# Adaptive Task Offloading over Wireless in Mobile Edge Computing

Xiaojie Zhang
xzhang6@gradcenter.cuny.edu
City University of New York
New York, NY

Saptarshi Debroy
saptarshi.debroy@hunter.cuny.edu
City University of New York
New York, NY

## ABSTRACT

In energy-aware mobile edge computing systems, offloading real-time application tasks to remote edge nodes may become counter-productive as frequent fluctuations in wireless channels that are used for task offloading cause overall task execution time to increase. In this paper, we propose an adaptive task offloading algorithm to optimize and balance energy consumption at end-devices and overall task execution time.

## CCS CONCEPTS

• **Networks** → **Cloud computing**; *Network management*; • **Human-centered computing** → *Mobile computing*; • **Hardware** → *Wireless devices.*

## KEYWORDS

Mobile edge computing, Task offloading, Real-time applications, Energy efficiency, Wireless spectrum.

**ACM Reference Format:**
Xiaojie Zhang and Saptarshi Debroy. 2019. Adaptive Task Offloading over Wireless in Mobile Edge Computing. In *SEC '19: ACM/IEEE Symposium on Edge Computing, November 7–9, 2019, Arlington, VA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3318216.3363328

## 1 INTRODUCTION

With real-time mobile applications becoming increasingly compute-intensive for many mission-critical use cases, the energy capacity of embedded mobile end-devices are proving to be insufficient to handle all in-device computation. Mobile edge computing (MEC) [1, 2] allows mobile devices to offload some or all of such real-time and compute-intensive tasks to edge nodes. The advantage of MEC is that it brings cloud-scale compute resources closer to the mobile devices. Thus ideally, in order to preserve the limited energy of mobile devices, all computing tasks should be offloaded to edge nodes.

In many mission-critical use-cases [3], such offloading often uses wireless channels/spectrum for data transfer between mobile devices and edge nodes. However, inherent fluctuations of wireless channel quality caused by phenomenons such as, interference, path loss, shadowing, and fading result in varying data rate. This in turn adds to the data offloading latency that eventually increases the overall task execution time of mission-critical real-time applications. Thus in energy-aware task offloading MEC, task execution time costs may outweigh the energy preservation benefits of remote computation, which is inconsistent with the motivation to offload tasks.

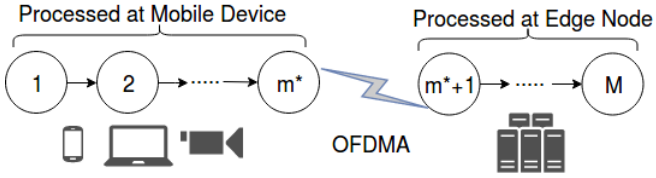


**Figure 1: Task partitioning and offloading model**

In this paper, we study the trade-off between task execution time and energy consumption at end-users under varying wireless channel conditions for soft real-time applications and involved tasks. We aim to find the optimal job partition between local (at mobile end-devices) verses remote edge nodes (as shown in Fig. 1) and optimal task assignment in terms of edge node selection that balances energy consumption and task execution time. We propose a Genetic Algorithm (GA) with constrained mutation for optimal job partitioning. We design a two-sided matching game for task assignment optimization and propose an Edge-Proposing Deferred Acceptance (EPDA) algorithm to solve the preference based matching game. Using simulation, we show how the GA and EPDA optimize and balance energy consumption and task execution time.

## 2 SYSTEM MODEL AND PROBLEM FORMULATION

### 2.1 Application Model

In our application model, a mobile application is represented as task $\tau_n$ with a Sporadic Directed Acyclic Graph (DAG) [4] $G_n = (V_n, E_n)$. Vertices set $V_n$ denotes $M_n$ *sequential* jobs executing task $\tau_n$. Each DAG-job $v_n^m = (\alpha_n^m, \omega_n^m, \beta_n^m) \in V_n$ has: i) computation requirement denoted by $\omega_{m,n}$ (i.e., the number of CPU cycles) and ii) input and output data denoted by $\alpha_n^m$ and $\beta_n^m$ (i.e., in bits) respectively. As shown in Fig. 1, each DAG is partitioned into two groups (local-only and remote-only) to achieve specific cost optimization. Edges describe the inter-job communication between jobs $m$ and $m+1$ with $\beta_n^m = \alpha_n^{m+1}$. The minimum release period and deadline for task $n$ are denoted by $T_n$ and $D_n$ respectively.


*SEC '19, November 7–9, 2019, Arlington, VA, USA*

ACM ISBN 978-1-4503-6733-2/19/11.
https://doi.org/10.1145/3318216.3363328

## 2.2 Transmission Model

For our application data transmission (from end-devices to edge nodes) model, we assume that every edge node has an Orthogonal Frequency Division Multiple Access (OFDMA) wireless channel of fixed bandwidth $B_k$ that can be equally divided into multiple orthogonal sub-channels and the node can allocate each such sub-channel to a data transmission from end-devies to that node [5, 6]. Given a fixed bandwidth $B_k$, the data rate is computed as $r_{up}(n,k) = \varphi_k B_k \log_2(1 + p_n h_{n,k}^2/\varphi_k N_0)$ where $\varphi_k = 1/\sum_{n=1}^{N} o_{n,k}$ is the fair bandwidth allocation coefficient, $p_n$ is the device transmission power, $h_{n,k}$ is the channel gain, and $N_0$ is the white Gaussian noise. Although $B_k$ is fixed, the value of $r_{up}(n,k)$ may vary based on channel and environment characteristics, such as, interference, path loss, shadowing, and fading. Thus, the data offloading/transmission time of each task $t_n(m_n^*) = \beta_n^{m^*}/r_{up}(n,k)$ and the energy consumption at mobile device in doing so $\varepsilon_d(m_n^*) = t_n(m_n^*)p_n$ can also fluctuate making the task offloading counter-productive.

## 2.3 Computation Model

In our model, we define $\mathcal{K} = \{1, 2, ..., K\}$ and $\mathcal{N} = \{1, 2, ..., N\}$ as the sets of edge nodes and end-devices (tasks) respectively. Every edge node and end-device have processing speeds denoted by $f_k^{edge}$ and $f_n^{local}$ respectively (i.e., in CPU cycles). We apply the computation model where the job scheduling follows well-known Earliest Deadline First (EDF) algorithm [7, 8]. For EDF, the task density must be no greater than 1 to ensure successful execution of all subsequent jobs. The task densities at mobile device and edge node are denoted as $\lambda_n^{local}$ and $\lambda_k^{edge} = \sum_{n=1}^{N} o_{n,k}\lambda_n^{remote}$ respectively. Here $o_{n,k} \in \{0, 1\}$ is the binary variable for the assignment of task $n$ to edge node $k$ where $o_{n,k} = 1$ denotes successful assignment and $o_{n,k} = 0$ otherwise. We denote the function $w(x) = \sum_{m=1}^{x} \omega_n^m$ as the accumulated computation from first to job $x$. The task density and accumulated computation follows the relation $\lambda = w(\cdot)/(f \times min(T, D))$. The overall computation time and energy consumption thus can be expressed as $c_n(m_n^*) = w(m_n^*)/f_n^{local} + (w(M_n) - w(m_n^*))/f_k^{edge}$ and $\varepsilon_c(m_n^*) = \kappa \times w(m_n^*) \times (f_n^{local})^2$ where $\kappa$ is a constant related to the chip architecture [5, 6].

## 2.4 Problem Formulation

For this paper, the two problems that we aim to solve in terms of cost optimization are: (1) DAG-jobs partition and (2) Task assignment. For DAG-jobs partition, each task has a decision tuple $< m_n^*, O_n >$ where $m_n^*$ denotes the DAG-job partition (i.e., offloading $w(M_n) - w(m_n^*)$ amount of computations to the edge) and $O_n = \{o_{n,k}\}$ is the binary vector for task assignment that follows the constraint $\sum_{k=1}^{K} o_{n,k} = 1$ (each task is assigned to only one edge node). In this paper, we assume that the first job is lightweight and is always executed locally (i.e., simple I/O pre-possessing). Whereas, the motivation of task offloading is to reduce the overall energy consumption that can be expressed as $\varepsilon(m_n^*) = \varepsilon_d(m_n^*) + \varepsilon_c(m_n^*)$ and to satisfy (i.e., reduce) the execution time constraint denoted by $L_n(m_n^*) = t_n(m_n^*) + c_n(m_n^*)$. We define the satisfaction of task offloading for individual task as,

$$\zeta_{n,k} = \underbrace{(1-\gamma)(1 - \frac{L_n(m_n^*)}{D_n})}_{\textbf{time reduction}} + \underbrace{\gamma(1 - \frac{\varepsilon(m_n^*)}{\varepsilon_c(M_n)})}_{\textbf{energy reduction}} \tag{1}$$

and we aim to maximize the overall satisfaction of all tasks under limited network resources (i.e., bandwidth and computation). Therefore, our joint DAG-jobs partition and task assignment optimization problem can be stated as:

$$\begin{aligned} \underset{m_n^*, o_{n,K}}{\text{Maximize}} \quad & \sum_{n=1}^{N}\sum_{k=1}^{K} o_{n,k}\zeta_{n,k} \\ \text{subject to} \quad & \sum_{k=1}^{K} o_{n,k} = 1, \ n \in \mathcal{N} \\ & \lambda_n^{local} \le 1, \ n \in \mathcal{N} \\ & \lambda_k^{edge} \le 1, \ k \in \mathcal{K} \\ & o_{n,k} \in \{0, 1\}, \ n \in \mathcal{N}, k \in \mathcal{K} \\ & 1 \le m_n^* \le M_n, \ n \in \mathcal{N} \end{aligned} \tag{2}$$

where $0 \le \gamma \le 1$ characterizes the trade-off between energy consumption and execution time (i.e., $\gamma_n = 1$ denoting energy-driven task offloading policy).

# 3 BASELINE APPROACH

Here we introduce a heuristic approach to solve the proposed optimization problem. To reduce the problem size, we define $\chi_n$ as the set of feasible candidates that satisfies the following two constraints: $\beta_n^m \le \beta_n^1$ and $\lambda_n^{local} \le 1$ ($m \in \chi_n$). These constrains signify that if $\beta_n^1$ is the smallest data size of inter-job communication, then $\chi_n = \{1\}$. The partition is solved by a Genetic Algorithm (GA) with constrained mutation range of partition candidates $\{\chi_n\}^N$. We apply a two-sided matching game (many-to-one) to perform task-to-edge matching. In the matching game, every edge node $k$ has a certain capacity $c_k$ and a strict preference $\succ_k$ over $\mathcal{N}$. Also, every task $n$ has strict preference $\succ_n$ over $\mathcal{K}$. We propose an Edge-Proposing Deferred Acceptance (EPDA) algorithm (Algo. 1) to solve the preference based matching game.

## 3.1 Adaptive Offloading Preference

For EPDA algorithm, we use an adaptive preference function for task offloading decision. The preference $\succ_n$ over $\mathcal{K}$ follows the order of satisfaction of task offloading $\zeta_{n,k}$. The preference $\succ_k$ over $\mathcal{N}$ is simply based on wireless channel quality $\zeta_{k,n} = h_{n,k}$. Both preference lists are established in a non-decreasing order of $\zeta_{n,k}$ or $\zeta_{k,n}$. Based on this, our EPDA algorithm chooses $s = \sum_{n=1}^{N}\sum_{k=1}^{K} \mu(n,k)\zeta_{n,k}$ as the fitness function where matching $\mu(n,k)$ equals $o_{n,k}$.

## 3.2 Load Balancing

On the other hand, the optimization problem solution should avoid any network traffic and computational congestion that could have a significant negative impact to the overall performance. In order to achieve this, EPDA sets the capacity $c_k = \lambda_k^{edge}$ and follows the concept of equal load measured by task density, i.e., each edge node gets a workload proportional to its capacity (e.g., bandwidth and

CPU cycle(s)). In EPDA, if assigning a task to an edge node $k$ does not result in a $\lambda_k^{edge} \geq 1$, the task is considered acceptable for that node.

**Algorithm 1:** Edge-Proposing Deferred Acceptance

1 Initialize temporary matching $\mu$ to be empty
2 Initialize the set $U = \{k\} \; \forall \; \lambda_k^{edge} < 1, \; k \in \mathcal{K}$
3 Calculate $\zeta_{n,k}$,$\zeta_{k,n}$ and build preference lists $\succ_n$ and $\succ_k$, $n \in \mathcal{N}$ and $k \in \mathcal{K}$
4 **while** *there exists* $k \in U$ *that still has acceptable tasks to propose* **do**
5 $k$ proposes to its favorite acceptable tasks among those it has yet to propose to.
6 $\mu(n) \leftarrow$ Match/re-match task $n$'s most favorite edge node among the ones which proposed to it.
7 Modify preference lists $\succ_n$ based on updated $\zeta_{n,k}$ caused by matched pairs (i.e. $\varphi_k$), $n \in \mathcal{N}$.
8 Calculate fitness score $s$ for $\mu$ and feed to GA.
9 **return** score $s$

## 4 PRELIMINARY RESULTS

We evaluate the performance of the proposed task offloading scheme using a simple yet realistic simulation. The system and network parameters are shown in Table 1 and are modeled based on [6]. The results in Fig. 2 show the efficiency of task offloading against different schemes. Compared to schemes like 1-RND (random assignment) and 1-EPDA which offload the entire DAG-jobs to edge nodes (i.e., $m_n^* = 1$), our scheme GA-EPDA selects a combination of DAG-jobs partition and task assignment based on highest fitness scores and achieves higher performance. In addition, GA-EPDA can characterize the trade-off between energy consumption and execution time as shown in load-unbalanced results (Fig. 2 (B) and (C)). It also shows that $\gamma$ should be set close to zero (denoting execution time-driven policy) when there exist failed tasks.

**Table 1: Simulation Parameters**

| Name | Value | Name | Value |
|---|---|---|---|
| K | 10 | N | 20 – 40 |
| $f_n^{local}$ | 0.5 – 1.5 GHz | $\kappa$ | $10^{-28}$ J/cycle |
| $f_k^{edge}$ | 10 – 20 GHz | $B_k$ | 50 – 100 MHz |
| $M_n$ | 3 – 6 stages | $\alpha, \beta$ | 50 – 1000 KB |
| $\omega$ | 5 – 250 cycles/bit | $T_n$,$D_n$ | 300 – 1000 ms |

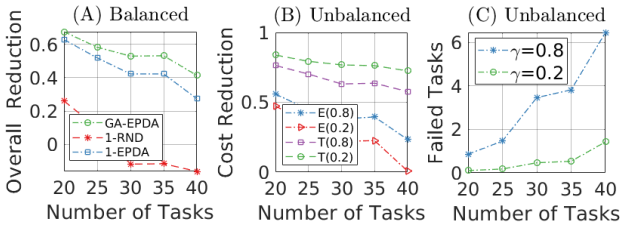


**Figure 2: Task offloading against different schemes.**

## REFERENCES

[1] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, Q. Li, "LAVEA: Latency-Aware Video Analytics on Edge Computing Platform," *Proc. of IEEE ICDCS*, 2017.
[2] Z. Dong, Y. Liu, H. Zhou, X. Xiao, Y. Gu, L. Zhang, C. Liu, "An energy-efficient offloading framework with predictable temporal correctness" *Proc. of ACM/IEEE SEC*, 2017.
[3] C-C. Hung, G. Ananthanarayanan, P. BodÃŋk, L. Golubchik, M. Yu, P. Bahl, M. Philipose, "VideoEdge: Processing Camera Streams using Hierarchical Clusters," *Proc. of ACM/IEEE SEC*, 2018.
[4] V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, A. Wiese, "Feasibility Analysis in the Sporadic DAG Task Model," *Proc. of Euromicro RTS*, 2013.
[5] A. Al-Shuwaili, O. Simeone, "Energy-Efficient Resource Allocation for Mobile Edge Computing-Based Augmented Reality Applications," *IEEE Wireless Communications Letters*, 6 (3), 398-401, 2017.
[6] C. You, K. Huang, H. Chae, B. Kim, "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," *IEEE Transactions on Wireless Communications*, 16 (3), 1397-1411, 2017.
[7] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, D. O. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel," *IEEE Transactions on Wireless Communications*, 12 (9), 4569-4581, 2013.
[8] J. M. Lopez, M. Garcia, J. L. Diaz, D. F. Garcia, "Worst-case utilization bound for EDF scheduling on real-time multiprocessor systems," *Proc. of Euromicro RTS*, 2000.
[9] Y. Mao, J. Zhang, S. H. Song, K. B. Letaief, "Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems," *IEEE Transactions on Wireless Communications*, 16 (9), 5994-6009, 2017.