Towards Multi-Criteria Heuristic Optimization for Computational Offloading in Multi-Access Edge Computing

Raghubir Singh[†], Student Member, IEEE, Simon Armour[†],
Aftab Khan[‡], Mahesh Sooriyabandara[‡], and George Oikonomou[†]
Communication Systems & Networks Research Group, University of Bristol, Bristol, UK[†]
Department of Electrical and Electronic Engineering, University of Bristol, UK[†]
Telecommunications Research Laboratory, Toshiba Research Europe Limited, Bristol, UK[‡]
E-mail: raghubir.singh@bristol.ac.uk

***Abstract*—In recent years, there has been considerable interest in computational offloading algorithms. The interest is mainly driven by the potential savings that offloading offers in task completion time and mobile device energy consumption. This paper builds on authors' previous work on computational offloading and describes a multi-objective optimization model that optimizes time and energy in a network with multiple Multi-Access Edge Computing servers (MECs) and Mobile Devices (MDs). Each MD has multiple computational jobs to process, and each task can be processed locally or offloaded to one of the MEC servers. Several heuristic offloading policies are proposed and tested with an objective function with a range of weightings for optimizing time and energy. The approaches are illustrated with the help of three test cases of varying complexity. The objective function shows a continuous variation as the emphasis is placed on either time or energy saving by the weighting factors. The numerical tests demonstrate that the proposed heuristic algorithms produce near-optimal computational offloading solutions while considering a combined weighted score for schedule task completion time and energy**

***Keywords*—Multi-Access Edge Computing, computation offloading, heuristic algorithms, energy savings**

I. Introduction

Computation offloading refers to the transfer of resource-intensive jobs to an external superior processing system to solve complex tasks [1]. Studies of computational offloading have focused on Mobile Cloud Computing [2] and Multi-Access Edge Computing (MEC) [3].

The existing literature on MEC has demonstrated that significant savings in time and energy usage can be accomplished by making use by offloading computational jobs in MCC or a MEC network [4]–[13]. However, most proposals for offloading mechanisms focus on a single objective function of minimizing time or energy as optimization problems. Majority of the existing literature consider task completion time and local energy as independent variables in the context of finding either the shortest total computation times or minimal energy usage by the a MD. In this context, an ambitious and a challenging question is: can time and energy savings be simultaneously maximized in a MEC network?

Authors previous work [13] concluded that minimization of time in offloading algorithm allocates jobs to mobile devices and MEC servers in parallel, hence saving overall computation time. However, this allocation may not yield a minimum overall energy consumption. The trade-off between time and energy should be taken into account in offloading algorithms, and these two are conflicting objectives, described as "divergent goals" in [14].

This main aim of this paper is two-fold: first, to build a heuristic scheduling algorithm that can provide near-optimal computational offloading solutions while considering both energy and time in the objective function; Secondly, to investigate how offloading solutions change as a user's preference for time and energy changes.

Some studies in the existing literature take-up the problem of multi-objective optimization. For example, in [12] multi-objective optimization of energy consumption, execution delay and price cost are considered in the objective function in a Mobile Edge Computing paradigm. The execution delay was a consequence of a queuing process for multiple MDs, and local energy consumption decreases as the probability of a job being offloaded increased. The authors of [12] aimed to find a global optimum that minimized an objective function that integrated all three parameters, each multiplied by an appropriate weighting factor. Only one set of weighting factors was used in numerical analyses in [12] and the sole evidence for such a global optimization was a minimization concerning transmission power (expressed, presumably as relative values) of the MD; however, no explanation was given as to what

978-1-6654-4005-9/21/$31.00

factors influenced the transmission power of the MD or if this was in any way modifiable by the user of the MD.

Authors' previous work [13] consider the minimization of time as an objective function. A critical insight from work was that the optimal offloading schedule made use of all the available resources in parallel and often decided not to offload jobs, mostly when link speeds were low. In this paper, we extend our existing work by considering an objective function that comprises time and energy. The contributions of the paper are incorporated both schedule task completion time and local (MD) energy:

- a novel distributed heuristic algorithm is proposed for computation offloading to identify minimal values of the weighted scales for task completion time and local (MD) energy use;
- a detailed investigation of how changes in the weighting factors affect the outcomes that maximize savings in computation time and local energy use investigation by evaluation

The rest of the paper is set out as follows. The problem formulation is provided in Section II, which includes the complete definition of a weighted objective function comprising two components: total completion time and energy consumption. Section III describes the heuristic approaches to finding offloading solutions in three different cases and compares their solutions with global optima achieved by linear programming. Section IV presents the numerical demonstration of the heuristic algorithms in three different test networks. Discussion and reflection on the results obtained are provided in Section V, and conclusions are projections for future work are presented in Section VI.

II. Problem Formulation

In this paper, we extend our existing work by introducing energy in the objective function. The reader is referred to [13] for details regarding the complete mathematical model. Here we present the extended objective function. The following subsections provide details of formulating total computation time and energy and an objective function that is a weighted sum of time and energy. The notations used in this paper are provided in Table I.

A. Computational Time

The overall computational time of processing the jobs is given as follows:

$$CT = \underbrace{\sum_{i \in M} T_i}_{\text{Local Time}} + \underbrace{\max\{T_c; c \in C\}}_{\text{A MEC maximum Time}} + \underbrace{\sum_{(i,c,j)} (2-\pi) T_{i,c,j}}_{\text{Transmission/ reception time}} \quad (1)$$

Table I: Notations used in the paper.

Symbol	Definition
$c \in C$	$\{c = 1, \ldots, n\}$ set of MECs
$i \in M$	$\{i = 1, \ldots, k\}$ set of mobile devices
$j \in J^i$	$\{j = 1, \ldots, m_i\}$ set of jobs on MD i
X_j^i	Data size of job j on a MD i
α_i	Computing capability of a MD i
β_c	Computing capability of a MEC c
CT	Total computational time of all jobs
CE	Total energy consumption of all jobs
T_i	Local processing time of a MD i
T_c	Offloading processing time of a MEC c
$T_{i,c,j}$	Data transmission time of a job j on a link connecting a MD i and a MEC c
π	Proportion of data size reduction after processing
P_i^{MD}	Power rating of the embedded processor on a MD i
P_i^{Trans}	Transmission power rating of a MD i
P_i^{Idle}	Idling power rating of a MD i
T_{max}	Worst computational offloading time to process all jobs
E_{max}	Worst energy consumption to process all jobs
O^{DFP}	Offloading decision based on fixed probability
O^{DPD}	Offloading decision based on probability distribution
O^{GS}	Offloading decision based on guided search
M^{J}	MECs allocation based on job size
M^{T}	MECs allocation based on computational time
M^{W}	MECs allocation based on minimum score

There are three components in equation 1: the first component defines the local computation time; the second component defines the MEC computational time, and the third component establishes the time of transmission and receiving the data. We assumed that π is the proportion of the data reduction when the data is solved on the MEC side and is transmitted back to the MD. Since the relationship between time and data is linear, same proportional reduction takes place in reception time [13].

B. Computational Energy

The overall computational energy consumption of all MDs to process all the jobs is given as follows:

$$CE = \underbrace{\sum_{i \in M} P_i^{\text{Idle}} \max\{0, \max\{T_c; c \in C\}\}}_{\text{Idling energy consumption}} + \underbrace{\sum_{(i,c,j)} (2-\pi) T_{i,c,j} P_i^{\text{Trans}}}_{\text{Transmission/receiving energy consumption}} + \underbrace{\sum_{i \in M} P_i^{MD} T_i}_{\text{Mobile energy consumption}} \quad (2)$$

There are three components in equation 2. The first component defines the energy consumption while the jobs are being processed on MEC servers and the MDs are in idle state. The second component defines the energy consumption due to transmission and receiving of data. We ignore the computational energy consumption of MEC servers. The third component is the energy consumption on the mobile side.

C. Multi-objective optimization formulation

The weighted combined optimization function is given as follows:

$$\min \quad \left(w_t \times \frac{CT}{T_{\max}} + w_e \times \frac{CE}{E_{\max}} \right) \quad (3)$$

where w_t and w_e are the weightings on computational time and computational energy consumption, respectively. The weights are defined as: $0 \leq w_t, w_e \leq 1$ such that $w_t + w_e = 1$. Further, $T_{\max}$ and $E_{\max}$ are the expected worse case computational times and energy, respectively. The worst-case options can be determined by solving all the jobs locally (for time) and (for energy) either when tasks are performed locally or when using slowest MEC link and processor speeds, depending on the numerical values selected. We assume that the weighting factor can be adjusted according to MD users' needs or by the MEC service provider at the local cell station when a "cluster" of MD users attempt to connect simultaneously.

To correct for large numerical discrepancies in the ranges of absolute values taken by the different variables, a Bias Correction Coefficient $\eta = \frac{E_{max}}{T_{max}}$ is introduced as follows.

$$\min \quad \left(w_t \times \frac{\eta CT}{T_{\max}} + w_e \times \frac{CE}{\eta E_{\max}} \right) \quad (4)$$

III. Heuristic algorithms for computation offloading

Figure 1 presents a generic flowchart of our heuristic approaches. The first step in the algorithm is guided by different assumptions regarding the probabilities of offloading on individual jobs; the second step defines the various possible policies for choosing a MEC server to which a job is loaded. The two best performing heuristic algorithms presented in [13] were extended to include both time and energy factors.

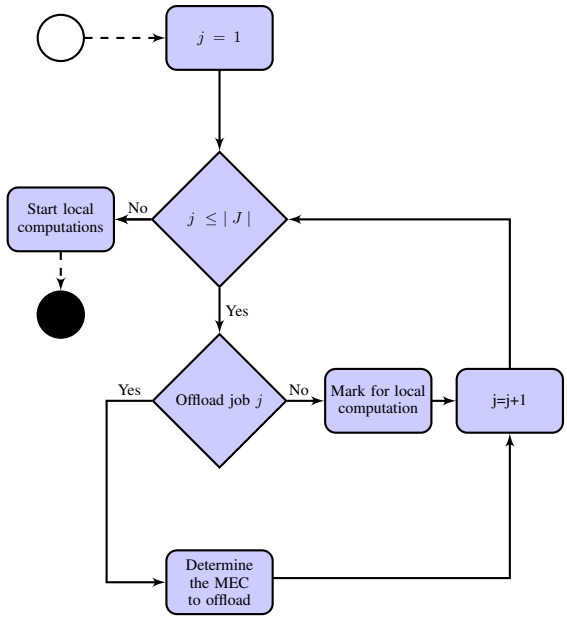

Figure 1: Flowchart of the heuristic algorithm for computation offloading.

The heuristic algorithms have two decision making stages: (a) whether to offload a job or not, (b) if a decision is made to offload a job, which MEC the job should be offloaded to? A set of probabilities govern the first-stage decision and two options are considered for offloading probabilities: offloading based on a single fixed probability ("O^{DFP}") and offloading based on a known probability distribution ("O^{DPD}"). If a job is offloaded then the decision on selection of MEC is based on a decision-based rule. Three policies of the decision-based rule are considered; two polices were described in [13]: offloading based on offloading based the job size ("M^{J}") and offloading based on the minimum remaining MEC computational time ("M^{T}"). Further, a new MEC offloading option is constructed on the fastest link MEC connection ("M^{W}"). The motivation is to offload a job from an individual MD to the MEC with the most agile bandwidth connection to reduce the minimum score.

In addition to the two approaches described above and in [13], an additional heuristic offloading approach, "(O^{GS})", is defined as follows.

1) A solution is obtained by one of the two heuristic approaches and scores for all the jobs are calculated
2) Allocation of the n^{fix} jobs with minimum scores is fixed and step 1 is repeated.
3) Step (1) and (2) are repeated until all the jobs have been allocated.

Each job's weighted score is calculated based on its time and energy and divided by the local worst time and energy. This strategy is a guided search, as the score reflects the computational load of mobile devices and MEC servers.

Table II: Parameters used for numerical simulations in Case 1,2 & 3

Entity	Parameter	Value	Unit
# of jobs	X_j^i	2-9(1), 1-7(2), 1-6(3)	MB
MDs	α_i	3.60×10^9	IPS
MEC1	β_1	1.40×10^{11}	IPS
MEC2	β_2	1.40×10^{11} (1),	IPS
		3.68×10^{10} (2 & 3)	
MEC3	β_3	1.40×10^{11} (2)	IPS
		3.68×10^{10} (1 & 3)	
Network	MDs - MECs	15-28	Mbps

IV. Numerical Results

To validate and to provide a comparison benchmark for results generated by the heuristic algorithms, linear programming optimization was performed using CPLEX[1]. Using this mathematical model, we compute the theoretical optimum job allocation with the minimum weighted multi-objective function. Sub-optimal job allocations generated by the heuristic algorithms were then compared to this theoretical optimal allocation to assess the performance of the heuristic under different combinations of MDs and MEC servers. All the heuristic algorithms were run for 100 iterations to identify a minimum weighted score.

A. Numerical Parameters

The numerical values for the parameters used in simulations are presented in Table II using processor speeds and power ratings from [15] and [16]. Three cases were considered: one with jobs offloaded from a single MD and two cases where multiple MDs attempted to offload; all three cases were in heterogeneous MEC networks.

B. Case 1: Offloading from a single MD

In this scenario, heuristic algorithms were simulated on a single MD that has 20 jobs and is connected to 2 MEC servers. Figure 2 presents the results as the weighting factor on time is increased. Recall that the weighting factors for time and energy are linked i.e., their sum is equal to 1. Therefore, as the weighting factor for the time is increased, this decreases the weighting factor for energy. The figure 2 presents the optimal solution from linear programming and the three heuristic approaches.

At low weighting factors for time (or high weighting factors for energy) the optimum schedule preferentially offloaded jobs to save MD energy use; this resulted in queuing on MEC servers, with an increase time for data transmission and data reception. After 100 iterations, the closest matches to the minimum weighted score with low weighting factors for time were those generated by (O^{GS}, M^{W}) (2% higher). Over the entire range of weighting factors, the closest approaches were those generated by (O^{GS}, M^{J}) and (O^{GS}, M^{W}), both 9% greater than the solution obtained by linear programming.

[1]IBM CPLEX Linear programming problem solver: https://www.ibm.com/pt-en/products/ilog-cplex-optimization-studio

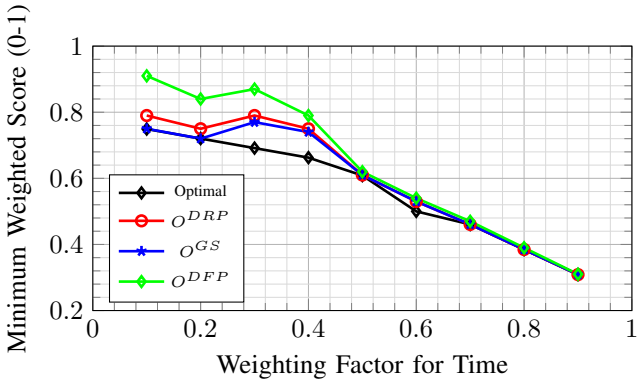

Figure 2: Performance of the heuristic approaches in a MEC network with 1 MD and 20 jobs. The solution for each heuristic approach was the best solution from 100 iterations. Solutions are presented for a MEC-offloading policy that yields best results.

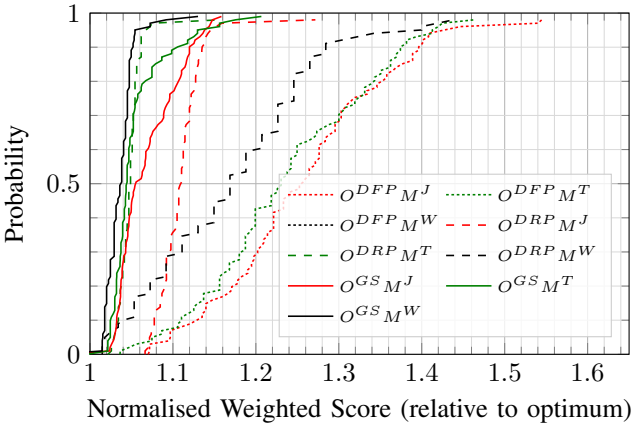

Figure 3: Cumulative Distribution Function graphs for heuristic algorithm outputs of schedule times and energy for Case 1 with w_t : 0.5 w_e : 0.5 (1 MD, 2 MECs servers, 20 jobs).

Cumulative Distribution Frequency plots for 8 of the algorithms with equal weighting factors for time and energy are shown in Figure 3. With this combination of weighting factors, 8 algorithms were within 7% of the guaranteed minimum weighted score. The use of (O^{DFP}, M^{W}) resulted in poor matches for minimum weighted scores.

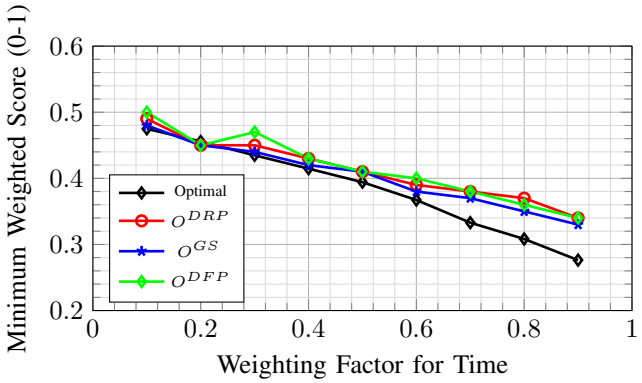

Figure 4: Performance of the proposed heuristic approaches in a MEC network with 10 MDs and 72 jobs. The solution for each heuristic approach was the best solution from 100 iterations. Solutions are presented for a MEC-offloading policy that yields best results.

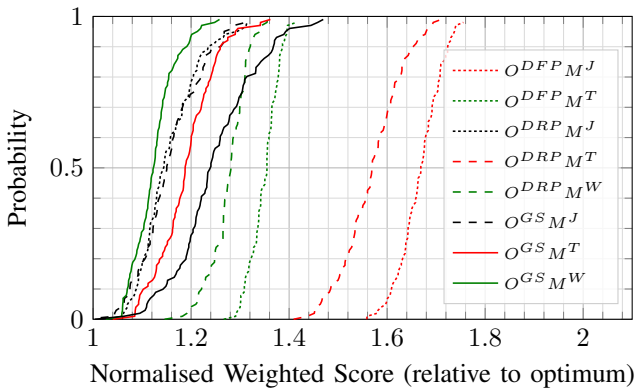

Figure 5: Cumulative Distribution Function graphs for heuristic algorithm outputs of schedule times and energy for Case 2 with w_t : 0.5 w_e : 0.5 (10 MDs, 3 MEC servers, 72 jobs).

C. Case 2: Offloading from 10 MDs

The heuristic algorithms were applied for offloading 72 jobs. The minimum normalised weighted score decreased continuously as the weighting factor for time increased, as shown in Figure 4. At the lowest and the highest weighting factors for time, the closest match to the minimum weighted scores were obtained by (O^{GS}, M^{W}). The difference between optimal solution and the heuristic solution in this case was +11% and +22%, respectively.

Cumulative Distribution Frequency plots for 8 of the algorithms with equal weighting factors for time and energy are shown in Figure 5. With this combination of weighting factors, 5 algorithms were within 8% of the guaranteed minimum weighted score.

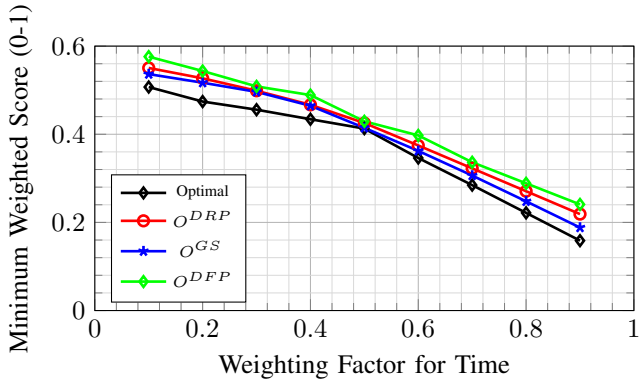

Figure 6: Performance of the proposed heuristic approaches in a MEC network with 13 MDs and 115 jobs. The solution for each heuristic approach was the best solution from 100 iterations. Solutions are presented for a MEC-offloading policy that yields best results.

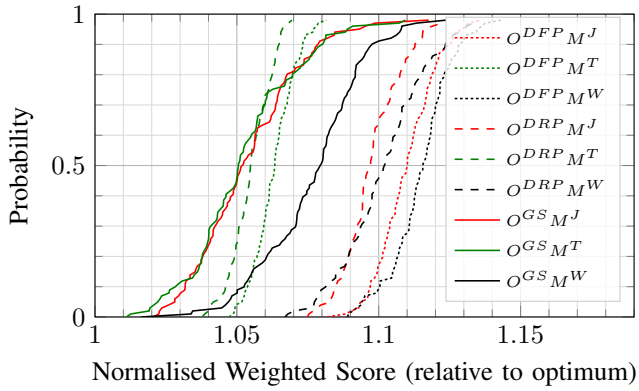

Figure 7: Cumulative Distribution Function graphs for heuristic algorithm outputs of schedule times and energy for Case 3 with w_t : 0.5 w_e : 0.5 (13 MDs, 3 MEC servers, 115 jobs).

D. Case 3: Offloading from 13 MDs

In this scenario, heuristic algorithms were simulated to solve the offloading problem of 115 jobs. As expected, Figure 6 shows the decreasing minimum normalised weighted score as the weighting factor for the time is increased.

Similar trend to case 2 is observed in case 3. That is at the lowest and at the highest weighting factor for time, the closest match to the optimal solution is produced by (O^{GS}, M^{W}). In this case, the difference between the optimal solution and the heuristic solution was +7% and +26%, respectively.

Cumulative Distribution Frequency plots for 8 of the algorithms with equal weighting factors for time and energy are shown in Figure 7. With this combination of weighting factors, all algorithms were within 10% of the guaranteed minimum weighted score.

V. Discussion

The heuristic algorithms that we have developed in this paper demonstrate that by using flexible approaches to computational offloading that incorporate both time and MD energy use. The time and energy weighting factors provide a convenient way to model users' preferences and check the sensitives of a solution. In general, low time weighting factors resulted in higher minimum weighted scores; greater emphasis on energy use encouraged more jobs to be offloading, and this caused queuing at MEC servers, which significantly increased task completion times. In contrast, higher emphasis on time-saving resulted in more parallel processing, using the full computational resources of the MEC network.

For the user of a MD, the choice between time and local energy use will be determined by the individual circumstances; for example, low battery charge will favour low MD energy use. For a network with multiple devices attempting to connect, time is more favorable a parameter because this will reduce time occupancy on the servers. For both an individual user and a network resource allocator, the choice of the best heuristic algorithm will depend on the parameter of higher value (total task completion time or local energy use).

VI. Conclusions and Future Work

The paper proposed a heuristic offloading method that is capable of producing near-optimal solutions. The proposed methodology is tested for a range of policy choices on offloading decisions and choosing MEC. An optimal choice of policy is dependent on the nature of MEC network. The methodology is flexible and can accommodate users' choice for decision making. Our numerical demonstrations clearly show that computational offloading can benefit not only in execution times but also in energy savings.

The choice of weighting factors are crucial because time and energy considerations can be balanced and assessed relative to each other, depending on the priorities of individual users of MDs or of the network, especially when usage is high. Single-choice weighting factors cannot adapt to changing circumstances and represent a sub-optimal approach [12].

A key advantage of heuristic approaches is its scalability. The linear programming approach provides an optimal solution, however, it does not scale well with the network size. Future work will demonstrate the value of proposed heuristic approaches on large networks. Furthermore, authors plan to extend our analysis to include financial costs of offloading to subscription services to further explore how trade-offs between different parameters affect the offloading process at varying levels of user demand on MEC networks.

VII. Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council grant number EP/P510427/1, Toshiba Research Europe Limited, and the University of Bristol.

References

[1] A. Juntunen, M. Kemppainen, and S. Luukkainen, "Mobile computation offloading - factors affecting technology evolution." in *International Conference on Mobile Business (ICMB)*, 2012, pp. 137–148.

[2] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 1–16, 2018.

[3] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, and L. Qian, "Computation offloading in multi-access edge computing networks: A multi-task learning approach," in *2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[4] D. Kovachev and K. Ralf, "Framework for computation offloading in mobile cloud computing," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 1, no. 7, pp. 6–15, 2012.

[5] Y.-D. Lin, E. T.-H. Chu, Y.-C. Lai, and T.-J. Huang, "Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds," *IEEE Systems Journal*, vol. 9, no. 2, pp. 393–405, 2013.

[6] A. Pawar, V. Jagtap, and M. Bhamare, "Time and energy saving through computation offloading with bandwidth consideration for mobile cloud computing," in *Proceedings of the Third International Symposium on Women in Computing and Informatics*, 2015, pp. 527–532.

[7] A. R. Khan, M. Othman, A. N. Khan, J. Shuja, and S. Mustafa, "Computation offloading cost estimation in mobile cloud application models," *Wireless Personal Communications*, vol. 97, no. 3, pp. 4897–4920, 2017.

[8] X. Gu, L. Jin, N. Zhao, and G. Zhang, "Energy-efficient computation offloading and transmit power allocation scheme for mobile edge computing," *Mobile Information Systems*, vol. 2019, 2019.

[9] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing," *Sustainable Computing: Informatics and Systems*, vol. 21, pp. 154–164, 2019.

[10] D. Xu, Q. Li, and H. Zhu, "Energy-saving computation offloading by joint data compression and resource allocation for mobile-edge computing," *IEEE Communications Letters*, vol. 23, no. 4, pp. 704–707, 2019.

[11] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, "Energy-efficient computation offloading in vehicular edge cloud computing," *IEEE Access*, vol. 8, pp. 37 632–37 644, 2020.

[12] L. Liu, Z. Chang, X. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 832–837.

[13] R. Singh, S. Armour, A. Khan, M. Sooriyabandara, and G. Oikonomou, "Heuristic approaches for computational offloading in multi-access edge computing networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2020 (PIMRC): Track 4: Applications and Business*, 2020, pp. 1–6.

[14] J. A. Suradkar and R. Bharati, "An effective computation offloading from mobile devices to cloud," *International Journal of Computer Science and Information Technologies*, vol. vol 7, pp. 1922–1927, 2016.

[15] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 160–164.

[16] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.