Geometric Programming for Lifetime Maximization in Mobile Edge Computing Networks

Sabyasachi Gupta, Jacob Chakareski
The University of Alabama, Tuscaloosa, AL 35487
Email: {sgupta6, jacob}@ua.edu. Web: http://lion.ua.edu

***Abstract*—Mobile edge computing has emerged as a promising technology to augment the computational capabilities of mobile devices. For a multi-user network in which its users periodically compute their tasks with the help of an edge cloud, we investigate the network lifetime maximization problem based on present user task information. We pursue this objective via a minimum energy efficiency maximization (MEEM) strategy that jointly optimizes the fraction of user task computations offloaded to the cloud and the respective allocation of edge computing and network communication resources across the users. We also investigate the network lifetime maximization problem for the case when the user task information is available for all future time slots, as well. This setting represents an upper bound for the MEEM strategy. Optimal solutions for both investigated strategies are formulated via feasibility testing and geometric programming. We show that MEEM can achieve a 70% lifetime improvement over the state-of-the-art and 450% lifetime improvement over the case of local user task computation only.**

I. Introduction

As mobile devices are gaining enormous popularity over the last decade, many new applications, e.g., face/fingerprint/iris recognition, augmented reality, natural language processing, and interactive gaming have emerged and attracted great attention. Due to the requirements of high reliability, intensive computing, and low latency for these applications, the concept of Mobile-Edge Computing (MEC) has emerged [1]. In MEC based system, small-scale cloud-computing facilities are available at the edge of pervasive radio access networks in close proximity to the mobile users [1].

Since wireless devices have limited battery energy, energy efficiency is a crucial design parameter for cooperative wireless networks. Significant effort has been made to date to investigate maximizing the lifetime of such networks [2–4]. It has been shown [3] that the wireless nodes' residual battery energy information must be taken into consideration to decide the transmit power control, relay selection, and channel allocation, so that the overall network lifetime is improved.

For wireless networks in which the nodes have computationally intensive tasks with low latency requirements, offloading them to the edge cloud may improve the network energy efficiency [1, 5–11]. [7] investigates a weighted sum energy consumption minimization scheme in mobile-edge computing networks, by jointly optimizing the load and communication resource allocation. For a multi-server mobile-edge computing network, [10] studies a joint computation resource allocation, transmit power allocation, and task offloading decision optimization, to minimize a system utility casted as a weighted function of task completion time and task energy consumption.

This work has been supported in part by NSF Awards CCF-1528030, ECCS-1711592, CNS-1836909, and CNS-1821875.

To improve the lifetime of a mobile-edge computing network with finite communication and computation resource, the decisions on resource allocation, for the users, need to be made based on their residual battery energy. For example, a node with low residual battery energy should be allocated high communication and computation resources, so that it can compute its task with low energy consumption. *No previous study has considered residual battery energy information, to allocate computation and communication resources in mobile edge computing networks. Similarly, to the best of our knowledge, the lifetime maximization problem has not been studied for such networks.* These are the objectives we pursue here.

Aiming to maximize the network lifetime, we investigate the joint optimization of sharing computation between the users and the edge cloud, and allocating communication and edge computing resources for each user. The network lifetime is defined as the time interval for which each user can compute his task within a maximum tolerable delay and none of the users is depleted of energy. Our main contributions are the following: i) Aiming to maximize the network lifetime based on user task information for the present time slot only, we explore an MEEM strategy for joint optimization of the fraction of user task computations offloaded to the cloud and the respective allocation of edge computing and network communication resources across the users; ii) We optimally solve the network lifetime maximization problem when future user task information is available; this setting is an upper bound for MEEM; iii) We show that MEEM performs close to the optimal network lifetime, for low initial user battery energy; and iv) We show that MEEM achieves significant network lifetime improvement over the state-of-the-art (70%) and the case of local user task computation only (450%).

II. System Model

Our multiuser network comprises K users denoted by the set $\mathcal{K} = \{1, .., K\}$ and a base station (BS) equipped with an edge cloud of limited computational capability. Each user k has a computation capability of f_k and residual battery energy E_k. Every n seconds, the edge cloud serves a set of users which have computationally intensive tasks. Let $\mathcal{K}_l \subseteq \mathcal{K}$ denote the set of users to be served by the cloud at slot $l \in \{1, 2, ..\}$. Let user $k \in \mathcal{K}_l$ has a task $\phi_k(l) = (\beta_k(l), b_k(l))$ to compute at the lth time slot, where $b_k(l)$ is the number of bits to be computed which include program codes, and input parameters and $\beta_k(l)$ is the required number of CPU cycles for 1 bit computation of the task. Therefore $\beta_k(l)b_k(l)$ denotes the total CPU cycles required to compute the task $\phi_k(l)$. The methods proposed in [12] can be applied to determine $b_k(l)$ and $\beta_k(l)$. Similar to [7], we consider splittable task. The tasks are needed to be executed within a maximum tolerable delay $\mathsf{T}^{th} \leq n$. An example of such network is internet of things (IoT) networks in which the edge cloud receives periodically splittable task, e.g., images from the IoT devices for processing.

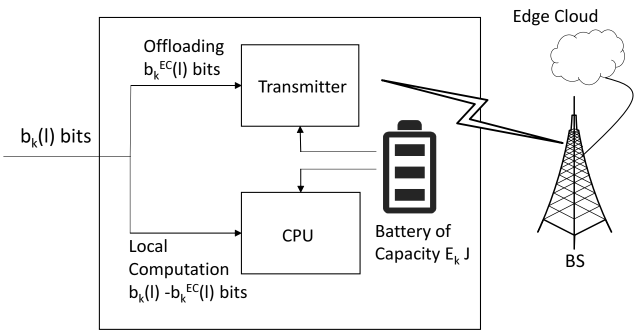

Fig. 1: Task computation of user k assisted by the edge cloud.

A. Local Computation

As shown in Fig. 1, user $k \in \mathcal{K}_l$ offloads $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud and computes $b_k(l) - b_k^{\mathrm{EC}}(l)$ bits at its own processor at time slot l. Thus, the local computation time is

$$T_k(l) = \frac{\beta_k(l)\left(b_k(l) - b_k^{\mathrm{EC}}(l)\right)}{f_k}. \tag{1}$$

Following the energy model in [11], the overall computation energy at user k to compute $b_k(l) - b_k^{\mathrm{EC}}(l)$ bits is

$$E_k(l) = \gamma_c \beta_k(l)\left(b_k(l) - b_k^{\mathrm{EC}}(l)\right) f_k^2, \tag{2}$$

where γ_c is the effective switched capacitance of the CPU.

B. Computation of Offloaded Tasks

Each user $k \in \mathcal{K}_l$ offloads $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud at time slot l, and then the edge cloud computes these bits at its processor and sends back the output to the users. Let the bandwidth allocated to user k at time slot l be $B_k(l)$. The spectral efficiency (in b/s/Hz) of the link between user k and the base station, for ergodic Rayleigh fading, is [13]:

$$R_{k,b} = \exp\left(\frac{N_0}{P_k\, g_{k,b}}\right)\mathbb{E}_1\left(\frac{N_0}{P_k\, g_{k,b}}\right)\log_2 e \tag{3}$$

where $\mathbb{E}_1(x) = \int_1^{\infty} m^{-1}e^{-xm}dm$ is an exponential integral, $g_{k,b}$ is the large-scale channel gain from user k to the BS, P_k is the transmit power density of user k, and N_0 is the noise power spectral density. Therefore, the delay in offloading $b_k^{\mathrm{EC}}(l)$ bits to the edge cloud becomes

$$\tau_{k,b}(l) = \frac{b_k^{\mathrm{EC}}(l)}{B_k(l)R_{k,b}}, \tag{4}$$

and the energy consumption at user k to offload $b_k^{\mathrm{EC}}(l)$ bits is

$$\mathcal{E}_k(l) = P_k \frac{b_k^{\mathrm{EC}}(l)}{R_{k,b}}. \tag{5}$$

Let the cloud allocate $F_k(l)$ of its computation resource to user k at time slot l. Thus, to compute the $b_k^{\mathrm{EC}}(l)$ bits for user k, the edge cloud requires time

$$T_{\mathrm{EC},k}(l) = \frac{\beta_k(l) b_k^{\mathrm{EC}}(l)}{F_k(l)}. \tag{6}$$

III. Problem Formulation

The overall completion time of task $\phi_k(l)$, $k \in \mathcal{K}_l$, is

$$\mathsf{T}_k(l) = \max\left(T_k(l), \tau_{k,b}(l) + T_{\mathrm{EC},k}(l)\right). \tag{7}$$

We disregard the time spent in sending back the results of the computation, as the size of the output data tends to be small relative to the input data [5]. Note that extension of the proposed resource allocation analysis for non-negligible output data size is straightforward.

The network lifetime is defined as the time duration for which all user tasks are executed within a maximum tolerable delay, while none of the users is depleted of energy. Thus, maximizing the lifetime of the network can be expressed as:

$$\begin{aligned} &\max_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \ \mathsf{T}, \\ \text{s.t.} \ &\sum_{l \in S_k^{\mathsf{T}}} \left(E_k(l) + \mathcal{E}_k(l)\right) \leq \mathsf{E_k}, \quad k \in \{1,..,K\}, \\ &\mathsf{T}_i(m) \leq \mathsf{T}^{th}, \quad i \in \mathcal{K}_m, \ m \in \{1,..,\mathsf{T}\}, \\ &\sum_{i \in \mathcal{K}_m} B_i(m) \leq B, \quad m \in \{1,..,\mathsf{T}\}, \\ &\sum_{i \in \mathcal{K}_m} F_i(m) \leq F, \quad m \in \{1,..,\mathsf{T}\}, \end{aligned} \tag{8}$$

where T denotes the network operating time in number of slots, S_k^{T} denotes the set of time slots when user k is activated within the network operating time T, B is the total available bandwidth in the system and F is the total processing capability of the cloud. In turn, $\boldsymbol{B}$, $\boldsymbol{F}$, and $\boldsymbol{b}$ are respectively the vectors of all values of $B_i(m)$, $F_i(m)$, and $b_i^{\mathrm{EC}}(m)$, for $i \in \mathcal{K}_m$, $m \in \{1,..,\mathsf{T}\}$.

The first constraint in (8) imposes that the energy consumption of user k (in local computation and offloading bits) over T be bounded by its residual battery energy $\mathsf{E_k}$. The second constraint imposes that the task completion time of user i at time slot m be bounded by the maximum tolerable delay T^{th}. The communication and computation resource allocations for the mobile users and the cloud are restricted by the total system bandwidth and the cloud's processing capability, respectively, as captured by the third and fourth constraints.

The above problem is hard to solve in practice, as task information for users in future time slots, $\beta_i(m), b_i(m), i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}\}$, may not be available, and the number of optimization variables is large (proportional to T), thus, finding the optimal solution requires high computational complexity. Aiming to maximize the network lifetime based on user task information for the present time slot only, we investigate the following optimization problem:

$$
\begin{aligned}
&\max_{\boldsymbol{F'},\boldsymbol{B'},\boldsymbol{b'}} \quad \min_{k\in\mathcal{K}_l} \quad \eta_k(l), && (9)\\
&\qquad \text{s.t.} \quad \mathsf{T}_k(l) \leq \mathsf{T}^{th}, \;\; k\in\mathcal{K}_l,\\
&\qquad \text{s.t.} \quad \sum_{k\in\mathcal{K}_l} B_k(l) \leq B, \quad \sum_{k\in\mathcal{K}_l} F_k(l) \leq F,
\end{aligned}
$$

where $\eta_k(l) = \mathsf{E}_k / \left(E_k(l) + \mathcal{E}_k(l)\right)$ is the energy efficiency of user $k \in \mathcal{K}_l$, and $\boldsymbol{B'}$, $\boldsymbol{F'}$, and $\boldsymbol{b'}$ are respectively the vectors of all values of $B_k(l)$, $F_k(l)$, and $b_k^{\mathrm{EC}}(l)$, for $k \in \mathcal{K}_l$. Minimum energy efficiency maximization (MEEM) of the network, as given in (9), leads to high computation and communication resource allocation for a user with low residual battery energy, and low communication and computation resource allocation for a user with high residual battery energy. *Thus, the resulting energy consumption of the user with low residual energy would be low, and the energy consumption of the user with high residual energy would be high, which result in network lifetime improvement.* Our experimental results in Section VI verify this induced property. The optimization problem in (9) can be solved efficiently and does not require knowledge of task information for future time slots. Note that (8), or (9) may be infeasible if T^{th} is very small. Next, we investigate solution methodologies for the problems (8), and (9).

IV. Minimum Energy Efficiency Maximization

Let V be a slack variable such that $1/V = \min_{k\in\mathcal{K}_l} \eta_k(l)$. Using (1)-(7), (9) can be expressed as

$$
\begin{aligned}
&\min_{\boldsymbol{F'},\boldsymbol{B'},\boldsymbol{b'}} \quad V, && (10)\\
&\text{s.t.} \quad \gamma_c \beta_k \left(b_k - b_k^{\mathrm{EC}}\right) f_k^2 + P_k \frac{b_k^{\mathrm{EC}}}{R_{k,b}} \leq \mathsf{E}_k V, \;\; k\in\mathcal{K}_l,\\
&\qquad \frac{\beta_k \left(b_k - b_k^{\mathrm{EC}}\right)}{f_k} \leq \mathsf{T}^{th}, \;\; k\in\mathcal{K}_l,\\
&\qquad \frac{\beta_k b_k^{\mathrm{EC}}}{F_k} + \frac{b_k^{\mathrm{EC}}}{B_k R_{k,b}} \leq \mathsf{T}^{th}, \;\; k\in\mathcal{K}_l,\\
&\qquad \sum_{k\in\mathcal{K}_l} B_k \leq B, \quad \sum_{k\in\mathcal{K}_l} F_k \leq F.
\end{aligned}
$$

We omit the time slot index l above for notation brevity. The problem (10) is nonconvex. It can be converted to a geometric programming problem via the single condensation method [14]. According to this method, for a constraint which is a ratio of posynomials, the denominator posynomial (say $\mathrm{f}(\mathbf{x})$) can be approximated into a monomial using the following inequality:

$$
\mathrm{f}(\mathbf{x}) = \sum_{\ell} \mathrm{f}_\ell(\mathbf{x}) \geq \hat{\mathrm{f}}(\mathbf{x}) = \prod_{\ell} \left[\frac{\mathrm{f}_\ell(\mathbf{x})}{\delta_\ell}\right]^{\delta_\ell}, \qquad (11)
$$

where $\delta_\ell > 0$ and $\sum_\ell \delta_\ell = 1$. Then, for $\delta_\ell = \mathrm{f}_\ell(\hat{\mathbf{x}})/\mathrm{f}(\hat{\mathbf{x}})$, $\hat{\mathrm{f}}(\hat{\mathbf{x}})$ is the best monomial approximation of $\mathrm{f}(\mathbf{x})$ near $\mathbf{x} = \hat{\mathbf{x}}$.

We formulate an iterative technique to optimally solve (10). At each iteration t, the first constraint in (10) is converted into a posynomial using (11) as

$$
\begin{aligned}
&\left(\frac{\mathsf{E}_k V(t)}{\delta_1(t)}\right)^{-\delta_1(t)} \left(\frac{\gamma_c \beta_k b_k^{\mathrm{EC}}(t) f_k^2}{\delta_2(t)}\right)^{-\delta_2(t)}\\
&\cdot \left(\gamma_c \beta_k b_k f_k^2 + P_k \frac{b_k^{\mathrm{EC}}(t)}{R_{k,b}}\right) \leq 1, \quad k\in\mathcal{K}_l, \qquad (12)
\end{aligned}
$$

where $\delta_1(t)$, and $\delta_2(t)$ are obtained from the solution at the $(t-1)$-th iteration as

$$
\begin{aligned}
\delta_1(t) &= \frac{\mathsf{E}_k V(t-1)}{\mathsf{E}_k V(t-1) + \gamma_c \beta_k b_k^{\mathrm{EC}}(t-1) f_k^2},\\
\delta_2(t) &= \frac{\gamma_c \beta_k b_k^{\mathrm{EC}}(t-1) f_k^2}{\mathsf{E}_k V(t-1) + \gamma_c \beta_k b_k^{\mathrm{EC}}(t-1) f_k^2}. \qquad (13)
\end{aligned}
$$

Similarly, at each iteration t, the second constraints therein is converted into a posynomial using (11) as

$$
\beta_k b_k \left(\frac{\mathsf{T}^{th} f_k}{\delta_3(t)}\right)^{-\delta_3(t)} \left(\frac{\beta_k b_k^{\mathrm{EC}}(t)}{\delta_4(t)}\right)^{-\delta_4(t)} \leq 1, \quad k\in\mathcal{K}_l, \quad (14)
$$

$$
\begin{aligned}
\text{where:} \quad \delta_3(t) &= \frac{\mathsf{T}^{th} f_k}{\mathsf{T}^{th} f_k + \beta_k b_k^{\mathrm{EC}}(t-1)}\\
\delta_4(t) &= \frac{\beta_k b_k^{\mathrm{EC}}(t-1)}{\mathsf{T}^{th} f_k + \beta_k b_k^{\mathrm{EC}}(t-1)}. \qquad (15)
\end{aligned}
$$

Thus, the overall optimization to be solved at iteration t is

$$
\begin{aligned}
&\min_{\substack{V(t),F_k(t),B_k(t)\\ b_k^{\mathrm{EC}}(t),k\in\mathcal{K}_l}} \quad V(t) && (16)\\
&\text{s.t.} \quad (12), \quad (14)\\
&\qquad \frac{\beta_k b_k^{\mathrm{EC}}(t)}{F_k(t)} + \frac{b_k^{\mathrm{EC}}(t)}{B_k(t) R_{k,b}} \leq \mathsf{T}^{th}, \;\; k\in\mathcal{K}_l\\
&\qquad \sum_{k\in\mathcal{K}_l} B_k(t) \leq B, \quad \sum_{k\in\mathcal{K}_l} F_k(t) \leq F.
\end{aligned}
$$

The above optimization problem is geometric programming and can be solved optimally. The iterative optimization is carried out until $|V(t) - V(t-1)| \leq \epsilon$ with $0 \leq \epsilon \ll 1$. An algorithmic implementation is included in Algorithm 1, which converges to the global solution of (10) [14].

Algorithm 1 Algorithm for MEEM.

```
Set t = 1, initialize V(t), F_k(t), B_k(t), b_k^EC(t), k ∈ K_l
such that the feasibility of (10) is preserved.
while true do                                  ▷ infinite loop
    t = t + 1
    Calculate δ_1(t), δ_2(t), δ_3(t) and δ_4(t)
    Find the optimum V(t), F_k(t), B_k(t), b_k^EC(t), k ∈ K_l
by solving (16) using GGPLAB [15]
    if |V(t) − V(t − 1)| ≤ ϵ then
        Break
    end if
end while
```

Implementation Of MEEM: The resource allocation according to MEEM strategy can be implemented in a centralized manner. To implement the MEEM strategy in a centralized manner, task information of the present time slot for all the users should be available at the BS which is similar to the centralized resource allocation strategies in literature [1, 5–10]. Additionally, the BS should also have the residual energy information of the users to implement the resource allocation. We assume that information of initial battery energy of the users is available at the BS which can be obtained with one time transmission from the users. Then, the BS calculate the energy consumption at each time slot and and find the available residual energy for the next time slot.

V. Optimal Lifetime Maximization

Using (1)-(7), the problem in (8) can be expressed as

$$\max_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad \mathsf{T}, \tag{17}$$

$$\text{s.t.} \quad \sum_{l\in S_k^{\mathsf{T}}} \left(\gamma_c \beta_k(l)\left(b_k(l)-b_k^{\text{EC}}(l)\right) f_k^2 + P_k \frac{b_k^{\text{EC}}(l)}{R_{k,b}}\right) \leq \mathsf{E_k}, \quad k\in\{1,..,K\},$$

$$\frac{\beta_i(m)\left(b_i(m)-b_i^{\text{EC}}(m)\right)}{f_i} \leq \mathsf{T}^{th},$$

$$\frac{\beta_i(m) b_i^{\text{EC}}(m)}{F_i(m)} + \frac{b_i^{\text{EC}}(m)}{B_i(m)R_{i,b}} \leq \mathsf{T}^{th}, \quad i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}\},$$

$$\sum_{i\in\mathcal{K}_m} B_i(m) \leq B, \quad \sum_{i\in\mathcal{K}_m} F_i(m) \leq F, \quad m\in\{1,..,\mathsf{T}\}.$$

Let $\mathsf{T} = \mathsf{T}'$ be a given value of T. The following feasibility test decides if the network will operate up to T' time slots:

$$\min_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad 0 \tag{18}$$

$$\text{s.t.} \quad \sum_{l\in S_k^{\mathsf{T}'}} \left(\gamma_c \beta_k(l)\left(b_k(l)-b_k^{\text{EC}}(l)\right) f_k^2 + P_k \frac{b_k^{\text{EC}}(l)}{R_{k,b}}\right) \leq \mathsf{E_k}, \quad k\in\{1,..,K\},$$

$$\frac{\beta_i(m)\left(b_i(m)-b_i^{\text{EC}}(m)\right)}{f_i} \leq \mathsf{T}^{th},$$

$$\frac{\beta_i(m) b_i^{\text{EC}}(m)}{F_i(m)} + \frac{b_i^{\text{EC}}(m)}{B_i(m)R_{i,b}} \leq \mathsf{T}^{th}, \quad i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}'\},$$

$$\sum_{i\in\mathcal{K}_m} B_i(m) \leq B, \quad \sum_{i\in\mathcal{K}_m} F_i(m) \leq F, \ m\in\{1,..,\mathsf{T}'\}$$

Thus, problem (17) can be solved in a two-nested search loops in which we vary the value of T' in the outer loop, and in the inner loop, check if (18) is feasible. The maximum value of T', for which (18) is feasible, is the optimal network lifetime. We consider the following optimization problem:

$$\min_{\boldsymbol{F},\boldsymbol{B},\boldsymbol{b}} \quad S,$$

$$\text{s.t.} \quad \sum_{l\in S_k^{\mathsf{T}'}} \left(\gamma_c \beta_k(l)\left(b_k(l)-b_k^{\text{EC}}(l)\right) f_k^2 + P_k \frac{b_k^{\text{EC}}(l)}{R_{k,b}}\right) \leq \mathsf{E_k}, \quad k\in\{1,..,K\}, \tag{19a}$$

$$\frac{\beta_i(m)\left(b_i(m)-b_i^{\text{EC}}(m)\right)}{f_i} \leq S, \quad i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}'\}, \tag{19b}$$

$$\frac{\beta_i(m) b_i^{\text{EC}}(m)}{F_i(m)} + \frac{b_i^{\text{EC}}(m)}{B_i(m)R_{i,b}} \leq S, \quad i\in\mathcal{K}_m, m\in\{1,..,\mathsf{T}'\}, \tag{19c}$$

$$\sum_{i\in\mathcal{K}_m} B_i(m) \leq B, \quad m\in\{1,..,\mathsf{T}'\}, \tag{19d}$$

$$\sum_{i\in\mathcal{K}_m} F_i(m) \leq F, \quad m\in\{1,..,\mathsf{T}'\}, \tag{19e}$$

Proposition 1. *The feasibility testing in (18) can be solved in two steps, first to solve (19) optimally, and then check if the optimal value of S for T' time slots, $S_{T'}$ which is obtained by solving (19), is less than or equal to T^{th}.*

Proof. See Appendix. A

Problem (19) can be converted into geometric programming, similarly to Section IV. We apply an iterative technique to solve it. At each iteration t, using (11), the first constraint in (19) is converted into a posynomial as

$$\left(\frac{\mathsf{E_k}}{\delta_5(t)}\right)^{-\delta_5(t)} \prod_{j\in S_k^{\mathsf{T}'}} \left(\frac{\gamma_c\beta_k(j) b_k^{\text{EC}}(j,t) f_k^2}{\delta_{6j}(t)}\right)^{-\delta_{6j}(t)} \cdot \sum_{l\in S_k^{\mathsf{T}'}} \left(\gamma_c\beta_k(l) b_k(l) f_k^2 + P_k \frac{b_k^{\text{EC}}(l,t)}{R_{k,b}}\right) \leq 1, \ k\in\{1,..,K\} \tag{20}$$

$$\text{where:} \quad \delta_5(t) = \frac{\mathsf{E_k}}{\mathsf{E_k} + \sum_{l\in S_k^{\mathsf{T}'}} \gamma_c\beta_k(l) b_k^{\text{EC}}(l,t-1) f_k^2}, \tag{21}$$

$$\delta_{6j}(t) = \frac{\gamma_c\beta_k(j) b_k^{\text{EC}}(j,t-1) f_k^2}{\mathsf{E_k} + \sum_{l\in S_k^{\mathsf{T}'}} \gamma_c\beta_k(l) b_k^{\text{EC}}(l,t-1) f_k^2},$$

and the second constraint is converted into a posynomial as

$$\beta_i(m) b_i(m) \left(\frac{S(t) f_i}{\delta_9(t)}\right)^{-\delta_9(t)} \left(\frac{\beta_i(m) b_i^{\text{EC}}(m,t)}{\delta_{10}(t)}\right)^{-\delta_{10}(t)} \leq 1, \quad i\in\mathcal{K}_m, \ m\in\{1,..,\mathsf{T}'\}, \tag{22}$$

$$\text{where:} \quad \delta_9(t) = \frac{S(t-1) f_i}{S(t-1) f_i + \beta_i(m) b_i^{\text{EC}}(m,t-1)}, \tag{23}$$

$$\delta_{10}(t) = \frac{\beta_i(m) b_i^{\text{EC}}(m,t-1)}{S(t-1) f_i + \beta_i(m) b_i^{\text{EC}}(m,t-1)}.$$

Thus, the overall optimization to be solved at time t is:

$$\min_{\substack{S(t), F_i(m,t),\\ B_i(m,t), b_i^{\text{EC}}(m,t)}} \quad S(t), \tag{24}$$

$$\text{s.t.} \quad (20), \ (22),$$

$$\frac{\beta_i(m) b_i^{\mathrm{EC}}(m,t)}{F_i(m,t)} + \frac{b_i^{\mathrm{EC}}(m,t)}{B_i(m,t) R_{i,b}} \leq S,$$
$$i \in \mathcal{K}_m, m \in \{1, .., \mathsf{T}'\},$$
$$\sum_{i \in \mathcal{K}_m} B_i(m,t) \leq B, \sum_{i \in \mathcal{K}_m} F_i(m,t) \leq F, m \in \{1, .., \mathsf{T}'\}.$$

The above optimization is geometric programming and can be solved optimally. Hence, the optimal solution of (19) is obtained by solving (24) iteratively, following similar steps as given in Algorithm 1 [14]. Thus, to solve (17), in the inner loop, we check if the network operates for T' time slots, by first solving (19), following the approach as stated above, and then checking the condition $S_{T'} \leq \mathsf{T}^{th}$, for the given value of T'. In the outer loop, we then use bisection search to find the maximum value of T' for which the network operates. The overall procedure is described in Algorithm 2. The output of the algorithm $T_{optimal}$ is the optimal network lifetime.

Algorithm 2 Finding the optimal network lifetime.

```
Initialize low and high (lower and upper bounds for bisection search)
while high > low do
    T' = ⌊(low+high)/2⌋
    Find S_T' by solving (19)
    if S_T' < T^th then
        low = T' + 1
    else
        high = T'
    end if
end while
T_optimal = low − 1
```

Even though this strategy may not be practically implementable due to its high computational complexity and the requirement for future user task information, it represents an upper bound for the performance of the MEEM approach.

VI. Performance Evaluation

Here we present simulation results that evaluate the network lifetime performance of the proposed strategies. Many of the works in literature [7, 8, 10] have considered sum energy minimization as objective to decide resource allocation for mobile-edge computing networks. Therefore, as a reference, we will compare our proposed strategies with sum energy minimization objective, i.e., minimizing total energy consumption of all the users $\sum_{k \in \mathcal{K}_l} (E_k(l) + \mathcal{E}_k(l))$ at each time slot l with the same constraints of the problem in (9). The resource allocation for sum energy minimization can be solved using geometric programming iteratively with similar steps as given in Algorithm 1. We refer this strategy as 'Reference Method'. For comparison purposes, we also consider the strategy 'Local Computation' in which the users compute the tasks at their own processors. We note that the users do not meet the maximum tolerable delay for 'Local Computation', since their processing capability is low.

For the evaluations that follow, ten users are uniformly distributed in a circular region of radius 50 m with a cloud-associated BS at the center. The system parameters are $f_i = 0.5$ GHz, $P_i = 10^{-8}$ W/Hz $B = 5$ MHz, $\mathsf{T}^{th} = 0.15$ s, $\gamma_c = 10^{-28}$ MHz, $N_0 = -147$ dBM/Hz, $\epsilon = 10^{-5}$, $b_i = 200$ Kb and β_i follow the uniform distribution with $[500, 1500]$ cycles/bit. Each user in the network is activated according to an activation probability p_i which follow the uniform distribution with $[0.3, 0.7]$. Therefore, the set of users which are activated at different time slots may be different. The obtained results are averaged over 500 network realizations.

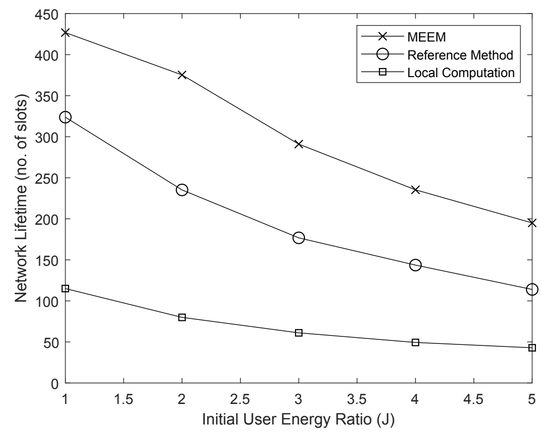

Fig. 2: Network lifetime vs. E_{ratio} for $E_{tot} = 5$ J.

In Fig. 2, we consider the performance of the proposed strategy in a network where the initial energy of the users is not identical. The network has a total of ten users among which five randomly chosen users have initial energy E_1 and the other five users have initial energy E_2 with $E_2 \leq E_1$. We fix the initial total network energy (i.e., the sum of battery energy of all users) as $E_{tot} = 5$ J. The network lifetime performance of the proposed strategies is evaluated when the initial user energy ratio $E_{ratio} = E_1/E_2$ varies from 1 to 5, while $F = 6$ GHz. If $E_{ratio} = 1$, we have identical initial energy for all the users, i.e., $E_1 = E_2 = 0.5$ J, and if $E_{ratio} = 5$, five users have initial energy $E_1 = 0.83$ and the other five users have initial energy $E_2 = 0.17$. As E_{ratio} is increased, E_1 increases more compared to E_2, and the energy balancing decreases in the network. Thus, the network lifetime decreases for all strategies. Since the MEEM strategy considers the residual battery energy information to decide on the task sharing and resource allocation, while the reference method does not consider the residual battery energy information to optimize the system parameters, MEEM achieves significant performance improvement compared to reference method for high values of E_{ratio}. For example, if $E_{ratio} = 5$, the MEEM strategy achieves 1.71 times longer network lifetime (70% improvement).

Here we analyze the performance of MEEM compared to the optimal network lifetime strategy described in Section V. For the latter, the number of optimization variables is proportional to the number of time slots the network operates. Thus, if the network lifetime is high, finding the optimal

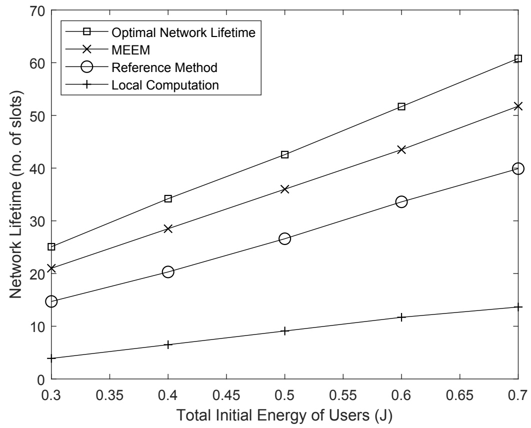

Fig. 3: Network lifetime vs. total initial energy for the users.

solution of (17) via geometric programming is hard with so many optimization variables. Hence, we show the performance of the proposed strategies when E_{tot} is low for which the network lifetime is low. Fig. 3 shows the network lifetime for a random network realization when the total initial energy in the network E_{tot} varies from 0.3 J to 0.7 J, while $E_{ratio} = 1$, $F = 6$ GHz. As E_{tot} increases, the network lifetime for all strategies increase. The optimal network lifetime strategy achieves 1.15 to 1.20 times better network lifetime compared to MEEM as E_{tot} varies.

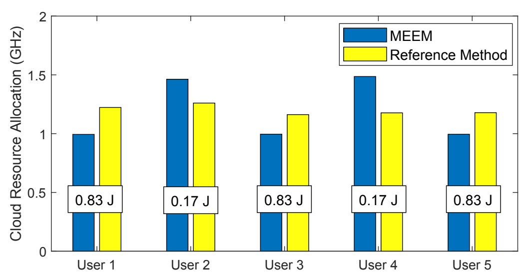

Fig. 4: Cloud computation power distribution among users.

In Fig. 4, we analyze the computation resource distribution among the users at a given time slot. We consider that f ve users are active at the time slot and the initial battery energy of the users 1 to 5 are 0.83 J, 0.17 J, 0.83 J, 0.17 J and 0.83 J, respectively. The initial battery energy for each user has been shown in rectangular box in the f gures. While the computation resource allocation are balanced across the users for reference method, MEEM allocates higher resource for the users with lower residual energy and lower resource for the user with higher residual energy and therefore performs better.

VII. Conclusion

We investigated the lifetime maximization problem in a network where its nodes/users periodically compute their task with the help of an edge cloud. Aiming to maximize the network lifetime based on the user task information for the present time slot only, we have proposed an MEEM strategy to decide the sharing of tasks between the users and the cloud, and the allocation of computation and communication resources. We further investigated network lifetime maximization when future user task information is available, as well, as an upper bound to MEEM. Though the optimization problem for MEEM is non-convex, we have shown that the global optimal solution can be obtained using feasibility testing and geometric programming. We have shown that the MEEM strategy performs close to the optimal network lifetime. For high value of the initial user energy ratio, MEEM achieves roughly 70% lifetime improvement over the state-of-the-art and 450% lifetime improvement relative to local user computation only.

Appendix A
Proof of Proposition 1

Let $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ be a feasible solution point of (19), i.e., the constraints (19a), (19d) and (19e) are met at this point. If $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ is also a feasible solution of (18), then the value of

$$S = \max_{i \in \mathcal{K}_m, m \in \{1,..,\mathsf{T}'\}} \left(\frac{\beta_i(m)\left(b_i(m) - b_i^{\mathrm{EC}}(m)\right)}{f_i}, \frac{\beta_i(m) b_i^{\mathrm{EC}}(m)}{F_i(m)} + \frac{b_i^{\mathrm{EC}}(m)}{B_i(m) R_{i,b}} \right)$$

is less than or equal to T^{th}. If $(\boldsymbol{F}, \boldsymbol{B}, \boldsymbol{b})$ is not a feasible solution of (18), then $S > \mathsf{T}^{th}$. Therefore $S_{T'}$ must be less than or equal to T^{th} if there exist a feasible solution of (17).

References

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing-A Key Technology Towards 5G," *ETSI White Paper*, vol. 11, 2015.

[2] D. Wu, Y. Cai, L. Zhou, and J. Wang, "A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks," *IEEE TWC*, vol. 11, no. 3, pp. 1190–1200, Mar. 2012.

[3] L. Pang, Y. Zhang, F. Gong, G. Ren, A. Wang, X. Liang, and J. Li, "Energy aware resource allocation for incremental AF-OFDM relaying," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1766–1769, Oct. 2015.

[4] P. Castiglione, S. Savazzi, M. Nicoli, and T. Zemen, "Partner selection in indoor-to-outdoor cooperative networks: An experimental study," *IEEE JSAC*, vol. 31, no. 8, pp. 1559–1571, Aug. 2013.

[5] X. Chen, "Decentralized computation off oading game for mobile cloud computing," *IEEE TPDS*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[6] H. Q. Le, H. Al-Shatri, and A. Klein, "Eff cient resource allocation in mobile-edge computation off oading: Completion time minimization," in *IEEE ISIT)*, June 2017, pp. 2513–2517.

[7] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-eff cient resource allocation for mobile-edge computation off oading," *IEEE TWC*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[8] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE TSIPN*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[9] S. Gupta and A. Lozano, "Computation-bandwidth trading for mobile edge computing," in *Proc. Annu. IEEE CCNC*, Jan 2019.

[10] T. X. Tran and D. Pompili, "Joint task off oading and resource allocation for multi-server mobile-edge computing networks," *IEEE TVT.*, 2018.

[11] C. You, K. Huang, and H. Chae, "Energy eff cient mobile cloud computing powered by wireless energy transfer," *IEEE JSAC*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[12] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *IEEE CLOUD*, Jun. 2012, pp. 794–802.

[13] W. C. Y. Lee, "Estimate of channel capacity in Rayleigh fading environment," *IEEE TVT*, vol. 39, no. 3, pp. 187–189, 1990.

[14] G. Xu, "Global optimization of signomial geometric programming problems," *Eur. J. Oper. Res.*, vol. 233, no. 3, pp. 500–510, 2014.

[15] GGPLAB: A simple MATLAB toolbox for geometric programming. [Online]. Available: http://www.stanford.edu/boyd/ggplab/