Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading

Changsheng You, *Student Member, IEEE*, Kaibin Huang, *Senior Member, IEEE*,
Hyukjin Chae, and Byoung-Hoon Kim, *Member, IEEE*

Abstract—*Mobile-edge computation offloading* (MECO) offloads intensive mobile computation to clouds located at the edges of cellular networks. Thereby, MECO is envisioned as a promising technique for prolonging the battery lives and enhancing the computation capacities of mobiles. In this paper, we study resource allocation for a multiuser MECO system based on time-division multiple access (TDMA) and orthogonal frequency-division multiple access (OFDMA). First, for the TDMA MECO system with infinite or finite cloud computation capacity, the optimal resource allocation is formulated as a convex optimization problem for minimizing the weighted sum mobile energy consumption under the constraint on computation latency. The optimal policy is proved to have a threshold-based structure with respect to a derived *offloading priority function*, which yields priorities for users according to their channel gains and local computing energy consumption. As a result, users with priorities above and below a given threshold perform complete and minimum offloading, respectively. Moreover, for the cloud with finite capacity, a sub-optimal resource-allocation algorithm is proposed to reduce the computation complexity for computing the threshold. Next, we consider the OFDMA MECO system, for which the optimal resource allocation is formulated as a mixed-integer problem. To solve this challenging problem and characterize its policy structure, a low-complexity sub-optimal algorithm is proposed by transforming the OFDMA problem to its TDMA counterpart. The corresponding resource allocation is derived by defining an *average offloading priority function* and shown to have close-to-optimal performance in simulation.

Index Terms—Mobile-edge computing, resource allocation, mobile computation offloading, energy-efficient computing.

I. INTRODUCTION

THE realization of Internet of Things (IoT) [1] will connect tens of billions of resource-limited mobiles, e.g., mobile devices, sensors and wearable computing devices, to Internet via cellular networks. The finite battery lives and limited computation capacities of mobiles pose significant challenges for designing IoT. One promising solution is to leverage mobile-edge computing [2] and offload intensive mobile computation to nearby clouds at the edges of cellular networks, called

edge clouds, with short latency, referred to as *mobile-edge computation offloading* (MECO). In this paper, we consider an MECO system with a single edge cloud serving multiple users and investigate the energy-efficient resource allocation.

A. Prior Work

Mobile computation offloading (MCO) [3] (or mobile cloud computing) has been extensively studied in computer science, including system architectures (e.g., MAUI [4]), virtual machine migration [5] and power management [6]. It is commonly assumed that the implementation of MCO relies on a network architecture with a central cloud (e.g., a data center). This architecture has the drawbacks of high overhead and long backhaul latency [7], and will soon encounter the performance bottleneck of finite backhaul capacity in view of exponential mobile traffic growth. These issues can be overcome by MECO based on a network architecture supporting distributed mobile-edge computing. Among others, designing energy-efficient control policies is a key challenge for the MECO system.

Energy-efficient MECO requires the joint design of MCO and wireless communication techniques. Recent years have seen research progress on this topic for both single-user [8]–[11] and multiuser [12]–[16] MECO systems. For a single-user MECO system, the optimal offloading decision policy was derived in [8] by comparing the energy consumption of optimized local computing (with variable CPU cycles) and offloading (with variable transmission rates). This framework was further developed in [9] and [10] to enable adaptive offloading powered by wireless energy transfer and energy harvesting, respectively. Moreover, dynamic offloading was integrated with adaptive LTE/WiFi link selection in [11] to achieve higher energy efficiency. For multiuser MECO systems, the control policies for energy savings are more complicated. In [12], distributed computation offloading for multiuser MECO at a single cloud was designed using game theory for both energy-and-latency minimization at mobiles. A multi-cell MECO system was considered in [13], where the radio and computation resources were jointly allocated to minimize the mobile energy consumption under offloading latency constraints. With the coexistence of central and edge clouds, the optimal user scheduling for offloading to different clouds was studied in [14]. In addition to total mobile energy consumption, cloud energy consumption for computation was also minimized in [15] by designing the mapping between clouds and mobiles for offloading using game theory. The cooperation among clouds was further investigated in [16] to maximize the revenues of clouds and meet mobiles' demands

via resource pool sharing. Prior work on MECO resource allocation focuses on complex algorithmic designs and yields little insight into the optimal policy structures. In contrast, for a multiuser MECO system based on time-division multiple access (TDMA), the optimal resource-allocation policy is shown in the current work to have a simple threshold-based structure with respect to a derived offloading priority function. This insight is used for designing the low-complexity resource-allocation policy for an orthogonal frequency-division multiple access (OFDMA) MECO system.

Resource allocation for traditional multiple-access communication systems has been widely studied, including TDMA (see e.g., [17]), OFDMA (see e.g., [18]) and code-division multiple access (CDMA) (see e.g., [19]). Moreover, it has been designed for existing networks such as cognitive radio [20] and heterogenous networks [21]. Note that all of them only focus on the radio resource allocation. In contrast, for the newly proposed MECO systems, both the computation and radio resource allocation at the edge cloud are jointly optimized for the maximum mobile energy savings, making the algorithmic design more complex.

B. Contribution and Organization

This paper considers resource allocation in a multiuser MECO system based on TDMA and OFDMA. Multiple mobiles are required to compute different computation loads with the same latency constraint. Assuming that computation data can be split for separate computing, each mobile can simultaneously perform local computing and offloading. Moreover, the edge cloud is assumed to have perfect knowledge of local computing energy consumption, channel gains and fairness factors at all users, which is used for designing centralized resource allocation to achieve the minimum weighted sum mobile energy consumption. In the TDMA MECO system, the optimal threshold-based policy is derived for both the cases of infinite and finite cloud capacities. For the OFDMA MECO system, a low-complexity sub-optimal algorithm is proposed to solve the mixed-integer resource allocation problem.

The contributions of current work are as follows.

- *TDMA MECO with infinite cloud capacity:* For TDMA MECO with infinite (computation) capacity, a convex optimization problem is formulated to minimize the weighted sum mobile energy consumption under the time-sharing constraint. To solve it, an *offloading priority function* is derived that yields priorities for users and depends on their channel gains and local computing energy consumption. Based on this, the optimal policy is proved to have a threshold-based structure that determines complete and minimum offloading for users with priorities above and below a given threshold, respectively.

- *TDMA MECO with finite cloud capacity:* The above results are extended to the case of finite capacity. Specifically, the optimal resource allocation policy is derived by defining an *effective offloading priority function* and modifying the threshold-based policy as derived for the infinite-capacity cloud. To reduce the complexity arising from a two-dimension search for Lagrange multipliers, a simple and low-complexity algorithm is proposed

based on the *approximated* offloading priority order. This reduces the said search to a one-dimension search, shown by simulation to have close-to-optimal performance.

- *OFDMA MECO:* For an infinite-capacity cloud based on OFDMA, the insight of *priority-based* policy structure of TDMA is used for optimizing its resource allocation. Specifically, to solve the corresponding mixed-integer optimization problem, a low-complexity sub-optimal algorithm is proposed. Using average sub-channel gains, the OFDMA resource allocation problem is transformed into its TDMA counterpart. Based on this, the initial resource allocation and offloaded data allocation can be determined by defining an *average offloading priority function*. Moreover, the integer sub-channel assignment is performed according to the offloading priority order, followed by adjustments of offloaded data allocation over assigned sub-channels. The proposed algorithm is shown to have close-to-optimal performance by simulation and can be extended to the finite-capacity cloud case.

The reminder of this paper is organized as follows. Section II introduces the system model. Section III presents the problem formulation for multiuser MECO based on TDMA. The corresponding resource allocation policies are characterized in Section IV and Section V for both the cases of infinite and finite cloud capacities, respectively. The above results are extended in Section VI for the OFDMA system. Simulation results and discussion are given in Section VII, followed by the conclusion in Section VIII.

II. SYSTEM MODEL

Consider a multiuser MECO system shown in Fig. 1 with K single-antenna mobiles, denoted by a set $\mathcal{K} = \{1, 2, \cdots, K\}$, and one single-antenna base station (BS) that is the gateway of an edge cloud. These mobiles are required to compute different computation loads under the same latency constraint.[1] Assume that the BS has perfect knowledge of multiuser channel gains, local computing energy per bit and sizes of input data at all users, which can be obtained by feedback. Using these information, the BS selects offloading users, determines the offloaded data sizes and allocates radio resource to offloading users with the criterion of minimum weighted sum mobile energy consumption.

A. Multiple-Access Model

Both the TDMA and OFDMA systems are considered as follows. For the TDMA system, time is divided into slots each with a duration of T seconds where T is chosen to meet the user-latency requirement. As shown in Fig. 1, each time slot comprises two sequential phases for 1) mobile offloading or local computing and 2) cloud computing and

---

[1]For asynchronous computation offloading among users, the maximum additional latency for each user is one time slot. Moreover, this framework can be extended to predictive computing by designing control policies for the coming data.

For the general case with heterogeneous user-latency constraints and finite cloud resource, the current work can be extended by integration with dynamic cloud control designed similarly as techniques such as demand scheduling and load shifting for smart grids [22], [23].
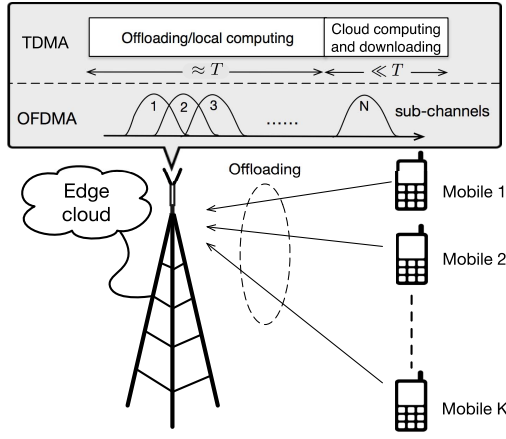
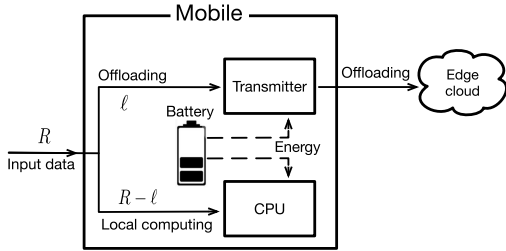Fig. 1.   Multiuser MECO systems based on TDMA and OFDMA.

Fig. 2.   Mobile computation offloading.

downloading of computation results from the edge cloud to mobiles. Cloud computing has small latency; the downloading consumes negligible mobile energy and furthermore is much faster than offloading due to relative smaller sizes of computation results. For these reasons, the second phase is assumed to have a negligible duration compared to the first phase and not considered in resource allocation. For the OFDMA system, the total bandwidth is divided into multiple orthogonal sub-channels and each sub-channel can be assigned to at most one user. The offloading mobiles will be allocated with one or more sub-channels.

Considering an arbitrary slot in TDMA/OFDMA, the BS schedules a subset of users for complete/partial offloading. The user with partial or no offloading computes a fraction of or all input data, respectively, using a local CPU.

B. Local-Computing Model

Assume that the CPU frequency is fixed at each user and may vary over users. Consider an arbitrary time slot. Following the model in [12], let C_k denote the number of CPU cycles required for computing 1-bit of input data at the k-th mobile, and P_k the energy consumption per cycle for local computing at this user. Then the product $C_k P_k$ gives computing energy per bit. As shown in Fig. 2, mobile k is required to compute R_k-bit input data within the time slot, out of which ℓ_k-bit is offloaded and $(R_k - \ell_k)$-bit is computed locally. Then the total energy consumption for local computing at mobile k, denoted as $E_{\text{loc},k}$, is given by $E_{\text{loc},k} = (R_k - \ell_k)C_k P_k$. Let F_k denote the computation capacity of mobile k that is measured by the number of CPU cycles per second. Under the computation

latency constraint, it has $C_k(R_k - \ell_k)/F_k \leq T$. As a result, the offloaded data at mobile k has the minimum size of $\ell_k \geq m_k^+$ with $m_k = R_k - F_k T/C_k$, where $(x)^+ = \max\{x, 0\}$.

C. Computation-Offloading Model

First, consider the TDMA system for an arbitrary time slot. Let h_k denote the channel gain for mobile k that is constant during offloading duration, and p_k its transmission power. Then the achievable rate (in bits/s), denoted by r_k, is:

$$r_k = B \log_2\left(1 + \frac{p_k h_k^2}{N_0}\right) \tag{1}$$

where B and N_0 are the bandwidth and the variance of complex white Gaussian channel noise, respectively. The fraction of slot allocated to mobile k for offloading is denoted as t_k with $t_k \geq 0$, where $t_k = 0$ corresponds to no offloading. For the case of offloading ($t_k > 0$), under the assumption of negligible cloud computing and result downloading time (see Section II-A), the transmission rate is fixed as $r_k = \ell_k/t_k$ since this is the most energy-efficient transmission policy under a deadline constraint [24]. Define a function $f(x) = N_0(2^{\frac{x}{B}} - 1)$. It follows from (1) that the energy consumption for offloading at mobile k is

$$E_{\text{off},k} = p_k t_k = \frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right). \tag{2}$$

Note that if either $\ell_k = 0$ or $t_k = 0$, $E_{\text{off},k}$ is equal to zero.

Next, consider an OFDMA system with N sub-channels, denoted by a set $\mathcal{N} = \{1, 2, \cdots, N\}$. Let $p_{k,n}$ and $h_{k,n}$ denote the transmission power and channel gain of mobile k on the n-th sub-channel. Define $\rho_{k,n} \in \{0, 1\}$ as the sub-channel assignment indicator variable where $\rho_{k,n} = 1$ indicates that sub-channel n is assigned to mobile k, and verse vice. Then the achievable rate (in bits/s) follows:

$$r_{k,n} = \rho_{k,n} \bar{B} \log_2\left(1 + \frac{p_{k,n} h_{k,n}^2}{\bar{N}_0}\right) \tag{3}$$

where $\bar{B}$ and $\bar{N}_0$ are the bandwidth and noise power for each sub-channel, respectively. Let $\ell_{k,n} = r_{k,n}T$ denote the offloaded data size over the offloading duration time T that can be set as the OFDMA symbol duration. The corresponding offloading energy consumption can be expressed as below, which is similar to that in [18], namely,

$$E_{\text{off},k,n} = \rho_{k,n} p_{k,n} T = \frac{\rho_{k,n}}{h_{k,n}^2} \bar{f}\left(\frac{\ell_{k,n}}{\rho_{k,n}}\right) \tag{4}$$

where $\bar{h}_{k,n}^2 = h_{k,n}^2/T$ and $\bar{f}(x) = \bar{N}_0(2^{\frac{x}{BT}} - 1)$.

D. Cloud-Computing Model

Considering an edge cloud with finite (computation) capacity, for simplicity, the finite capacity is reflected in one of the following two constraints.[2] The first one upper-bounds

---

[2]For simplicity, we consider either a computation-load or a computation-time constraint at one time but not both simultaneously. However, note that the two constraints can be considered equivalent. Specifically, limiting the cloud computation load allows the computation to be completed within the required time and vice versa. The current resource-allocation policies can be extended to account for more elaborate constraints, which are outside the scope of the paper.

CPU cycles of sum offloaded data that can be handled by the cloud in each time slot. Let F represent the cloud computation capacity measured by CPU cycles per time slot. Then it follows: $\sum_{k=1}^{K} C_k \ell_k \leq F$. This constraint ensures negligible cloud computing latency. The other one considers non-negligible computing time at the cloud that performs load balancing as in [25], given as $t_{\text{comp}} = (\sum_k^K \ell_k C_k)/F'$, where F' is the cloud computation capacity measure by CPU cycles per second. Note that t_{comp} is factored into the latency constraint in the sequel.

III. MULTIUSER MECO FOR TDMA: PROBLEM FORMULATION

In this section, resource allocation for multiuser MECO based on TDMA is formulated as an optimization problem. The objective is to minimize the weighted sum mobile energy consumption: $\sum_{k=1}^{K} \beta_k (E_{\text{off},k} + E_{\text{loc},k})$, where the positive weight factors $\{\beta_k\}$ account for fairness among mobiles. Under the constraints on time-sharing, cloud computation capacity and computation latency, the resource allocation problem is formulated as follows:

$$\min_{\{\ell_k, t_k\}} \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k) C_k P_k \right]$$

$$\text{s.t.} \sum_{k=1}^{K} t_k \leq T, \quad \sum_{k=1}^{K} C_k \ell_k \leq F,$$

$$t_k \geq 0, \quad m_k^+ \leq \ell_k \leq R_k, \quad k \in \mathcal{K}. \qquad \textbf{(P1)}$$

First, it is easy to observe that the feasibility condition for Problem P1 is: $\sum_{k=1}^{K} m_k^+ C_k \leq F$. It shows that whether the cloud capacity constraint is satisfied determines the feasibility of this optimization problem, while the time-sharing constraint can always be satisfied and only affects the mobile energy consumption. Next, one basic characteristic of Problem P1 is given in the following lemma, proved in Appendix A.

Lemma 1: Problem P1 is a convex optimization problem.

Assume that Problem P1 is feasible. The direct solution for Problem P1 using the dual-decomposition approach (the Lagrange method) requires iterative computation and yields little insight into the structure of the optimal policy. To address these issues, we adopt a two-stage solution approach that requires first solving Problem P2 below, which follows from Problem P1 by relaxing the constraint on cloud capacity:

$$\min_{\{\ell_k, t_k\}} \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k) C_k P_k \right]$$

$$\text{s.t.} \sum_{k=1}^{K} t_k \leq T,$$

$$t_k \geq 0, \quad m_k^+ \leq \ell_k \leq R_k, \quad k \in \mathcal{K}. \qquad \textbf{(P2)}$$

If the solution for Problem P2 violates the constraint on cloud capacity, Problem P1 is then incrementally solved building on the solution for Problem P2. This approach allows the optimal policy to be shown to have the said threshold-based structure

and also facilitates the design of low-complexity close-to-optimal algorithm. It is interesting to note that Problem P2 corresponds to the case where the edge cloud has infinite capacity. The detailed procedures for solving Problems P1 and P2 are presented in the two subsequent sections.

IV. MULTIUSER MECO FOR TDMA: INFINITE CLOUD CAPACITY

In this section, by solving Problem P2 using the Lagrange method, we derive a threshold-based policy for the optimal resource allocation.

To solve Problem P2, the partial Lagrange function is defined as

$$L = \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k) C_k P_k \right] + \lambda \left(\sum_{k=1}^{K} t_k - T \right)$$

where $\lambda \geq 0$ is the Lagrange multiplier associated with the time-sharing constraint. For ease of notation, define a function $g(x) = f(x) - x f'(x)$. Let $\{\ell_k^{*(2)}, t_k^{*(2)}\}$ denote the optimal solution for Problem P2 that always exists satisfying the feasibility condition.

Then applying KKT conditions leads to the following necessary and sufficient conditions:

$$\frac{\partial L}{\partial \ell_k^{*(2)}} = \frac{\beta_k f'\left(\frac{\ell_k^{*(2)}}{t_k^{*(2)}}\right)}{h_k^2} - \beta_k C_k P_k \begin{cases} > 0, & \ell_k^{*(2)} = m_k^+ \\ = 0, & \ell_k^{*(2)} \in (m_k^+, R_k) \\ < 0, & \ell_k^{*(2)} = R_k, \end{cases}$$

$$(5a)$$

$$\frac{\partial L}{\partial t_k^{*(2)}} = \frac{\beta_k g\left(\frac{\ell_k^{*(2)}}{t_k^{*(2)}}\right)}{h_k^2} + \lambda^* \begin{cases} > 0, & t_k^{*(2)} = 0 \\ = 0, & t_k^{*(2)} > 0, \end{cases} \quad \forall k \in \mathcal{K},$$

$$(5b)$$

$$\sum_{k=1}^{K} t_k^{*(2)} \leq T, \quad \lambda^* \left(\sum_{k=1}^{K} t_k^{*(2)} - T \right) = 0. \qquad (5c)$$

Note that for $\ell_k^{*(2)} \in (m_k^+, R_k)$ and $t_k^{*(2)} > 0$, it can be derived from (5) and (5b) that

$$\frac{\ell_k^{*(2)}}{t_k^{*(2)}} = f'^{-1}\left(C_k P_k h_k^2\right) = g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right). \qquad (6)$$

Based on these conditions, the optimal policy for resource allocation is characterized in the following sub-sections.

A. Offloading Priority Function

Define a (mobile) *offloading priority function*, which is essential for the optimal resource allocation, as follows:

$$\varphi(\beta_k, C_k, P_k, h_k) = \begin{cases} \frac{\beta_k N_0}{h_k^2} \left(v_k \ln v_k - v_k + 1\right), & v_k \geq 1 \\ 0, & v_k < 1, \end{cases}$$

$$(7)$$

with the constant v_k defined as

$$v_k = \frac{B C_k P_k h_k^2}{N_0 \ln 2}. \qquad (8)$$

This function is derived by solving a useful equation as shown in the following lemma.

Lemma 2: Given $v_k \geq 1$, the offloading priority function $\varphi(\beta_k, C_k, P_k, h_k)$ in (7) is the root of following equation with respect to x:

$$f'^{-1}\left(C_k P_k h_k^2\right) = g^{-1}\left(\frac{-h_k^2 x}{\beta_k}\right).$$

Lemma 2 is proved in Appendix B. The function generates an offloading priority value, $\varphi_k = \varphi(\beta_k, C_k, P_k, h_k)$, for mobile k, depending on corresponding variables quantifying fairness, local computing and channel. The amount of offloaded data by a mobile grows with an increasing offloading priority as shown in the next sub-section. It is useful to understand the effects of parameters on the offloading priority that are characterized as follows.

Lemma 3: Given $v \geq 1$, $\varphi(\beta, C, P, h)$ is a *monotone increasing function* for β, C, P and h.

Lemma 3 is proved in Appendix C by deriving the first derivatives of φ with respect to each parameter. This lemma is consistent with the intuition that, to reduce energy consumption by offloading, the BS should schedule those mobiles having high computing energy consumption per bit (i.e., large C and P) or good channels (i.e., large h).

Remark 1 (Effects of Parameters on the Offloading Priority): It can be observed from (7) and (8) that the offloading priority scales with local computing energy per bit CP approximately as $(CP)\ln(CP)$ and with the channel gain h approximately as $\ln h$. The former scaling is much faster than the latter. This shows that the computing energy per bit is dominant over the channel on determining whether to offload.

B. Optimal Resource-Allocation Policy

Based on conditions in (5)-(5c) and Lemma 2, the main result of this section is derived, given in the following theorem.

Theorem 1 (Optimal Resource-Allocation Policy): Consider the case of infinite cloud computation capacity. The optimal policy solving Problem P2 has the following structure.

1) If $v_k \leq 1$ and the minimum offloaded data size $m_k^+ = 0$ for all k, none of these users performs offloading, i.e.,

$$\ell_k^{*(2)} = t_k^{*(2)} = 0 \quad k \in \mathcal{K}.$$

2) If there exists mobile k such that $v_k > 1$ or $m_k^+ > 0$, for $k \in \mathcal{K}$,

$$\ell_k^{*(2)} \begin{cases} = m_k^+, & \varphi_k < \lambda^* \\ \in [m_k^+, R_k], & \varphi_k = \lambda^* \\ = R_k, & \varphi_k > \lambda^*, \end{cases}$$

and

$$t_k^{*(2)} = \frac{\ln 2}{B\left[W_0\left(\frac{\lambda^* h_k^2/\beta_k - N_0}{N_0 e}\right) + 1\right]} \times \ell_k^{*(2)}$$

where $W_0(x)$ is the Lambert function and λ^* is the optimal value of the Lagrange multiplier satisfying the active time-sharing constraint: $\sum_{k=1}^{K} t_k^{*(2)} = T$.

Proof: See Appendix D. $\square$

Theorem 1 reveals that the optimal resource-allocation policy has a threshold-based structure when offloading saves energy. In other words, since the exact case of $\varphi_k = \lambda^*$ rarely occurs in practice, the optimal policy makes a *binary offloading decision* for each mobile. Specifically, if the corresponding offloading priority exceeds a given threshold, namely λ^*, the mobile should offload all input data to the edge cloud; otherwise, the mobile should offload only the minimum amount of data under the computation latency constraint. This result is consistent with the intuition that the greedy method can lead to the optimal resource allocation. Note that there are two groups of users selected to perform the minimum offloading. One is the group of users for which it has positive minimum offloading data, i.e., $m_k > 0$, and offloading cannot save energy consumption since they have bad channels or small local computing energy such that $v_k \leq 1$ and $\varphi_k = 0$. The second group is the set of users for which offloading is energy-efficient, i.e., $v_k > 1$, however, have relatively small offloading priorities, i.e., $\varphi_k < \lambda^*$; they cannot perform complete offloading due to the limited radio resource.

Remark 2 (Offloading or Not?): For a conventional TDMA communication system, continuous transmission by at least one mobile is always advantageous under the criterion of minimum sum energy consumption [17]. However, this does not always hold for a TDMA MECO system where no offloading for all users may be preferred as shown in Theorem 1. Offloading is not necessary expect for two cases. First, there exists at least one mobile whose input-data size is too large such that complete local computing fails to meet the latency constraint. Second, some mobile has a sufficient high value for the product $C_k P_k h_k^2$, indicating that energy savings can be achieved by offloading because of high channel gain or large local computing energy consumption.

Remark 3 (Offloading Rate): It can be observed from Theorem 1 that the offloading rate, defined as $\ell_k^{*(2)}/t_k^{*(2)}$ for mobile k, is determined only by the channel gain and fairness factor while other factors, namely C_k and P_k, affect the offloading decision. The rate increases with a growing channel gain and vice versa since a large channel gain supports a higher transmission rate or reduces transmission power, making offloading desirable for reducing energy consumption.

Remark 4 (Low-Complexity Algorithm): The traditional method for solving Problem P2 is the block-coordinate descending algorithm which performs iterative optimization of the two sets of variables, $\{\ell_k\}$ and $\{t_k\}$, resulting in high computation complexity. In contrast, by exploiting the threshold-based structure of the optimal resource-allocation policy in Theorem 1, the proposed solution approach, described in Algorithm 1, needs to perform only a *one-dimension* search for λ^*, reducing the computation complexity significantly. To facilitate the search, next lemma gives the range of λ^*, which can be easily proved from Theorem 1.

Lemma 4: When there is at least one offloading mobile, λ^* satisfies: $0 \leq \lambda^* \leq \lambda_{\max} = \max_k \varphi_k$.

Algorithm 1 Optimal Algorithm for Solving Problem P2

- **Step 1** [Initialize]: Let $\lambda_\ell = 0$ and $\lambda_h = \lambda_{\max}$. According to Theorem 1, obtain $T_\ell = \sum_{k=1}^{K} t_{k,\ell}^{*(2)}$ and $T_h = \sum_{k=1}^{K} t_{k,h}^{*(2)}$, where $\{t_{k,\ell}^{*(2)}\}$ and $\{t_{k,h}^{*(2)}\}$ are the allocated fractions of slot for the cases of λ_ℓ and λ_h, respectively.
- **Step 2** [Bisection search]: While $T_\ell \neq T$ and $T_h \neq T$, update $\{\lambda_\ell, \lambda_h\}$ as follows.
  (1) Define $\lambda_m = (\lambda_\ell + \lambda_h)/2$ and compute T_m.
  (2) If $T_m = T$, then $\lambda^* = \lambda_m$ and the optimal policy can be determined. Otherwise, if $T_m < T$, let $\lambda_h = \lambda_m$ and if $T_m > T$, let $\lambda_\ell = \lambda_m$.

Furthermore, with the assumption of infinite cloud capacity, the effects of finite radio resource (i.e., the TDMA time-slot duration) are characterized in the following two propositions in terms of the number of offloading users, which can be easily derived using Theorem 1.

Proposition 1 (Exclusive Mobile Computation Offloading): For TDMA MECO with offloading users, only one mobile can offload computation if $T \leq \dfrac{R_m}{B \log_2\left(\frac{BC_m P_m h_m^2}{N_0 \ln 2}\right)}$ where $m = \arg\max_k \varphi_k$.

It indicates that short time slot limits the number of offloading users. From another perspective, it means that if the *winner* user m has excessive data, it will take up all the resource.

Proposition 2 (Inclusive Mobile Computation Offloading): All offloading-desired mobiles (defined as for which, it has $\varphi_k > 0$) will completely offload computation if

$$T \geq \sum_{k \in O_1} \frac{R_k \ln 2}{B\left[W_0\left(\frac{\lambda_{\min} h_k^2/\beta_k - N_0}{N_0 e}\right) + 1\right]}$$
$$+ \sum_{k \in O_2} \frac{m_k^+ \ln 2}{B\left[W_0\left(\frac{\lambda_{\min} h_k^2/\beta_k - N_0}{N_0 e}\right) + 1\right]}$$

where $O_1 = \{k \mid \varphi_k > 0\}$, $O_2 = \{k \mid \varphi_k = 0\}$ and $\lambda_{\min} = \min_{k \in O_1} \varphi_k$.

Proposition 2 reveals that when T exceeds a given threshold, the offloading-desired mobiles for which offloading brings energy savings, will offload all computation to the cloud.

Remark 5 (Which Resource is Bottleneck?): Proposition 1 and 2 suggest that as the radio resource continuously increases, the cloud will become the performance bottleneck and the assumption of infinite cloud capacity will not hold. For a short time-slot duration, only a few users can offload computation. This just requires a fraction of computation such that the cloud can be regarded as having infinite capacity. However, when the time-slot duration is large, it not only saves energy consumption by offloading but also allows more users for offloading, which potentially exceeds the cloud capacity. The case of finite-capacity cloud will be considered in the sequel.

C. Special Cases

The optimal resource-allocation policies for several special cases considering equal fairness factors are discussed as follows.

1) Uniform Channels and Local Computing: Consider the simplest case where $\{h_k, C_k, P_k\}$ are identical for all k. Then all mobiles have uniform offloading priorities. In this case, for the optimal resource allocation, all mobiles can offload arbitrary data sizes so long as the sum offloaded data size satisfies the following constraint: $\sum_{k=1}^{K} \ell_k^{*(2)} \leq TB \log_2\left(\frac{BCP h^2}{N_0 \ln 2}\right)$.

2) Uniform Channels: Consider the case of $h_1 = h_2 \cdots = h_K = h$. The offloading priority for each mobile, say mobile k, is only affected by the corresponding local-computing parameters P_k and C_k. Without loss of generality, assume that $P_1 C_1 \leq P_2 C_2 \cdots \leq P_K C_K$. Then the optimal resource-allocation policy is given in the following corollary of Theorem 1.

Corollary 1: Assume infinite cloud capacity, $h_1 = h_2 \cdots = h_K = h$ and $P_1 C_1 \leq P_2 C_2 \cdots \leq P_K C_K$. Let k_t denote the index such that $\varphi_k < \lambda^*$ for all $k < k_t$ and $\varphi_k > \lambda^*$ for all $k \geq k_t$, neglecting the rare case where $\varphi_k = \lambda^*$. The optimal resource-allocation policy is given as follows: for $k \in \mathcal{K}$,

$$\ell_k^{*(2)} = \begin{cases} R_k, & k \geq k_t \\ m_k^+, & \text{otherwise,} \end{cases}$$

and

$$t_k^{*(2)} = \frac{\ln 2}{B\left[W_0\left(\frac{\lambda^* h^2/\beta - N_0}{N_0 e}\right) + 1\right]} \times \ell_k^{*(2)}.$$

The result shows that the optimal resource-allocation policy follows a *greedy* approach that selects mobiles in a descending order of energy consumption per bit for complete offloading until the time-sharing duration is fully utilized.

3) Uniform Local Computing: Consider the case of $C_1 P_1 = C_2 P_2 \cdots = C_K P_K$. Similar to the previous case, the optimal resource-allocation policy can be shown to follow the greedy approach that selects mobiles for complete offloading in the descending order of channel gains.

V. MULTIUSER MECO FOR TDMA: FINITE CLOUD CAPACITY

In this section, we consider the case of finite cloud capacity and analyze the optimal resource-allocation policy for solving Problem P1. The policy is shown to also have a threshold-based structure as the infinite-capacity counterpart derived in the preceding section. Both the optimal and sub-optimal algorithms are presented for policy computation. The results are extended to the finite-capacity cloud with non-negligible computing time.

A. Optimal Resource-Allocation Policy

To solve the convex Problem P1, the corresponding partial Lagrange function is written as

$$\tilde{L} = \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k) C_k P_k \right]$$
$$+ \lambda \left(\sum_{k=1}^{K} t_k - T \right) + \mu \left(\sum_{k=1}^{K} C_k \ell_k - F \right) \qquad (9)$$

where $\mu \geq 0$ is the Lagrange multiplier associated with the cloud capacity constraint. Using the above Lagrange function, it is straightforward to show that the corresponding KKT conditions can be modified from their infinite-capacity counterparts in (5)-(5c) by replacing P_k with $\tilde{P}_k = P_k - \mu$, called the *effective computation energy per cycle*. The resultant *effective offloading priority function*, denoted as $\tilde{\varphi}_k$, can be modified accordingly from that in (7) as

$$\tilde{\varphi}(\beta_k, C_k, P_k, h_k, \tilde{\mu}^*)$$
$$= \begin{cases} \dfrac{\beta_k N_0}{h_k^2} (\tilde{v}_k \ln \tilde{v}_k - \tilde{v}_k + 1), & \tilde{v}_k \geq 1 \\ 0, & \tilde{v}_k < 1, \end{cases} \quad (10)$$

where $\tilde{v}_k = \dfrac{BC_k(P_k - \tilde{\mu}^*)h_k^2}{N_0 \ln 2}$.

Moreover, it can be easily derived that a cloud with smaller capacity F leads to a larger Lagrange multiplier $\tilde{\mu}^*$. It indicates that compared with φ_k in (7) for the case of infinite-capacity cloud, the effective offloading priority function here is also determined by the cloud capacity. Based on above discussion, the main result of this section follows.

Theorem 2: Consider the finite-capacity cloud with upper-bounded offloaded computation. The optimal policy solving Problem P1 has the same structure as that in Theorem 1 and is expressed in terms of the priority function $\tilde{\varphi}_k$ in (10) and optimized Lagrange multipliers $\{\tilde{\lambda}^*, \tilde{\mu}^*\}$.

Remark 6 (Variation of Offloading Priority Order): Since $\tilde{\mu}^* > 0$, it has $\tilde{\varphi}_k < \varphi_k$ for all k. Therefore, the offloading priority order may be different with that of infinite-capacity cloud, due to the varying decreasing rates of offloading priorities. The reason is that the finite-capacity cloud should make the tradeoff between energy savings and computation burden. To this end, it will select the mobiles for offloading that can save significant energy and require less computation for each bit of data.

Computing the threshold for the optimal resource-allocation policy requires a *two-dimension search* over the Lagrange multipliers $\{\tilde{\lambda}^*, \tilde{\mu}^*\}$, described in Algorithm 2. For an efficient search, it is useful to limit the range of $\tilde{\lambda}^*$ and $\tilde{\mu}^*$ shown as below, which can be easily proved.

Lemma 5: When there is at least one offloading mobile, the optimal Lagrange multipliers $\{\tilde{\lambda}^*, \tilde{\mu}^*\}$ satisfy:

$$0 \leq \tilde{\lambda}^* \leq \lambda_{\max}, \quad \text{and} \quad 0 \leq \tilde{\mu}^* \leq \mu_{\max} = \max_k \left\{ P_k - \frac{N_0 \ln 2}{BC_k h_k^2} \right\}$$

where $\lambda_{\max}$ is defined in Lemma 4.

Note that $\tilde{\mu}^* = 0$ corresponds to the case of infinite-capacity cloud and $\tilde{\mu}^* = \mu_{\max}$ to the case where offloading yields no energy savings for any mobile.

B. Sub-Optimal Resource-Allocation Policy

To reduce the computation complexity of Algorithm 2 due to the two-dimension search, one simple sub-optimal policy is proposed as shown in Algorithm 3. The key idea is to decouple the computation and radio resource allocation. In Step 2, based on the *approximated* offloading priority in (7) for the case of infinite-capacity cloud, we allocate the computation

---

Algorithm 2 Optimal Algorithm for Solving Problem P1

- **Step 1** [Check solution for Problem P2]: Perform Algorithm 1. If $\sum_{k=1}^{K} \ell_k^{*(2)} \leq F$, the optimal policy is given in Theorem 1. Otherwise, go to Step 2.
- **Step 2** [Initialize]: Let $\mu_\ell = 0$ and $\mu_h = \mu_{\max}$. Based on Theorem 2, obtain $F_\ell = \sum_{k=1}^{K} C_k \ell_{k,\ell}^*$ and $F_h = \sum_{k=1}^{K} C_k \ell_{k,h}^*$, where $\{\ell_{k,\ell}^*\}$ and $\{\ell_{k,h}^*\}$ are the offloaded data sizes for μ_ℓ and μ_h, respectively, involving the one-dimension search for $\tilde{\lambda}^*$.
- **Step 3** [Bisection search]: While $F_\ell \neq F$ and $F_h \neq F$, update $\{\mu_\ell, \mu_h\}$ as follows.
  (1) Define $\mu_m = (\mu_\ell + \mu_h)/2$ and compute F_m.
  (2) If $F_m = F$, then $\tilde{\mu}^* = \mu_m$ and the optimal policy can be determined. Otherwise, if $F_m < F$, let $\mu_h = \mu_m$ and if $F_m > F$, let $\mu_\ell = \mu_m$.

Algorithm 3 Sub-Optimal Algorithm for Solving Problem P1

- **Step 1**: Perform Algorithm 1. If $\sum_{k=1}^{K} \ell_k^{*(2)} \leq F$, Theorem 1 gives the optimal policy. Otherwise, go to Step 2.
- **Step 2**: Based on offloading priorities in (7), offload the data from mobiles in the descending order of offloading priority until the cloud computation capacity is fully occupied, i.e., $\sum_{k=1}^{K} C_k \ell_k^* = F$.
- **Step 3**: With $\{\ell_k^*\}$ derived in Step 2, perform one-dimension search for λ^* such that $\sum_{k=1}^{K} t_k^* = T$ where
$$t_k^* = \frac{\ell_k^* \ln 2}{B[W_0(\frac{\lambda^* h_k^2/\beta_k - N_0}{N_0 e}) + 1]}.$$

---

resource to mobiles with high offloading priorities. Step 3 optimizes the corresponding fractions of slot given offloaded data. This sub-optimal algorithm has low computation complexity. Specifically, let d denote the largest bisection-search interval. Given a solution accuracy $\varepsilon > 0$, the bisection method will call for $\log_2(d/\epsilon)$ times of comparison operations, and thus has the order of complexity $O(\log(1/\varepsilon))$. For each iteration, the resource-allocation complexity is $O(K)$. Therefore, the total computation complexity for the sub-optimal algorithm is $O(K \log(1/\varepsilon))$. Moreover, its performance is shown by simulation to be close-to-optimal in the sequel.

C. Extension: MECO With Non-Negligible Computing Time

Consider another finite-capacity cloud for which the computing time is non-negligible. Surprisingly, the resultant optimal policy is also threshold based, with respect to a different offloading priority function.

Assume that the edge cloud performs load balancing for the uploaded computation as in [25]. In other words, the CPU cycles are proportionally allocated for each user such that all users experience the same computing time: $(\sum_{k=1}^{K} C_k \ell_k)/F'$ (see Section II-D). Then the latency constraint is reformulated as $(\sum_{k=1}^{K} C_k \ell_k)/F' + \sum_{k=1}^{K} t_k \leq T$, accounting for both the data transmission and cloud computing time. The resultant optimization problem for minimizing weighted sum mobile

energy consumption is re-written by

$$\min_{\{\ell_k, t_k\}} \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k)C_k P_k \right]$$

$$\text{s.t.} \quad \frac{\sum_{k=1}^{K} C_k \ell_k}{F'} + \sum_{k=1}^{K} t_k \leq T,$$

$$t_k \geq 0, \quad m_k^+ \leq \ell_k \leq R_k, \quad k \in \mathcal{K}. \qquad \textbf{(P3)}$$

The key challenge of Problem P3 is that the amount of offloaded data size for each user has effects on offloading energy consumption, offloading duration and cloud computing time, making the problem more complicated.

The feasibility condition for Problem P3 can be easily obtained as: $(\sum_{k=1}^{K} C_k m_k^+)/F' < T$. Note that the case $(\sum_{k=1}^{K} C_k m_k^+)/F' = T$ makes Problem P3 infeasible since the resultant offloading time ($t_k = 0$) cannot enable computation offloading.

Similarly, to solve Problem P3, the partial Lagrange function is written as

$$\widehat{L} = \sum_{k=1}^{K} \beta_k \left[\frac{t_k}{h_k^2} f\left(\frac{\ell_k}{t_k}\right) + (R_k - \ell_k)C_k P_k \right]$$

$$+ \lambda \left(\frac{\sum_{k=1}^{K} C_k \ell_k}{F'} + \sum_{k=1}^{K} t_k - T \right).$$

Define two sets of important constants: $a_k = \frac{F' \ln 2}{B C_k}$ and $b_k = \frac{F' P_k h_k^2}{N_0}$ for all k. Using KKT conditions, we can obtain the following offloading priority function

$$\widehat{\varphi}(\beta_k, C_k, P_k, h_k, F')$$
$$= \begin{cases} \frac{\beta_k N_0}{h_k^2} (\widehat{v}_k \ln \widehat{v}_k - \widehat{v}_k + 1), & \widehat{v}_k \geq 1 \\ 0, & \widehat{v}_k < 1, \end{cases} \qquad (11)$$

where

$$\widehat{v}_k = \frac{b_k - 1}{W_0((b_k - 1)e^{(a_k - 1)})}. \qquad (12)$$

This function is derived by solving a equation in the following lemma, proved in Appendix E.

Lemma 6: Given $\widehat{v}_k \geq 1$, the offloading priority function $\widehat{\varphi}_k = \widehat{\varphi}(\beta_k, C_k, P_k, h_k, F')$ in (11) is the root of the following equation with respect to x:

$$f'^{-1}\left(C_k P_k h_k^2 - \frac{x C_k h_k^2}{\beta_k F'}\right) = g^{-1}\left(\frac{-h_k^2 x}{\beta_k}\right). \qquad (13)$$

Recall that for a cloud that upper-bounds the offloaded computation, its offloading priority (i.e., $\widetilde{\varphi}_k$ in (10)) is function of a Lagrange multiplier $\widetilde{\mu}^*$ which is determined by F. However, for the current cloud with non-negligible computing time, the offloading priority function $\widehat{\varphi}_k$ in (11) is directly affected by the finite cloud capacity F' via $\widehat{v}_k$.

In the following, the properties of $\widehat{v}_k$, which is the key component of $\widehat{\varphi}_k$, are characterized.

Lemma 7: $\widehat{v} > 1$ if and only if $v > 1$, where v is defined in (8).

It is proved in Appendix F and indicates that the condition that offloading saves energy comsumption for this kind of finite-capacity cloud is same as that of infinite-capacity cloud.

Lemma 8: Given $\widehat{v} \geq 1$, $\widehat{\varphi}(\beta, C, P, h, F')$ is a monotone increasing function for β, C, P, h and F', respectively.

Lemma 8 can be proved by deriving the first derivatives of $\widehat{\varphi}$ with respect to each parameter. It shows that enhancing the cloud capacity will increase the offloading priority for all users that is same as the result of a cloud with upper-bounded offloaded computation.Based on above discussion, the main result of this section is presented in the following theorem.

Theorem 3: Consider the finite-capacity cloud with non-negligible computing time. The optimal resource allocation policy solving Problem P3 has the same structure as that in Theorem 1 and is expressed in terms of the priority function $\widehat{\varphi}_k$ in (11) and optimized Lagrange multipliers $\widehat{\lambda}^*$.

The optimal policy can be computed with a one-dimension search for $\widehat{\lambda}^*$, following a similar procedure in Algorithm 1.

VI. MULTIUSER MECO FOR OFDMA

In this section, consider resource allocation for MECO OFDMA. Both OFDM sub-channels and offloaded data sizes are optimized for the energy-efficient multiuser MECO. To solve the formulated mixed-integer optimization problem, a sub-optimal algorithm is proposed by defining an average offloading priority function from its TDMA counterpart and shown to have close-to-optimal performance in simulation.

A. Multiuser MECO for OFDMA: Infinite Cloud Capacity

Consider an OFDMA system (see Section II) with K mobiles and N sub-channels. The cloud is assumed with infinite cloud capacity. Given time-slot duration T, the latency constraint for local computing is rewritten as $C_k(R_k - \sum_{n=1}^{N} \ell_{k,n})/F_k \leq T$. Moreover, the time-sharing constraint is replaced by sub-channel constraints, expressed as $\sum_{k=1}^{K} \rho_{k,n} \leq 1$ for all n. Then the corresponding optimization problem for the minimum weighted sum mobile energy consumption based on OFDMA is readily re-formulated as:

$$\min_{\{\ell_{k,n}, \rho_{k,n}\}} \sum_{k=1}^{K} \beta_k \left[\sum_{n=1}^{N} \frac{\rho_{k,n}}{\overline{h}_{k,n}^2} \bar{f}\left(\frac{\ell_{k,n}}{\rho_{k,n}}\right) + (R_k - \sum_{n=1}^{N} \ell_{k,n})C_k P_k \right]$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \rho_{k,n} \leq 1, \quad n \in \mathcal{N};$$

$$m_k^+ \leq \sum_{n=1}^{N} \ell_{k,n} \leq R_k, \quad k \in \mathcal{K};$$

$$\rho_{k,n} \in \{0, 1\}, \quad n \in \mathcal{N} \text{ and } k \in \mathcal{K}. \qquad \textbf{(P4)}$$

Observe that Problem P4 is a mixed-integer programming problem that is difficult to solve. It involves the joint optimization of both continuous variables $\{\ell_{k,n}\}$ and integer variables $\{\rho_{k,n}\}$. One common solution method is relaxation-and-rounding, which firstly relaxes the integer constraint $\rho_{k,n} \in \{0, 1\}$ as the real-value constraint $0 \leq \rho_{k,n} \leq 1$ [18], and then determines the integer solution using rounding techniques. Note that the integer-relaxation problem is a convex

problem which can be solved by powerful convex optimization techniques. An alternative method is using dual decomposition as in [26], which has been proved to be optimal when the number of sub-channels goes to infinity. However, both algorithms performing extensive iterations shed little insight on the policy structure.

To reduce the computation complexity and characterize the policy structure, a low-complexity sub-optimal algorithm is proposed below by a decomposition method, motivated by the following existing results and observations. First, for traditional OFDMA systems, low-complexity sub-channel allocation policy was designed in [27] and [28] via defining average channel gains, which was shown to achieve close-to-optimal performance in simulation. Next, for the integer-relaxation resource allocation problem, applying KKT conditions directly can lead to its optimal solution. It can be observed that for each sub-channel, users with higher offloading priorities should be allocated with more radio resource. Therefore, in the proposed algorithm, the initial resource and offloaded data allocation is firstly determined by defining average channel gains and an *average offloading priority function*. Then, the integer sub-channel assignment is performed according to the offloading priority order, followed by the adjustment of offloaded data allocation over assigned sub-channels for each user. The main procedures of this *sequential* algorithm are as follows.

- **Phase 1** [Sub-Channel Reservation for Offloading-Required Users]: Consider the offloading-required users that have $m_k^+ > 0$. The offloading priorities for these users are ordered in the descending manner. Based on this, the available sub-channels with high priorities are assigned to corresponding users sequentially and each user is allocated with one sub-channel.
- **Phase 2** [Initial Resource and Offloaded Data Allocation]: For the unassigned sub-channels, using average channel gain over these sub-channels for each user, the OFDMA MECO problem is transformed into its TDMA counterpart. Then, by defining an average offloading priority function, the optimal total sub-channel number and offloaded data size for each user are derived. Note that the resultant sub-channel numbers may not be integer.
- **Phase 3** [Integer Sub-Channel Assignment]: Given constraints on the rounded total sub-channel numbers for each user derived in Phase 2, specific integer sub-channel assignment is determined by the offloading priority order. Specifically, each sub-channel is assigned to the user that requires sub-channel assignment and has higher offloading priority than others.
- **Phase 4** [Adjustment of Offloaded Data Allocation]: For each user, based on the sub-channel assignment in Phase 3, the specific offloaded data allocation is optimized.

Before stating the algorithm, let $\varphi_{k,n}$ define the offloading priority function for user k at sub-channel n. It can be modified from the TDMA counterpart in (7) by replacing h_k, N_0 and v_k with $h_{k,n}$, $\bar{N}_0$ and $v_{k,n} = \frac{\bar{B}TC_kP_k\bar{h}_{k,n}^2}{\bar{N}_0\ln 2}$, respectively. Let Φ reflect the offloading priority order, which is constituted by

---

Algorithm 4 Sub-Channel Reservation for Offloading-Required Users

While $\mathcal{K}_1 \neq \emptyset$, reserve sub-channels as follows.

(1) Let $\rho_{k',n'} = 1$ where $\{k', n'\} = \underset{k \in \mathcal{K}_1, n \in \mathcal{N}_2}{\arg\max} \varphi_{k,n}$.

(2) Update sets: $\mathcal{S}_{k'} = \mathcal{S}_{k'} \cup \{n'\}$; $\mathcal{K}_1 = \mathcal{K}_1 \setminus \{k'\}$; $\mathcal{N}_1 = \mathcal{N}_1 \cup \{n'\}$; $\mathcal{N}_2 = \mathcal{N} \setminus \mathcal{N}_1$.

$\{\varphi_{k,n}\}$, arranged in the descending manner, e.g., $\{\varphi_{2,3} \geq \varphi_{1,4} \geq \cdots \varphi_{5,2}\}$. The set of offloading-required users is denoted by $\mathcal{K}_1$, given as $\mathcal{K}_1 = \{k, \ |m_k^+ > 0\}$. The sets of assigned and unassigned sub-channels are denoted by $\mathcal{N}_1$ and $\mathcal{N}_2$, initialized as $\mathcal{N}_1 = \emptyset$ and $\mathcal{N}_2 = \mathcal{N}$. For each user, say user k, the assigned sub-channel set is represented by $\mathcal{S}_k$, initialized as $\mathcal{S}_k = \emptyset$. In addition, sub-channel assignment indicators are set as $\{\rho_{k,n} = 0\}$ at the beginning.

Using these definitions, the detailed control policies are elaborated as follows.

1) Sub-Channel Reservation for Offloading-Required Users: The purpose of this phase is to guarantee that the computation latency constraints are satisfied for all users. This can be achieved by reserving one sub-channel for each offloading-required user as presented in Algorithm 4.

Observe that Step 1 in the loop searches for the highest offloading priority $\varphi_{k',n'}$ over unassigned sub-channels $\mathcal{N}_2$ for the remaining offloading-required users $\mathcal{K}_1$; and then allocates sub-channel n' to user k'. This sequential sub-channel assignment follows the descending offloading priority order. Moreover, the condition for the loop ensures that all offloading-required users will be allocated with one sub-channel. This phase only has a complexity of $O(K)$ since it just performs the max operation for at most K iterations.

2) Initial Resource and Offloaded Data Allocation: This phase determines the *total* allocated sub-channel number and offloaded data size for each user. Note that the integer constraint on sub-channel allocation makes Problem P4 challenging, which requires an exhaustive search. To reduce the computation complexity, we first derive the non-integer total number of sub-channels for each user as below.

Using a similar method in [28], for each user, say user k, let H_k denote its average sub-channel gain, give by $H_k = \sqrt{(\sum_{n \in \mathcal{N}_2} \bar{h}_{k,n}^2)/|\mathcal{N}_2|}$ where $|\mathcal{N}_2|$ gives the cardinality of unassigned sub-channel set $\mathcal{N}_2$ resulted from Phase 1. Then, the MECO OFDMA resource allocation Problem P4 is transformed into its TDMA counterpart Problem P5 as:

$$\min_{\{\ell_k, n_k\}} \sum_{k=1}^{K} \beta_k \left[\frac{n_k}{H_k^2} \bar{f}\left(\frac{\ell_k}{n_k}\right) + (R_k - \ell_k)C_kP_k \right]$$

$$\text{s.t.} \ \sum_{k=1}^{K} n_k \leq |\mathcal{N}_2|,$$

$$n_k \geq 0, \quad m_k^+ \leq \ell_k \leq R_k, \ k \in \mathcal{K} \qquad \textbf{(P5)}$$

where $\{\ell_k, n_k\}$ are the allocated total sub-channel numbers and offloaded data sizes.

Define an *average offloading priority function* as in (7) by replacing h_k with H_k. The optimal control policy, denoted by $\{\ell_k^*, n_k^*\}$, can be directly obtained following the same method as for Theorem 1. Note that this phase only invokes the bisection search. Similar to Section V-B, the computation complexity can be represented by $O(K \log(1/\varepsilon))$.

3) Integer Sub-Channel Assignment: Given the non-integer total sub-channel number allocation obtained in Phase 2, in this phase, users are assigned with specific integer sub-channels based on offloading priority order. Specifically, it includes the following two steps as in Algorithm 5.

In the first step, to guarantee that sub-channels are enough for allocation, each user is allocated with $\tilde{n}_k^* = \lfloor n_k^* \rfloor$ sub-channels. However, allocating specific sub-channels to users given the rounded numbers is still hard, for which the optimal solution can be obtained using the Hungarian Algorithm [29] that has the complexity of $O(N^3)$. To further reduce the complexity, a priority-based sub-channel assignment is proposed as follows. Let $\tilde{\mathcal{K}}$ denote the set of users that require sub-channel assignment, which is initialized as $\tilde{\mathcal{K}} = \{k, |\tilde{n}_k^* > 0\}$ and will be updated as in Step 1.(3), by deleting the user that has been allocated with the maximum sub-channels. During the loop, for users in set $\tilde{\mathcal{K}}$ and available sub-channels $\mathcal{N}_2$, we search for the highest offloading priority function, indexed as $\varphi_{k',n'}$, and assign sub-channel n' to user k'.

In the second step, all users compete for remaining sub-channels since $\tilde{n}_k^*$ is the lower-rounding of n_k^* in the first step. In particular, each unassigned sub-channel in $\mathcal{N}_2$ is assigned to the user with highest offloading priority. In total, the computation complexity of this phase is $O(N)$.

4) Adjustment of Offloaded Data Allocation: Based on results from Phase 1–3, for each user, say k, this phase allocates the total offloaded data ℓ_k^* over assigned sub-channels $\mathcal{S}_k$ for minimizing the individual mobile energy consumption. The corresponding optimization problem is formulated as below with the solution given in Proposition 3.

$$\min_{\{\ell_{k,n}\}} \sum_{n \in \mathcal{S}_k} \frac{1}{\bar{h}_{k,n}^2} \bar{f}\left(\ell_{k,n}\right)$$
$$\text{s.t.} \sum_{n \in \mathcal{S}_k} \ell_{k,n} = \ell_k^*,$$
$$\ell_{k,n} \geq 0, \quad n \in \mathcal{S}_k. \qquad \textbf{(P6)}$$

Proposition 3: For user k, the optimal offloaded data allocation solving Problem P6 is

$$\ell_{k,n}^* = \left[\bar{B}T \log_2\left(\frac{\xi_k \bar{B}T\bar{h}_{k,n}^2}{\bar{N}_0 \ln 2}\right)\right]^+ \quad \text{for } n \in \mathcal{S}_k$$

where ξ_k satisfies $\sum_{n \in \mathcal{S}_k} \ell_{k,n}^* = \ell_k^*$.

Note that it is possible that some sub-channels are allocated to user k but without offloaded data allocation due to their poor sub-channel gains. For each user, the optimal solution is obtained by performing one-dimension search for ξ_k, whose computation complexity is $O(N \log(1/\varepsilon))$ since $|\mathcal{S}_k| \leq N$. Thus, the total complexity of this phase is $O(KN \log(1/\varepsilon))$, considering offloaded data allocation for all users.

Remark 7 (Low-Complexity Algorithm): Based on above discussion, the total complexity for the proposed sequential

---

Algorithm 5 Integer Sub-Channel Assignment

Step 1: While $\tilde{\mathcal{K}} \neq \emptyset$, assign sub-channels as follows.
(1) Let $\rho_{k',n'} = 1$ where $\{k', n'\} = \underset{k \in \tilde{\mathcal{K}}, n \in \mathcal{N}_2}{\arg \max} \varphi_{k,n}$.
(2) Update sets: $\quad \mathcal{S}_{k'} = \mathcal{S}_{k'} \cup \{n'\}; \quad \mathcal{N}_1 = \mathcal{N}_1 \cup \{n'\};$
$\mathcal{N}_2 = \mathcal{N} \setminus \mathcal{N}_1$.
(3) If $|\mathcal{S}_{k'}| = \tilde{n}_{k'}^*$, then $\tilde{\mathcal{K}} = \tilde{\mathcal{K}} \setminus \{k'\}$.
Step 2: If $\mathcal{N}_2 \neq \emptyset$, assign remaining sub-channels as follows. For each $n \in \mathcal{N}_2$, let $\rho_{k',n} = 1$ where $k' = \arg \max_{k \in \mathcal{K}} \varphi_{k,n}$.

---

sub-optimal algorithm is up to $O(K + N + KN \log(1/\varepsilon))$. It significantly reduces the computation complexity compared with that of relaxation-and-rounding policy having the complexity order of $O((KN)^{3.5} \log(1/\varepsilon) + N)$ accounting for the worst case. Specifically, the relaxation-and-rounding policy can be derived by solving a relaxation problem using CVX followed by a rounding technique. The CVX solver is based on the interior-point method which has complexity order $O((KN)^{3.5} \log(1/\varepsilon))$ [30], where KN is the total number of variables, $(KN)^{3.5}$ characterizes the complexity order of dominant Hession matrix calculation, and $\log(1/\varepsilon)$ is the iteration complexity order. Moreover, the rounding technique allocates each sub-channel to the user with largest sub-channel ratio, and thus has complexity order of $O(N)$.

B. Multiuser MECO for OFDMA: Finite Cloud Capacity

For the case of finite-capacity cloud based on OFDMA, the corresponding sub-optimal low-complexity algorithm can be derived by modifying that of infinite-capacity cloud as follows.

Recall that for TDMA MECO, modifying the offloading priority function of infinite-capacity cloud leads to the optimal resource allocation for the finite-capacity cloud. Therefore, by the similar method, modifying Phase 2 to account for the finite computation capacity will give the new optimal initial resource and offloaded data allocation for all users. Other phases in Section VI-A can be straightforwardly extended to the current case and are omitted for simplicity.

VII. SIMULATION RESULTS

In this section, the performance of the proposed resource-allocation algorithms for both the TDMA and OFDMA systems is evaluated by simulation based on 200 channel realizations. The simulation settings are as follows unless specified otherwise. There are 30 users in the system with equal fairness factors, i.e., $\beta_k = 1$ for all k such that the weighted sum mobile energy consumption represents the total mobile energy consumption. The time slot $T = 100$ ms. Both channel h_k in TDMA and sub-channel $h_{k,n}$ in OFDAM are modeled as independent Rayleigh fading with average power loss set as 10^{-3}. The variance of complex white Gaussian channel noise $N_0 = 10^{-9}$ W. Consider mobile k. The CPU computation capacity F_k is uniformly selected from the set $\{0.1, 0.2, \cdots, 1.0\}$ GHz and the local computing energy per cycle P_k follows a uniform distribution in the range $(0, 20 \times 10^{-11})$ J/cycle similar to [12]. For the computing task, both the data size and required number of CPU cycles per bit follow the uniform distribution with $R_k \in [100, 500]$ KB

and $C_k \in [500, 1500]$ cycles/bit. All random variables are independent for different mobiles, modeling heterogeneous mobile computing capabilities. Last, the finite-capacity cloud is modeled by the one with upper-bounded offloaded computation, set as $F = 6 \times 10^9$ cycles per slot.[3]

A. Multiuser MECO for TDMA

Consider an MECO system where the bandwidth $B = 10$ MHz. For performance comparison, a baseline *equal resource-allocation* policy is considered, which allocates equal offloading time duration for mobiles that satisfy $v_k > 1$ and based on this, the offloaded data sizes are optimized.

Fig. 3(a) shows the curves of total mobile energy consumption versus the time slot duration T. Several observations can be made. First, the total mobile energy consumption reduces as the time-slot duration grows. Next, the sub-optimal policy computed using Algorithm 3 is found to have close-to-optimal performance and yields total mobile energy consumption less than half of that for the equal resource-allocation policy. The energy reduction is more significant for a shorter time slot duration. The reason is that without the optimization on time fractions, the offloading energy of baseline policy grows exponentially as the allocated time fractions decrease.
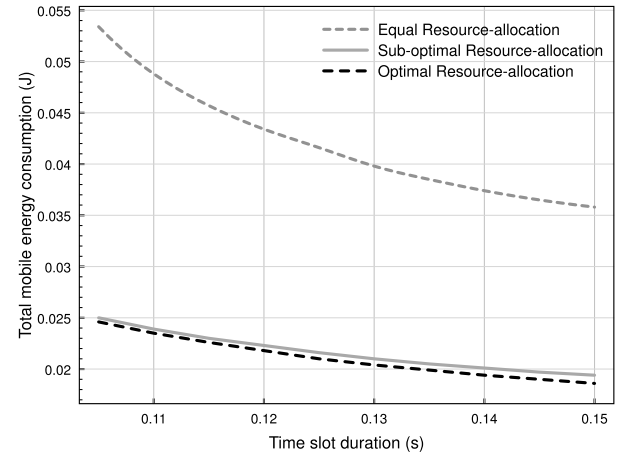
The curves of total mobile energy consumption versus the cloud computation capacity are displayed in Fig. 3(b). It can be observed that the performance of the sub-optimal policy approaches to that of the optimal one when the cloud computation capacity increases and achieves substantial energy savings gains over the equal resource-allocation policy. Furthermore, the total mobile energy consumption is invariant after the cloud computation capacity exceeds some threshold (about 6×10^9). This suggests that there exists some critical value for the cloud computation capacity, above which increasing the capacity yields no reduction on the total mobile energy consumption.

Last, Fig. 3(c) plots the curves of total energy consumption versus the number of mobiles given fixed cloud computation capacity set as $F = 6 \times 10^9$ cycles per slot. It shows the total energy consumption of the proposed policy grows with the number of mobiles at a much slower rate than that of the equal-allocation policy. Again, the designed sub-optimal policy is observed to have close-to-optimality.

B. Multiuser MECO for OFDMA

Consider an OFDMA system where $F = 5 \times 10^{15}$ cycles per slot (modeling large cloud capacity), $\bar{B} = 1$ MHz and $\bar{N}_0 = 10^{-9}$ W. The proposed low-complexity sub-optimal resource allocation policy is compared with two baseline policies. One is the *relaxation-and-rounding resource-allocation* policy, for which the integer-relaxation convex problem is computed by a convex problem solver, CVX in Matlab, and the integer solution is determined by rounding technique. The other one is a *greedy resource-allocation* policy. It assigns each sub-channel to the user that has highest offloading priority over this sub-channel, followed by the optimal data allocation over assigned sub-channels for each user. However, this policy does not consider the effect of heterogeneous computation loads.
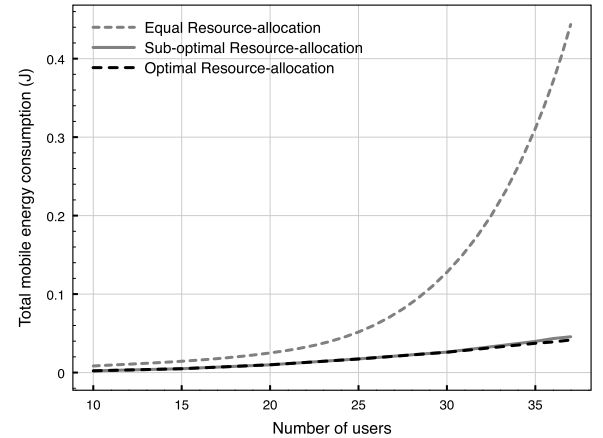

(a) Effect of the time slot duration.
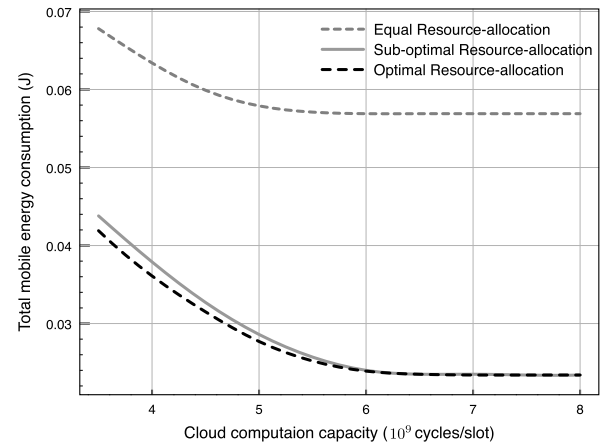
(b) Effect of the cloud computation capacity.

(c) Effect of the number of users.

Fig. 3. (a) Total mobile energy consumption vs. time slot duration for a TDMA system. (b) Total mobile energy consumption vs. cloud computation capacity for a TDMA system. (c) Total mobile energy consumption vs. number of users for a TDMA system.

Table I presents the results of total mobile energy consumption versus the number of sub-channels in an OFDMA MECO system with 8 users, where RA is short for

---

[3]The performance of finite-capacity cloud with non-negligible computing time has similar observations and is omitted due to limited space.

TABLE I

TOTAL MOBILE ENERGY CONSUMPTION (J) VS. NUMBER OF SUB-CHANNELS

Number of Sub-channels	8	16	32	64	128	256
Relaxation-and-rounding RA	0.0041	0.0013	0.0008	0.0006	0.0005	0.0004
Sub-optimal RA	0.0073	0.0021	0.0010	0.0007	0.0006	0.0005
Greedy RA	0.0129	0.0099	0.0078	0.0064	0.0054	0.0046

TABLE II

TOTAL MOBILE ENERGY CONSUMPTION (J) VS. NUMBER OF USERS

Number of Users	4	8	12	16	20
Relaxation-and-rounding RA	0.0002	0.0005	0.0008	0.0011	0.0014
Sub-optimal RA	0.0003	0.0006	0.0010	0.0013	0.0016
Greedy RA	0.001	0.0051	0.0109	0.0181	0.0258

TABLE III

AVERAGE RUNNING TIME (s) VS. NUMBER OF SUB-CHANNELS

Number of Sub-channels	8	16	32	64	128	256
Relaxation-and-rounding RA	9.5367	15.6099	29.1621	54.6303	114.6334	247.4548
Sub-optimal RA	0.3085	0.4019	0.4582	0.5369	0.5679	0.6338
Greedy RA	0.3476	0.4179	0.4822	0.5241	0.5653	0.6118

TABLE IV

AVERAGE RUNNING TIME (s) VS. NUMBER OF USERS

Number of Users	4	8	12	16	20
Relaxation-and-rounding RA	52.2648	113.3511	180.5420	255.2736	355.2360
Sub-optimal RA	0.3978	0.5374	0.7229	0.9313	1.1758
Greedy RA	0.4123	0.5519	0.6576	0.7518	0.8607

resource-allocation. It can be observed that the performance of proposed sub-optimal resource allocation is close to that of relaxation-and-rounding policy, especially when the number of sub-channels is large (e.g., 256). However, the proposed sub-optimal policy has much smaller computation complexity as discussed in Remark 7. The comparison of average running time of different policies will also be given in the sequel. In addition, the proposed policy has significant energy-savings gain over the greedy policy. The reason is that it considers the varying computation loads over users and allocates more sub-channels to heavy-loaded users, while the greedy policy only offloads computation from users with high priorities. Moreover, it suggests that increasing the number of sub-channels has little effect on the energy savings for the system if this number is above a threshold (about 64), but otherwise it decreases the total mobile energy consumption significantly.

Table II gives the results of total mobile energy consumption versus the number of users for an OFDMA system with 128 sub-channels. It shows that the total mobile energy consumptions for three policies increase with the number of users in the same trend that is almost linear. However, the proposed policy has much smaller increasing rate than the greedy one and approaches the performance of the relaxation-and-rounding policy.

Last, we compare the computation complexity of different policies by evaluating their average running time using Matlab on a computer equipped with Intel Core i5-4570, 3.20GHz processor and 8GB RAM memory. Table III gives the results of average running time versus the number of sub-channels in an OFDMA MECO system with 8 users. It can be observed

that the proposed policy has much shorter average running time than the relaxation-and-rounding policy, especially when the number of sub-channels is large (e.g., 256). This is due to the fact that the proposed policy is computed by the simple sequential algorithm while the latter involves complicated matrix calculations and iterations. On the other hand, the proposed policy consumes comparable running time with the greedy one but it can achieve significant energy-savings gain (see Table I). Similar observations can be found on Table IV that shows the results of average running time versus the number of users for an OFDMA system with 128 sub-channels.

VIII. CONCLUSION

This work studies resource allocation for a multiuser MECO system based on TDMA/OFDMA, accounting for both the cases of infinite and finite cloud computation capacities. For the TDMA MECO system, it shows that to minimize weighted sum mobile energy consumption, the optimal resource allocation policy should have a threshold-based structure. Specifically, we derive an offloading priority function that depends on the local computing energy and channel gains. Based on this, the BS makes a binary offloading decision for each mobile, where users with priorities above and below a given threshold will perform complete and minimum offloading. Then, we extend this threshold-based policy structure to the OFDMA system and design a low-complexity algorithm to solve the formulated mixed-integer optimization problem, which has close-to-optimal performance in simulation.

APPENDIX

A. Proof of Lemma 1

Since $f(x)$ is a convex function, its perspective function [31], i.e., $t_k f(\frac{\ell_k}{t_k})$, is still convex. Using the same technique in [17], jointly considering the cases for $t_k = 0$ and $t_k > 0$, $f(x)$ is still convex. Thus, the objective function, the summation of a set of convex functions, preserves the convexity. Combining it with the linear convex constraints leads to the result. ∎

B. Proof of Lemma 2

First, we derive a general result that is the root of equation: $f'^{-1}(p) = g^{-1}(y)$ with respect to y as follows. According to the definitions of $f(x)$ and $g(x)$, it has

$$f'(x) = \frac{N_0 \ln 2}{B} 2^{\frac{x}{B}} \text{ and } f'^{-1}(y) = B \log_2\left(\frac{By}{N_0 \ln 2}\right). \quad (14)$$

Therefore, the solution for the general equation is

$$
\begin{aligned}
y = g(f'^{-1}(p)) &= f(f'^{-1}(p)) - f'^{-1}(p) \times f'(f'^{-1}(p)) \\
&= f(f'^{-1}(p)) - f'^{-1}(p) \times p \\
&= \frac{Bp}{\ln 2} - N_0 - pB \log_2\left(\frac{Bp}{N_0 \ln 2}\right).
\end{aligned} \quad (15)
$$

Note that to ensure $\ell_k^{*(2)} \geq 0$ in Problem P1, it requires $f'^{-1}(C_k P_k h_k^2) \geq 0$ from (6). Combining this with (14), it leads to $v_k \geq 1$ where v_k is defined in (8). Then, substituting $p = C_k P_k h_k^2$ and $y = \frac{-h_k^2 x}{\beta_k}$ to (15) and making arithmetic operations gives the desired result as in (7). ∎

C. Proof of Lemma 3

First, the monotone increasing property in terms of β is straightforward, since the offloading priority function in (7) is linear to β. Next, by rewriting (7) as

$$\varphi(\beta, C, P, h) = \beta BCP\left[\log_2\left(\frac{BCPh^2}{N_0 \ln 2}\right) - \frac{1}{\ln 2}\right] + \frac{\beta N_0}{h^2},$$

it is easy to conclude that $\varphi(\beta, C, P, h)$ is monotone increasing with respect to C and P. Last, the first derivative of $\varphi(\beta, C, P, h)$ for h can be derived as:

$$\frac{\partial \varphi(\beta, C, P, h)}{\partial h} = \frac{2\beta(BCPh^2 - N_0 \ln 2)}{h^3 \ln 2}.$$

For $v = \frac{BCPh^2}{N_0 \ln 2} \geq 1$, we have $\frac{\partial \varphi(\beta, C, P, h)}{\partial h} \geq 0$, leading to the desired results. ∎

D. Proof of Theorem 1

First, to prove this theorem, we need the following two lemmas which can be easily proved using the definition of Lambert function and its property.

Lemma 9: The function $g^{-1}(y)$ can be expressed as

$$g^{-1}(y) = \frac{B\left[W_0(\frac{y+N_0}{-N_0 e}) + 1\right]}{\ln 2}.$$

Lemma 10: The function $g^{-1}(y)$ is a monotone decreasing function for $y < 0$.

Then, consider case 1) in Theorem 1. Note that for mobile k, if $m_k^+ = 0$ and $v_k \leq 1$, it results in $\ell_k^{*(2)} = 0$ derived from (5). Thus, if these two conditions are satisfied for all k, it leads to $\ell_k^{*(2)} = t_k^{*(2)} = 0$.

For case 2), if there exists mobile k such that $v_k > 1$ or $m_k^+ > 0$, it leads to $\ell_k^{*(2)} > 0$. And the time-sharing constraint should be active since remaining time can be used for extending offloading duration so as to reduce transmission energy. Moreover, consider each user, say user k. If $v_k \geq 1$, then from (5) and (5b), $\{\ell_k^{*(2)}, t_k^{*(2)}\}$ should satisfy the following:

$$\frac{\ell_k^{*(2)}}{t_k^{*(2)}} = \min\left\{ \max\left[\frac{m_k^+}{t_k^{*(2)}}, f'^{-1}\left(C_k P_k h_k^2\right)\right], \frac{R_k}{t_k^{*(2)}}\right\} \quad (16a)$$

$$= \max\left\{ \frac{m_k^+}{t_k^{*(2)}}, \min\left[f'^{-1}\left(C_k P_k h_k^2\right), \frac{R_k}{t_k^{*(2)}}\right]\right\} \quad (16b)$$

$$= g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right). \quad (16c)$$

Using Lemma 2 and Lemma 10, we have the following:

1) If $\varphi_k > \lambda^* \geq 0$, it has $-h_k^2 \varphi_k < -h_k^2 \lambda^* \leq 0$. Then, from (16a), it gives

$$
\begin{aligned}
\max&\left[\frac{m_k^+}{t_k^{*(2)}}, f'^{-1}\left(C_k P_k h_k^2\right)\right] \\
&\geq f'^{-1}\left(C_k P_k h_k^2\right) \\
&= g^{-1}\left(\frac{-h_k^2 \varphi_k}{\beta_k}\right) > g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right).
\end{aligned} \quad (17)
$$

From (16a), (16c) and (17), it follows that $\ell_k^{*(2)} = R_k$.

2) If $\varphi_k = \lambda^*$, it has $f'^{-1}\left(C_k P_k h_k^2\right) = g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right)$.

3) If $0 \leq \varphi_k < \lambda^*$, it has $-h_k^2 \varphi_k > -h_k^2 \lambda^*$. Combining it with (16b) leads to

$$
\begin{aligned}
\min&\left[f'^{-1}\left(C_k P_k h_k^2\right), \frac{R_k}{t_k^{*(2)}},\right] \\
&\leq f'^{-1}\left(C_k P_k h_k^2\right) \\
&= g^{-1}\left(\frac{-h_k^2 \varphi_k}{\beta_k}\right) < g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right).
\end{aligned} \quad (18)
$$

From (16b), (16c) and (18), it follows that $\ell_k^{*(2)} = m_k^+$.

Furthermore, if $v_k < 1$, it has $\ell_k^{*(2)} = m_k^+$. Note that this case can be included in the scenario of $\varphi_k < \lambda^*$ with the definition of φ_k in (7). Last, from (16c), it follows that

$$t_k^{*(2)} = \frac{\ell_k^{*(2)}}{g^{-1}\left(\frac{-h_k^2 \lambda^*}{\beta_k}\right)} = \frac{\ell_k^{*(2)} \ln 2}{B\left[W_0(\frac{\lambda^* h_k^2 / \beta_k - N_0}{N_0 e}) + 1\right]} \quad (19)$$

where (19) is obtained using Lemma 9, ending the proof. ∎

E. Proof of Lemma 6

First, by arithmetic operations with the Lambert function, it can be proved that the solution for a general equation $x \ln x + px = q$ is $x = \dfrac{q}{W_0(q \times e^p)}$.

Next, to solve equation (13), let $y_k = C_k P_k h_k^2 - \dfrac{x C_k h_k^2}{\beta_k F'}$ and use the derivation method in Lemma 2, it has

$$\frac{F' y_k}{C_k} - F' P_k h_k^2 = \frac{B y_k}{\ln 2} - N_0 - y_k B \log_2 \left(\frac{B y_k}{N_0 \ln 2} \right). \quad (20)$$

Defining $z_k = \dfrac{B y_k}{N_0 \ln 2}$, (20) can be rewritten as

$$z_k \ln z_k + (a_k - 1) z_k = b_k - 1, \quad (21)$$

where a_k and b_k are defined in Lemma 6. Using Lambert function, the solution for (21) can be obtained: $z_k = \widehat{v}_k$ where $\widehat{v}_k$ is defined in (12). Then, it follows that

$$\begin{aligned}
x &\overset{(a)}{=} \beta_k F' P_k (1 - \frac{N_0 \ln 2}{B P_k C_k h_k^2} z_k) \\
&\overset{(b)}{=} \beta_k F' P_k (1 - \frac{a_k}{b_k} z_k) \\
&\overset{(c)}{=} \frac{\beta_k N_0}{h_k^2} (z_k \ln z_k - z_k + 1) \quad (22)
\end{aligned}$$

where (a) comes from the relationship among x, y_k and z_k; (b) follows the definition of a_k and b_k; (c) is derived from (21). This leads to the desired result. ∎

F. Proof of Lemma 7

It is equivalent to proved as below that when $\widehat{v} > 1$, it has $b \geq a$. According to the definition of Lambert function, it has $b - 1 = W_0((b-1)e^{(b-1)})$. Then, it leads to

$$\widehat{v} = \frac{b-1}{W_0((b-1)e^{(a-1)})} = \frac{W_0((b-1)e^{(b-1)})}{W_0((b-1)e^{(a-1)})} \geq 1. \quad (23)$$

Using the monotone increasing property of Lambert function, (23) is equivalent to $b \geq a$. ∎

REFERENCES

[1] M. Swan, "Sensor mania! The Internet of things, wearable computing, objective metrics, and the quantified self 2.0," *J. Sens. Actuator Netw.*, vol. 1, no. 3, pp. 217–253, 2012.

[2] "Mobile-edge computing introductory technical white paper," Dept. Mobile-Edge Comput. Ind. Initiative, ETSI, Sophia Antipolis, France, White Paper, Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

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

[4] E. Cuervo, *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM MobiSys*, San Francisco, CA, USA, Mar. 2010, pp. 49–62.

[5] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[6] H. N. Van, F. D. Tran, and J.-M. Menaud, "Performance and power management for cloud infrastructures," in *Proc. IEEE Cloud Comput.*, Jun. 2010, pp. 329–336.

[7] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. IEEE ISCO*, Jan. 2016, pp. 1–8.

[8] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[9] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[10] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[11] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Commmu. Lett.*, vol. 3, no. 2, pp. 153–156, Apr. 2014.

[12] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

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

[14] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing," in *Proc. IEEE Globecom*, Dec. 2015, pp. 1–6.

[15] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. IEEE ISLPED*, Jul. 2012, pp. 279–284.

[16] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[17] X. Wang and G. B. Giannakis, "Power-efficient resource allocation for time-division multiple access over fading channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1225–1240, Mar. 2008.

[18] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, "Multiuser OFDM with adaptive subcarrier, bit, and power allocation," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1747–1758, Oct. 1999.

[19] S.-J. Oh, D. Zhang, and K. M. Wasserman, "Optimal resource allocation in multiservice CDMA networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 811–821, Jul. 2003.

[20] L. B. Le and E. Hossain, "Resource allocation for spectrum underlay in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5306–5315, Dec. 2008.

[21] Y. Choi, H. Kim, S.-W. Han, and Y. Han, "Joint resource allocation for parallel multi-radio access in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3324–3329, Nov. 2010.

[22] Y. Liu, C. Yuen, N. Ul Hassan, S. Huang, R. Yu, and S. Xie, "Electricity cost minimization for a microgrid with distributed energy resource under different information availability," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2571–2583, Apr. 2015.

[23] X. Wang, C. Yuen, X. Chen, N. Ul Hassan, and Y. Ouyang, "Cost-aware demand scheduling for delay tolerant applications," *J. Netw. Comput. Appl.*, vol. 53, pp. 173–182, Jul. 2015.

[24] B. Prabhakar, E. U. Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proc. IEEE INFOCOM*, vol. 1. Apr. 2001, pp. 386–394.

[25] S.-C. Wang, K.-Q. Yan, W.-P. Liao, and S.-S. Wang, "Towards a load balancing in a three-level cloud computing network," in *Proc. IEEE Int. Conf. Comput. Sci. Inf. Technol.*, vol. 1. Jul. 2010, pp. 108–113.

[26] M. Tao, Y.-C. Liang, and F. Zhang, "Resource allocation for delay differentiated traffic in multiuser OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2190–2201, Jun. 2008.

[27] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 226–234, Feb. 2009.

[28] D. Kivanc, G. Li, and H. Liu, "Computationally efficient bandwidth allocation and power control for OFDMA," *IEEE Trans. Wireless Commun.*, vol. 2, no. 6, pp. 1150–1158, Nov. 2003.

[29] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

[30] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, vol. 2. Philadelphia, PA, USA: SIAM, 2001.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Changsheng You (S'12) received the B.S. degree in electronic engineering and information science from the University of Science and Technology of China in 2014. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with The University of Hong Kong. His research interests include mobile-edge computing, fog computing, wireless power transfer, energy harvesting systems, and convex optimization.

Kaibin Huang (M'08–SM'13) received the B.Eng. (Hons.) and M.Eng. degrees from the National University of Singapore and the Ph.D. degree from The University of Texas at Austin (UT Austin), all in electrical engineering.

Since 2014, he has been an Assistant Professor with the Department of Electrical and Electronic Engineering (EEE), The University of Hong Kong. He is an Adjunct Professor with the School of EEE, Yonsei University, South Korea. He had been a Faculty Member with the Department of Applied Mathematics (AMA), Hong Kong Polytechnic University (PolyU), and with the Department of EEE, Yonsei University. He had been a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, Hong Kong University of Science and Technology, from June 2008 to February 2009, and an Associate Scientist with the Institute for Infocomm Research, Singapore, from November 1999 to July 2004. His research interests include the analysis and design of wireless networks using stochastic geometry and multi-antenna techniques.

Dr. Huang frequently serves on the technical program committees of major IEEE conferences in wireless communications. He has been the Technical Chair/Co-Chair for the IEEE CTW 2013, the Communication Theory Symposium of the IEEE GLOBECOM 2014, and the Advanced Topics in Wireless Communication Symposium of the IEEE/CIC ICCC 2014, and has been the Track Chair/Co-Chair of the IEEE PIMRC 2015, the IEE VTC Spring 2013, the Asilomar 2011, and the IEEE WCNC 2011. He is currently an Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Series on Green Communications and Networking, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE WIRELESS COMMUNICATIONS LETTERS. He was also a Guest Editor of the JSAC Special Issue on Communications Powered by Energy Harvesting and an Editor of the IEEE/KICS JOURNAL OF COMMUNICATION AND NETWORKS (2009–2015). He is an elected member of the SPCOM Technical Committee of the IEEE Signal Processing Society. He received the 2015 IEEE ComSoc Asia Pacific Outstanding Paper Award, the Outstanding Teaching Award from Yonsei, Motorola Partnerships in Research Grant, the University Continuing Fellowship from UT Austin, and the Best Paper Award from the IEEE GLOBECOM 2006 and PolyU AMA in 2013.

Hyukjin Chae received the B.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea. He joined LG Electronics, South Korea, as a Senior Research Engineer, in 2012. His research interests include interference channels, multiuser MIMO, D2D, V2X, and full duplex radio. Since 2012, he has contributed to and participated as a delegate in 3GPP RAN1 with interests in ePDCCH, eIMTA, FD MIMO, indoor positioning, D2D, and V2X communications. He is an inventor of over 100 patents.

Byoung-Hoon Kim (S'95–M'01) received the B.S. and M.E. degrees in electronics engineering, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 1994, 1996, and 2000, respectively. From 2000 to 2003, he was with GCT Semiconductor, Seoul, developing W-CDMA and WLAN chip sets. From 2003 to 2008, he was with QUALCOMM Inc., San Diego, CA, USA, where he was responsible for MIMO technology development and 3GPP LTE standard and design works. He was also involved in the IEEE 802.11 standard works and assumed the role of a member of the board of directors of the Wi-Fi Alliance from 2011 to 2012. Since 2008, he has been with LG Electronics as the Vice President and a Research Fellow, developing advanced wireless technologies, including 5G mobile communications and 3GPP LTE-Advanced/5G standards. He has co-authored *Scrambling Techniques for CDMA Communications* (Springer, 2001). His current research interest includes advanced channel coding, multiple access, V2X, massive MIMO, flexible and full duplex radio, and mmWave technologies. He was elected as the First IEEE Communications Society Asia-Pacific Best Young Researcher in 2001.