

Received January 31, 2020, accepted February 12, 2020, date of publication February 18, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974809

# Partial Offloading in Energy Harvested Mobile Edge Computing: A Direct Search Approach

ASAD MAHMOOD<sup>1,2</sup>, ASHFAQ AHMED<sup>2</sup>, MUHAMMAD NAEEM<sup>2</sup>, AND YUE HONG<sup>1</sup>

<sup>1</sup>College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China

<sup>2</sup>Department of Electrical and Computer Engineering, COMSATS University Islamabad at Wah Campus, Wah Cantt 47040, Pakistan

Corresponding author: Yue Hong (yhong210@gmail.com)

This work was supported by funding Natural Science Foundation of SZU (grant no. 2019035, 860-000002110271).

**ABSTRACT** In the next generation wireless communication paradigm, the number of devices are expected to increase exponentially after the concept of Internet of Things (IoT). These devices are power constrained, with limited processing capability. Therefore, in order to get the maximum advantage from these low power IoT sensing devices, it is of utmost need to empower them. Similarly, the devices are not able to process the computationally intensive applications. In this work, Wireless Power Mobile Edge Cloud (WPMEC) is considered, which is an integration of Wireless Power Transfer (WPT) and Mobile Edge Cloud (MEC) to address low power devices' battery and computational capabilities. The WPMEC is charging the devices in the first phase using the WPT and in the second phase, the devices are offloading their computational intensive data to the MEC. Partial offloading scheme is first time introduced and analyzed with WPMEC. Performance of proposed solution is evaluated in terms of overall network computational energy efficiency. Extensive simulations have been carried out to validate the proposed solution. It is shown that the proposed partial offloading scheme with WPMEC outperforms the binary and local computational schemes.

**INDEX TERMS** Wireless Power Transfer, Mobile Edge Cloud Computing, Energy Efficiency.

## I. INTRODUCTION

The demands for data rates and Quality of Services (QoS) is exponentially increasing with the rapid evolution of information technology devices, such as smart phones, tablets and laptops. While new mobile devices are more and more powerful in terms of computing capabilities, they are still not able to handle applications that need real-time processing such as wearable Virtual Reality (VR), self driving vehicular systems, mobile health care, mobile governance and etc [1]. In addition, the recent development of Internet of Things (IoT) technology is a key step towards smart and autonomous control in industrial and business processes, such as smart grids and smart home automation. The devices (e.g. sensors) often are equipped with a limited battery and a low performance processor because of the small size and to reduce the production costs [2]. As a consequence, devices with limited computing capabilities are unable to accommodate and process applications requiring scalable and high-performance computations. Therefore, for the development of modern IoT technology, it is of utmost requirement to address the

issues related to power and computational capability of IoT devices [3].

The concept of Wireless Power Transmitter (WPT) is emerged as an effective solution to address the power constraints of low power devices, including the IoT devices. The devices are continuously charged via WPT, where the WPT uses the dedicated radio frequency for energy transmission. The technology is able to transfer tens of micro watts at a distance of more than ten meters. Therefore, WPT effectively can charge low power devices. Using the advance signal processing techniques, the improvement in the performance of WPT in terms of energy transmission is foreseen in the near future [4]. Therefore, one of the critical issue related to low power devices (IoTs) can be addressed using the WPT.

The other main critical limitation of IoT devices is related to their computational capabilities. Usually the IoT devices act as sensors, where they sense data from their surrounding environment, but are unable to process the data because of limited battery and limited processing resources. To assist the low power devices a concept of edge computing is addressed by the researchers. Therefore, under a contract with a centralized infrastructure, a new paradigm namely Mobile Edge Cloud (MEC) is proposed, where the MEC is located in the

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Guo.

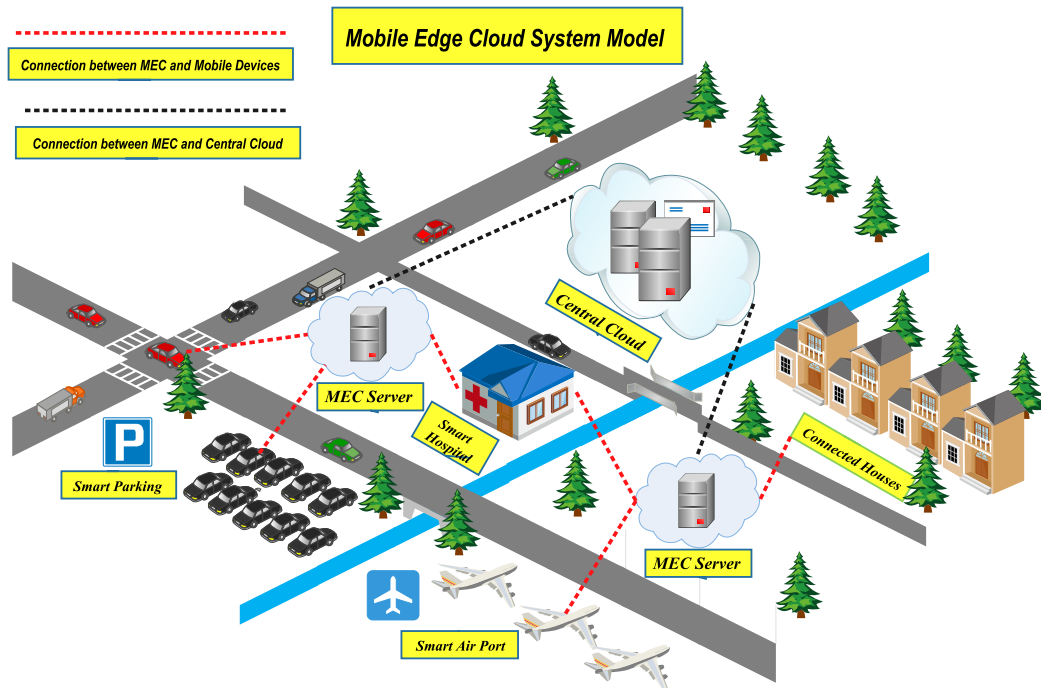


FIGURE 1. System Model For Mobile Edge Cloud.

vicinity of the devices, as shown in Figure 1. The position of the MEC, in vicinity to the devices overcomes the latency issue [5]. The MEC has less resource as compared to the cloud. MEC, not only, provides the computational facility but also the storage and caching services at the edge of mobile devices. MEC is an emerging concept of 5G and B5G which ensures the end to end delay requirement of less than 1ms. In MEC and WPT assisted IoT network, the devices are able to re-charge their batteries and to offload the computational intensive applications to the MEC [5]. In this way the energy consumption of the devices would be significantly reduced. Therefore, for a better computational efficiency and to reduce the end to end delay, the computational and communication resources should be optimally allocated among the devices and the MEC.

In the recent research work, two schemes for data offloading to MEC are proposed. These schemes are named as the binary offloading scheme and the partial offloading scheme. In the binary offloading scheme, the whole data is either computed locally on the device or is offloaded to the edge server. In the partial offloading scheme, the complete task is split into two parts, where one part is computed locally, whereas the remaining part is offloaded to the MEC. One advantage incurred from the partial offloading is the level of achieved parallelism, where a portion of the task is computed locally and in parallel at the same time other portion is computed on the MEC. In [6], a weighted sum computational rate maximization of MEC network is proposed, where the binary offloading scheme is deployed with one server and several mobile devices. In [7], an MEC is proposed with one energy harvesting device and a dynamic offloading algorithm using partial offloading scheme to minimize the execution cost.

In [8], an energy consumption minimization problem of Access Point is addressed. In [9], the author presented with Energy Harvesting the Reinforcement Learning-based computation offloading system for IoT devices to achieve the optimal offloading strategy without knowledge of the MEC model, the computation latency model and the energy consumption model. In [10], a concept of cooperative communication is studied. Here the authors have considered three nodes, where one of them is acting a relay and edge computational device. A computational energy efficiency maximization problem is discussed in [11] using MEC. In [12] author formulate the cost minimization function in order to construct a intelligent offloading framework for 5G enabled Vehicular systems. In [13] author model an intelligent offloading for vehicular edge computing by using the concept of deep reinforcement learning taken mobility and non orthogonal multiple access into consideration, The proposed solution addresses optimal allocation of resources between mobile users and MEC.

In this work, the concept of Wireless Power Mobile Edge Cloud (WPMEC) [6] is considered. The WPMEC is an integration of WPT and MEC. WPMEC is located at the Access Point (AP), where the AP could be a base station, a Wi-Fi router etc. The WPMEC is used for energy transfer to low power devices and at the same time to receive computational tasks from the devices. This work addresses partial offloading scheme, Which is a direct measure of computational capability of power constrained devices. To the best of our knowledge, it is the first effort to study the WPMEC with partial offloading scheme to overcome the computational capability and battery limitation problems. In this work the terms users, user devices and IoT are interchangeably used to address the

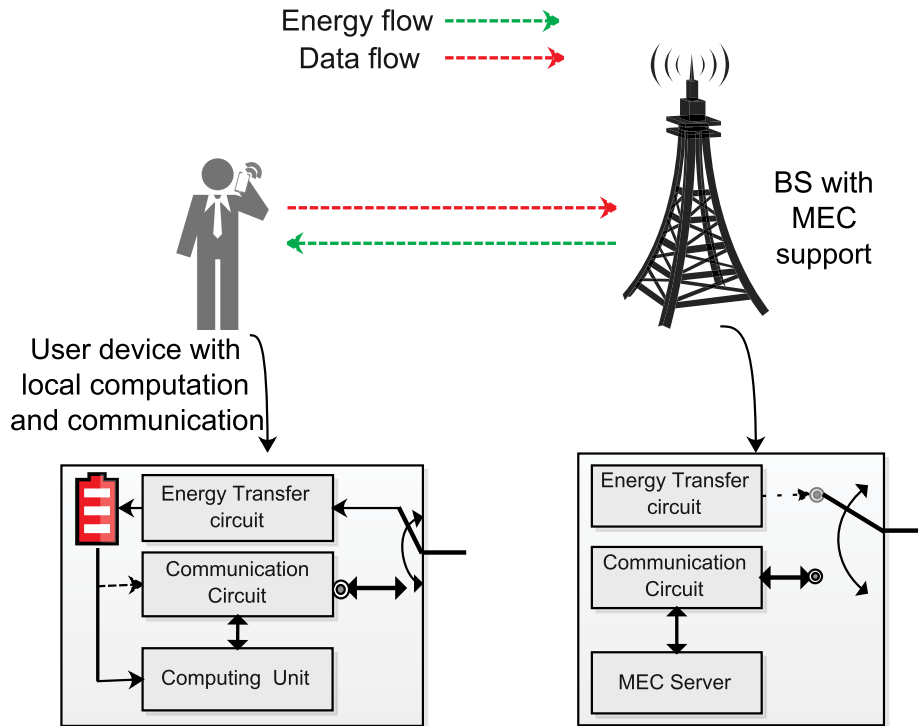


FIGURE 2. Wireless Power Mobile Edge Cloud (WPMEC).

low power devices. The computational energy efficiency is used as performance metric. The main contributions in this work are as follow:

- 1) We formulate our problem as a joint optimal allocation of transmission power, local computing chip frequency as well as transmission time among mobile devices and WPMEC in order to maximize the computational energy efficiency of low power devices, which is a direct measure of systems computing capabilities.
- 2) An algorithm is proposed which works on the principle of contraction and expansion, and is based on Mesh Adaptive Direct Search (MADS) algorithm. The proposed algorithm reaches the  $\epsilon$ -optimal solution.
- 3) The fundamental trade-off between data size, offloading and local computation is analyzed through extensive simulations. Impact of user devices with different priorities is analyzed in different offloading schemes.

Rest of the paper is organized as follow. The system model for WPMEC is explained in Section II. While the proposed solution is discussed in Section III. Results are discussed in Section IV, and finally conclusion is drawn in Section V.

## II. SYSTEM MODEL

In this work a Wireless Power Mobile Edge Cloud (WPMEC), proposed in [6], is considered with one Access Point (AP) and  $N$  mobile devices. The mobile devices contain single antenna and are distributed uniformly. WPMEC is integration of WPT and MEC, as shown in Figure 2. The WPT uses radio frequency transmitter to continuously charge the battery of mobile devices. Mobile devices harvest

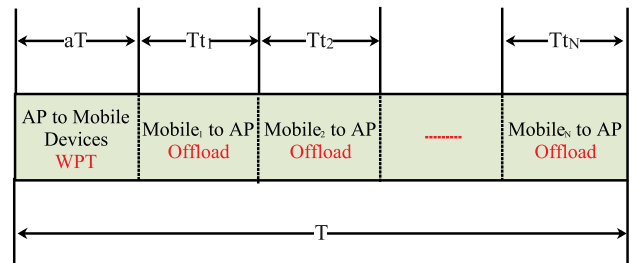


FIGURE 3. Time Allocation in Mobile Edge Cloud (WPMEC).

and store energy for further processing and offload some of their computational tasks to MEC located at AP. It is assumed that WPT and MEC are using the same frequency for energy transferring and data offloading. To avoid their mutual interference, Time Division Multiplexing (TDM) technique is applied, as shown in Figure 3, to separate communication and energy transmission within mobile devices.

In order to implement TDM, the time frame  $T$  is divided into two parts, where in time sub-frame  $aT$  the mobile devices harvest energy from AP, whereas during the remaining time sub-frames  $T - aT$  the devices offload their computation intensive task to MEC cloud using partial offloading scheme. Using partial offloading parallel execution at the mobile and MEC server is introduced, which means edge computing and local computation takes place at the same time.

### A. EDGE COMPUTING

The  $N$  uniformly distributed devices are represented by set  $N = \{1, 2, 3 \dots, N\}$ . These devices can offload some part of their computational tasks to MEC. In order to reduce

their mutual interference, devices offload their task using TDM approach, as shown in Figure 3. The allocated time, power and the channel between mobile devices and wireless power MEC is represented by  $t_n$ ,  $p_n$  and  $g_n$ , respectively. The total number of bits that need to be offloaded to cloud is represented by  $r_n$ , whereas noise power and bandwidth is represented by  $\delta^2$  and  $B$ , respectively. The total number of offloaded bits in the allocated time by a mobile device  $n$  is given as,

$$r_n = B \log_2 \left( 1 + \frac{p_n g_n}{\delta^2} \right) t_n \quad (1)$$

Similarly, the corresponding energy consumption during data offloading is given as,

$$e_n = p_n t_n + p_r t_n \quad (2)$$

where,  $p_n$  represents the transmission power of  $n^{\text{th}}$  user in time  $t_n$  and  $p_r$  represents the constant energy consumption for signal processing and is same for all devices.

### B. LOCAL COMPUTATION

In parallel to MEC computation, some part of the tasks are computed locally by the corresponding mobile devices. Number of computational cycles needed to compute one bit of data is represented by  $C_n$ . For local computation, the mobile devices take entire time frame  $T$  to complete their task. The total number locally computed bits by a mobile device  $n$  is given as,

$$r_n^l = \frac{T f_n}{C_n} \quad (3)$$

where  $f_n$  represents the computational capability of the mobile device, i.e. cycles per second. The energy consumption by device  $n$  during local computation is given as,

$$E_n^l = \epsilon_n f_n^3 T \quad (4)$$

where  $\epsilon_n$  represents the computational energy efficiency.

### C. PROBLEM FORMULATION

The objective of this work is to maximize the overall computational efficiency of all the mobile devices connected to WPMEC located at the AP. Mathematically, this problem expressed as,

$$\begin{aligned} & \max_{\substack{a, t_n, f_n, p_n \\ \forall n \in \{1, \dots, N\}}} \sum_{n=1}^N w_n \left( \frac{B \log_2 \left( 1 + \frac{p_n g_n}{\delta^2} \right) t_n + \frac{T f_n}{C_n}}{\epsilon_n f_n^3 T + p_n t_n + p_r t_n} \right) \\ & C_1 : a + \sum_{n=1}^N t_n \leq T \\ & C_2 : B \log_2 \left( 1 + \frac{p_n g_n}{\delta^2} \right) t_n + \frac{T f_n}{C_n} \geq L_n, \quad \forall n \\ & C_3 : \epsilon_n f_n^3 T + p_n t_n + p_r t_n \leq E_n, \quad \forall n \\ & C_4 : 0 \leq f_n \leq f_n^{\max}, \quad \forall n \\ & C_5 : t_n \geq 0, p_n \geq 0 \quad \forall n \\ & C_6 : a \in (0, 1) \end{aligned} \quad (5)$$

In (5),  $w_n$  is a weighting factor used to prioritize the users devices based on their QoS requirements. The objective in (5) is a to maximize the overall network energy efficiency, through optimal allocation of resources. The resources are the offloading time, the power assigned to each user device, and the devices' computation capabilities. Constraint  $C_1$  states that all the offloaded tasks must be completed with the allowed time frame  $T$ . Here only transmission from device to AP is considered, whereas the receiving time from AP to device is ignored as the size of received data is very less and comparatively it would take very little time.  $C_2$  states that the total number of bits computed locally and at the MEC, should be greater than minimum number of data bits  $L_n$ . Finally,  $C_3$  states that total amount of energy consumed by the user device should be less than or equal to the amount of energy harvested by the user device form WPT located at AP.

### III. OPTIMAL RESOURCE ALLOCATION

The problem in (5) is a non-linear non-convex problem. Because of the non-convex aspect of the optimization problem, the traditional convex optimizers can't be used to solve (5).

An effective  $\epsilon$ -optimal algorithm is proposed that is based on Mesh Adaptive Direct Search (MADS) algorithm. The proposed algorithm uses exploration and exploitation method to maximize the problem search space. The proposed algorithm is named as Joint Power Profile, Time Splitting and Chip Computing Frequency Optimization Algorithm (JPTFA), as shown in Algo. 1.

The problem given in (5) is solved using the proposed JPTFA, where JPTFA contains trail points that have been evaluated using the function of black box. Results from these tests are further analyzed and used to generate new trail points. JPTFA iteration involves three phases, i.e. review, search and poll phases. Theoretical search is being performed by search phase with one trail point generated in conjunction with previous successful route. JPTFA has the ability to handle different types of inequality constraints, but can not accommodate the constraints of equality.

JPTFA applies contraction and expansion iteratively in the entire search space in order to find the optimum solution. JPTFA depends on polls to determine the optimal solution for the current location of the iteration. Iteration number  $j$  starts with '0'.  $\Psi_j^p$  determines the size of the poll and  $\Psi_j^m$  specifies the width of the mesh at  $j^{\text{th}}$  iteration. The  $j^{\text{th}}$  iteration mesh points are defined by  $M_j$ , where  $M_j$  is identified by putting the stencil at the current position  $z$  and moving  $\Psi_j^m$  (mesh steps) in compliance with the  $D_j^m$ . Set  $M$  includes all previously visited points lying on the mesh and the new trial points around any of the previously visited points using the instructions in  $D_j^m$  at distances  $\Psi_j^m$ . On set  $M$ , the problem (5) is solved and the results are stored in set  $x_j^m$ . The value of objective function at previous iteration, i.e.  $f(x_{j-1}^m)$ , is compared with the updated objective function value at current points, i.e.  $f(x_j^m)$ . If the value of objective function obtained

**Algorithm 1** Joint Power Profile, Time Splitting and Computing Chip, Frequency Optimization Algorithm (JPTFA)

```

1 Initialization  $j \leftarrow 0, \Psi_j^m > \Psi_j^p > 0, x_j^p, x_j^m;$ 
2  $0 < \Theta^m < \Theta^p < 1;$ 
3  $f \leftarrow (5);$ 
4 Execution;
5 while Terminate criterion not met do
6    $M_j = \bigcup_{z \in \beta_j} \{z + \Psi_j^m D_j^m\};$ 
7    $x_j^m = \begin{cases} \operatorname{argmin} f \\ z \in M_j \end{cases};$ 
   Subject to: C1 – C5 of (5)
8   if  $f(x_j^m) > f(x_{j-1}^m)$  then
9      $\zeta_j = \bigcup_{z \in \beta_j} \{z + \Psi_j^p D_j^p\};$ 
10     $x_j^p = \begin{cases} \operatorname{argmin} f \\ z \in \zeta_j \end{cases};$ 
    Subject to: C1 – C5 of (5)
11    if  $f(x_j^p) > f(x_{j-1}^p)$  then
12       $\Psi_{j+1}^p = \Psi_j^p \cdot \Theta^p;$ 
13    end
14  else
15     $\Psi_{j+1}^m = \frac{\Psi_j^m}{\Theta^m};$ 
16  end
17   $j \leftarrow j + 1;$ 
18 end

```

at previous iteration is smaller than that of updated iteration, it indicates an improvement. In case of no improvement, the size of mesh in the next iteration, i.e.  $\Psi_{j+1}^m$ , is increased by a factor of  $\Theta^m$ . This phenomenon is called expansion.

Usually polling stage is initiated after an improvement in the mesh analysis. The polling points are attained by placing the stencil at present location  $z$  and moving  $\Psi_j^p$  (poll steps) in accordance with the direction mesh vector  $D_j^p$ . As in the mesh analysis, once again the optimization problem (5) is solved on the set of polling points, i.e.  $\zeta_j$ , and the results are stored in set  $x_j^p$ . The value of objective function on polling points in previous iteration, i.e.  $f(x_{j-1}^p)$ , is compared with the objective function value on the updated point, i.e.  $f(x_j^m)$ . In case of improvement, the poll size  $\Psi_{j-1}^p$  is further reduced by a factor of  $\Theta^p$ . This phenomenon is called contraction. The algorithm follows the contraction in the iterations when there will be improvement found, whereas the algorithm expands its search space when the improvement could not be achieved on current mesh points. The algorithm continues until it achieves the  $\epsilon$ -optimality.

As discussed earlier that MADS is basically pattern search algorithm based on polling to find the near optimal solution. The algorithm starts from a random location and attempts to search for optimal solution in the proximity locations. As shown in the Figure 4, the algorithm starts at point ‘a’

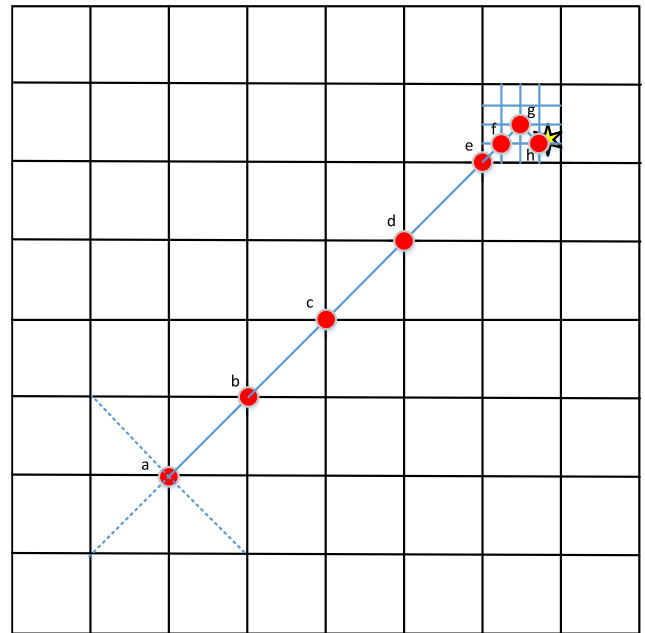


FIGURE 4. MADs Algorithm's polling steps.

TABLE 1. Simulation parameters.

Parameter Name	Symbols	Values
Bandwidth	$B$	20kHz
Users devices	$N$	2, 5
Time block	$T$	1 sec
Chip Computing Efficiency	$\epsilon_n$	$10^{-24}$
Maximum Computation Capacity	$f_n^{\max}$	$10^9$ cycles/sec.
Harvested Energy	$E_n$	2 Joules
Cycle per bit	$C_n$	$10^3$
Static Circuit Power	$p_r$	50mW

in the first iteration and it has to reach the optimal solution which is denoted by a star in the grid. The stencil at point ‘a’ has four direction and out of these four possibilities, one is called the direction of descent where the objective function converges. In the successive iterations the stencil moves towards the solution where the objective function shows convergence. For example, in the example it moves from ‘a’ to ‘b’, then to ‘c’ and continues its direction in a straight line till ‘e’. After ‘e’, the point reaches closer to the optimal point. Now the grid is zoomed and the stencil starts taking small steps in order to ensure near optimal solution.

**IV. RESULTS AND DISCUSSION**

The proposed solution is validated through extensive simulations carried out in *Matlab*. The parameters used in the simulations are shown in Table 1 and taken from [11]. Channel between AP and mobile devices is assumed to block fading, means channel between AP and mobile devices remain constant in entire duration of time  $T$ .

Figure 5 shows the comparison between complete local computations and our proposed scheme in terms of the achieved energy efficiency for  $N = 2$ . In this case,  $L_1 = L_2$  and  $w_1 = w_2 = 1$ , which means minimum requirement of

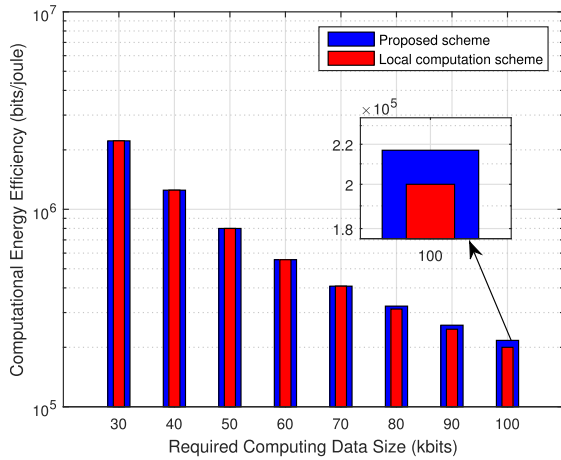


FIGURE 5. Computational energy efficiency vs. required computing data size with  $N = 2$ .

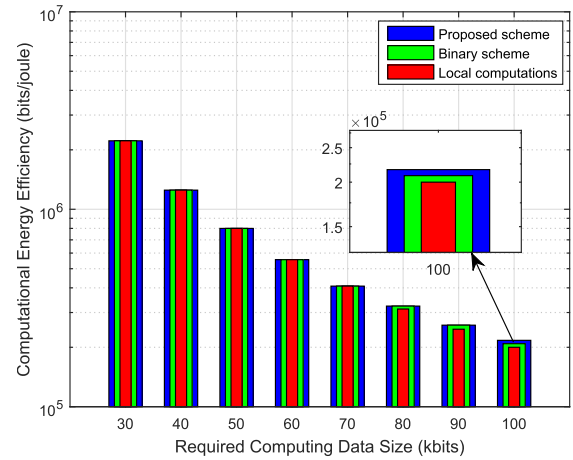


FIGURE 7. Computational energy efficiency vs. required computing data size for binary scheme, complete local computations and the proposed partial offloading scheme with  $N = 2$ .

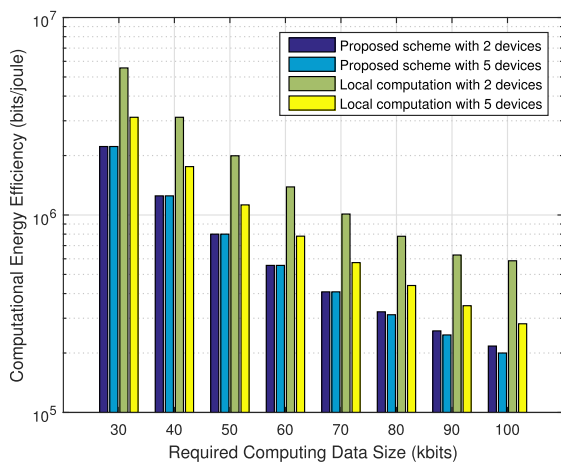


FIGURE 6. Computational energy efficiency vs. required computing data size with  $N = 2$  and  $N = 5$ .

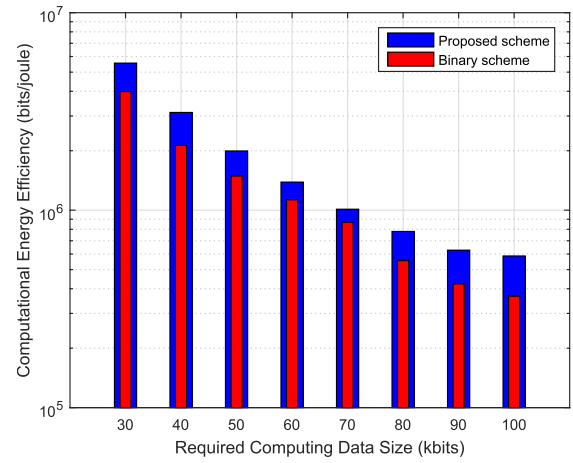


FIGURE 8. Computational energy efficiency vs. required computing data size for binary and proposed schemes with  $N = 5$ .

bits computation is same for both devices. It is clearly shown that the computational energy efficiency decreases with the increase of data requirements. This behaviour reflects that more energy is required to compute large amount of data. In this scenario, the proposed scheme performs better than the complete local computations.

Figure 5 represents the fundamental trade off between data size and computational complexity with partial offloading and complete local computations. For small data size computational energy efficiency for local computation and proposed scheme is almost same. This implies that if the energy required to compute task locally is less than the energy required to offload, then it is better to perform task locally. On the other hand if the data size and number of users increase, computation at WPMEC become a better choice. This effect is more prominent for  $N = 5$ , as shown in Figure 6, where it is illustrated that the proposed scheme performs better even for small data size, when the number of devices increase.

Other than complete local computations and partial offloading, there is another scheme known as binary offloading. In the partial offloading scheme, the partial data is computed locally and whereas the partial part of data is offloaded to the MEC. In the binary offloading scheme, the decision is to either compute the data locally or to compute it on MEC. Therefore, in binary scheme complete data is either computed on the user devices, or the complete data is computed on the MEC using offloading. Figure 7 illustrates the comparison between the proposed partial offloading scheme and the binary offloading scheme. It is quite clear from the Figure. 7 that the performance of proposed scheme and binary offloading scheme is same for small number of user devices and small data size. As soon as, the number of user devices and the data size increase the proposed scheme outperforms the binary offloading scheme even for small data size.

The computational energy efficiency of individual user device increases if the priority increases. In this case, the priority is set using the weight factors. Figure 9 shows the computational energy efficiency of individual devices, and

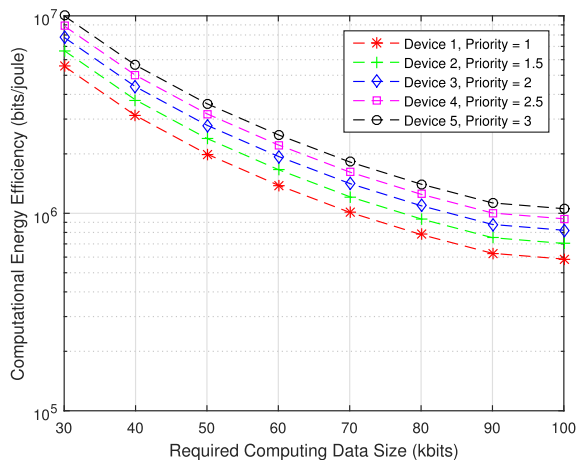


FIGURE 9. Computational energy efficiency vs. required computing data size for proposed schemes with different weights for each user device with  $N = 5$ .

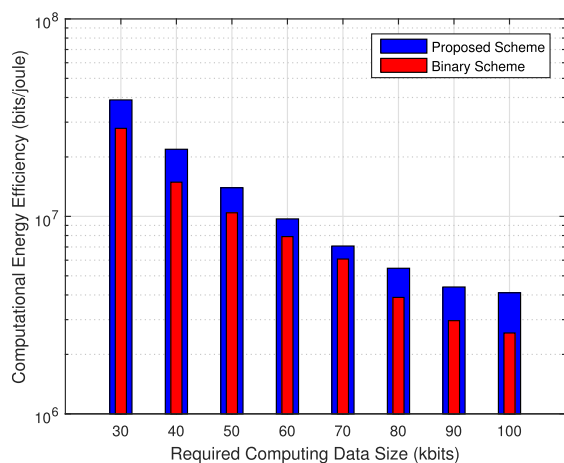


FIGURE 10. Computational energy efficiency vs. required computing data size for binary and proposed schemes with different weights for each user device with  $N = 5$ .

the results demonstrate that devices with high priority (weight factor) have high computational energy efficiency as compared to users having low priorities. This tremendous behavior of model can be utilized in emergency scenario where life of individual user is more important than overall computational energy efficiency. Figure 10 illustrates the comparison among binary and proposed schemes with respect to the priorities, where the priority is set equal in both cases. Results depicts that computational energy efficiency using proposed scheme dominate binary offloading scheme.

### V. CONCLUSION

In this article an energy efficient maximization problem is presented using the Wireless Power Mobile Edge Cloud (WPMEC), by optimal allocation of resources amount user devices. The proposed model is validated through extensive simulations. The proposed solution outperforms other schemes, like the binary scheme and local computation schemes in terms of energy efficiency for different data sizes and different priorities of devices. The results with different priorities have endorsed the importance of WPMEC system in emergency scenario.

### REFERENCES

- [1] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
- [2] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.
- [3] Z. Chen, R. Klatzky, D. Stewiorek, M. Satyanarayanan, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, and P. Pillai, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, 2017, p. 14.
- [4] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018.
- [5] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.
- [6] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.
- [7] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
- [8] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.
- [9] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.
- [10] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing," in *Proc. 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018, pp. 1–6.
- [11] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3052–3056, Jan. 2019.
- [12] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.
- [13] Z. Ning, P. Dong, X. Wang, J. J. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–24, 2019.

...