Optimal Code Partitioning Over Time and Hierarchical Cloudlets

Abbas Kiani, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*— This letter proposes a task scheduling scheme designed for code partitioning over time and the hierarchical cloudlets in a mobile edge network. To this end, we define the so called energy-time cost parameters to optimally schedule tasks over time and hierarchical cloudlet locations. Accordingly, we investigate two different optimization scenarios. In particular, the first scenario aims at finding the optimal task scheduling for given radio parameters. In the second scenario, we carry out the optimization of both the task scheduling and the mobile device's transmission power. More importantly, we show that by adopting the proposed code partitioning scheme in this letter, the transmission power optimization problem becomes a disjoint problem from the task scheduling problem.**

***Index Terms*— Hierarchical mobile edge computing, computation offloading.**

I. Introduction

THE Mobile Edge Computing (MEC) is recognized as one of the key emerging technologies for 5G networks and aims at providing computing capabilities within the Radio Access Network (RAN) and in proximity of mobile users [1]. In the past few years, the MEC architecture and service management in MEC has been widely researched, and a variety of policies and algorithms such as [2] have been proposed. However, it is still desirable to investigate new networking architectures and efficient computation offloading models that better suits the MEC concept.

Computation offloading requires code partitioning to decide which tasks should be executed locally and which tasks should be offloaded to the mobile edge depending on different parameters such as energy and delay. Existing computation offloading problems in the literature such as [3] and [4] propose joint optimization framework for the code partitioning problem and the radio resource optimization. Such joint optimization frameworks lead to Mixed Integer Nonlinear Programming (MINLP) models in which finding the optimal solution requires an exhaustive search over all the useful call graph partitions, i.e, all the configurations that satisfy the feasibility conditions. Accordingly, these schemes propose to find sub-optimal solutions for code partitioning and then optimize the radio resources for a given partitioning. A message-passing approach for the same problem is proposed in [5] which reduces the complexity of the computation offloading problem. However, the proposed model in [5] considers the code partitioning problem between a mobile device and only one remote location. In this letter, inspired by distributed processing systems [6], we propose to use the shortest tree algorithm to optimally schedule tasks in mobile edge networks. More importantly, we extend the code partitioning problem to scheduling problem over time and a hierarchical mobile edge. To this end, we investigate two different optimization scenarios. In particular. the first scenario aims at finding an optimal task scheduling for given radio parameters. In the second scenario, we investigate joint optimization of task scheduling and the mobile device's transmission power, and show that by using the proposed scheduling scheme, the transmission power optimization problem becomes a disjoint problem from the task scheduling problem.

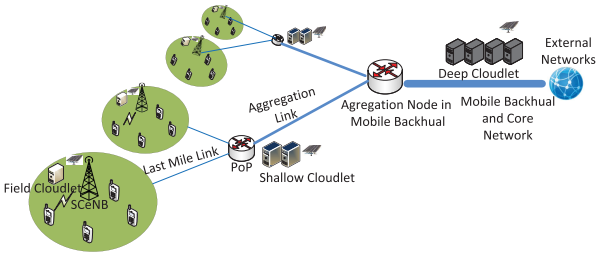

Fig. 1. System model.

The rest of the letter is organized as follows. Section II describes the system model and problem formulation. We investigate our task scheduling scheme and the corresponding scenarios in Section III. Finally, Sections IV and V present numerical results and conclude the letter, respectively.

II. System Model and Problem Formulation

We consider a Hierarchical Mobile Edge Computing (HI-MEC) architecture [7] shown in Fig. 1. The HI-MEC architecture consists of field, shallow and deep cloudlets. In particular, in a HI-MEC environment, the field cloudlets as the resource-poor facilities are co-located with Small Cell enhanced Node Bs (SCeNBs). The shallow cloudlets as the resource-modest facilities are also hosted at the first level of aggregation nodes, i.e., at Point of Presences (PoPs). Moreover, a resource-rich facility called the deep cloudlet is located at the mobile backhaul. We consider a two-time scale model in which the running time of the HI-MEC environment is divided into a sequence of time frames at equal length, T, e.g., five minutes. Each time frame itself is also divided into a sequence of time slots at equal length, τ, e.g., one minute. We assume one time frame consists of N time slots and denote $t_0, \ldots, t_{N-1}$ as the set of time slots in a time frame. At the beginning of each time frame, each SCeNB broadcasts the available computational

Manuscript received September 6, 2017; revised September 21, 2017; accepted October 17, 2017. Date of publication October 20, 2017; date of current version January 8, 2018. This work was supported in part by NSF under grant no. CNS-1647170. The associate editor coordinating the review of this paper and approving it for publication was D. W. K. Ng. *(Corresponding author: Abbas Kiani.)*

The authors are with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: abbas.kiani@njit.edu; nirwan.ansari@njit.edu).

Digital Object Identifier 10.1109/LCOMM.2017.2764904

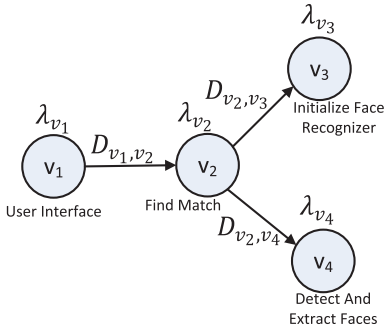

Fig. 2. Call tree.

capacities at the field, shallow and deep level to their MUs. In fact, a centralized controller at the deep cloudlet equipped with a data management model as well as a global view of the network predicts the workloads during the next few time slots and accordingly allocates the resources to the MUs. The centralized controller informs the SCeNBs about the allocated resources to each MU. The allocated resources within a time slot are assumed to be fixed but changing from a time slot to the next time slot. A MU's application is described by a call graph, i.e., a directed acyclic graph as $G = (\mathcal{V}; \mathcal{E})$. The call graph represents the relation among the tasks in which the MU's application can be partitioned. For example, the call graph of a face recognition application [8] is shown in Fig. 2. Each vertex represents a task v_i in the call stack and each edge $e = (v_i; v_j)$ shows an invocation of task v_j from task v_i. Each task node v_i is characterized by its workload, λ_{v_i}, i.e., the number of CPU cycles required to complete the execution of the task. Each edge $(v_i; v_j) \in \mathcal{E}$ is also characterized by the number of bits (D_{v_i,v_j}) that must be transferred from the parent task v_i to child task v_j. In the rest of the letter, we consider a given MU of interest in defining the corresponding notations. The MU can decide to execute a task locally at the mobile device or remotely at the available cloudlet locations. The MU's decision depends on two factors, energy and delay. The MU's energy consumption is the energy required to execute a task locally or to transmit the required bits to the remote cloudlet (when the parent task is executed locally and the child remotely), or to receive the required bits from the cloudlet (when the parent task is executed remotely and the child locally). On the other hand, the delay is the time required to execute the task locally or to transmit the required bits to the remote location. Therefore, not only the local parameters but also the remote parameters are contributing to the corresponding cost of each task, i.e, the energy and time costs. Here, the local parameters include transmit power P_{up}, reception power P_{rx}, local computational capacity μ_{loc} (in CPU cycles per second), and local processing power P_{loc}. Unlike the remote parameters, we assume that the local parameters are not changing time slot by time slot.

In terms of the wireless access parameters, we define C_{dl} and $C_{up}(P_{up})$ as the capacities of the downlink and the uplink channels between the MU and its associated SCeNB, respectively. The remote parameters are the available cloudlet locations for the MU, the computational capacities at the cloudlets and the data rates on the corresponding links. Let's assume the MU is associated with SCeNB s and $\mathcal{AC}$ is the set of all available remote locations, which provision a field, a shallow and the deep cloudlet. Let $\mu_x^{t_n}$ also be the remote computational capacity that can be assigned to the MU at cloudlet $x \in \mathcal{AC}$ during time slot t_n. Moreover, we assume $C_{s,x}^{t_n}$ is the maximum data rate that can be allocated to the MU between SCeNB s and the cloudlet x during time slot t_n. Similarly, $C_{x,y}^{t_n}$ is the maximum data rate that can be allocated to the MU between two cloudlets $x \in \mathcal{AC}$ and $y \in \mathcal{AC}$. It is assumed that a task is allowed to start execution only at the beginning of a time slot. However, the data from a parent task to a child task is assumed to be transferred as soon as the parent task execution is completed. We also assume that once a task starts executing during a time slot (once a data from a parent task to a child task starts transferring over the network), it is allowed to execute to completion (to transfer to completion) even if the time slot ends during execution (transfer) but with the same allocated computational capacity (data rate). Based on the defined local and remote parameters, we can translate the computation and communication requirements of the tasks and the edges on the MU's call graph to an energy-time cost parameter as follows,

$$ETC = \zeta_1(energy\ cost) + \zeta_2(time\ cost) \tag{1}$$

where ζ_1 and ζ_2 are two coefficients as the weights of the energy and time, respectively. The MU can flexibly choose the coefficients that favors more their demands. For example, a user with a low battery level may like to put more weight on the energy [9]. According to the proposed model, the MU not only has the option to execute a task at $|\mathcal{AC}|+1$ different local and remote computing locations (including the mobile device) but also in N time slots. Thus, the task offloading decision problem can be modeled as an assignment problem in a distributed processors system with $(|\mathcal{AC}|+1) \times N$ processors. In terms of the local ETC of a task, let's define $ETC_{loc}^{t_n}(v_i)$ as the ETC of task v_i when executed locally at the mobile device in time slot t_n. $ETC_x^{t_n}(v_i)$ is also defined as the ETC of task v_i when executed at location x during time slot t_n. Moreover, $ETC_{loc,x}^{t_n,t_m}(D_{v_i,v_j})$ is assumed to be the ETC between two tasks v_i executed locally at the mobile device during time slot t_n and v_j executed remotely at cloudlet x during time slot t_m for $m > n$ where D_{v_i,v_j} is the number of bits that must be transferred from task node v_i to v_j. $ETC_{x,loc}^{t_n,t_m}(D_{v_i,v_j})$ indicates the same ETC but v_i executed remotely at cloudlet x and v_j locally at the mobile device. Similarly, let $ETC_{x,y}^{t_n,t_m}(D_{v_i,v_j})$ be the ETC between two tasks v_i executed remotely at cloudlet x during time slot t_n and v_j executed at cloudlet y during time slot t_m. Then, we can calculate the following ETCs,

$$ETC_{loc}^{t_n}(v_i) = \zeta_1 P_{loc}(\frac{\lambda_{v_i}}{\mu_{loc}}) + \zeta_2(n\tau + \frac{\lambda_{v_i}}{\mu_{loc}}) \tag{2}$$

$$ETC_x^{t_n}(v_i) = \zeta_2(n\tau + \frac{\lambda_{v_i}}{\mu_x^{t_n}}) \tag{3}$$

$$ETC_{loc,x}^{t_n,t_m}(D_{v_i,v_j}) = \begin{cases} \zeta_1 P_{up}(\frac{D_{v_i,v_j}}{C_{up}(P_{up})}) + \zeta_2(D_{v_i,v_j} \\ (\frac{1}{C_{up}(P_{up})} + \frac{1}{C_{s,x}^{tl}})), & m \geq k+1 \\ \infty, & m < k+1 \end{cases} \tag{4}$$

where $l = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu_{loc}} \rfloor$ and $k = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu_{loc}} + \frac{D_{v_i,v_j}}{\tau}(\frac{1}{C_{up}(P_{up})} + \frac{1}{C^{l_l}_{s,x}}) \rfloor$.

$$ETC^{t_n,t_m}_{x,loc}(D_{v_i,v_j}) = \begin{cases} \zeta_1 P_{rx}(\frac{D_{v_i,v_j}}{C_{dl}}) + \zeta_2(D_{v_i,v_j}(\frac{1}{C_{dl}} + \frac{1}{C^{l_{l'}}_{s,x}})), & m \geq k'+1 \\ \infty, & m < k'+1 \end{cases} \quad (5)$$

$$ETC^{t_n,t_m}_{x,y}(D_{v_i,v_j}) = \begin{cases} \zeta_2(\frac{D_{v_i,v_j}}{C^{l_{l'}}_{x,y}}), & m \geq k''+1 \\ \infty, & m < k''+1 \end{cases} \quad (6)$$

where $l' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu^{t_n}_x} \rfloor$, $k' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu^{t_n}_x} + \frac{D_{v_i,v_j}}{\tau}(\frac{1}{C_{dl}} + \frac{1}{C^{l_{l'}}_{s,x}}) \rfloor$ and $k'' = \lfloor n + \frac{\lambda_{v_i}}{\tau \mu^{t_n}_x} + \frac{D_{v_i,v_j}}{\tau C^{l_{l'}}_{x,y}} \rfloor$.

$$ETC^{t_n,t_{m\geq l+1}}_{local,local}(D_{v_i,v_j}) = ETC^{t_n,t_{m\geq l'+1}}_{x,x}(D_{v_i,v_j}) = 0 \quad (7)$$

Moreover, some of the tasks in a call graph are required to be executed locally. For example, the user interface task in Fig. 2 which initiates the application must be executed locally at the mobile device. Therefore, the ETC of executing such tasks remotely is set to infinity. Based on the defined ETC parameters, the code partitioning problem over time and hierarchical cloudlets can be formulated as the following MINLP

$$\begin{aligned} \underset{0\leq P_{up}\leq P_{max},\ I^{x,t_n}_{v_i}\in\{0,1\}}{\text{minimize}} \quad & \sum_{v_i\in\mathcal{V}}\sum_{x\in\mathcal{AC}'}\sum_{n=0}^{N-1} I^{x,t_n}_{v_i} ETC^{t_n}_{x}(v_i) \\ & + \sum_{(v_i;v_j)\in\mathcal{E}}\sum_{x\in\mathcal{AC}'}\sum_{y\in\mathcal{AC}'}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} I^{x,t_n}_{v_i} \\ & \times I^{y,t_m}_{v_j} ETC^{t_n,t_m}_{x,y}(D_{v_i,v_j}) \\ s.t. \quad & \sum_{x\in\mathcal{AC}'}\sum_{n=0}^{N-1} I^{x,t_n}_{v_i} = 1 \quad \forall v_i \in \mathcal{V} \\ & \sum_{v_i\in\mathcal{V}} I^{x,t_n}_{v_i} = 1 \quad \forall x \in \mathcal{AC}',\ n = 1,\ldots,N-1 \end{aligned} \quad (8)$$

where $I^{x,t_n}_{v_i} = 1$ if task v_i is executed at cloudlet x during time slot t_n, and $I^{x,t_n}_{v_i} = 0$ otherwise. Set $\mathcal{AC}'$ is also the set of all cloudlets plus the mobile device.

III. Optimal Hierarchical Task Scheduling

Note that (8) defines a mixed integer program which involves binary and real variables. Finding an optimal solution to this problem requires an exhaustive search over all the useful code partitions and entails a complexity that is exponential in the number of tasks. Therefore, we investigate an optimal scheduling scheme to solve problem (8) for two optimization scenarios. In the first scenario, we are interested in finding an optimal task scheduling for given radio parameters, i.e., the case that variable P_{up} in the optimization problem (8) is fixed. In the second scenario, beside finding the optimal scheduling, we also optimize the transmission power at the mobile device, i.e., P_{up}. Note that in the scheduling scheme to be presented in this section, it is assumed that the MU's call graph is a directed tree.

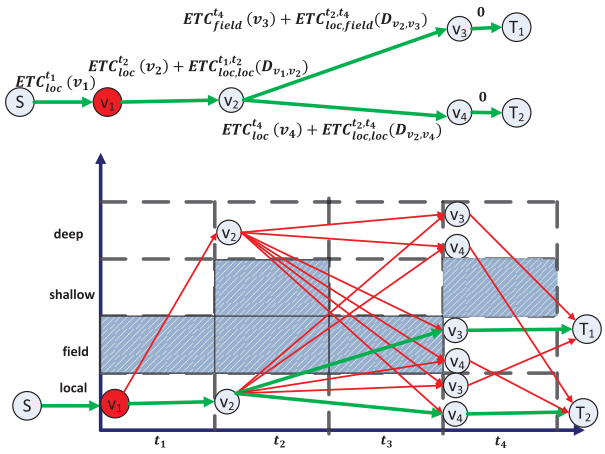

Fig. 3. Scheduling graph and one of the corresponding assignment trees.

A. Optimal Scheduling for Given Radio Parameters

Fig. 3 shows an scheduling graph for a time frame consisting of four time slots and one of its corresponding assignment trees. Each node of the scheduling graph corresponds to the execution of a task in a given time slot and at a given cloudlet location. As shown in Fig. 3, in time slot t_1, the local, shallow and deep cloudlets are all available to execute task v_1. However, as task v_1 initiates the application, it is required to be executed locally. Therefore, task v_1 is scheduled only at the local location. In time slot t_2, while the local and the deep locations are available to execute task v_2, the field and shallow locations are unavailable due to for example peak load at the corresponding SCeNBs. Accordingly, task v_2 is scheduled to be executed either locally or at the deep cloudlet. Moreover, we assume that the execution of task v_2 takes more than the duration of one time slot. Therefore, no matter which locations are available during time slot t_3, child tasks v_3 and v_4 have to wait until the execution of parent task v_2 is completed, i.e., time slot t_4. Then, tasks v_3 and v_4 can be scheduled at the local, field and deep locations. We assume that two tasks cannot be scheduled at the same cloudlet location in one time slot. In fact, if task v_3 is scheduled to be executed locally, task v_4 has to be executed either at the field cloudlet or the deep cloudlet. An assignment graph also has some distinguished nodes including one source node and several terminal nodes. In particular, there is one terminal node for each leaf node of the call tree.

Note that each scheduling of the tasks to different cloudlet locations and different time slots corresponds to a subgraph of the scheduling graph. The subgraph plus the source and the terminal nodes is called an assignment tree, and it connects the source node to all the terminal nodes. The weight of an edge on the assignment tree connecting parent task v_i, executed at cloudlet x during time slot t_n, to child task v_j, executed at cloudlet y during time slot t_m, is equal to $ETC^{t_m}_{y}(v_j) + ETC^{t_n,t_m}_{x,y}(D_{v_i,v_j})$. The ETC of the source and the terminal nodes as well as the weight of the edges that connect the leaf tasks to the terminal nodes are assumed to be zero (see the assignment tree in Figure 3). Moreover, the weight of each assignment tree which indicates the ETC of that assignment is established by the sum of the weights of

all edges in it. Therefore, the optimal assignment corresponds to the assignment tree which has the minimum weight. The minimum weight assignment tree of an application, which involves M tasks, N time slots, and $|\mathcal{AC}|+1$ cloudlet locations, can be found by dynamic programming with complexity $O(M \times N^2 \times (|\mathcal{AC}|+1)^2)$ [6].

B. Optimal Scheduling While Optimizing the Transmission Power

In this section, we are interested in both finding the optimal scheduling and optimizing the transmission power at the mobile device, i.e., P_{up}. We show in the following theorem that the optimal scheduling and the optimization of the transmission power are disjoint optimization problems that can be solved independently.

Theorem 1: The scheduling optimization problem and the transmission power optimization are disjoint optimization problems.

Proof: We first assume that the transmission power is given. Then, following the optimal scheduling scheme for given radio parameters, the optimal scheduling corresponds to the assignment tree that has the minimum weight. On the other hand, according to the defined ETCs, factors $\frac{\zeta_1 P_{up}+\zeta_2}{C_{up}(P_{up})}$ appear on the weight of an assignment tree. Therefore, one can first minimize $\frac{\zeta_1 P_{up}+\zeta_2}{C_{up}(P_{up})}$ by optimizing P_{up} and then find the optimal scheduling for the given optimal P_{up}. The proof is complete. ■

Therefore, we propose a disjoint optimization framework in which we first solve the following optimization problem to find the optimal transmission power,

$$\underset{0\le P_{up}\le P_{max}}{\text{minimize}} \quad \frac{\zeta_1 P_{up}+\zeta_2}{C_{up}(P_{up})} \tag{9}$$

Then, we follow the optimal scheduling scheme in the previous section for the given optimal transmission power. The optimization problem in (9) becomes strictly convex with the change of variables $Z = C_{up}(P_{up})$ [4] and thus can be solved by efficient convex optimization techniques.

IV. Simulation Results

In this section, we evaluate the results of the proposed optimal code partitioning scheme. To this end, we consider the call tree of Fig. 2 and assume $\lambda_{v_1} = 2$ M cycles. We also set $\lambda_{v_2} = 18.1$ M cycles, $\lambda_{v_3} = 92.6$ M cycles, $\lambda_{v_4} = 256.1$ M cycles, $D_{v_1,v_2} = 182$ kB, $D_{v_2,v_3} = 4675$ kB and $D_{v_2,v_4} = 13860$ kB [8]. Moreover, we consider an scheduling graph consisting of four time slots and we assume the computational capacity of a cloudlet location during a time slot is fixed and between 10 to 14 G cycles per second if it is available, and is equal to zero otherwise. The local computational capacity is also set to 100 M cycles per second. In terms of the uplink channel, we set the channel bandwidth to 5 MHz, the transmission power budget constraint to 100 mW, and the background noise to -100 dBm [9]. For performance evaluations, we define the normalized energy-time gain as $\frac{ETC_{all\ \ locall}-ETC_{code\ \ partitioning}}{ETC_{all\ \ locall}}$ where $ETC_{all\ \ locall}$ is the ETC incurred if all the tasks are executed locally. Fig. 4 compares the normalized energy-time gain of the proposed scheme with MINLP model. The OPTI toolbox combined with NOMAD, which is an excellent derivative free MINLP solver, is used to solve the MINLP problem. As demonstrated in Fig. 4, the proposed scheme performs better than the MINLP model. This result is attributed to the fact that in the proposed scheme we first optimize P_{up} and then carry out the scheduling optimization for the optimal value of P_{up}.

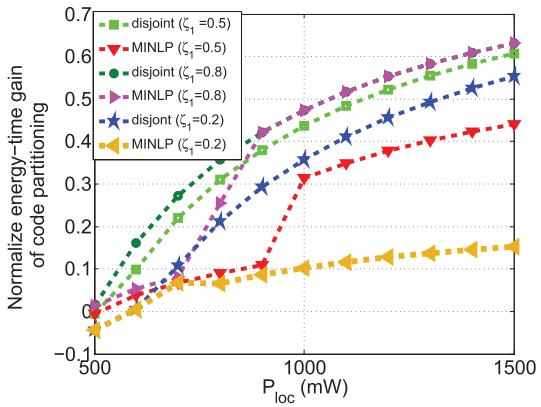

Fig. 4. Normalized energy-time gain of code partitioning versus local processing power.

V. Conclusion

In this letter, we have proposed a task scheduling scheme for offloading computation over time and the hierarchical mobile edge. To this end, we have studied two different optimization scenarios. In particular, in the first scenario, we have found an optimal task scheduling for given radio parameters. In the second scenario, we have investigated the joint optimization of task scheduling and the mobile device's transmission power in which we have showed that by using the scheduling task in this letter, the problem of optimizing the transmission power becomes a disjoint problem from the scheduling problem.

References

[1] Y. C. Hun, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper 11, 2015.

[2] X. Sun and N. Ansari, "Adaptive avatar handoff in the cloudlet network," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2017.2701794.

[3] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[4] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti. (2016). "Joint optimization of radio resources and code partitioning in mobile edge computing." [Online]. Available: https://arxiv.org/abs/1307.3835

[5] S. Khalili and O. Simeone, "Inter-layer per-mobile optimization of cloud mobile computing: A message-passing approach," *Trans. Emerg. Telecom. Tech.*, vol. 27, no. 6, pp. 814–827, 2016.

[6] S. H. Bokhari, "A shortest tree algorithm for optimal assignments across space and time in a distributed processor system," *IEEE Trans. Softw. Eng.*, vol. SE-7, no. 6, pp. 583–589, Nov. 1981.

[7] A. Kiani and N. Ansari, "Towards hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2017.2750030.

[8] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 49–62.

[9] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.