Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

An Energy Efficient And Deadline Constrained Heuristic Algorithm For Dynamic Offloading Of Tasks In Mobile Cloud Computing Environment

Ramesh Babu A[1] , Dr. Niraj Upadhayaya[2]

[1]Associate Professor, Department of CSE, J.B. Institute of Engineering and Technology, Moinabad, Hyderabad-75.

[2]Professor (CSE) & Dean (R&D), J.B. Institute of Engineering and Technology, Moinabad, Hyderabad-75.

Abstract

In MCC environments, mobile applications run based on battery power of smart hand held devices. It the presence of constrained resources, dynamic offloading of tasks to public cloud is essential for mobile applications to meet task deadlines and enhance performance of the system. When all the tasks are executed locally, it leads to high resource consumption and inability in meeting deadlines. Similarly, when all the tasks are offloaded to cloud, it results in more execution time which may sometimes even lead to missing deadlines. Therefore, it is essential to strike balance between with optimal offloading decisions selectively. Towards this end, in this paper, we proposed an algorithm named Energy Efficient and Deadline Constrained Heuristic Algorithm for Dynamic Offloading (EEDCHA-DO). It considers the problem of deadline constrained tasks offloaded to cloud based on execution time and energy consumption in presence of varied bandwidth of wireless network. The algorithm performs desired computations dynamically for each task and the problem of whether to offload or not to offload the task is solved optimally. The algorithm is implemented in MATLAB and the simulation results revealed that EEDCHA-DO is efficient in task offloading when compared with its predecessors such as LAC and LC. The proposed algorithm achieved minimal energy consumption while meeting the deadline of tasks with its optimized offloading decisions.

Keywords – MCC, cloud computing, computational offloading, local execution, cloud execution

1. INTRODUCTION

Computation offloading is the process used in MCC where some of the tasks are offloaded to remote cloud for execution. This has potential to save resources of mobile devices while reaping benefits of resource-rich cloud infrastructure at the same time [1]. When all the tasks

are executed locally, it leads to high resource consumption and inability in meeting deadlines. Similarly, when all the tasks are offloaded to cloud, it results in more execution time which may sometimes even lead to missing deadlines. Therefore, it is essential to strike balance between with optimal offloading decisions selectively. The goal of any offloading algorithm is to reduce energy consumption and meet deadline constraint associated with given task [3]. Many existing algorithms in [1], [6], [8], [9] and [14] followed game theoretic approach to solve the problem. They made a potential game model that works in multi-user environment for computational offloading dynamically. They made experiments with increased number of users and found the method to be scalable.

The concept of decentralized computation of data offloading is explored in [5], [8], [9] and [14]. In [5], their approach is known as selfish decentralized computation offloading where selfish mobile nodes are involved in a dense network in offloading computations through base station of many access points. They proposed an algorithm known as Join and Play Best Replies (JPBR) that plays it role in the game of computational offloading. In [8], they proposed a data offloading technique that is decentralized in nature based on game theory. It is the game involved in the system model, offloading and congestion game, which reduces congestion and improves performance of the system by offloading appropriately. In [9] also game based offloading approach is proposed in MCC to improve capabilities of mobile devices. In [14], decentralized algorithms are designed to achieve joint management of computing and wireless resources for offloading in MCC. In [20], they investigated different offloading mechanisms with a stochastic perspective covering MCC and edge computing scenarios. Their study involved different variants of Markov processes that are widely used in computational offloading approaches. From the literature, it is found that different techniques came into existence. However, this paper focuses on an energy efficient and deadline constrained heuristic algorithm for dynamic offloading of tasks in mobile cloud computing environment. Our contributions in this paper are as follows.

1. A system model is defined and problem formulation is made to arrive at the required solution.
2. An algorithm named Energy Efficient and Deadline Constrained Heuristic Algorithm for Dynamic Offloading (EEDCHA-DO) is defined to have dynamic offloading which minimizes energy consumption and meets task deadlines.
3. A prototype is implemented using MATLAB for simulation study and evaluate the proposed algorithm.

The remainder of the paper is structured as follows. Section 2 reviews state of the art on different offloading tasks. Section 3 presents the preliminaries to understand the proposed system in terms of the system model and problem formulation. Section 4 presents the algorithm design with its details. Section 5presents experimental results while section 6 concludes the paper with details on future work.

2. RELATED WORK

With regard to MCC, it is essential to exercise dynamic computation offloading for performance enhancement. Zheng et al. [1] formulated dynamic computation offloading

process as a stochastic game using a game-theoretic approach and defined an algorithm named multi-agent stochastic learning algorithm to realize the same. They could see its effectiveness under dynamic environments. Enzai and Tang [2] opined that dynamic offloading is very important for MCC and explored the process of it in general and specific approaches such as coarse-grained and fine-grained techniques for offloading. Chen et al. [3] considered a multi-user environment where multi-channel wireless interference is expected. They also used mobile-edge cloud computing for their study. They made a potential game model that works in multi-user environment for computational offloading dynamically. They made experiments with increased number of users and found the method to be scalable. In MCC settings, Kuang et al. [4] proposed an agent-based MCC framework where computation offloading takes place in presence of multiple users. They investigated on the optimization problem which is considered to be maximum energy conservation problem. They proposed an algorithm known as Dynamic Programming After Filtering (DPAF) to realize it. There is an offloading task filtering process that is based on the benefits of offloading. The algorithm is based on dynamic programming that makes decisions based on runtime situations. They intended to use cloudlet system in future for improving it.

Josilo and Dan [5] proposed an offloading phenomenon for MCC. Their approach is known as selfish decentralized computation offloading where selfish mobile nodes are involved in a dense network in offloading computations through base station of many access points. They proposed an algorithm known as Join and Play Best Replies (JPBR) that plays it role in the game of computational offloading. The convergence time for offloading is found linear to number of mobile devices. Cardellini et al. [6] proposed a game theoretic approach for realizing computational offloading in MCC. They designed a three tier architecture for MCC where mobile devices come under local tier, cloudlets come under middle tier and remote tier is made up of cloud servers in a remote place. There is dispatcher module used in middle tier to achieve the offloading process to cloud infrastructure. Shiraz and Gani [7] proposed an active service migration framework that takes care of the task offloading. Its runtime distribution platform is based on lightweight architecture. There is communication between the cloud server node and smart mobile device as per the system model used. Wireless medium is the link between the two to realize offloading process. They intended to examine intensive mobile applications with their framework in future.

Liu et al. [8] proposed a data offloading technique that is decentralized in nature based on game theory. It is the game involved in the system model, offloading and congestion game, which reduces congestion and improves performance of the system by offloading appropriately. It also supports multi-item auction based offloading that could reduce cellular traffic and leverage benefits to mobile subscribers. Chen [9] also proposed game based offloading approach in MCC to improve capabilities of mobile devices. Their mechanism includes interference measurement, decision on contention and algorithm for decentralized computation offloading. They intended to improve it by considering churn rate in future. Goudarzi et al. [10] explored challenges in offloading computations in the context of MCC. Then they proposed a mechanism known as fast hybrid multi-site computation offloading that converges better with respect to time. Alam et al. [11] focused on mobile edge or fog computing to investigate the

dynamics of offloading computations in MCC. They considered different factors such as geographical distribution of mobile nodes, heterogeneity and mobility. A Markov decision process (MDP) is modelled besides using a reinforcement learning to have optimized decision making in offloading procedures.

Enzai and Tang [12] investigated the problem of multi-site computation offloading in MCC. They defined a heuristic algorithm to have good performance in offloading. Liu et al. [13] also modelled MDP in order to have offloading computations. In the process, they proposed a hybrid algorithm for mobile data offloading based on the prediction. Misra et al. [15] followed an action based approach for optimal task offloading in MCC. It has three tier architecture and different ways in approaching particular destination such as local mobile device, cloudlet, or the cloud infrastructure for offloading. Jo et al. [16] considered the problem of video streaming using MCC environment with 5G cellular network. The burden of video enhancement is offloaded to public cloud by a mobile client. Their methodology was found to be energy efficient. Barbera et al. [17] investigated on the bandwidth and energy of mobile nodes and formulated a solution that renders decision as to whether to offload a task or not. Wu and Huang [18] proposed an offloading mechanism that considers different aspects such as risk, multi-site and multi-factor. Upadhyay [19] proposed a Service Oriented Architecture (SOA) based approach for computational offloading while Jesilo and Dan [14] defined decentralized algorithms to achieve joint management of computing and wireless resources for offloading in MCC.

Shakaramiet al. [20] investigated different offloading mechanisms with a stochastic perspective covering MCC and edge computing scenarios. Their study involved different variants of Markov processes that are widely used in computational offloading approaches. Mastoi et al. [21] considered heterogeneous MCC environments to build content and failure aware task offloading which is dynamic in nature. It investigates on the noise, signal and bandwidth while making offloading decisions. Lai et al. [22] developed a scheme known as fairness-oriented task offloading for mobile cloudlet networks. Their algorithm is named as FairEdge that measures network fairness as well. Silva et al. [23] proposed a modular and adaptive approach for computational offloading that is suitable for hybrid cloud scenarios. It exploits scheduling strategy and corresponding metrics to arrive at offloading decisions. Li et al. [24] used deep reinforcement learning approach to realize dynamic offloading. It makes use of both MDP and Deep Q Network (DQN) where fine-tuning of offloading process is made. Gao et al. [25] proposed a method known as Q-learning based task offloading that exploits resource optimisation with optimized decision making. Masdarl and Kherl [26] reviewed different Markov based models that contributed to dynamic offloading in MCC. Zaman et al. [27] also reviewed offloading models but focused on mobility-aware approaches. Meng et al. [28] investigated on security-aware dynamic scheduling approaches in different industrial applications. Ibrar et al. [29] proposed an Artificial Intelligence (AI) based framework for task offloading in an environment where there is Internet of vehicular networks. It makes use of fog computing nodes for task offloading. From the literature, it is found that different techniques came into existence. However, this paper focuses on an energy efficient and deadline

constrained heuristic algorithm for dynamic offloading of tasks in mobile cloud computing environment.

3. PRELIMINARIES

3.1 The System Model

A real word application may have some offloadable and some unoffloadable tasks or local tasks. The tasks in mobile application that directly takes user inputs and perform input and output tasks on I/O devices must be considered local and offloading such tasks is not possible. Therefore, it is essential to identify the tasks that are offloadable. We considered a mobile application that has many independent tasks that can be run either locally or they can be offloaded to public cloud for remote execution to lessen burden on mobile device.

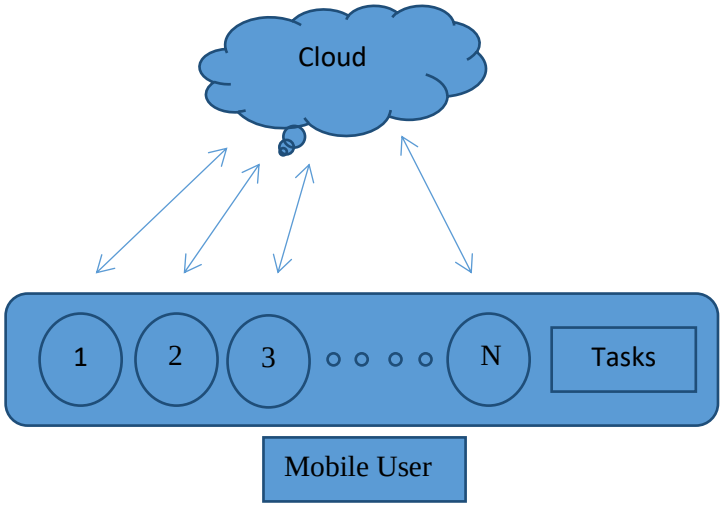

Figure 1: Illustrates the task offloading scenario

The mobile device has WiFi connectivity but network bandwidth may vary dynamically. Based on the network bandwidth and other considerations, the mobile device needs to determine whether a given task is to be run locally or offload it to cloud. This is illustrated in Figure 1. As each task has a deadline, it is essential to consider the time required to send a task from mobile device to cloud as it affect task's total execution time. Therefore, the proposed algorithm needs to consider network bandwidth as well in order to arrive at a decision.

3.2 Problem Definition

A mobile application has M number of tasks that are to be executed. M_i represents i^{th} task which needs to be executed. The expression Mi $\in$ {0, 1} indicates two possibilities with regard to task execution. The value 1 indicates local execution and 0 indicates remote execution. In either case, there is energy consumption of mobile device which is denoted as E_{li} and E_{ri} respectively. With respect to remote execution E_{ti} represents energy consumption of mobile device for transmission of task i to cloud. With regard to execution time T_{li}, T_{ri}, T_{ti} denote local execution time, remote execution time and time taken for transferring task from mobile device to cloud respectively.

Notation	Description
i	Task
M_i	Execution of task i with regard to offloading
Er_i	Energy consumption of mobile device to execute task i with given transmission rate.
Et_i	Energy cost associated with the transmission of task i.
Tl_i	Local execution time
Tr_i	Remote execution time
Tt_i	Transmission time
T	Execution time
$T_{constraint}$	Execution time requirement of given task
M	Vector with offloading decisions
El_i	Energy consumption when task is executed locally.

Table 1: Notations used in the mathematical formulations

The energy (E_{ti}) and transmission time (T_{ti}) required for transmission of task i to cloud depends on network bandwidth. Therefore, the network bandwidth changes can affect decisions pertaining to offloading. The energy consumption and execution time are computed as in Eq. 1 and Eq. 2 respectively.

$$E = \sum_{i \in N}(M_i El_i + (1 - M_i)Er_i + (1 - M_i)Et_i) \quad (1)$$

$$T = \sum_{i \in N}(M_i Tl_i + (1 - M_i)Tr_i + (1 - M_i)Tt_i) \quad (2)$$

Since each task has deadline constraint with respect to execution time, T needs to comply with $T_{constraint}$ as expressed in Eq. 3.

$$T \leq T_{constraint} \quad (3)$$

When different offloading decisions for the tasks is denoted as M=[M_1, M_2, M_3, …, Mn], the problem being solved is expressed as in Eq. 4.

$$\begin{aligned} &\min \ E \\ &\text{subject to: } T \leq T_{cnostraint} \end{aligned} \quad (4)$$

Search for optimal solution grows exponentially based on the number of tasks. The aim of the system is to determine the tasks that can be uploaded that minimize energy and execution time satisfying the expression in Eq. 4.

4. ALGORITHM DESIGN

We defined an algorithm named Energy Efficient and Deadline Constrained Heuristic Algorithm for Dynamic Offloading (EEDCHA-DO). It is based on the system model presented in Figure 1 and the problem formulation discussed in Section 3. It makes use of a table with N*N size to hold the offloading decisions made by the algorithm where N value is equal to the

number of tasks involved in the given mobile application. The table keeps track of the decisions and each one has associated energy computations and execution time computations.

Algorithm: Energy Efficient and Deadline Constrained Heuristic Algorithm for Dynamic Offloading

Inputs: Set of tasks M with deadline constraint

Output: Dynamic offloading decisions D

1. Initialize energy matrix E'
2. Initialize time matrix T'
3. Initialize task-decision map D
4. Find transmission rate r
5. For each task i in M
6. Compute energy $E = \sum_{i \in N} (M_i El_i + (1 - M_i) Er_i + (1 - M_i) Et_i)$
7. Compute execution time $T = \sum_{i \in N} (M_i Tl_i + (1 - M_i) Tr_i + (1 - M_i) Tt_i)$
8. Update E'
9. Update T'
10. IF i satisfies $\min E$ subject to: $T \leq T_{cnostraint}$ Then
11. Update D with i and 0 (offloading)
12. Else
13. Update D with i and 1 (local)
14. End If
15. End For
16. Return D

Algorithm 1: Energy efficient and deadline constrained heuristic algorithm for dynamic offloading

As presented in Algorithm 1, it takes number of tasks associated with a mobile application where each task can be offloaded based on the benefits in terms of saving energy and meeting task deadline. In presence of Service Level Agreements (SLAs), it is essential to meet deadline of task every time. Keeping this in mind $T_{cnostraint}$ is given high importance in the proposed algorithm. It is the basic condition that needs to be met and then other possibilities are explored. Each task is different in terms of its size and required time for execution and consume certain energy. Therefore, there is dynamic computation of energy required and execution time where the computations depend on bandwidth of the wireless network that is somehow reduced to a variable r denoting transmission rate. Since, the execution time and energy depend on r for offloading, it is indispensable to consider it in the computations of T and E. When a task

satisfies the minimal E constraint and $T_{cnostraint}$, that is eligible for offloading which makes the mobile applications empowered in MCC scenario.

5. EXPERIMENTAL RESULTS

The proposed algorithm EEDCHA-DO is executed and evaluated with a simulation study made in MATLAB. The total cost is expressed in terms of Joules which includes the weighted sum of total energy, the cost of cloud execution and transmission delays. This is modelled as expressed in [30] from which two existing offloading methods are used for comparison in this paper. The existing methods include LC (local-cloud) and LAC (local-access-cloud).

Transmission Rate (Mbps)	3	4	5	6	7	8	9
Total Cost of EEDCHA-DO	1650	1500	1400	1390	1380	1360	1300
Total Cost of Local Execution	1300	1310	1320	1320	1320	1340	1350
Total Cost of Offloading All	1700	1600	1450	1400	1380	1370	1300

Table 2: Total cost comparison when $T_{cnostraint}$is set to 700

As presented in Table 2, the total cost of local execution and offloading all tasks and offloading as per the proposed algorithm EEDCHA-DO against different rates of transmission are provided.

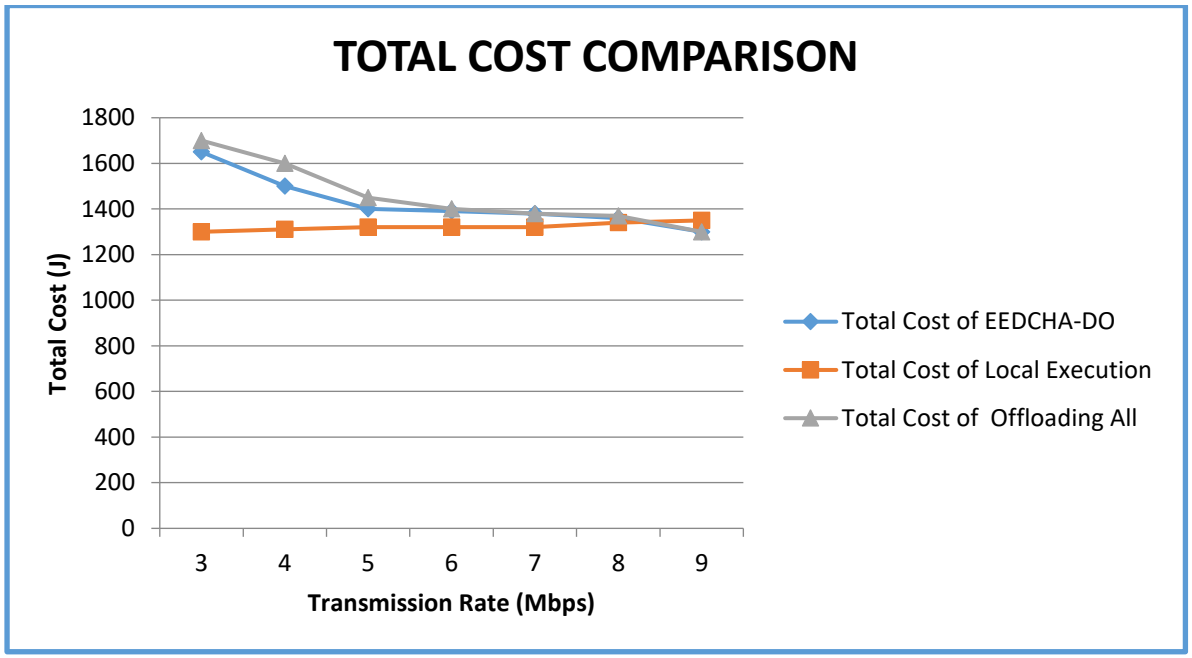

Figure 2: Total cost comparison with different rates of transmission

As presented in Figure 2, the rates of transmission are provided in horizontal axis. As the mobile applications need to depend on WiFi (wireless media), the bandwidth will not be same all the time. Therefore, varying rate of transmission is considered. The total cost reflects the weighted sum of total energy, the cost of cloud execution and transmission delays. The total cost is provided in vertical axis that indicates the performance of each approach. When all tasks are executed locally, the cost appears less. However, it forfeits the offloading and chances of

saving energy. The proposed algorithm EEDCHA-DO shows reduced cost when compared with the case where all tasks are offloaded to cloud. This slight difference is actually big difference as the tasks are executed in large scale in the spectrum of MCC. It reveals performance benefits when the proposed algorithm is used for offloading.

Transmission Rate (Mbps)	3	4	5	6	7	8	9	10
Total Execution time from EEDCHA-DO	600	500	390	370	350	300	280	250
Local Execution Time	750	750	750	750	750	750	750	750
Execution Time For Offloading All	650	550	400	360	350	300	280	250
Time Constraint	700	700	700	700	700	700	700	700

Table 3: Total cost execution time when $T_{cnostraint}$is set to 700

As presented in Table 3, the total execution time such as local execution, offloading all tasks and offloading as per the proposed algorithm EEDCHA-DO against different rates of transmission are provided.

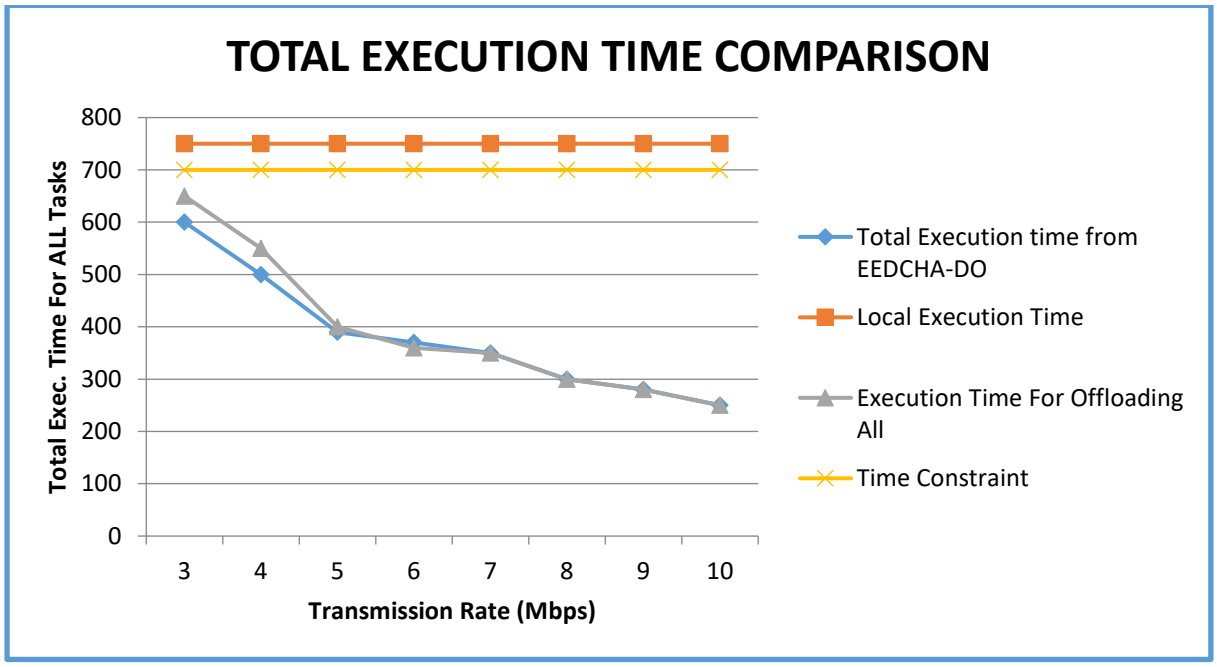

Figure 3: Total execution time comparison with different rates of transmission

As presented in Figure 3, the rates of transmission are provided in horizontal axis. The total execution time is provided in vertical axis that indicates the performance of each approach. When all tasks are executed locally, the execution time is more and it could not meet deadlines of tasks. Therefore, it demonstrates that dynamic offloading is indispensable to satisfies possible SLAs in MCC environments. The total execution time achieved with the proposed algorithm EEDCHA-DO is less than the case where all the tasks are offloaded to public cloud. This reveals the benefits of using the algorithm for making offloading decisions.

Number of Components (N)	10	15	20	25	30
Execution Time of EEDCHA-DO Algorithm (ms)	120	140	150	220	250

Table 4: Execution time of EEDCHA-DO against number of components

As presented in Table 4, the execution time of the proposed algorithm is provided against number of components.

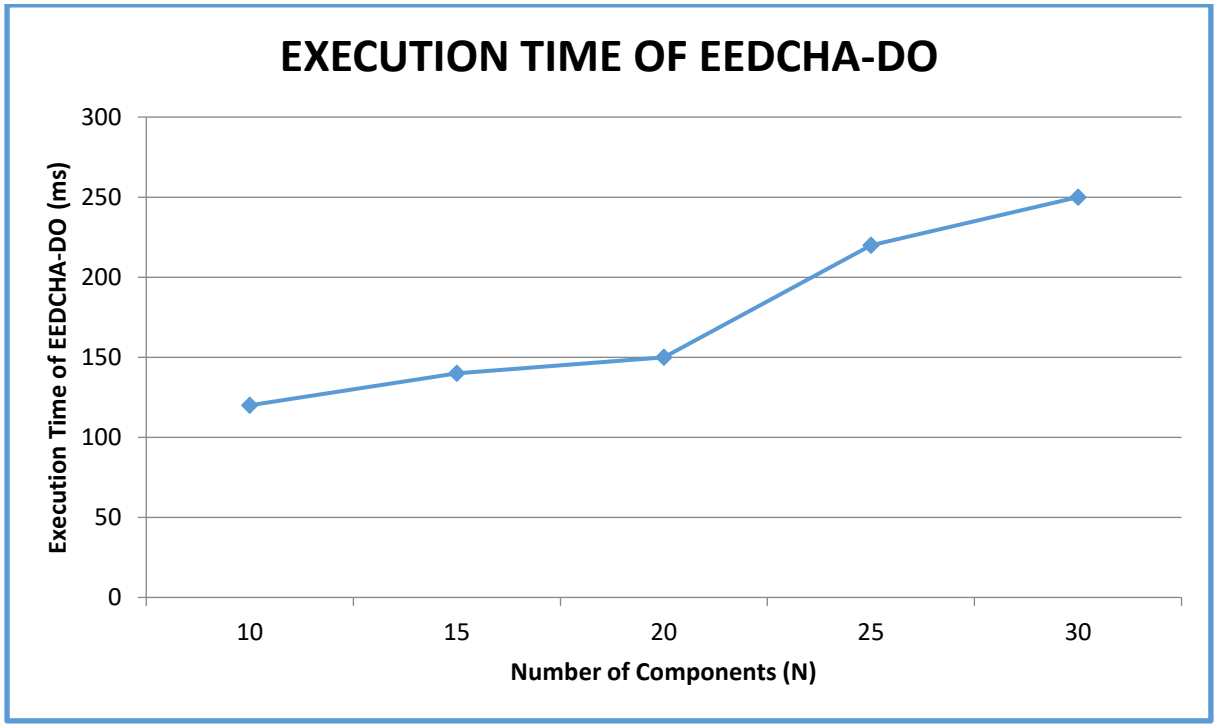

Figure 4: Number of components versus the execution time of the proposed algorithm

As presented in Figure 4, the number of components is provided in horizontal axis and the vertical axis shows execution time of the proposed algorithm. The results revealed that the execution time is influenced by the number of components.

	LAC	EEDCHA-DO (proposed)	LC
Total Cost (J)	1115	1175	1125

Table 5: Total cost of different methods

As presented in Table 5, the total cost of the proposed method is compared with that of LC and LAC methods taken from [30].

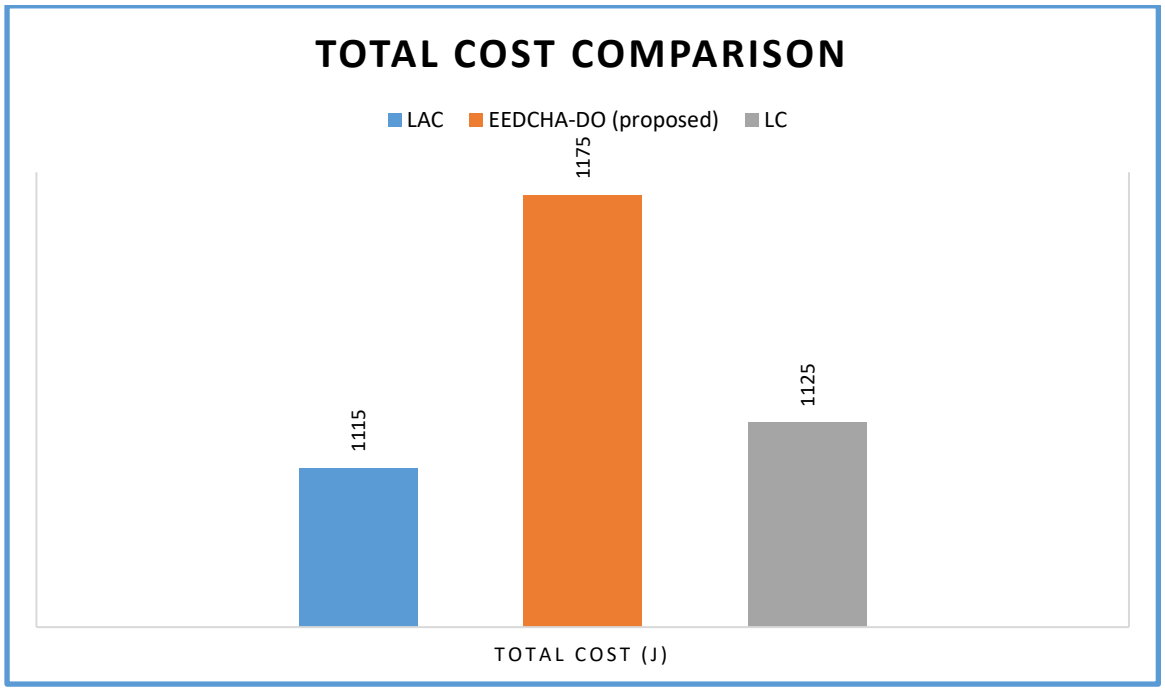

Figure 5: Comparison of total cost by different methods

As presented in Figure 5, the total cost of each method is compared. The cost of the proposed method is more which indicates that there is offloading of tasks dynamically as needed without compromising $C_{onstraint}$. The proposed algorithm dynamically computes the execution time and energy every time based on the bandwidth of the wireless network so as to make offloading decisions beneficial. All the tasks when executed locally, it will not be able to meet task deadlines. At the same time, it all the tasks are offloaded, it has issues with more execution time. Therefore, the usage of EEDCHA-DO revealed better performance.

6. CONCLUSION AND FUTURE WORK

In this paper, a heuristics based algorithm is proposed for dynamic offloading of tasks in MCC environment. The algorithmis named as Energy Efficient and Deadline Constrained Heuristic Algorithm for Dynamic Offloading (EEDCHA-DO). It considers the problem of deadline constrained tasks offloaded to cloud based on execution time and energy consumption in presence of varied bandwidth of wireless network. The algorithm performs desired computations dynamically for each task and the problem of whether to offload or not to offload the task is solved optimally. The wireless network's bandwidth is considered in execution time and energy computations. The rationale behind this is that the task when offloaded to cloud, it needs to be transmitted from mobile device to cloud with certain rate of transmission that essentially depends on bandwidth. As the bandwidth varies, it leads to varied execution time and energy consumption. The algorithm is implemented in MATLAB and the simulation results revealed that EEDCHA-DO is efficient in task offloading when compared with its predecessors such as LAC and LC. The proposed algorithm achieved minimal energy consumption while meeting the deadline of tasks with its optimized offloading decisions. The algorithm tries to have quick convergence to an optimal solution for faster execution while minimizing energy consumption. However, it needs further exploration to have more insights

of its usage in MCC. For instance, it can be enhanced to have a hybrid offloading algorithm that considers increasing the spectrum of offloading possibilities. This will be carried out in our future work.

References

[1] Zheng, J., Cai, Y., Wu, Y., and Shen, X. S. (2018). Dynamic Computation Offloading for Mobile Cloud Computing: A Stochastic Game-Theoretic Approach. IEEE Transactions on Mobile Computing, 1–1. P1-17.

[2] Md Enzai, Nur Idawati and Tang, Maolin (2014) A taxonomy of computation offloading in mobile cloud computing. In Gao, J, Zhu, H, & Wirtz, G (Eds.) Proceedings of the 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile Cloud). IEEE, United States of America, pp. 19-28.

[3] Xu Chen, Lei Jiao, Wenzhong Li and Xiaoming Fu. (2015). Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. IEEE., p1-14.

[4] Kuang, Z., Guo, S., Liu, J., and Yang, Y. (2018). A quick-response framework for multi-user computation offloading in mobile cloud computing. Future Generation Computer Systems, 81, 166–176.

[5] Sladana Jošilo and György Dán . (2018). Selfish Decentralized Computation Offloading for Mobile Cloud Computing in Dense Wireless Networks. IEEE , p1-13.

[6] Valeria Cardellini, Vittoria De Nitto Persone, Valerio Di Valerio, Francisco Facchinei ·Vincenzo Grassi, Francesco Lo Presti and Veronica Piccialli. (2013). A game-theoretic approach to computation offloading in mobile cloud computing. Springer, p1-25.

[7] Muhammad Shiraz and Abdullah Gani. (2014). A lightweight active service migration framework for computational offloading in mobile cloud computing. Springer , p1-18.

[8] Dongqing Liu, Lyes Khoukhi And Abdelhakim Hafid. (2017). Decentralized Data Offloading for Mobile Cloud Computing Based on Game Theory. IEEE, p1-7.

[9] Xu Chen. (2015). Decentralized Computation Offloading Game For Mobile Cloud Computing. IEEE, p1-12.

[10] Mohammad Goudarzi, Mehran Zamania, Abolfazl Toroghi and Haghighatb. (2016). A Fast Hybrid Multi-site Computation Oï¬„oading for Mobile Cloud Computing. Springer, p1-20.

[11] MdGolam Rabiul Alama, Mohammad Mehedi Hassan, Md.Zia Uddinc, Ahmad Almogrenb, Giancarlo Fortinod. (2019). Autonomic computation offloading in mobile edge for IoT applications . Elsevier, p149-157.

[12] Nur Idawati Md Enzai and Maolin Tang. (2016). A Heuristic Algorithm for Multi-Site Computation Offloading in Mobile Cloud Computing. Elsevier. 80, p1232–1241.

[13] Dongqing Liu, Lyes Khoukhi and Abdelhakim Hafid. (2018). Prediction-Based Mobile Data Offloading in Mobile Cloud Computing. IEEE. 17 (7), p1-14.

[20] Slaldana Jošilo and György Dán . (2019). Joint Management of Wireless and Computing Resources for Computation Offloading in Mobile Edge Clouds. IEEE, p1-14.

[15] Sudip Misra, Bernd E. Wolfinger. (2019). Auction-Based Optimal Task-Offloading in Mobile Cloud Computing, p1-9.

[16] Bokyun Jo, Md. Jalil Piran, Daeho Lee and Doug Young Suh. (2019). Efficient Computation Offloading in Mobile Cloud Computing for Video Streaming Over 5G. Computers, Materials &Continua . 62 (2), p439-463.

[17] Marco V. Barbera, Sokol Kosta, Alessandro Mei and Julinda Stefa. (2013). To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing, p1-12.

[18] Huijun Wu and Dijiang Huang . (2014). Modeling Multi-factor Multi-site Risk-based Offloading for Mobile Cloud Computing . CNSM Mini-Conference Paper . . (.), P1-7.

[19] Upadhyay and Rajasi D., "An SOA-Based Framework of Computational Offloading for Mobile Cloud Computing" (2019). Electronic Theses and Disser tations . 8185. P1-73.

[20] Shakarami, A., Ghobaei-Arani, M., Masdari, M., & Hosseinzadeh, M. (2020). A Survey on the Computation Offloading Approaches in Mobile Edge/Cloud Computing Environment: A Stochastic-based Perspective. Journal of Grid Computing. p1-33.

[21] Qurat-ul-Ain Mastoi, Abdullah Lakhan, Fawad Ali Khan, Qammer H. Abbasi. (2020). Dynamic Content and Failure Aware Task Offloading in Heterogeneous Mobile Cloud Networks. IEEE, p1-6.

[22] Lai, S., Fan, X., Ye, Q., Tan, Z., Zhang, Y., He, X., & Nanda, P. (2020). FairEdge: A Fairness-Oriented Task Offloading Scheme for Iot Applications in Mobile Cloudlet Networks. IEEE Access, 8, p13516–13526.

[23] Joaquim Silva, Eduardo R. B. Marques, Lu´ıs M. B. Lopes, and Fernando Silva. (2020). JAY: Adaptive Computation Offloading for Hybrid Cloud Environments. IEEE, p54-61.

[24] Chao Li , Junjuan Xia , Fagui Liu , Dong Li , Lisheng Fan , George K. Karagiannidis , and Arumugam Nallanathan. (2021). Dynamic Offloading for Multiuser Muti-CAP MEC Networks: A Deep Reinforcement Learning Approach. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. 70 (3), p2921-2927.

[25] Gao, Z., Hao, W., Han, Z., & Yang, S. (2020). Q-learning Based Task Offloading and Resources Optimization for a Collaborative Computing System. IEEE Access, 8, p149011–149024.

[26] Mohammad Masdari, Hemn Khezri. (2020). Efficient offloading schemes using Markovian models: a literature review. Springer, p1-44.

[27] Sardar Khaliq uz Zaman, Ali Imran Jehangiri, Tahir Maqsood, Zulfiqar Ahmad, Arif Iqbal Umar, Junaid Shuja, Eisa Alanazi, Waleed Alasmary. (2021). Mobility-aware computational offloading in mobile edge networks: a survey. Springer, p1-22.

[28] Meng, S., Huang, W., Yin, X., Khosravi, M. R., Li, Q., Wan, S., & Qi, L. (2020). Security-aware Dynamic Scheduling for Real-time Optimization in Cloud-based Industrial Applications. IEEE Transactions on Industrial Informatics, p1–9.

[29] Muhammad Ibrar, Aamir Akbar, Syed Rooh Ullah Jan, Mian Ahmad Jan§ , Lei Wang, Houbing Song, Nadir Shah. (2021). ART Net: AI-based Resource Allocation and Task Offloading in a Reconfigurable Internet of Vehicular Networks. IEEE, p1-10.

[30] M. Chen, B. Liang, M. Dong, "A Semidefinite Relaxation Approach to Mobile Cloud Offloading with Computing Access Point", IEEE Signal Processing Advances in Wireless Communications, pp. 186-190, 2015.