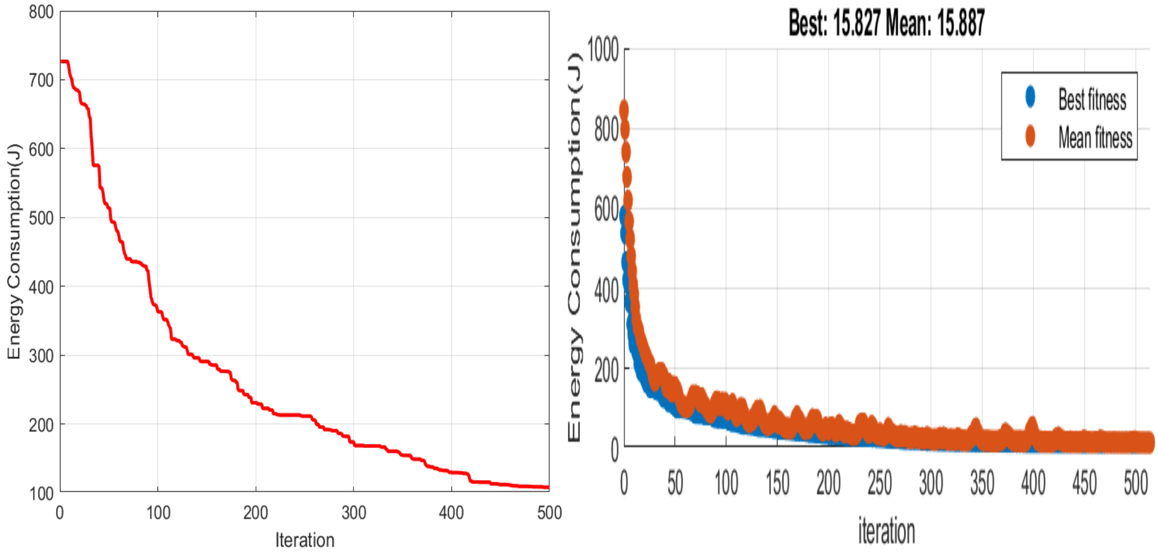

Figure 5. Comparative analysis on energy consumption.

5.2. Comparative analysis on execution delay

We test the execution delay with a variety of mobile users. The load balancing rate indicates the offloading techniques' unload status. In this section, we compare the proposed ETOSP algorithm to the JRORS baseline algorithm. ETOSP represents only cloud computing and MEC resources; based on partial offloading that indicates selected modules from the set of modules are executed on the UEs. Because execution delay is the optimization target. The performance of execution delay using baseline algorithm i.e., JRORS based on PSO under different number of users is shown in **Figure 6**.

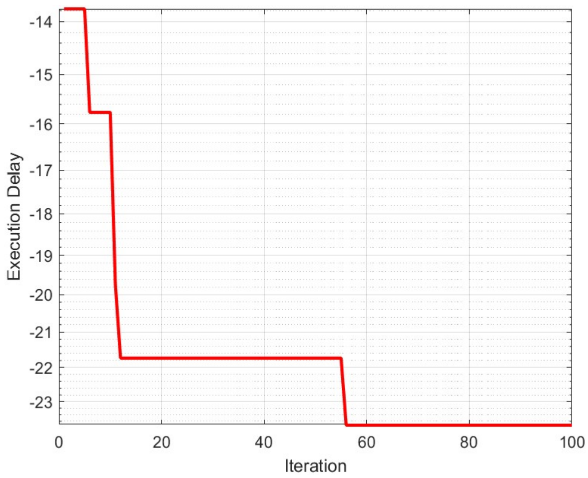

Figure 6. Computation of Execution Delay using JRORS algo based on PSO.

Therefore, **Figure 7** presents a ETOSP algorithm based on GA approach. It may be inferred that ETOSP demonstrates improved load balancing efficiency in situations where the ECDs are experiencing underutilization. The total execution delay of ETOSP is reduced as compared to baseline algorithm.

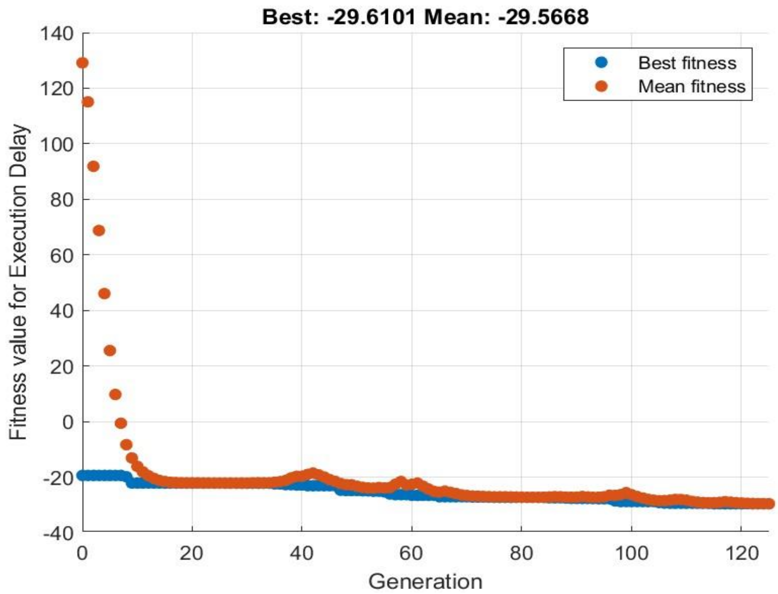

Figure 7. Computation of Execution Delay using ETOSP algo based on GA approach.

We compare the JRORS and ETOSP's performance under different number of UEs. The total execution delay can be reduced more than the baseline algorithm. Thus, the performance of the proposed algorithm is always better than the baseline algorithm.

In summary, **Figure 8** illustrates ETOSP provides better performance than baseline algorithm and is able to provide a better performance by the full utilization of MEC resources.

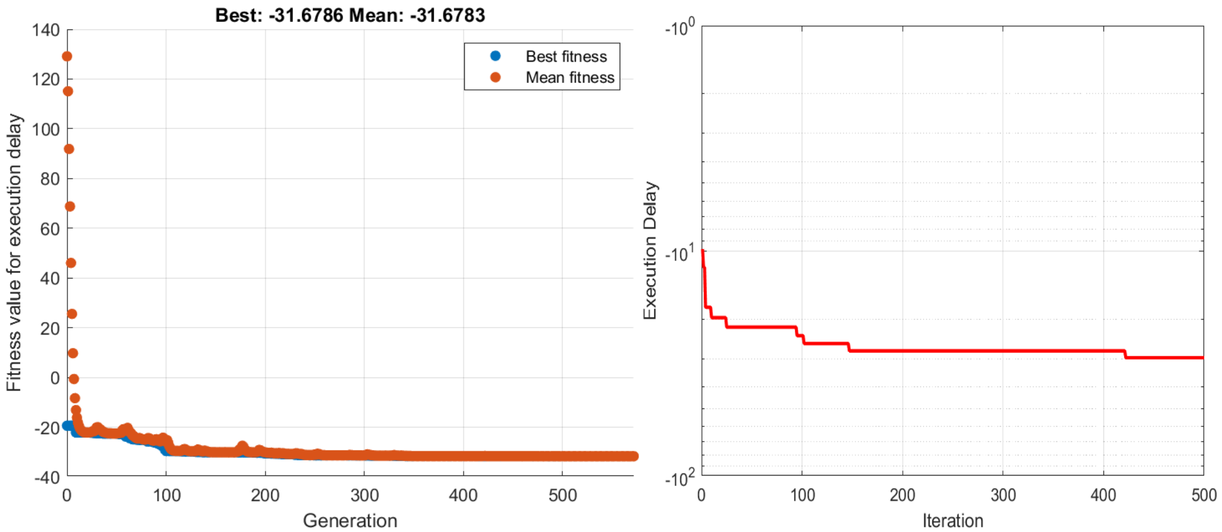

Figure 8. Comparative analysis on Execution Delay.

6. Conclusion and future research

The potential for supporting upcoming 5G applications is promising through the combination of 5G and MEC, as a result of the hardware limitations and computational capacity constraints of UEs. By transferring computational workloads from user equipment (UEs) to edge nodes (ENs), enough computing resources are made available to UEs, therefore overcoming the constraints imposed by limited resources. Due to the uneven distribution of resources, the operational performance of entrepreneurial ventures is rarely ensured. In addition, the transmission latency and data loss incurred during the process of compute offloading have a significant impact on the quality of service experienced by the user. Given the above-mentioned problem, we suggest a computational migration approach called Energy-efficient Task Offloading Strategy based on Partial Offloading (ETOSP). The utilization of Genetic Algorithms (GA) is implemented in order to get the optimal and well-balanced offloading techniques. Ultimately, the feasibility and efficiency of ETOSP are validated by the execution of systematic experiments and comparative assessments.

In the future, the proposed ETOSP will be modified to align with the specific demands of real-world situations. Additionally, the suggested solution takes into consideration a wider range of user Quality of Service (QoS) preferences in order to enhance its practicality and efficiency.

Author contributions

Conceptualization, AS and CD; methodology, AS; software, AS; validation, AS and CD; formal analysis, CD; investigation, AS; resources, AS; data curation, AS; writing—original draft preparation, AS; writing—review and editing, AS; visualization, CD; supervision, AS; project administration, AS; funding acquisition, CD. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Casado-Vara R, Sittón-Candanedo I, De la Prieta F, et al. Edge Computing and Adaptive Fault-Tolerant Tracking Control Algorithm for Smart Buildings: A Case Study. Cybernetics and Systems. 2020, 51(7): 685-697. doi: 10.1080/01969722.2020.1798643
2. Huang PQ, Wang Y, Wang K, et al. A Bilevel Optimization Approach for Joint Offloading Decision and Resource Allocation in Cooperative Mobile Edge Computing. IEEE Transactions on Cybernetics. 2020, 50(10): 4228-4241. doi: 10.1109/tcyb.2019.2916728

3. Ebrahim Pourian R, Fartash M, Akbari Torkestani J. A New Approach to the Resource Allocation Problem in Fog Computing Based on Learning Automata. Cybernetics and Systems. Published online November 23, 2022: 1-20. doi: 10.1080/01969722.2022.2145653
4. Sabella D, Vaillant A, Kuure P, et al. Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things. IEEE Consumer Electronics Magazine. 2016, 5(4): 84-91. doi: 10.1109/mce.2016.2590118
5. Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. IEEE Communications Surveys & Tutorials. 2017, 19(3): 1628-1656. doi: 10.1109/comst.2017.2682318
6. Carella GA, Pauls M, Magedanz T, et al. Prototyping nfv-based multi-access edge computing in 5G ready networks with open baton. 2017 IEEE Conference on Network Softwarization (NetSoft). Published online July 2017. doi: 10.1109/netsoft.2017.8004237
7. Xiong Z, Zhang Y, Niyato D, et al. When Mobile Blockchain Meets Edge Computing. IEEE Communications Magazine. 2018, 56(8): 33-39. doi: 10.1109/mcom.2018.1701095
8. Wang CX, Haider F, Gao X, et al. Cellular architecture and key technologies for 5G wireless communication networks. IEEE Communications Magazine. 2014, 52(2): 122-130. doi: 10.1109/mcom.2014.6736752
9. Rappaport TS, Sun S, Mayzus R, et al. Millimeter Wave Mobile Communications for 5G Cellular: It Will Work! IEEE Access. 2013, 1: 335-349. doi: 10.1109/access.2013.2260813
10. Hsin-Hung Cho, Chin-Feng Lai, Shih TK, et al. Integration of SDR and SDN for 5G. IEEE Access. 2014, 2: 1196-1204. doi: 10.1109/access.2014.2357435
11. Zaidi AA, Baldemair R, Tullberg H, et al. Waveform and Numerology to Support 5G Services and Requirements. IEEE Communications Magazine. 2016, 54(11): 90-98. doi: 10.1109/mcom.2016.1600336cm
12. Hu YC, Patel M, Sabella D, et al. Mobile Edge Computing—A Key Technology Towards 5G. ETSI white paper no. 11. Sophia Antipolis, France: European Telecommunications Standards Institute; 2015.
13. Mao Y, You C, Zhang J, et al. A Survey on Mobile Edge Computing: The Communication Perspective. IEEE Communications Surveys & Tutorials. 2017, 19(4): 2322-2358. doi: 10.1109/comst.2017.2745201
14. Rimal BP, Van DP, Maier M. Mobile Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era. IEEE Communications Magazine. 2017, 55(2): 192-200. doi: 10.1109/mcom.2017.1600156cm
15. Fajardo JO, Liberal F, Giannoulakis I, et al. Introducing Mobile Edge Computing Capabilities through Distributed 5G Cloud Enabled Small Cells. Mobile Networks and Applications. 2016, 21(4): 564-574. doi: 10.1007/s11036-016-0752-2
16. Zhang K, Mao Y, Leng S, et al. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks. IEEE Access. 2016, 4: 5896-5907. doi: 10.1109/access.2016.2597169
17. Chen X, Jiao L, Li W, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. IEEE/ACM Transactions on Networking. 2016, 24(5): 2795-2808. doi: 10.1109/tnet.2015.2487344
18. Prashanth Joshi A, Han M, Wang Y. A survey on security and privacy issues of blockchain technology. Mathematical Foundations of Computing. 2018, 1(2): 121-147. doi: 10.3934/mfc.2018007
19. Zheng Z, Xie S, Dai H-N, Wang H. Blockchain Challenges and Opportunities: A Survey. Working paper. 2016.
20. Xiong Z, Feng S, Niyato D, et al. Optimal Pricing-Based Edge Computing Resource Management in Mobile Blockchain. 2018 IEEE International Conference on Communications (ICC). Published online May 2018. doi: 10.1109/icc.2018.8422517
21. Sarkar S, Chatterjee S, Misra S. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. IEEE Transactions on Cloud Computing. 2018, 6(1): 46-59. doi: 10.1109/tcc.2015.2485206
22. Wang X, Yang LT, Li H, et al. NQA. ACM Transactions on Embedded Computing Systems. 2019, 18(4): 1-21. doi: 10.1145/3330139
23. Roy SS, Puthal D, Sharma S, et al. Building a Sustainable Internet of Things: Energy-Efficient Routing Using Low-Power Sensors Will Meet the Need. IEEE Consumer Electronics Magazine. 2018, 7(2): 42-49. doi: 10.1109/mce.2017.2776462
24. Bany Salameh HA, Almajali S, Ayyash M, et al. Spectrum Assignment in Cognitive Radio Networks for Internet-of-Things Delay-Sensitive Applications Under Jamming Attacks. IEEE Internet of Things Journal. 2018, 5(3): 1904-1913. doi: 10.1109/jiot.2018.2817339
25. Dolui K, Datta SK. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. 2017 Global Internet of Things Summit (GIoTS). Published online June 2017. doi: 10.1109/giots.2017.8016213
26. Satria D, Park D, Jo M. Recovery for overloaded mobile edge computing. Future Generation Computer Systems. 2017, 70: 138-147. doi: 10.1016/j.future.2016.06.024
27. Yu Y, Li X, Qian C. Sdlb: A scalable and dynamic software load balancer for fog and mobile edge computing, in: Proceedings of the Workshop on Mobile Edge Communications. ACM; 2017. pp. 55–60.
28. Liu M, Yu FR, Teng Y, et al. Distributed Resource Allocation in Blockchain-Based Video Streaming Systems with Mobile Edge Computing. IEEE Transactions on Wireless Communications. 2019, 18(1): 695-708. doi: 10.1109/twc.2018.2885266