Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems

Feng Wang¹, Jie Xu¹*, Xin Wang², and Shuguang Cui³

¹School of Information Engineering, Guangdong University of Technology, Guangzhou, China
²Key Lab of EMW information (MoE), Fudan University, Shanghai, China
³Department of Electrical and Computer Engineering, University of California, Davis, USA

Email: fengwang.nl@gmail.com, jiexu.ustc@gmail.com, xwang11@fudan.edu.cn, robert.cui@gmail.com

Abstract—Integrating mobile-edge computing (MEC) and wireless power transfer (WPT) is a promising technique in the Internet of Things (IoT) era. It can provide massive low-power mobile devices with enhanced computation capability and sustainable energy supply. In this paper, we consider a wireless powered multiuser MEC system, where a multi-antenna access point (AP) (integrated with an MEC server) broadcasts wireless power to charge multiple users and each user node relies on the harvested energy to execute latency-sensitive computation tasks. With MEC, these users can execute their respective tasks locally by themselves or offload all or part of them to the AP following a time division multiple access (TDMA) protocol. Under this setup, we pursue an energy-efficient wireless powered MEC system design by jointly optimizing the transmit energy beamformer at the AP, the central processing unit (CPU) frequency and the offloaded bits at each user, as well as the time allocation among different users. In particular, we minimize the energy consumption at the AP over a particular time block subject to the computation latency and energy harvesting constraints per user. By formulating this problem into a convex framework and employing the Lagrange duality method, we obtain its optimal solution in a semi-closed form. Numerical results demonstrate the benefit of the proposed joint design over alternative benchmark schemes in terms of the achieved energy efficiency.

Index Terms—Mobile-edge computing, wireless power transfer, computation offloading, transmit energy beamforming, convex optimization.

I. INTRODUCTION

The recent advancement of Internet of Things (IoT) has motivated various new applications (such as autonomous driving, virtual reality, and tele-surgery) to provide real-time machine-to-machine and machine-to-human interactions [1]. These emerging applications critically rely on the real-time communication and computation of massive mobile devices (e.g., sensors). As extensive existing works focus on improving their communication performances [1], how to provide these devices with enhanced computation capability is a crucial but challenging task to be tackled, especially when they are of small size and low power. To resolve this issue, mobile-edge computing (MEC) has emerged as a promising technique by providing cloud computing at the edge of mobile networks via integrated MEC servers at wireless access points (APs) and base stations (BSs) [2]. Some practical examples of MEC efforts include the mobile assistance using infrastructure (MAUI) [3], ThinkAir [4], and the new MEC standardization group launched by the European telecommunications standards institute (ETSI) [5]. With MEC, resource-limited mobile devices can offload partial or all their computation tasks to APs; then MEC servers integrated there can compute these tasks remotely on their behalf. The MEC technique facilitates the real-time implementation of computation-extensive tasks at massive low-power devices, and thus has attracted growing research interests. For example, the authors in [2], [6], [7] investigated the energy-efficient MEC design for mobile devices to minimize their energy consumption subject to the computation requirements. In these works, each mobile device decides whether a task should be offloaded to the AP or executed locally by itself, or how many bits should be offloaded, so as to balance the energy consumption tradeoff between offloading and local computing.

On the other hand, how to provide sustainable and cost-effective energy supply to massive computation-heavy devices is another challenge facing IoT. Radio-frequency (RF) signal based wireless power transfer (WPT) provides a viable solution by deploying dedicated energy transmitters to broadcast energy wirelessly [8]. Recently, emerging wireless powered communication networks (WPCNs) and simultaneous wireless information and power transfer (SWIPT) paradigms have been proposed to achieve ubiquitous wireless communications in a self-sustainable way [9]–[12], where WPT and wireless communications are combined into a joint design. Motivated by these approaches, in this paper we pursue a joint design of WPT and MEC to facilitate self-sustainable computing for a large number of mobile devices by considering wireless powered multiuser MEC systems.

We consider a wireless powered multiuser MEC system consisting of a multi-antenna AP and multiple users. Each user is equipped with two antennas: one for WPT and the other for computation offloading. These two antennas operate over different frequency bands such that WPT and computation offloading can be performed simultaneously without co-channel interference. Suppose a block-based operation, where each user relies on its harvested wireless energy to execute the latency-sensitive computation tasks per time block via local computing or (partial) offloading to the MEC server. The computation task per user needs to be accomplished within the block. We assume the computation bits can be arbitrarily partitioned to be computed separately and each user can perform local computing and offloading at the same time. We also consider a time division multiple access (TDMA)
It is assumed that the WPT and the wireless communications (for offloading) are operated over orthogonal frequency bands simultaneously without co-channel interference. The AP is assumed to be equipped with $N$ antennas, each of which can transmit energy signals in the downlink and receive the offloading signals in the uplink simultaneously over different frequency bands. Each user is equipped with two antennas: one connecting to the energy harvesting circuit to harvest wireless energy from the AP and the other connecting to information transceivers to communicate with the AP (including offloading tasks and receiving computed results).

We consider a block-based model for both the WPT and the MEC. Let $T$ denote the length of each block. Each of the $K$ users relies on its harvested wireless energy in each block to execute the corresponding computation task. The latency-sensitive computation task at each user must be accomplished before the end of this block. It is assumed that the AP knows the perfect channel state information from/to the $K$ users, as well as the perfect computation information of all the $K$ users. Suppose that user $i$ has a computation task with $R_i$ input bits, which can either be computed locally by itself, or remotely computed at the MEC server via offloading. In practice, the computation task can be separated into various modules and computed in a distributed manner [2]–[4]. This enables partial offloading, such that $\ell_i$ bits of the task are offloaded to the MEC server remotely, while the remaining $(R_i - \ell_i)$ bits are computed locally. For simplicity of analysis, we consider that the $R_i$ bits at each user can be arbitrarily segmented as considered in the prior work [7]. Accordingly, $\ell_i$ can be any real number within $[0, R_i]$.

\section{System Model and Problem Formulation}

Consider a wireless powered multiuser MEC system in Fig. 1, where a multi-antenna AP (with an integrated MEC server) employs RF signal based transmit energy beamforming to charge $K$ users. Let $\mathcal{K} \triangleq \{1, \ldots, K\}$ denote the set of the users. The $K$ users utilize the harvested energy to accomplish their computation tasks by computing locally and offloading partial/all their respective tasks remotely to the MEC server.

\subsection{Transmit Energy Beamforming from AP to Users}

Let $s \in \mathbb{C}^{N \times 1}$ denote the energy-bearing signal transmitted by the AP and $Q = \mathbb{E}[ss^H] \in \mathbb{C}^{N \times N}$ denote its transmit covariance matrix, where $\mathbb{E}[\cdot]$ stands for the statistical expectation and the superscript $H$ stands for the conjugate transpose. In general, the AP can use multiple transmit beams to deliver the wireless energy, i.e., $Q$ can be of any rank. Supposing $d = \text{rank}(Q) \leq N$, there are a total of $d$ energy beams that can be obtained via the eigenvalue decomposition (EVD) of $Q$ [8]. In this case, the total transmit power at the AP is given as $\text{tr}(Q)$, where $\text{tr}(\cdot)$ denotes the matrix trace operation. Let $h_i \in \mathbb{C}^{N \times 1}$ denote the energy channel vector from the AP to user $i \in \mathcal{K}$, and we define $H_i \triangleq h_i h_i^H, \forall i \in \mathcal{K}$. Then the amount of energy harvested by user $i$ is

\begin{equation}
E_i = T\zeta \mathbb{E}\left[|h_i^H s|^2\right] = T\zeta \text{tr}(Q H_i), \quad \forall i \in \mathcal{K},
\end{equation}

where $0 < \zeta \leq 1$ is the energy conversion efficiency and $|\cdot|$ denotes the absolute value of a scalar. The harvested energy $E_i$ is used by user $i$ for both computation offloading and local computing.

\footnote{The simultaneous energy transmission and information reception over different frequency bands are implemented via a diplexer, similarly as for the conventional frequency-division-duplexing (FDD) information transceivers.}

\footnote{Each input bit can be treated as the smallest task unit, which includes the needed program codes and input parameters.}
B. Computation Offloading from Users to AP

We next consider the computation offloading from the $K$ users to the AP. In order for the MEC server to successfully compute the task on behalf of the users, each user should first offload the computation bits to the AP, and then the AP transmits the computation results back to the users after MEC computations. Practically, the AP with an integrated MEC server could provide sufficient CPU capability and high transmit power, while the computation results are usually of small sizes. Therefore, the computing time consumed at the MEC server and that consumed for delivering the computation results are relatively small. For each user, the energy required to receive its computation result from the AP is generally negligible. For these reasons, we only consider the uplink offloading time as the total MEC latency time and ignore users’ energy consumption for receiving computation results throughout the paper.

We consider a TDMA protocol for the uplink offloading. The whole time block is divided into a total of $K$ time slots, where user $i$ offloads its task in the $i$-th time slot with a length of $t_i \in [0, T]$. Let $g_i \in \mathbb{C}^{N \times 1}$ denote the uplink channel from user $i$ to the AP and $p_i$ the transmission power for offloading. Assume further that the AP employs the maximum ratio combining (MRC) receiver to decode the information. The achievable communication rate from user $i$ to the AP is given by

$$r_i = B \log_2 \left( 1 + \frac{p_i \tilde{g}_i}{\sigma^2} \right), \quad \forall i \in K,$$

where $B$ denotes the bandwidth for offloading, $\tilde{g}_i \triangleq \|g_i\|^2$, and $\sigma^2$ is the receiver noise power at the AP. $\|\cdot\|$ denotes the Euclidean norm of a vector. Since each user $i$ needs to offload a total number of $\ell_i$ bits over the time slot of length $t_i$, without loss of generality, we have

$$\ell_i = r_i t_i, \quad \forall i \in K.$$

For each user $i$, the power consumption consists of the transmit power $p_i$ (for offloading) and a constant circuit power $p_{c.i}$ (by the digital-to-analog converter (DAC), filter, and etc.). Here, the transmit power $p_i$ can be calculated as

$$p_i = \frac{1}{\tilde{g}_i} \beta \left( \frac{\ell_i}{t_i} \right), \quad \forall i \in K,$$

where $\beta(\cdot) \triangleq \sigma^2(2^x - 1)$ is a monotonically increasing and convex function of $x \geq 0$; we define $\beta(0) = 0$ when either $\ell_i = 0$ or $t_i = 0$ holds. By combining $p_i$ and $p_{c.i}$, the total energy consumption at user $i$ to offload $\ell_i$ bits over the $i$-th time slot is then

$$E_{\text{off},i} \triangleq (p_i + p_{c.i}) t_i = \frac{t_i}{\tilde{g}_i} \beta \left( \frac{\ell_i}{t_i} \right) + p_{c.i} t_i, \quad \forall i \in K. \tag{5}$$

Note that the computation offloading also incurs additional energy consumption at the AP, as it needs to receive the offloaded bits from the $K$ users, execute the computation sub-tasks (by the integrated MEC server), and send the computation results back to the users [2]. As the AP and its integrated MEC server are supposed to have sufficient communication and computation capacities, we assume that it consumes constant large receive/transmit power levels and a constant high CPU frequency to minimize the communication and computation time. We further assume the computation result transmission length is proportional to the computation bits. Therefore, we simply adopt a linear model for the offloading related energy consumption at the AP:

$$E_{\text{MEC}} = \alpha \sum_{i=1}^{K} \ell_i,$$

where $\alpha$ denotes the energy consumption per offloaded bit at the AP. In practice, $\alpha$ is generally determined by the transceiver structure of the AP, the chip structure of the MEC server, and its operated CPU frequencies [2].

C. Local Computing at $K$ Users

We now address the local computing for the remaining $(R_i - \ell_i)$ bits at each user $i \in K$. Denote by $C_i$ the number of CPU cycles required for computing one input bit at user $i$. Let $f_{i,n} \in (0, f_{\text{max}}]$ be the CPU frequency for the $n$-th CPU cycle required for user $i$; then $f'_{i,n}$ denotes the CPU frequency. Then the total number of CPU cycles for local computing at each user $i$ is $C_i (R_i - \ell_i)$ and the corresponding delay is $\sum_{i=1}^{K} C_i (R_i - \ell_i) / f_{i,n}$. Since all the local computing should be accomplished before the end of each given time block, the computation latency for executing these $C_i (R_i - \ell_i)$ CPU cycles by user $i$ should satisfy

$$\sum_{n=1}^{\ell_i} \frac{1}{f_{i,n}} \leq T, \quad \forall i \in K. \tag{7}$$

Under the assumption of a low CPU voltage that normally holds for low-power devices, the consumed energy for local computing is expressed as [14]

$$E_{\text{loc},i} = \sum_{n=1}^{\ell_i} \kappa_i f_{i,n}^2, \quad \forall i \in K, \tag{8}$$

where $\kappa_i$ is the effective capacitance coefficient that depends on the chip architecture at user $i$.

By combining the computation offloading energy in (5) and the local computation energy in (8), the total energy consumed by user $i$ within the block is given as $E_{\text{off},i} + E_{\text{loc},i}$. Note that user $i$ is powered by the wireless charging from the AP. In order for each user $i \in K$ to achieve self-sustainable operation, the total consumed energy $E_{\text{loc},i} + E_{\text{off},i}$ cannot exceed its harvested energy $E_{i}$ as in (1) per block. Therefore, we have

$$E_{\text{loc},i} + E_{\text{off},i} \leq E_{i}, \quad \forall i \in K. \tag{9}$$

Note that for the energy constraints in (9) to be sufficient for sustainable operation, we assume that the initial energy stored at the battery of each user is sufficiently large, such that the energy at the user will never be used up at any time within each
block and the energy storage level will be refilled by the end of each block. Therefore, the “energy causality” constraints (see, e.g., [11]) could be satisfied automatically.

D. Problem Formulation

Targeting an energy-efficient design, our objective is to minimize the energy consumption at the AP while ensuring the successful execution of the $K$ users’ computation tasks per time block. To this end, we jointly optimize the energy transmit covariance matrix $Q$ at the AP, the local CPU frequencies $\{f_{i,1}, \ldots, f_{i,C_i(R_i - \ell_i)}\}$ and the number $\ell_i$ of the offloaded bits at each user, as well as the time allocation $t_i$ among different users. Let $t \triangleq [t_1, \ldots, t_K]$, $\ell \triangleq [\ell_1, \ldots, \ell_K]$, and $f \triangleq [f_{1,1}, \ldots, f_{K,C_K(R_K - \ell_K)}]$, where the superscript $\dagger$ denotes the transpose operation. Mathematically, the joint computing and offloading problem is formulated as:

\[
\begin{align}
(P_1) : \quad & \min_{Q \succeq 0, t, \ell, f} \quad T \text{tr}(Q) + \sum_{i=1}^{K} \alpha \ell_i \\
& \quad \text{s.t.} \quad \sum_{n=1}^{i} \frac{1}{f_{i,n}} \leq T \quad \text{and} \quad 0 \leq f_{i,n} \leq f_{i,\text{max}}, \ \forall n, \ \forall i \in \mathcal{K}, \quad (10a) \\
& \quad \quad \quad \quad C_i(R_i - \ell_i) \leq T \quad \text{and} \quad 0 \leq f_{i,n} \leq f_{i,\text{max}}, \ \forall n, \ \forall i \in \mathcal{K}, \quad (10b) \\
& \quad \quad \quad \quad \sum_{n=1}^{i} \kappa_i f_{i,n} + \frac{t_i}{\gamma_i} \sum_{i=1}^{K} \beta(\ell_i) + p_c t_i - T \zeta \text{tr}(QH_i) \leq 0, \quad \forall i \in \mathcal{K}, \quad (10c) \\
& \quad \quad \quad \quad 0 \leq \ell_i \leq R_i, \ \forall i \in \mathcal{K}, \quad (10d) \\
& \quad \quad \quad \quad \sum_{i=1}^{K} t_i \leq T, \quad t_i \geq 0, \ \forall i \in \mathcal{K}. \quad (10e)
\end{align}
\]

The two sets of the constraints in (10b) represent the local computation latency and CPU frequency constraints at user $i$, respectively. The $i$-th constraint of (10c) represents the energy harvesting constraint for user $i \in \mathcal{K}$. Furthermore, (10d) and (10e) correspond to the constraints about the users’ offloading bits and their TDMA offloading time allocation, respectively. Suppose that each user has a sufficient computing capacity, i.e., $f_{i,\text{max}} \geq C_i R_i / T$, $\forall i \in \mathcal{K}$. Then problem (P1) is always feasible since one can always scale $Q$ to guarantee that the energy harvesting constraints in (10c) hold for any $(t, \ell, f)$. Note that problem (P1) is non-convex due to the non-convexity of the constraints in (10b) and (10c).

III. Optimal Solution

In this section, we provide the optimal solution to problem (P1). To cope with the non-convex constraints (10b) and (10c), we first establish the following lemma.

Lemma 1: Given the offloaded bits $\ell$, the optimal solution of the CPU frequencies $f$ to (P1) satisfies

\[
f_{i,1} = \ldots = f_{i,C_i(R_i - \ell_i)} = C_i(R_i - \ell_i) / T, \quad \forall i \in \mathcal{K}. \quad (11)
\]

Proof: See Appendix A.

Lemma 1 indicates that at each user $i \in \mathcal{K}$, the local CPU frequencies for different CPU cycles are identical in the optimal strategy. Building on (11), problem (P1) can be equivalently reformulated as:

\[
(P_{1.1}) : \quad \min_{Q \succeq 0, t, \ell} \quad T \text{tr}(Q) + \sum_{i=1}^{K} \alpha \ell_i \quad (12a)
\]

\[
\text{s.t.} \quad \sum_{i=1}^{K} t_i \leq T, \quad (12b) \\
\quad \quad \frac{\kappa_i C_i^3 (R_i - \ell_i)^3}{T^2} + \frac{t_i}{\gamma_i} \beta(\ell_i) + p_c t_i - T \zeta \text{tr}(QH_i) \leq 0, \quad \forall i \in \mathcal{K}, \quad (12c)
\]

\[
0 \leq \ell_i \leq R_i, \quad \forall i \in \mathcal{K}, \quad (12d) \\
\quad t_i \geq 0, \quad \forall i \in \mathcal{K}. \quad (12e)
\]

As $\beta(x)$ is a convex function of $x \geq 0$, it can be shown that its perspective $\frac{1}{\gamma_i} \beta(\frac{\ell_i}{t_i})$ is a joint convex function of $t_i$ and $\ell_i$ [16]. As a result, the energy harvesting constraints in (12c) become convex. Furthermore, since the objective function in (12a) is affine and the other constraints are all convex, problem (P_{1.1}) is a convex problem which can be then efficiently solved by standard convex optimization techniques such as the interior-point method [16]. To gain more insights, we next derive its semi-closed solution by leveraging the Lagrange duality method [16].

Let $\mu \geq 0$ and $\lambda_i \geq 0$ denote the dual variables associated with the time-allocation constraint in (12b) and the $i$-th energy harvesting constraint in (12c), respectively. Then the partial Lagrangian of (P_{1.1}) is expressed as

\[
\mathcal{L}(Q, t, \ell, \lambda, \mu) = T \text{tr}(Q) + \sum_{i=1}^{K} \alpha \ell_i \\
+ \sum_{i=1}^{K} \lambda_i \left( \frac{\kappa_i C_i^3 (R_i - \ell_i)^3}{T^2} + \frac{t_i}{\gamma_i} \beta(\ell_i) + p_c t_i - T \zeta \text{tr}(QH_i) \right) + \mu \left( \sum_{i=1}^{K} t_i - T \right), \quad (13)
\]

where $\lambda \triangleq [\lambda_1, \ldots, \lambda_K]$. The dual function $g(\lambda, \mu)$ is then

\[
g(\lambda, \mu) = \min_{Q \succeq 0, t, \ell} \mathcal{L}(Q, t, \ell, \lambda, \mu) \quad (14)
\]

\[
\text{s.t.} \quad 0 \leq \ell_i \leq R_i, \quad t_i \geq 0, \quad \forall i \in \mathcal{K}. \quad (14)
\]

Consequently, the dual problem of (P_{1.1}) is

\[
(D_{1.1}) : \quad \max_{\lambda, \mu} \quad g(\lambda, \mu) \quad (15a)
\]

\[
\text{s.t.} \quad F(\lambda, \mu) \triangleq I_N - \sum_{i=1}^{K} \lambda_i H_i \succeq 0, \quad (15b)
\]

\[
\lambda_i \geq 0, \quad \forall i \in \mathcal{K}, \quad (15c)
\]

\[
\mu \geq 0, \quad (15d)
\]

where $I_N$ denotes the $N \times N$ identity matrix. Note that $F(\lambda) \succeq 0$ is needed to ensure that the dual function is bounded below (as proved in Appendix B). We denote the feasible set of $(\lambda, \mu)$, characterized by (15b)–(15d), as $\mathcal{X}$.

Since problem (P_{1.1}) is convex and satisfies the Slater’s condition, strong duality holds between (P_{1.1}) and (D_{1.1})
Lemma 2: For a given \( \lambda, \mu \) and \( g \) solution to (14) under given \( \lambda \) and \( \mu \), while \((Q_{opt}, \mu_{opt})\) denotes the primary solution to (P1.1) (or equivalently, (P1)) and \((\Lambda_{opt}, \mu_{opt})\) denotes the optimal dual solution to problem (D1.1).

A. Evaluating the Dual Function \( g(\lambda, \mu) \)

For the given \( \lambda \) and \( \mu \), problem (14) can be decomposed into \( K+1 \) subproblems, one for optimizing \( Q \) and the other \( K \) for optimizing \( t_i \)'s and \( \ell_i \)'s. In particular, we have

\[
\min_Q \text{ tr } (QF(\lambda)) \quad \text{s.t.} \quad Q \succeq 0, \tag{16}
\]

and

\[
\min_{t_i, \ell_i} \; \alpha \ell_i + \frac{\lambda_i K_i C_i^T(R_i - \ell_i)^3}{T^2} + \frac{\lambda_i t_i}{g_i} \beta \left( \frac{\ell_i}{t_i} \right) + \lambda_i p_i c_i t_i + \mu_i t_i \\
\text{s.t.} \; 0 \leq \ell_i \leq R_i, \; t_i \geq 0, \tag{17}
\]

where the \( i \)-th subproblem in (17) is for user \( i \). Under the condition of \( F(\lambda) \succeq 0 \), it is evident that the optimal value of (16) is zero and the optimal solution \( Q^* \) to (16) can be any positive semi-definite matrix in the null space of \( F(\lambda) \). Here, we simply set \( Q^* = 0 \) to obtain the dual function \( g(\lambda, \mu) \), which is only for the purpose of computing the optimal dual solution (see Section III-B). Note that \( Q^* = 0 \) is not a unique solution to (16) when \( F(\lambda) \) is rank-deficient, i.e., rank\( (F(\lambda)) < N \), and it is not the primary solution to \( (P1.1) \) since it violates the energy harvesting constraints in (12c).

We will show how to retrieve the primary solution of \( Q_{opt} \) to problem (P1.1) later in Section III-C.

For the \( i \)-th subproblem in (17), it is convex and satisfies the Slater’s condition. Based on the Karush-Kuhn-Tucker (KKT) conditions [16], one can obtain a semi-closed solution \((\ell_i^*, \ell_i^*)\) to (17), as stated formally in the next lemma.

**Lemma 2:** For a given \( \lambda_i \geq 0 \) and \( \mu_i \geq 0 \), the optimal solution to (17) can be obtained as follows.

- If \( \lambda_i = 0 \), we have \( \ell_i^* = 0 \) and \( t_i^* = 0 \);
- If \( \lambda_i > 0 \), we have

\[
\begin{align*}
\ell_i^* & = \left[ R_i - \frac{T^2}{3 \lambda_i C_i} \left( \frac{1}{G_i} + \frac{\pi^2 \ln 2}{B G_i^2} \right) \right]^{+}, \\
\ell_i^* & = \ell_i^* / r_i,
\end{align*}
\]

where \( r_i = \frac{1}{\int_0^{1/2} \left( W_0 \left( \frac{1}{2 e x} \left( \frac{e x}{\mu_i} + p_i c_i, i \right) - \frac{1}{\mu_i} \right) + 1 \right) \) denotes the offloading rate of user \( i \), \( W_0(x) \) is the principal branch of the Lambert \( W \) function defined as the solution for \( W_0(x) e^{W_0(x)} = x \) [15], \( e \) is the base of the natural logarithm, and \( |x|^{+} = \max(x, 0) \).

**Proof:** See Appendix C.

By combining Lemma 2 and \( Q^* = 0 \), the dual function \( g(\lambda, \mu) \) can be evaluated for any \( (\lambda, \mu) \).

B. Obtaining the Optimal \( \lambda_{opt} \) and \( \mu_{opt} \) to Maximize \( g(\lambda, \mu) \)

Note that the dual function \( g(\lambda, \mu) \) is concave but non-differentiable in \( \lambda \) and \( \mu \). As a result, we use subgradient based methods, such as the ellipsoid method [17], to obtain the optimal \( \lambda_{opt} \) and \( \mu_{opt} \) for (D1.1). To start, we generate a sequence of ellipsoids of decreasing volumes, each of which is guaranteed to contain the optimal \( \lambda_{opt} \) and \( \mu_{opt} \), until the objective value of (D1.1) converges. Per iteration, if \( \lambda \) and \( \mu \) are feasible to \( (D1.1) \), then we form a new ellipsoid using the subgradient of the objective function; otherwise, we form a new one using the subgradient of the violated constraint function. Therefore, it remains to determine the subgradients of both the objective function in (15a) and the constraints in (15b)-(15d). Clearly, with \( (t^*, \ell^*) \), one subgradient for the objective function \( g(\lambda, \mu) \) in (15a) is given by

\[
\kappa_1 C_i^T (R_i - \ell_i^*)^3 + t_i^* \frac{\beta}{g_i} \left( \frac{\ell_i^*}{t_i^*} \right) + p_i c_i t_i + \mu_i t_i, \tag{18}
\]

where the first \( K \) components and the last one of (19) correspond to the first-derivative of \( g(\lambda, \mu) \) with respect to \( \lambda_i \) and \( \mu_i \), respectively. For the positive semidefinite constraint \( F(\lambda) \succeq 0 \), we establish the following lemma.

**Lemma 3:** Let \( v \in \mathbb{C}^{N \times 1} \) be the eigenvector corresponding to the smallest eigenvalue of \( F(\lambda) \), i.e., \( v = \arg\min_{\|\xi\| = 1} \xi^H F(\lambda) \xi \). Then the constraint \( F(\lambda) \succeq 0 \) is equivalent to the constraint of \( v^H F(\lambda) v \geq 0 \). In this case, the subgradient of \( v^H F(\lambda) v \) at the given \( \lambda \) and \( \mu \) is

\[
\begin{bmatrix}
\zeta_i v^H H_1 v, \ldots, \zeta_i v^H H_K v, 0^{T}
\end{bmatrix} \tag{19}
\]

**Proof:** See Appendix D.

Furthermore, the subgradient of the \( i \)-th constraint in (15c) is given by the elementary vector \( e_i \in \mathbb{R}^{(K+1) \times 1} \) (i.e., \( e_i \) is of all zero entries except for the \( i \)-th entry being one), while that of (15d) is \( e_{K+1} \). With (19), (20), and the subgradients of (15c)-(15d), we can then apply the ellipsoid method to efficiently update \( \lambda \) and \( \mu \) towards the optimal \( \lambda_{opt} \) and \( \mu_{opt} \) for (D1.1).

C. Finding the Optimal Primary Solution to \( (P1) \)

With \( \lambda_{opt} \) and \( \mu_{opt} \) obtained, it remains to determine the optimal primary solution to \( (P1.1) \) (or equivalently to \( P1 \)). Specifically, by replacing \( \lambda \) and \( \mu \) with \( \lambda_{opt} \) and \( \mu_{opt} \) in Lemma 2, respectively, one can obtain the optimal \((t^*, \ell^*)\) for \( (P1) \) in a semi-closed form. Based on \((t^*, \ell^*)\), one can then obtain the optimal local CPU frequencies \( \{f_{i,n}^{opt}\} \) per user and the optimal transmit covariance matrix \( Q_{opt} \) of the AP for \( (P1) \). We can then readily establish the following proposition.

**Proposition 1:** The optimal solution \((\{f_{i,n}^{opt}\}, Q_{opt}, t^*, \ell^*)\) for problem \((P1)\) is given by

\[
l_{i,n}^{opt} = \begin{cases} R_i - \sqrt{\frac{T^2}{3e_i C_i} \left( \frac{1}{G_i} + \frac{\pi^2 \ln 2}{B G_i^2} \right)}, \quad \lambda_i^{opt} > 0, \\
0, \quad \lambda_i^{opt} = 0, \tag{20}
\end{cases}
\]

\[
\]
\begin{align}
    \ell_i^\text{opt} &= \left\{ \begin{array}{ll}
    \ell_i^\text{opt} / j_i^\text{opt}, & \lambda_i^\text{opt} > 0, \\
    0, & \lambda_i^\text{opt} = 0,
    \end{array} \right. \\
    j_i^\text{opt} = \ldots = j_i, C_i(R_i - \ell_i^\text{opt}) &= C_i(R_i - \ell_i^\text{opt}) / T, \quad \forall i \in K, \tag{23}
\end{align}

and

\begin{align}
    Q^\text{opt} &= \arg\min_{Q \succeq 0} \quad T \text{tr}(Q) \\
    \text{s.t.} & \quad \frac{k_i C_i^3 (R_i - \ell_i^\text{opt})^3}{T^2} + \frac{\ell_i^\text{opt}}{\tilde{g}_i} \beta \left( \frac{\ell_i^\text{opt}}{\tilde{g}_i} \right) + p_c, t_i^\text{opt} \\
    & \quad - T \text{tr}(Q H_i) \leq 0, \quad \forall i \in K, \tag{24}
\end{align}

where

\begin{align}
    r_i^\text{opt} &\triangleq \frac{B}{\ln 2} \left( \frac{W_0 \left( \frac{\tilde{g}_i}{\sigma T i} \left( \frac{\mu_i^\text{opt}}{\lambda_i^\text{opt}} + p_{c,i} \right) - \frac{1}{e} \right) + 1}{1} \right) \
\end{align}

corresponds to the offloading rate for user \( i \), \( \forall i \in K \).

Proposition 1 can be verified by simply combining Lemmas 1 and 2; hence, we omit its detailed proof for conciseness. Note that (24) is an instance of semidefinite program (SDP), which can thus be efficiently solved by off-the-shelf solvers, e.g., CVX [18]. Summarizing, we now present Algorithm 1 to obtain the optimal solution \( \{ f_i, Q^\text{opt}, \ell^\text{opt}, E^\text{opt} \} \) to the joint offloading and computing problem \( (P1) \).

\begin{algorithm}
1: **Initialization:** Given an ellipsoid \( E((\lambda, \mu), A) \) containing \( (\lambda^\text{opt}, \mu^\text{opt}) \), where \( (\lambda, \mu) \) is the center point of \( E \) and \( A > 0 \) characterizes the size of \( E \).
2: **Repeat**
3: \quad **For** each user \( i \in K \)
4: \quad \quad Obtain \( (t_i^\text{opt}, \ell_i^\text{opt}) \) with \( \lambda_i \) and \( \mu \) according to Lemma 2.
5: \quad **End For**
6: \quad Compute the subgradient of \( g(\lambda, \mu) \) as in (19) and that of the constraints in (15b) by (20), then update \( \lambda \) and \( \mu \) using the ellipsoid method [17].
7: **Until** \( \lambda \) and \( \mu \) converge within a prescribed accuracy.
8: \quad Set \( (\lambda^\text{opt}, \ell^\text{opt}) \leftarrow (\lambda, \mu) \).
9: **Output:** Obtain \( (\ell^\text{opt}, E^\text{opt}) \) by (21) and (22), \( \{ f_i^\text{opt} \} \) by (23), and \( Q^\text{opt} \) by (24).

Based on the optimal solution to \( (P1) \) obtained above, we have the following remark.

**Remark 1:** Proposition 1 provides helpful insights about the optimal joint computing and offloading design.

- First, if \( \lambda_i^\text{opt} = 0 \), we have \( \ell_i^\text{opt} = 0 \) as shown in (21). Due to the complementary slackness condition [16], i.e.,

\begin{align}
    \lambda_i^\text{opt} \left( \frac{k_i C_i^3 (R_i - \ell_i^\text{opt})^3}{T^2} + \frac{\ell_i^\text{opt}}{\tilde{g}_i} \beta \left( \frac{\ell_i^\text{opt}}{\tilde{g}_i} \right) + p_{c,i} t_i^\text{opt} \\
    - T \text{tr}(Q^\text{opt} H_i) \right) &= 0, \quad \forall i \in K, \tag{26}
\end{align}

it implies that if the energy harvesting constraint is not tight for user \( i \) (i.e., user \( i \) harvests sufficient wireless energy), no computation offloading is required and user \( i \) computes all the tasks locally. This is intuitive: when the user has sufficient energy to accomplish the tasks locally, there is no need to employ computation offloading that can incur additional energy consumption at the AP.

- Next, if \( \lambda_i^\text{opt} > 0 \), we have \( 0 \leq \ell_i^\text{opt} < R_i \) (see (21)), which is due to the non-zero energy related to \( \alpha \) at the MEC server, together with the fact that \( r_i^\text{opt} > 0 \) (see (25)). In other words, offloading all the bits to the AP is always suboptimal in this case. This is because the marginal energy consumption of local computing is nearly zero when \( \ell_i^\text{opt} \rightarrow R_i \). It is then always beneficial to leave some bits for local computing. Interestingly, (21) further show that a larger \( \lambda_i^\text{opt} \) admits a larger \( \ell_i^\text{opt} \) and a smaller \( r_i^\text{opt} \), leading to a larger \( \ell_i^\text{opt} \) allocated for user \( i \). This means that more stringent the energy harvesting constraint is, more bits should be offloaded.

- Finally, the offloading bit number \( \ell_i \) and the offloading rate \( r_i \) for user \( i \) are affected by the channel gain \( \tilde{g}_i \), the block length \( T \), the circuit power \( p_{c,i} \), and the MEC energy consumption \( \alpha \) per offloaded bit in the following way: 1) when the channel condition becomes better (i.e., \( \tilde{g}_i \) becomes larger), user \( i \) is more likely to offload more bits with a higher offloading rate; 2) a higher circuit power \( p_{c,i} \) at the user leads to a higher offloading rate; 3) when \( T \) or \( \alpha \) increases, fewer bits are offloaded over the uplink. In addition, a smaller \( r_i \) leads to more offloading bits, such that a longer time duration is allocated for offloading.

\section{IV. Numerical Results}

In this section, we provide numerical results to gauge the performance of the proposed optimal joint offloading and computing design. For comparison, we consider the following three baseline schemes:

1. **Local computing only:** In this approach, each user \( i \in K \) accomplishes its computation task by only local computing, i.e., by setting \( \ell_i = 0 \), \( \forall i \in K \). The CPU frequencies at each user \( i \) are obtained as \( f_i = C_i R_i / T_i \); the transmit energycovariance matrix \( Q^\text{opt} \) is obtained by solving problem (24) with \( \ell_i^\text{opt} = 0 \) and \( r_i^\text{opt} = 0 \), \( \forall i \in K \).

2. **Computation offloading only:** In this approach, each user \( i \in K \) accomplishes its computation task by fully offloading to the AP, i.e., by setting \( \ell_i = R_i \), \( \forall i \in K \). The CPU frequencies at all the users are set as zero; the allocated time slot \( t_i \) and the transmit covariance matrix \( Q^\text{opt} \) are determined by solving problem \( (P1.1) \) with \( \ell_i^\text{opt} = R_i \), \( \forall i \in K \).

3. **Joint computing and offloading design with isotropic WPT:** In this approach, we set the transmit covariance matrix \( Q = p I_N \), where \( p \) denotes the transmit power to be optimized at each antenna. The CPU frequencies \( \{ f_i,n \} \) at each user, the offloading bit number \( \ell_i \), and the allocated time slot \( t_i \), \( \forall i \in K \), are obtained by solving problem \( (P1.1) \) with \( Q^\text{opt} = p I_N \).

In the simulations, we consider a \( K \)-user MEC system, where the AP is equipped with \( N = 4 \) antennas. The energy conversion efficiency \( \xi \) is 0.8. All channels are modeled as independent Rayleigh fading with an average power loss of \( 5 \times 10^{-6} \) (i.e., \(-53 \text{ dB}\)) which corresponds to a distance of about 5 meters from users to the AP in urban environment. We set the same computation task of \( R = R_i \) input bits for
Computation task size for each user, $R$ (10 kbits)

Average energy consumption at the AP (Joule)

Fig. 2. The average energy consumption at the AP versus the computation task size $R$ at each user.

all users, $C_i = 10^3$ cycles/bit, and $\kappa_i = 10^{-28}$ [7]. The circuit power is $p_{c,i} = 10^{-4}$ W and the energy per bit consumed by the MEC server is $\alpha = 10^{-5}$ Joule/bit. In addition, we set the receiver noise power as $\sigma^2 = 10^{-9}$ W and the bandwidth as $B = 2$ MHz. We average the energy consumption over 500 Monte-Carlo runs.

Fig. 2 shows the average energy consumption at the AP versus the computation task of $R$ at each user for different schemes with $K = 4$ and $T = 0.1$ s. It is observed that the proposed scheme outperforms the three baseline schemes. The average energy consumption increases for all schemes as $R$ increases, especially for the local computing scheme. For the task size $R$ as shown in Fig. 2, the full offloading scheme outperforms the local computing one especially when $R$ becomes large. Therefore, for large $R$ values, it is desirable to offload a large proportion of computation tasks to the AP to improve the system energy efficiency. As expected, the baseline full offloading scheme achieves a near-optimal performance as with the proposed one for large $R$. Due to the loss of multi-antenna array gain, the isotropic WPT scheme is strictly sub-optimal.

Fig. 3 depicts the average energy consumption versus the user number $K$, where $R = 10^4$ bits and $T = 0.5$ s. It is seen that the proposed optimal joint scheme enjoys substantial performance gains over all the three baseline schemes. As $K$ increases, these performance gains become more significant. This suggests that joint offloading and computing becomes more beneficial in the large user cases to facilitate an energy-efficient design.

By fixing $R = 10^4$ bits and $K = 4$, Fig. 4 shows the average energy consumption at the AP versus the time block length $T$. Interestingly, for small $T$ (e.g., $T = 0.05$ s), the full offloading scheme consumes almost the same energy as the proposed one. However, when $T$ increases, the energy consumption for the full offloading scheme remains almost unchanged. This suggests that there exists some critical value for the time block length to enable energy saving for full offloading. By contrast, the energy consumption of the local computing only scheme is observed to decrease significantly as $T$ increases. This is because as $T$ increases, we could lower down the CPU frequency such that the power consumption for local computing becomes smaller than that for offloading at each user. For a large $T$ (e.g., $T \geq 0.2$ s), the local computing scheme outperforms the full offloading scheme, and even approaches the optimal performance with $T \geq 0.4$ s.

V. CONCLUSION

In this paper, we investigated the joint offloading and computing design in emerging wireless powered multiuser MEC systems. We developed a new energy-efficient design principle for the AP to minimize its energy consumption while maintaining the self-sustainable computation at mobile devices, by jointly optimizing the transmit energy beamformer at the AP, the local CPU frequency and the offloaded bits for each user, as well as the TDMA time allocation among different users. Leveraging the Lagrange duality method, we
obtained the optimal solution in a semi-closed form. Numerical results demonstrated the benefit of the proposed design over alternative baseline schemes in terms of the achieved energy efficiency.

APPENDIX

A. Proof of Lemma 1

First, consider the case when there exists some user \( i \) with \( \ell_i = R_i \), which implies that user \( i \) offloads all of its computation task bits to the AP. Clearly, user \( i \) does not perform local computing; hence, the local CPU frequency of user \( i \) is zero.

We next consider the nontrivial case of \( 0 \leq \ell_i < R_i, \forall i \in K \). Define \( f_i \triangleq \sum_{n=1}^{C_i(R_i-\ell_i)} f_{i,n} \), \( \forall i \in K \). Since that both \( 1/x \) and \( x^2 \) are convex functions with respect to \( x > 0 \), based on Jensen’s inequality [16], it follows that

\[
\begin{align*}
C_i(R_i - \ell_i)/f_i &\leq \sum_{n=1}^{C_i(R_i-\ell_i)} 1/f_{i,n}, \\
C_i(R_i - \ell_i) \kappa_i f_i^2 &\leq \sum_{n=1}^{C_i(R_i-\ell_i)} \kappa_i f_{i,n}^2,
\end{align*}
\]

where both equalities in (27) hold if and only if

\[ f_{i,1} = \ldots = f_{i,C_i(R_i-\ell_i)}, \forall i \in K. \]  (28)

As a result, the optimality of problem (P1) is achieved when (28) holds. Therefore, by replacing \( f_i \triangleq f_{i,n}, \forall n \), problem (P1) is equivalently expressed as

\[
\begin{align*}
\min_{Q \succeq 0, t \geq (f_{i,n})} & \quad T r(Q) + \sum_{i=1}^{K} \alpha \ell_i \\
\text{s.t.} & \quad C_i(R_i - \ell_i)/f_i \leq T, \quad \forall i \in K, \\
& \quad \kappa C_i(R_i - \ell_i) f_i^2 + \frac{1}{g_i} \beta \left( \frac{\ell_i}{t_i} \right) p_{c,i} t_i - T \gamma t_i \leq 0, \\
& \quad 0 \leq \ell_i \leq R_i, \quad \forall i \in K, \\
& \quad \sum_{i=1}^{K} t_i \leq T, \quad t_i \geq 0, \quad \forall i \in K.
\end{align*}
\]

(29a) (29b) (29c) (29d) (29e)

For a given \((t, \ell)\), it is evident that the optimal \( f_i \)'s for (29) (equivalent (P1)) should be as small as possible by (29c). Since \( f_i \) is bounded below by \( C_i(R_i - \ell_i)/T \) in (29b), it follows that the optimal \( f_i \)'s are

\[ f_i = C_i(R_i - \ell_i)/T, \quad \forall i \in K. \]  (30)

It readily follows that, at optimum of (P1),

\[ f_{i,1} = \ldots = f_{i,C_i(R_i-\ell_i)} = C_i(R_i - \ell_i)/T, \quad \forall i \in K. \]  (31)

B. Proof of \( F(\lambda) \geq 0 \)

It can be verified by contradiction. Assume that \( F(\lambda) \) is not positive semidefinite. Denote by \( \xi \) one eigenvector corresponding to the negative eigenvalue of \( F(\lambda) \). By setting \( Q = \tau \xi \xi^H \succeq 0 \) with \( \tau \) going to infinity (which is feasible for (14)), it follows that

\[ \lim_{\tau \to +\infty} T r(QF(\lambda)) = \lim_{\tau \to +\infty} \tau \xi^H F(\lambda) \xi = -\infty, \]  (32)

which in turn implies that the objective value in (14) is unbounded below over \( Q \succeq 0 \). Therefore, in order for the dual function value \( g(\mu, \lambda) \) to be bounded below, we need \( F(\lambda) \succeq 0 \).

C. Proof of Lemma 2

Given \( \lambda \) and \( \mu \), we solve subproblem (17) for each \( \lambda_i, i \in K \). When \( \lambda_i = 0 \), the objective function in (17) becomes

\[ \alpha \ell_i + \mu \ell_i, \]  (33)

It is evident that \( t_i^* = 0 \) and \( \ell_i^* = 0 \) are optimal for (17).

For \( \lambda_i > 0 \), we consider the Lagrangian of (17), denoted by \( L_i \), which is given by

\[
L_i = \alpha \ell_i + \kappa_i C_i^3(R_i - \ell_i)^3/2 + \lambda_i \gamma_i \left( \ell_i - \ell_i - \eta_i - \eta_i t_i \right), \]

where \( \gamma_i, \nu_i, \) and \( \eta_i \) are the non-negative Lagrangian multipliers associated with \( \ell_i \leq R_i, \ell_i \geq 0, \) and \( t_i \geq 0, \) respectively.

Based on the KKT conditions [16], the necessary and sufficient conditions for the optimal primal-dual point \((\ell_i^*, \ell_i, \gamma_i^*, \nu_i^*, \eta_i^*)\) are as follows.

\[
\begin{align*}
t_i^* &\geq 0, \quad 0 \leq \ell_i^* \leq R_i, \\
\gamma_i^* &\geq 0, \quad \nu_i^* \geq 0, \quad \eta_i^* \geq 0, \\
\gamma_i^*(\ell_i^* - R_i) &\geq 0, \quad \nu_i^* \ell_i^* = 0, \quad \eta_i^* t_i^* = 0, \\
\frac{\lambda_i}{g_i} \left( \frac{\ell_i^*}{t_i^*} \right)^2 - \frac{\nu_i^*}{t_i^*} - \frac{\eta_i^*}{t_i^*} = 0, \\
\alpha - 3\lambda_i \kappa_i C_i^3(R_i - \ell_i)^3/2 &+ \frac{\lambda_i}{g_i} \beta \left( \frac{\ell_i^*}{t_i^*} \right) + \gamma_i^* - \nu_i^* = 0.
\end{align*}
\]

where \( \beta'(x) \triangleq \frac{\sigma^2 \ln 2}{2 \pi} \) is the first-order derivative of \( \beta(x) \) with respect to \( x \). Note that (35c) denotes the complementary slackness condition, while the left-hand side terms of (35d) and (35e) are the first-order derivatives of \( L_i \) with respect to \( \ell_i^* \) and \( t_i^* \), respectively. For the function \( y = \beta(x) - x \beta'(x) \) of \( x > 0 \), its inverse function can be shown to be [15]

\[ x = \frac{B}{\ln 2} \left( W_0 \left( -\frac{y}{\sigma^2 e} - \frac{1}{e} \right) + 1 \right). \]  (36)

Let \( r_i^* \triangleq \ell_i^*/t_i^* \). From (35b) and (35d), we have

\[ \beta(r_i^*) - r_i^* \beta'(r_i^*) = -\frac{\lambda_i}{g_i} + p_{c,i} \]  (37)

Based on (36), it then follows that

\[ r_i^* = \frac{B}{\ln 2} \left( W_0 \left( \frac{\mu}{\sigma^2 e} + p_{c,i} - \frac{1}{e} \right) + 1 \right). \]  (38)

Since \( W_0(x) \) is a monotonically increasing function of \( x \geq -1/e \) and \( W_0(-1/e) = -1 \), it follows that \( r_i^* > 0 \) with non-zero \( p_{c,i} \). From (35c) and (35e), we readily have

\[ \ell_i^* = \left[ R_i - \frac{1}{3\kappa_i C_i^3} \left( \frac{\alpha}{\lambda_i} + \frac{\sigma^2 \ln 2}{B g_i} \right) \right]^+. \]  (39)

With (38) and (39), the optimal \( \ell_i^* \) is then obtained as

\[ \ell_i^* = \ell_i^*/r_i^*. \]  (40)
D. Proof of Lemma 3

The positive semidefinite constraint $F(\lambda) \succeq 0$ can be equivalently expressed as a scalar inequality constraint as [16]

$$\pi(\lambda) \triangleq \pi_{\min}(F(\lambda)) \geq 0,$$

where $\pi_{\min}(\cdot)$ denotes the smallest eigenvalue. Furthermore, the constraint (41) is equivalently written as

$$\pi(\lambda) = \min_{\|\xi\|=1} \xi^H F(\lambda) \xi \geq 0. \quad (42)$$

Given a query point $\lambda_1 \triangleq [\lambda_{1,1}, \ldots, \lambda_{1,K}]^T$, one can find the normalized eigenvector $v_1$ of $F(\lambda_1)$ corresponding to $\pi_{\min}(F(\lambda_1))$. Consequently, we can determine the value of the scalar constraint at a query point as $\pi(\lambda_1) = v_1^H F(\lambda_1) v_1 = \pi_{\min}(F(\lambda_1))$. To obtain a subgradient, we have the following

$$\pi(F(\lambda)) - \pi(F(\lambda_1)) = \min_{\|\xi\|=1} \xi^H F(\lambda) \xi - v_1^H F(\lambda_1) v_1,$$

$$\leq v_1^H (F(\lambda) - F(\lambda_1)) v_1, \quad (43a)$$

$$= \sum_{i=1}^K (\lambda_{1,i} - \lambda_i) \zeta^T H_i v_1, \quad (43b)$$

where the last equality follows from the affine structure of $F(\cdot)$ in (15b). By the weak subgradient calculus [16], the subgradient of $F(\lambda)$ at the given $\lambda$ and $\mu$ is then

$$[\zeta v_1^H H_1 v, \ldots, \zeta v_1^H H_K v, 0]^T, \quad (44)$$

where $v$ is the eigenvector corresponding to the smallest eigenvalue of $F(\lambda)$, and the last zero entry follows from the fact that $\pi(F(\lambda))$ is independent of $\mu$.

REFERENCES

[1] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” To appear in *IEEE Internet Thing J.*, 2016.

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

[3] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: Making smartphones last longer with code offload,” in *Proc. ACM Int. Conf. Mobile Systems, Applications, and Services*, San Francisco, CA, USA, Jun. 15–18, 2010, pp. 49–62.

[4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 25–30, 2012, pp. 945–953.

[5] ETSI. (2014). *ETSI First Meeting of New Standardization Group on Mobile-edge Computing*. [Online]. Available: http://www.etsi.org/news-events/news/388-2014-10-news-etsi-announces-first-meeting-of-new-standardization-group-on-mobile-edge-computing

[6] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *Proc. IEEE Int. Symp. Information Theory*, Barcelona, Spain, Jun. 2016.

[7] C. You, K. Huang, H. Chae, and B. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” 2016. [Online]. Available: http://arxiv.org/abs/1605.08518

[8] J. Xu and R. Zhang, “Energy beamforming with one-bit feedback,” *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5370–5381, Oct. 2014.

[9] S. Bi, R. Zhang, and C. Ho, “Wireless powered communication: Opportunities and challenges,” *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 117–125, Apr. 2015.

[10] J. Xu, L. Liu, and R. Zhang, “Multiuser MISO beamforming for simultaneous wireless information and power transfer,” *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4798–4810, Sep. 2014.

[11] H. Li, J. Xu, R. Zhang, and S. Cui, “A general utility optimization framework for energy harvesting based wireless communications,” *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 79–85, Apr. 2015.

[12] F. Wang, T. Peng, Y. Huang, and X. Wang, “Robust transceiver optimization for power-splitting based downlink MISO SWIPT systems,” *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1492–1496, Sep. 2015.

[13] 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. 1757–1770, May 2016.

[14] T. D. Burd and R. W. Brodersen, “Processor design for portable systems,” *Kluwer J. VLSI Signal Process. Syst.*, vol. 13, no. 2, pp. 203–221, Aug. 1996.

[15] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, “On the Lambert W function,” *Adv. Comput. Math.*, vol. 5, pp. 329-359, 1996.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[17] S. Boyd, “Ellipsoid method,” Stanford Univ., Stanford, CA, USA. [Online]. Available: http://stanford.edu/class/ee364b/lectures/ellipsoid_method.pdf

[18] M. Grant, S. Boyd, and Y. Ye, “CVX: Matlab software for disciplined convex programming,” 2009. [Online]. Available: http://cvxr.com/cvx/