Article:
Ye, Y., Shi, L., Chu, X. et al. (2022) Resource allocation in backscatter-assisted wireless powered MEC networks with limited MEC computation capacity. IEEE Transactions on Wireless Communications, 21 (12). pp. 10678-10694. ISSN: 1536-1276

https://doi.org/10.1109/twc.2022.3185825

Resource Allocation in Backscatter-Assisted Wireless Powered MEC Networks with Limited MEC Computation Capacity

Yinghui Ye, Liqin Shi, Xiaoli Chu, *Senior Member, IEEE*, Rose Qingyang Hu, *Fellow, IEEE*, and Guangyue Lu

***Abstract*—In this paper, we consider a backscatter-assisted wireless powered mobile edge computing (MEC) network, where multiple Internet-of-Things (IoT) nodes harvest energy from the energy signals transmitted by a power beacon (PB) and utilize the harvested energy for local computing and task offloading via hybrid backscatter communication (BackCom) and active transmission (AT). Considering the limited computation capacity of the MEC server and the quality-of-service (QoS) and energy-causality constraints per IoT node, we propose two resource allocation schemes to maximize the total computation bits of all the IoT nodes and the system computation energy efficiency (EE), respectively, by jointly optimizing the computation frequency and time of the MEC server and each IoT node, the transmit power of the PB and each IoT node, and the BackCom power reflection coefficient and the time for energy harvesting (EH), BackCom, and AT of each IoT node. The non-convex computation bits maximization problem is transformed to a convex one by introducing a series of auxiliary variables and proof by contradiction, and then solved by the existing convex tools. The system computation EE maximization is a non-convex nonlinear programming problem. We propose a two-layer iterative algorithm to solve it optimally and devise a reduced-complexity iterative algorithm to solve it sub-optimally by leveraging the block coordinate decent technique. Computer simulations validate the convergence of the proposed iterative algorithms and their superior performance over the benchmark schemes in terms of the computation bits or EE.**

***Index Terms*—Backscatter communications, computation bits, computation energy efficiency, energy harvesting, mobile edge computing.**

I. Introduction

THE Internet of Things (IoT) will play an important role in future intelligent services. However, owing to the cost limitation, IoT devices are usually energy- and computation-constrained, making it very challenging for them to timely handle computation-intensive tasks [1]. To address this challenge, wireless powered mobile edge computing (WPMEC), which integrates wireless power transfer (WPT) [2] with mobile edge computing (MEC) [3]–[5], has been proposed as a promising solution. The key idea of WPMEC is to let IoT devices harvest energy from radio frequency (RF) signals transmitted by a dedicated energy source, e.g., a power beacon (PB), and then utilize the harvested energy to support the local processing and/or task offloading of tasks [1], [6].

Resource allocation schemes based on binary or partial computation offloading have been proposed for WPMEC. In binary computation offloading, an IoT node's task is either executed locally or completely offloaded to an MEC server [3], while in partial computation offloading, a portion of the IoT node's task is offloaded to the MEC server and the rest is computed locally [5]. In [6], the authors maximized the computation success probability by jointly optimizing the binary computation offloading decision, the user's computation frequency and the time allocation for energy harvesting (EH) and task offloading under the energy-causality and delay constraints. In [7], the total computation rate of all the users in a multi-user WPMEC network was maximized by jointly optimizing user's binary computation offloading decision, computation frequency and time allocation for EH and task offloading. In [8], an online algorithm, which jointly selects the binary computation offloading mode and adjusts the time resource for WPT and task offloading according to the time-varying channel, was proposed to maximize the weighted sum-computation rate of all the IoT nodes in a multi-user WPMEC network. Under partial computation offloading, the weighted sum-computation bits of all the IoT nodes were maximized and the PB's energy consumption was minimized respectively in [9] and [10], by jointly optimizing the PB's beamforming and the IoT nodes' computation frequency, task bits for offloading, and offloading time. In [11], the authors maximized the computation energy efficiency (EE) of all the IoT nodes in a WPMEC network by jointly optimizing the transmit power of the PB and each IoT node, and each IoT node's computation frequency and time for task offloading and EH. In [12], a resource allocation scheme was proposed for cooperation-assisted WPMEC, where the IoT node close to the AP servers as relay, to minimize the PB's energy consumption by jointly optimizing the time for EH and cooperative data offloading, and the transmit power of each IoT node.

In the above works [6]–[12], the IoT nodes offload task bits to the MEC server following the harvest-then-transmit

This work was supported in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JQ-713, in part by the Young Talent fund of University Association for Science and Technology in Shaanxi under Grant 20210121, in part bythe Scientific Research Program Funded by Shaanxi Provincial Education Department under Grant 21JK0914, and in part by the Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application under Grant 2017KCT-30-02. *(Corresponding Author: Liqin Shi)*

Yinghui Ye, Liqin Shi, and Guangyue Lu are with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, China. (e-mail: connectyyh@126.com, liqinshi@hotmail.com, tonylugy@163.com)

Xiaoli Chu is with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. (e-mail: x.chu@sheffield.ac.uk)

Rose Qingyang Hu is with the Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84332 USA (e-mail: rose.hu@usu.edu).

protocol via active transmission (AT), which enjoys a high transmission rate for data offloading but at the cost of a high power consumption [13]. Under the energy-causality constraint, each IoT node has to be allocated with a long period for harvesting sufficient energy to support AT, leaving only a short period for AT offloading within each time block. This will limit their task offloading performance. Different from AT, backscatter communication (BackCom) allows an IoT node to modulate and reflect the incident signals by adjusting the antenna load impedance instead of generating RF signals by itself [14]. Such an approach avoids the use of active components, resulting in a much lower power consumption as well as a lower offloading rate than AT. Accordingly, AT and BackCom have different tradeoffs[1] between data transmission and power consumption [15], [16], which can be exploited to achieve efficient task offloading in WPMEC. Driven by this, hybrid BackCom-AT has been employed for computation offloading in WPMEC [17].

Hybrid BackCom-AT offloading brings new challenges in resource allocation as AT offloading and BackCom offloading share the same pool of resources within each time block and the optimization of resource allocation needs to consider the tradeoff between them. In [18], under a complete offloading strategy that offloads all task bits of an IoT node to the MEC server, the authors minimized the energy consumption of the PB by jointly optimizing the IoT nodes' transmit power for AT, and their time for BackCom offloading and AT offloading. Under partial computation offloading, the total computation bits of all the IoT nodes were maximized by jointly optimizing the BackCom power reflection coefficient, transmit power for AT, BackCom and AT offloading time, local computation frequency and computation time of each IoT node [19]. Under the same setting as in [19], a max-min computation EE problem and a computation EE maximization problem were studied in [20] and [21], respectively. In [22], to minimize the total delay for tasks offloading and computing of all the IoT nodes, the authors jointly optimized the transmission time of the PB, and each IoT node's transmit power for AT, and their portions of task bits for BackCom offloading, AT offloading and local computing.

We note that the MEC server's energy consumption and computation time and practically limited computation capacity have not been considered in the existing works on hybrid BackCom-AT WPMEC networks [18]–[22]. However, the computation time of the MEC server may not be negligible for ultra-delay-sensitive applications [5]. Besides, assuming a sufficiently high computation frequency of the MEC server, thus ignoring its computation time, makes it impossible to determine the computation energy consumption of the MEC server, which however should be considered when designing energy efficient resource allocation schemes for MEC networks [5], [23]. Please note that considering the limited computation capacity for the MEC server introduces additional variables, e.g., the MEC server's computation frequency and time, which are potentially coupled in the optimization of resource, thus, bringing new challenges in resource allocation.

In this paper, we design two resource allocation schemes for a hybrid BackCom-AT WPMEC network, where a PB and an MEC server are deployed close to multiple IoT nodes to provide them with energy and computational services on demand, in order to maximize the total computation bits of all the IoT nodes and the system computation EE, respectively, while considering the MEC server's energy consumption and computation time and capacity. The main contributions are summarized as follows.

- Considering the computation resource allocation at the MEC server and partial computation offloading at each IoT node, we maximize the sum computation bits of all the IoT nodes by jointly optimizing the transmit power of the PB, the MEC server's computation frequency and time, and each IoT node's local computation frequency and time, EH time, BackCom time, AT time, BackCom power reflection coefficient and AT transmit power, subject to the quality-of-service (QoS) and other necessary constraints. This joint optimization is formulated into a non-convex problem. By using the proof by contradiction and introducing a series of auxiliary variables, the original problem is transformed into a convex one and is then solved by the convex tools.
- The system computation EE is defined as the ratio of the total computation bits of all the IoT nodes to the total energy consumption of the PB, MEC server and all the IoT nodes in the system. The system computation EE maximization problem is a non-convex fractional programming problem of the same optimization variables as in the sum computation bits maximization problem. To solve it, based on the Dinkelbach's method and the bisection method, we first propose a two-layer iterative algorithm to obtain the optimal resource allocation scheme. By leveraging the block coordinate decent (BCD) technique, we decompose the original problem into two subproblems, i.e., 1) given the transmit power of the PB, jointly optimizing all the other optimization variables, and 2) optimizing the transmit power of the PB while all the other variables are given. We further develop a reduced-complexity iterative algorithm to a suboptimal resource allocation.
- Our analytical and numerical results show that under the limited computation capacity of the MEC server, each IoT node should always perform local computing throughout each time block in order to achieve the maximum total computation bits or system computation EE; the total computation bits are maximized when the MEC server always uses the maximum allowed computing frequency to complete their computing tasks; the system computation EE maximization is maximized when the available time in each time block is used up for BackCom/AT offloading, tasks computing at the IoT node and the MEC server, while the computing frequency of the MEC server is jointly optimized with its computing time and the offloaded task bits; the proposed total computation bits/EE maximization schemes outperform the bench-

[1]The offloading rate of AT is higher than that of BackCom, but the energy consumption of BackCom is much lower than that of AT [15].

mark schemes in terms of computation bits/EE and the proposed reduced-complexity iterative algorithm achieves the suboptimal performance.

The rest of the paper is organized as follows. In Section II, we introduce the system model and analyze the working flow. Section III studies the computation bits maximization problem. In Section IV, the system computation EE maximization problem is studied and two iterative algorithms are developed to obtain the optimal and suboptimal solutions, respectively. Simulation results and the conclusion are given in Sections V and VI, respectively.

In the existing works on backscatter-assisted wireless powered networks, the sum throughput [16], [24] and the EE [25], [26] have been maximized by optimizing the allocation of communication resources between the BackCom and AT under the energy-causality constraint of each IoT node. Our work is different in that we consider not only the tradeoff between the BackCom and AT for task offloading, but also that between the different computation capacities of the IoT nodes and the MEC server.

II. System Model and Working Flow

As shown in Fig. 1, we consider a hybrid BackCom-AT WPMEC network that consists of one PB, K IoT nodes and one MEC server. Each node is equipped with a single antenna and works in the half-duplex mode. The K IoT nodes harvest energy from the RF signals emitted by the PB and use the harvested energy to offload their tasks to the MEC server as well as compute tasks locally. In order to avoid consuming the battery power and thereby to prolong the operation time of each IoT node, the energy consumed at each IoT node for computing and offloading their tasks is assumed to be less than or equal to its harvested energy in each time block T. Following [19]–[22], we assume that the bits of each task are bit-wise independent so that the partial offloading scheme can be applied at each IoT node. Each IoT node is equipped with separate circuits for backscattering, AT, EH, and computing, respectively, and thus can perform local computation when offloading tasks or harvesting energy. A quasi-static fading channel model is assumed, that is, all the channels remain static within one time block but may vary across adjacent time blocks. We also assume that perfect channel state information (CSI) is available at the MEC server for obtaining the performance bound. The MEC server can obtain the CSI of all links at the beginning of each transmission block by using channel estimation methods such as the least-square estimation [27], and exploiting the channel reciprocity, as detailed in Section II-E of [28].

Each time block T is divided into five phases, which are the EH phase, the BackCom phase, the AT phase, the task execution phase, and the downloading phase. In the EH phase, the PB broadcasts energy signals and each IoT node harvests energy from the received signals. In the BackCom phase and the AT phase, the K IoT nodes offload parts of their tasks to the MEC server following the time division multiple access (TDMA) protocol. Specifically, in the BackCom phase, the PB keeps broadcasting energy signals and the K IoT nodes take turns to offload task bits by modulating and backscattering the energy signals to the MEC server. In the AT phase, the PB keeps silent[2] [18]–[22] and the K IoT nodes take turns to transmit task bits to the MEC server using the harvested energy. In the task execution phase, the PB and all the IoT nodes keep silent, while the MEC server executes all the received computation tasks. The MEC server sends the computation results to the IoT nodes in the downloading phase. In this work, we consider the scenarios where the computation result bits are much less than the task bits, e.g., automatic manufacturing systems. Thus, the downloading phase is significantly shorter than the other four phases, and is ignored hereafter for simplicity.

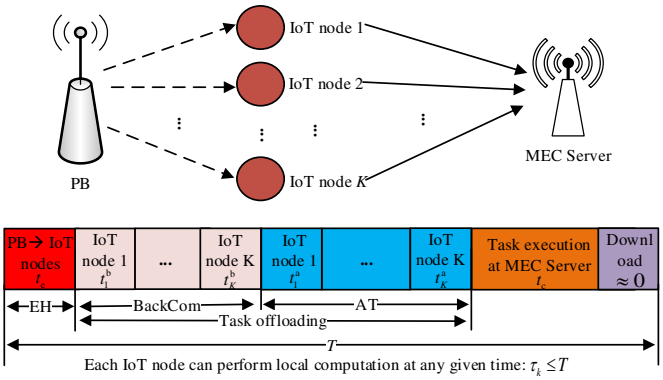

Fig. 1. Hybrid BackCom-AT WPMEC and its frame structure.

A. EH Phase

Let $t_{\rm e}$ denote the duration of the EH phase and $P_{\rm t}$ denote the transmit power of the PB, where $0 < P_{\rm t} \le P_{\max}$ and $P_{\max}$ is the maximum allowed transmission power at the PB. Then, the harvested energy of the k-th ($k \in \mathcal{K} = \{1, 2, ..., K\}$) IoT node during the EH phase can be computed as $E_k^{\rm h} = t_{\rm e}\eta P_{\rm t} g_k$, where g_k is the channel power gain between the k-th IoT node and the PB, η ($0 < \eta < 1$) denotes the energy conversion efficiency[3].

B. BackCom Phase

The BackCom phase is divided into K sub-phases. In the k-th sub-phase of duration $t_k^{\rm b}$, the k-th IoT node offloads its task bits to the MEC server via BackCom while the others perform EH. The task bits offloaded by the k-th IoT node via BackCom are given by

$$R_k^{\rm b} = t_k^{\rm b} B \log_2 \left(1 + \frac{\xi \rho_k P_{\rm t} g_k h_k}{B\sigma^2}\right), \tag{1}$$

[2]Please note that our work can be extended to the scenario where the PB continues to broadcast energy signal in the AT phase, in which case, the IoT nodes can harvest more energy in each time block but the PB consumes more energy and the MEC server needs to remove the PB energy signal as interference.

[3]For analytical tractability, this paper considers the linear EH model. Please note that this work can be extended to the scenarios with a non-linear EH model by using the approach adopted in [15] or [29].

where B is the channel bandwidth, ρ_k ($0 \leq \rho_k \leq 1$) denotes the power reflection coefficient of the k-th IoT node, h_k represents the channel power gain between the k-th IoT node and the MEC server, σ^2 is the noise power spectral density, and ξ denotes the performance gap[4] between the BackCom capacity and the Shannon capacity.

From (1), we can see that the portion $\rho_k P_{\rm t} g_k$ of the received power is used for backscattering the k-th IoT node's task bits to the MEC server; while the rest of the received power, $(1-\rho_k)P_{\rm t}g_k$, is flowed into the EH circuit. During the sub-phase t_k^b, the harvested energy of the k-th IoT node and the i-th ($i \in \mathcal{K}\backslash k$) IoT node can be, respectively, computed as

$$E_k^{\rm b} = t_k^{\rm b}\eta\left(1-\rho_k\right)P_{\rm t}g_k, \tag{2}$$

$$E_{k,i}^{\rm b} = t_k^{\rm b}\eta P_{\rm t} g_i. \tag{3}$$

Accordingly, at the end of the BackCom phase, the total harvested energy of the k-th IoT node is determined by

$$\begin{aligned} E_k^{\rm total} &= E_k^{\rm h} + E_k^{\rm b} + \sum\nolimits_{i\in\mathcal{K}, i\neq k} E_{i,k}^{\rm b} \\ &= \eta P_{\rm t} g_k \left(t_{\rm e} + \sum\nolimits_{i\in\mathcal{K}, i\neq k} t_i^{\rm b}\right) + \eta P_{\rm t} g_k t_k^{\rm b}\left(1-\rho_k\right) \\ &= \eta P_{\rm t} g_k \left(t_{\rm e} + \sum\nolimits_{i=1}^{K} t_i^{\rm b}\right) - \eta P_{\rm t} g_k \rho_k t_k^{\rm b}. \end{aligned} \tag{4}$$

The energy consumption of the k-th IoT node for BackCom is given by $P_{\rm c} t_k^{\rm b}$, where $P_{\rm c}$ denotes the circuit power consumption for BackCom [19]–[21].

C. AT Phase

The AT phase is also divided into K subphases. In the k-th subphase of duration t_k^a, the k-th IoT node transmits its task bits to the MEC server. Letting p_k denote the transmit power of the k-th IoT node, the task bits offloaded by the k-th IoT node during t_k^a can be calculated as

$$R_k^{\rm a} = t_k^{\rm a} B \log_2\left(1 + \frac{p_k h_k}{B\sigma^2}\right). \tag{5}$$

Note that the value of p_k is limited by the total energy harvested by the k-th IoT node during the EH and BackCom phases minus the energy consumed for offloading and local computing so far.

The energy consumption of the k-th IoT node for offloading via AT can be calculated as $\left(p_k + p_{\rm c}\right) t_k^{\rm a}$, where $p_{\rm c}$ denotes the constant circuit power consumption for AT.

[4]During t_k^b, the received signal at the MEC server is given by $\sqrt{\rho_k P_{\rm t} g_k h_k} x_e x_k^b + \sqrt{P_0 g_{TR}} x_e + n$, where x_e, x_k^b and n are the energy signal transmitted by the PB, the offloaded data via BackCom and the additive Gaussian noise at the MEC server, respectively. Assuming that the signal received from the PB, $\sqrt{P_0 g_{TR}} x_e$, is removed by performing successive interference cancellation, as the PB's energy signal can be predefined and known by the MEC server [13]. The remaining signal for decoding x_k^b can be written as $\sqrt{\rho_k P_{\rm t} g_k h_k} x_e x_k^b + n$. The Shannon capacity is for Gaussian inputs, but for BackCom, the multiplicative signal $x_e x_k^b$ does not necessarily follow the Gaussian distribution and thus a performance gap exists between the Shannon and BackCom capacities. In this work, we use ξ ($0 < \xi < 0$) to represent this performance gap following [15], [16], [19].

At the end of the AT phase, the total task bits offloaded by all the IoT nodes are given as

$$\begin{aligned} R_{\rm sum}^{\rm o} &= \sum\nolimits_{k=1}^{K} \left(R_k^{\rm b} + R_k^{\rm a}\right) \\ &= \sum_{k=1}^{K} \left(t_k^{\rm b} B \log_2\left(1 + \frac{\xi\rho_k P_{\rm t} g_k h_k}{B\sigma^2}\right) + t_k^{\rm a} B \log_2\left(1 + \frac{p_k h_k}{B\sigma^2}\right)\right). \end{aligned} \tag{6}$$

D. Task Execution Phase

After successfully receiving the task bits offloaded by the IoT nodes, the MEC server starts to execute the received tasks.

Let $f_{\rm m}$ (in Hz unit) and $t_{\rm c}$ (in s unit) denote the computation frequency and time of the MEC server, respectively. The MEC server's computation capacity can be computed as

$$R_{\rm m} = \frac{t_{\rm c} f_{\rm m}}{C_{\rm cpu}}, \tag{7}$$

where $C_{\rm cpu}$ represents the number of CPU cycles required for computing one bit at the MEC server.

The number of task bits computed by the MEC server is determined by the minimum between the received task bits and the computation capacity of the MEC server, i.e.,

$$\begin{aligned} R_{\rm m}^{\rm e} &= \min\left\{R_{\rm sum}^{\rm o}, R_{\rm m}\right\} \\ &= \min\left\{ \sum\nolimits_{k=1}^{K} \left(t_k^{\rm b} B \log_2\left(1 + \frac{\xi\rho_k P_{\rm t} g_k h_k}{B\sigma^2}\right) \right.\right. \\ &\qquad \left.\left. + t_k^{\rm a} B \log_2\left(1 + \frac{p_k h_k}{B\sigma^2}\right)\right), \frac{t_{\rm c} f_{\rm m}}{C_{\rm cpu}}\right\}. \end{aligned} \tag{8}$$

Letting $\varepsilon_{\rm m}$ denote the effective capacitance coefficient of the processor chip at the MEC server, the energy consumption of the MEC server for task executing is given by [30]

$$E_{\rm m}^{\rm e} = \varepsilon_{\rm m} f_{\rm m}^3 t_{\rm c}. \tag{9}$$

E. Local Computation

At any time during a time block, each IoT node can compute its own tasks locally. Letting f_k (in Hz unit) and τ_k (in s unit) denote the computation frequency and time of the k-th IoT node, respectively, the task bits computed locally by the k-th IoT node are given by

$$R_k^{\rm e} = \frac{\tau_k f_k}{C_{{\rm cpu},k}}, \tag{10}$$

where $C_{{\rm cpu},k}$ denotes the number of CPU cycles required for computing one bit at the k-th IoT node.

The consumed energy for local computation at the k-th IoT node is given by

$$E_k^{\rm e} = \varepsilon_k f_k^3 \tau_k, \tag{11}$$

where ε_k is the effective capacitance coefficient of the processor chip at the k-th IoT node. Note that the values of E_k^e and τ_k are limited by the harvested energy minus the energy consumed for offloading by the k-th IoT node.

III. Computation Bits Maximization

In this section, we design an optimal resource allocation scheme to maximize the total computation bits for the hybrid BackCom-AT WPMEC network by jointly optimizing the transmit power of the PB $P_{\rm t}$, the EH time $t_{\rm e}$, the MEC server's computation frequency $f_{\rm m}$ and time $t_{\rm c}$, and the IoT nodes' BackCom time vector $\mathbf{t}^{\rm b} = \left[t_1^{\rm b}, \ldots, t_K^{\rm b}\right]$, AT time vector $\mathbf{t}^{\rm a} = [t_1^{\rm a}, \ldots, t_K^{\rm a}]$, power reflection coefficient vector $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_K]$, transmit power vector $\mathbf{p} = [p_1, \ldots, p_K]$, local computation frequency vector $\mathbf{f} = [f_1, \ldots, f_K]$ and time vector $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_K]$. Specifically, we formulate a computation bits maximization problem subject to the QoS, energy causality, latency, computational budget and transmit power constraints. Then, we transform the formulated problem into a convex one and obtain the optimal solution using the existing convex optimization tools.

A. Problem Formulation

The total computation bits include the task bits computed by both the MEC server and the IoT nodes, and are given by

$$R_{\rm total}\left(P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}\right) = R_{\rm m}^{\rm e} + \sum_{k=1}^{K} R_k^{\rm e}, \quad (12)$$

where $\mathbf{t}^{\rm b} = \left[t_1^{\rm b}, \ldots, t_K^{\rm b}\right]$, $\mathbf{t}^{\rm a} = [t_1^{\rm a}, \ldots, t_K^{\rm a}]$, $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_K]$, $\mathbf{p} = [p_1, \ldots, p_K]$, $\mathbf{f} = [f_1, \ldots, f_K]$ and $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_K]$. The maximization of the total computation bits should consider the following constraints.

1) QoS Constraint: Letting $L_{{\rm min},k}$ and β_k ($0 \le \beta_k \le 1$), $\forall k \in \mathcal{K}$, denote the minimum required task bits and the portion of task bits offloaded to the MEC server for the k-th IoT node, respectively, the following three inequalities should be satisfied for each IoT node to guarantee that the minimum required task bits of each IoT node are computed successfully,

$$R_k^{\rm b} + R_k^{\rm a} \ge \beta_k L_{{\rm min},k}, \forall k, \quad (13)$$

$$\sum\nolimits_{k=1}^{K} \beta_k L_{{\rm min},k} \le R_{\rm m}, \quad (14)$$

$$R_k^{\rm e} \ge \left(1-\beta_k\right) L_{{\rm min},k}, \forall k, \quad (15)$$

where (13) and (14) jointly ensure that at least $\beta_k L_{{\rm min},k}$ task bits are offloaded to and computed successfully by the MEC server, while (15) ensures that at least $\left(1-\beta_k\right) L_{{\rm min},k}$ task bits are computed successfully at the k-th IoT node.

2) Energy-Causality Constraint: To ensure that the consumed energy of each IoT node for offloading and computing is less than its harvested energy within each time block, the energy-causality constraint is expressed as

$$P_{\rm c} t_k^{\rm b} + \left(p_k + p_{\rm c}\right) t_k^{\rm a} + \varepsilon_k f_k^3 \tau_k \le E_k^{\rm total}, \forall k. \quad (16)$$

Based on (12)-(16), the computation bits maximization problem can be formulated as

$$\begin{aligned}
\mathbf{P_0}: \max_{P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}, \boldsymbol{\beta}} & R_{\rm total}\left(P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}\right) \\
{\rm s.t.}\ & {\rm C1}: (13) - (15), \\
& {\rm C2}: (16), \\
& {\rm C3}: 0 \le f_{\rm m} \le f_{\max}, 0 \le f_k \le f_k^{\max},\ \forall k, \\
& {\rm C4}: t_{\rm e} + \textstyle\sum_{k=1}^{K}\left(t_k^{\rm b} + t_k^{\rm a}\right) + t_{\rm c} \le T, t_{\rm e}, t_k^{\rm b}, t_k^{\rm a}, t_{\rm c} \ge 0, \forall k, \\
& {\rm C5}: 0 \le \tau_k \le T,\ \forall k, \\
& {\rm C6}: 0 < P_{\rm t} \le P_{\max}, p_k \ge 0, \forall k, \\
& {\rm C7}: 0 \le \rho_k \le 1, \forall k, \\
& {\rm C8}: 0 \le \beta_k \le 1, \forall k,
\end{aligned}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_K]$, and $f_k^{\max}$ and $f_{\max}$ are the maximum computing frequencies for the k-th IoT node and the MEC server, respectively.

In $\mathbf{P_0}$, C1 is the QoS constraint, C2 is the energy-causality constraint. C3 constrains the maximum computing frequencies at the MEC server and IoT nodes, C4 and C5 are the latency constraints, which ensure that all the task bits should be executed within T, C6 and C7 are the constraints on the transmit power of the PB and the IoT nodes and the power reflection coefficient of each IoT node, respectively, and C8 constrains the portion of task bits offloaded to the MEC server for each IoT node. The transmit power and computation capacity constraints together with the minimum QoS requirement $L_{{\rm min},k}$ may cause an infeasibility issue, i.e., it may be impossible to find a solution to the formulated total computation bits maximization problem (as well as the system computation EE maximization problem in Section IV) under certain values of those constraints and $L_{{\rm min},k}$. It is reasonable to assume that the probability of an infeasibility event is very low for practical settings of the maximum PB transmit power $P_{\max}$ or the maximum computing frequencies of the IoT nodes and the MEC server.

It is observed that $\mathbf{P_0}$ is a typical non-convex optimization problem since there are coupling relationships among several optimization variables (i.e., $P_{\rm t}$, $t_{\rm e}$, ρ_k and $t_k^{\rm b}$, f_k and τ_k, $f_{\rm m}$ and $t_{\rm c}$, etc.) in the objective function and constraints. We note that directly introducing auxiliary variables into $\mathbf{P_0}$ may not decouple all the coupling optimization variables. Because there will always be at least one non-convex constraint remaining. In order to solve $\mathbf{P_0}$, we first derive the closed-form expressions for parts of the optimal solutions and then transform $\mathbf{P_0}$ into a convex one by introducing auxiliary variables.

B. Solution

In order to remove the min function in the objective, we introduce a slack variable λ and let $\lambda = \min\left\{R_{\rm sum}^{\rm o}, R_{\rm m}\right\}$. Accordingly, the optimization problem $\mathbf{P_0}$ can be reformulated as

$$\begin{aligned}
\mathbf{P_1}: \max_{P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}, \boldsymbol{\beta}, \lambda} & \lambda + \textstyle\sum_{k=1}^{K} R_k^{\rm e} \\
{\rm s.t.}\ & {\rm C1} - {\rm C8}, \\
& {\rm C9}: R_{\rm sum}^{\rm o} \ge \lambda, \\
& {\rm C10}: R_{\rm m} \ge \lambda.
\end{aligned}$$

As for $\mathbf{P_1}$, we provide the following proposition to determine the optimal transmit power of the PB $P_{\rm t}^*$, the optimal

computation frequency of the MEC server $f_{\rm m}^*$ and the optimal computing time of the k-th IoT node τ_k^*.

Proposition 1: For the considered network, the maximum total computation bits are achieved when the PB transmits energy signals with the maximum transmit power, the MEC server executes all the received tasks with its maximum computation frequency and each IoT node performs local computation throughout a time block, namely $P_{\rm t}^* = P_{\max}$, $f_{\rm m}^* = f_{\max}$ and $\tau_k^* = T$.

Proof. Please see Appendix A. ∎

Substituting $P_{\rm t} = P_{\max}$, $f_{\rm m} = f_{\max}$ and $\tau_k = T$ into $\mathbf{P_1}$, we have

$$\begin{aligned}
\mathbf{P}_2: \max_{t_{\rm e},\mathbf{t}^{\rm b},\mathbf{t}^{\rm a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_{\rm c},\boldsymbol{\beta},\lambda} \ & \lambda + \textstyle\sum_{k=1}^K \frac{f_k T}{C_{{\rm cpu},k}} \\
\text{s.t.} \mathrm{C1}': \ & t_k^{\rm b} B \log_2\left(1+\tfrac{\xi\rho_k P_{\max} g_k h_k}{B\sigma^2}\right) + R_k^{\rm a} \geq \beta_k L_{{\min},k}, \forall k, \\
& \textstyle\sum_{k=1}^K \beta_k L_{{\min},k} \leq \frac{t_{\rm c} f_{\max}}{C_{\rm cpu}}, \\
& \tfrac{f_k T}{C_{{\rm cpu},k}} \geq (1-\beta_k) L_{{\min},k}, \forall k, \\
\mathrm{C2}': \ & P_{\rm c} t_k^{\rm b} + (p_k + p_{\rm c}) t_k^{\rm a} + \varepsilon_k f_k^3 T \\
& \leq \eta P_{\max} g_k \left(t_{\rm e} + \textstyle\sum_{i=1}^K t_i^{\rm b}\right) - t_k^{\rm b} \eta \rho_k P_{\max} g_k, \forall k, \\
\mathrm{C3}': \ & 0 \leq f_k \leq f_k^{\max}, \ \forall k, \\
\mathrm{C4, C7, C8}, & \\
\mathrm{C9}': \ & \textstyle\sum_{k=1}^K \left(t_k^{\rm b} B \log_2\left(1+\tfrac{\xi\rho_k P_{\max} g_k h_k}{B\sigma^2}\right) + R_k^{\rm a}\right) \geq \lambda, \\
\mathrm{C10}': \ & \tfrac{t_{\rm c} f_{\max}}{C_{\rm cpu}} \geq \lambda.
\end{aligned}$$

It can be observed that $\mathbf{P}_2$ is still non-convex due to the non-convex constraints, C1′, C2′ and C9′, which contain coupled variables, e.g., $t_k^{\rm b}$ and ρ_k. To overcome the non-convexity, we introduce the auxiliary variables, $x_k = \rho_k t_k^{\rm b}$ and $y_k = t_k^{\rm a} p_k$, and use $\frac{x_k}{t_k^{\rm b}}$ and $\frac{y_k}{t_k^{\rm a}}$ to replace ρ_k and p_k, respectively. Thus, $\mathbf{P}_2$ is equivalently transformed into $\mathbf{P}_3$, as follows,

$$\begin{aligned}
\mathbf{P}_3: \max_{t_{\rm e},\mathbf{t}^{\rm b},\mathbf{t}^{\rm a},\mathbf{x},\mathbf{y},\mathbf{f},t_{\rm c},\boldsymbol{\beta},\lambda} \ & \lambda + \textstyle\sum_{k=1}^K \frac{f_k T}{C_{{\rm cpu},k}} \\
\text{s.t.} \mathrm{C1}'': \ & t_k^{\rm b} B \log_2\left(1+\tfrac{\xi x_k P_{\max} g_k h_k}{t_k^{\rm b} B\sigma^2}\right) \\
& + t_k^{\rm a} B \log_2\left(1+\tfrac{y_k h_k}{t_k^{\rm a} B\sigma^2}\right) \geq \beta_k L_{{\min},k}, \forall k, \\
& \textstyle\sum_{k=1}^K \beta_k L_{{\min},k} \leq \frac{t_{\rm c} f_{\max}}{C_{\rm cpu}}, \\
& \tfrac{f_k T}{C_{{\rm cpu},k}} \geq (1-\beta_k) L_{{\min},k}, \forall k, \\
\mathrm{C2}'': \ & P_{\rm c} t_k^{\rm b} + y_k + p_{\rm c} t_k^{\rm a} + \varepsilon_k f_k^3 T \\
& \leq \eta P_{\max} g_k \left(t_{\rm e} + \textstyle\sum_{i=1}^K t_i^{\rm b}\right) - \eta x_k P_{\max} g_k, \forall k, \\
\mathrm{C3}', \mathrm{C4, C8, C10}', & \\
\mathrm{C7}': \ & 0 \leq x_k \leq t_k^{\rm b}, y_k \geq 0, \forall k, \\
\mathrm{C9}'': \ & \textstyle\sum_{k=1}^K \left(t_k^{\rm b} B \log_2\left(1+\tfrac{\xi x_k P_{\max} g_k h_k}{t_k^{\rm b} B\sigma^2}\right)\right. \\
& \left. + t_k^{\rm a} B \log_2\left(1+\tfrac{y_k h_k}{t_k^{\rm a} B\sigma^2}\right)\right) \geq \lambda,
\end{aligned}$$

where $\mathbf{x} = [x_1, \ldots, x_K]$ and $\mathbf{y} = [y_1, \ldots, y_K]$.

Proposition 2: $\mathbf{P}_3$ is convex and can be efficiently solved by using the convex optimization tool, e.g., CVX.

Proof. Please see Appendix B. ∎

IV. System Computation EE Maximization

In this section, we maximize the system computation EE maximization for the hybrid BackCom-AT WPMEC system under the QoS, energy causality, latency, computational budget and transmit power constraints, by jointly optimizing the same variables as of the computation bits maximization problem $\mathbf{P_0}$. The formulated joint optimization problem is highly non-convex and difficult to solve. We devise two iterative algorithms to solve it optimally and sub-optimally (with a reduced complexity), respectively.

A. Problem Formulation

Following [31], [32], the system computation EE for the considered network is defined as the ratio of the total computation bits of all the IoT nodes to the total energy consumption of the PB, MEC server and all the IoT nodes in the system. The total consumed energy consists of three parts: the consumed energy of the PB for wireless energy transfer, the energy consumption of all the IoT nodes for task offloading and computing, and the energy consumed at the MEC server. Accordingly, the total energy consumption can be computed as

$$\begin{aligned}
E_{\rm total} = & \underbrace{(P_{\rm t} + P_{\rm sc})\left(t_{\rm e} + \sum_{k=1}^K t_k^{\rm b}\right) - \sum_{k=1}^K E_k^{\rm total}}_{\text{Part 1}} + \underbrace{E_{\rm m}^{\rm e}}_{\text{Part 3}} \\
& + \underbrace{\sum_{k=1}^K \left(P_{\rm c} t_k^{\rm b} + (p_k + p_{\rm c}) t_k^{\rm a} + \varepsilon_k f_k^3 \tau_k\right)}_{\text{Part 2}},
\end{aligned} \tag{17}$$

where $P_{\rm sc}$ denotes the constant circuit power consumption at the PB.

The computation EE of the system is given by

$$q_s\left(P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}\right) = \frac{R_{\rm total}}{E_{\rm total}}. \tag{18}$$

Based on (18) and the QoS and energy-causality constraints in (13)-(16), the system computation EE maximization problem can be formulated as

$$\begin{aligned}
\mathbf{P}_4: \max_{P_{\rm t},t_{\rm e},\mathbf{t}^{\rm b},\mathbf{t}^{\rm a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},\boldsymbol{\tau},t_{\rm c},f_{\rm m},\boldsymbol{\beta}} \ & q_s\left(P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\rm c}, f_{\rm m}\right) \\
\text{s.t. } & \mathrm{C1-C8}.
\end{aligned}$$

Similar to $\mathbf{P}_1$, $\mathbf{P}_4$ is also highly non-convex and difficult to solve due to multiple coupled optimization variables in both the objective function and constraints, which can not be tackled by means of variable substitution. Besides, $\mathbf{P}_4$ is more complicated than $\mathbf{P}_1$ since the objective function of $\mathbf{P}_4$ is a ratio of two functions. The following section is devoted to solving $\mathbf{P}_4$.

B. Solution and Iterative Algorithms

We provide the following proposition to determine the optimal computation time of each IoT node, as part of the optimal solution to $\mathbf{P}_4$.

Proposition 3: In order to achieve the maximum computation EE for the considered system, each IoT node performs local computation throughout a time block, i.e., $\tau_k^* = T$.

Proof. Proposition 3 can be proven by using contradiction and the detailed process is similar to Appendix A. Thus, the proof is omitted here for brevity. ∎

By substituting $\tau_k = T$ into $\mathbf{P}_4$, we have

$$\begin{aligned}\mathbf{P}_5: \max_{P_\mathrm{t},t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_\mathrm{c},f_\mathrm{m},\boldsymbol{\beta}} & q_s\left(P_\mathrm{t},t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},T,t_\mathrm{c},f_\mathrm{m}\right)\\ \text{s.t.}\ & \mathrm{C1}^\dagger: (13),(14),\ \tfrac{f_k T}{C_{\mathrm{cpu},k}} \geq (1-\beta_k)\, L_{\min,k}, \forall k,\\ & \mathrm{C2}^\dagger: P_\mathrm{c} t_k^\mathrm{b} + (p_k + p_\mathrm{c})\, t_k^\mathrm{a} + \varepsilon_k f_k^3 T\\ & \quad \leq \eta P_\mathrm{t} g_k \left(t_\mathrm{e} + \textstyle\sum_{i=1}^K t_i^\mathrm{b}\right) - t_k^\mathrm{b}\eta\rho_k P_\mathrm{t} g_k, \forall k,\\ & \mathrm{C3}, \mathrm{C4}, \mathrm{C6}-\mathrm{C8}.\end{aligned}$$

We can see that $\mathbf{P}_5$ is still non-convex mainly because P_t is coupled with ρ_k, t_k^b and t_e. In the following, we first solve $\mathbf{P}_5$ for given P_t.

1) Solving $\mathbf{P}_5$ for given P_t: To remove the *min* function in (8) from the objective function of $\mathbf{P}_5$, we introduce the slack variable $\lambda = \min\{R_\mathrm{sum}^\mathrm{o}, R_\mathrm{m}\}$ into $\mathbf{P}_5$ and have

$$\begin{aligned}\mathbf{P}_6: \max_{t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_\mathrm{c},f_\mathrm{m},\boldsymbol{\beta},\lambda} & \frac{\lambda + \sum_{k=1}^K \frac{f_k T}{C_{\mathrm{cpu},k}}}{E_\mathrm{total}(P_\mathrm{t},t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},T,t_\mathrm{c},f_\mathrm{m})}\\ \text{s.t.}\ & \mathrm{C1}^\dagger, \mathrm{C2}^\dagger, \mathrm{C3}, \mathrm{C4}, \mathrm{C7}-\mathrm{C10}.\end{aligned}$$

Next, we use the following lemma from the Dinkelbach's method to transform the objective function of $\mathbf{P}_6$ from a fractional form into a subtractive form.

Lemma 1. The optimal solution to $\mathbf{P}_6$ is achieved if and only if the following equation holds.

$$\begin{aligned}&\max_{t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_\mathrm{c},f_\mathrm{m},\boldsymbol{\beta},\lambda} \lambda + \sum_{k=1}^K \frac{f_k T}{C_{\mathrm{cpu},k}}\\ &-q^+ E_\mathrm{total}\left(P_\mathrm{t},t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},T,t_\mathrm{c},f_\mathrm{m}\right) = \lambda^+ + \sum_{k=1}^K \frac{f_k^+ T}{C_{\mathrm{cpu},k}}\\ &-q^+ E_\mathrm{total}\left(P_\mathrm{t},t_\mathrm{e}^+,\mathbf{t}^{\mathrm{b}+},\mathbf{t}^{\mathrm{a}+},\boldsymbol{\rho}^+,\mathbf{p}^+,\mathbf{f}^+,T,t_\mathrm{c}^+,f_\mathrm{m}^+\right) = 0, \quad (19)\end{aligned}$$

where q^+ denotes the maximum objective function value of $\mathbf{P}_6$ and the superscript $+$ indicates the optimal solution to $\mathbf{P}_6$. The proof of Lemma 1 can be readily obtained based on the generalized fractional programming theory [33] and the detailed proof is omitted here for brevity.

Based on Lemma 1, we propose a Dinkelbach-based iterative algorithm to solve $\mathbf{P}_6$ as summarized in Algorithm 1. Denoting the given value of the system computation EE by q, in each iteration of Algorithm 1, the following problem $\mathbf{P}_7$ is solved for the given q and the obtained optimal solution is used to update the value of q, which is used as the given q for the next iteration until the value of q converges.

$$\begin{aligned}\mathbf{P}_7: \max_{t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_\mathrm{c},f_\mathrm{m},\boldsymbol{\beta},\lambda} & \lambda + \textstyle\sum_{k=1}^K \frac{f_k T}{C_{\mathrm{cpu},k}}\\ & -q E_\mathrm{total}\left(P_\mathrm{t},t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},T,t_\mathrm{c},f_\mathrm{m}\right)\\ \text{s.t.}\ & \mathrm{C1}^\dagger, \mathrm{C2}^\dagger, \mathrm{C3}, \mathrm{C4}, \mathrm{C7}-\mathrm{C10}.\end{aligned}$$

Note that $\mathbf{P}_7$ is still a non-convex optimization problem due to the existence of coupling relationships between multiple variables, i.e., ρ_k and t_k^b, etc. In order to tackle $\mathbf{P}_7$, we introduce the auxiliary variables x_k, y_k, $\phi = f_\mathrm{m} t_\mathrm{c}$ and $\varphi = f_\mathrm{m}^3 t_\mathrm{c}$ into $\mathbf{P}_7$ and transform it into

$$\begin{aligned}\mathbf{P}_8: \max_{t_\mathrm{e},\mathbf{t}^\mathrm{b},\mathbf{t}^\mathrm{a},\mathbf{x},\mathbf{y},\mathbf{f},t_\mathrm{c},f_\mathrm{m},\boldsymbol{\beta},\lambda} & \lambda + \textstyle\sum_{k=1}^K \frac{f_k T}{C_{\mathrm{cpu},k}} - q\Bigg(\left(t_\mathrm{e} + \sum_{k=1}^K t_k^\mathrm{b}\right)\\ & \times \left(P_\mathrm{t} + P_\mathrm{sc} - \eta P_\mathrm{t} \textstyle\sum_{k=1}^K g_k\right) + \eta P_\mathrm{t} \sum_{k=1}^K x_k g_k + \varepsilon_\mathrm{m}\varphi\\ & + \textstyle\sum_{k=1}^K \left(P_\mathrm{c} t_k^\mathrm{b} + y_k + p_\mathrm{c} t_k^\mathrm{a} + \varepsilon_k f_k^3 T\right)\Bigg)\\ \text{s.t.}\ & \mathrm{C1}^{\dagger\dagger}: t_k^\mathrm{b} B \log_2\left(1 + \tfrac{\xi x_k P_\mathrm{t} g_k h_k}{t_k^\mathrm{b} B \sigma^2}\right)\\ & \quad + t_k^\mathrm{a} B \log_2\left(1 + \tfrac{y_k h_k}{t_k^\mathrm{a} B \sigma^2}\right) \geq \beta_k L_{\min,k}, \forall k,\\ & \quad \textstyle\sum_{k=1}^K \beta_k L_{\min,k} \leq \frac{\phi}{C_\mathrm{cpu}},\\ & \quad \tfrac{f_k T}{C_{\mathrm{cpu},k}} \geq (1-\beta_k)\, L_{\min,k}, \forall k,\\ & \mathrm{C2}^{\dagger\dagger}: P_\mathrm{c} t_k^\mathrm{b} + y_k + p_\mathrm{c} t_k^\mathrm{a} + \varepsilon_k f_k^3 T\\ & \quad \leq \eta P_\mathrm{t} g_k \left(t_\mathrm{e} + \textstyle\sum_{i=1}^K t_i^\mathrm{b}\right) - \eta x_k P_\mathrm{t} g_k, \forall k,\\ & \mathrm{C3}^\dagger: 0 \leq \varphi \leq \phi f_\mathrm{max}^2, 0 \leq f_k \leq f_k^\mathrm{max},\ \forall k,\\ & \mathrm{C4}^\dagger: t_\mathrm{e} + \textstyle\sum_{k=1}^K \left(t_k^\mathrm{b} + t_k^\mathrm{a}\right) + \sqrt{\frac{\phi^3}{\varphi}} \leq T, t_\mathrm{e}, t_k^\mathrm{b}, t_k^\mathrm{a} \geq 0, \forall k,\\ & \mathrm{C7}^\dagger: 0 \leq x_k \leq t_k^\mathrm{b}\,, y_k \geq 0, \forall k, \mathrm{C8},\\ & \mathrm{C9}^\dagger: \textstyle\sum_{k=1}^K \Bigg(t_k^\mathrm{b} B \log_2\left(1 + \tfrac{\xi x_k P_\mathrm{t} g_k h_k}{t_k^\mathrm{b} B \sigma^2}\right)\\ & \quad + t_k^\mathrm{a} B \log_2\left(1 + \tfrac{y_k h_k}{t_k^\mathrm{a} B \sigma^2}\right)\Bigg) \geq \lambda,\\ & \mathrm{C10}^\dagger: \tfrac{\phi}{C_\mathrm{cpu}} \geq \lambda,\end{aligned}$$

where $f_\mathrm{m} = \sqrt{\frac{\varphi}{\phi}}$ and $t_\mathrm{c} = \sqrt{\frac{\phi^3}{\varphi}}$.

It is easy to prove that $\mathbf{P}_8$ is convex and can be solved by using existing convex methods (e.g., interior point method, Lagrange duality, etc.) efficiently. The proof is similar to Appendix B and is omitted here for brevity.

In order to gain more insights into the optimal solutions, we use the Lagrange duality method to solve $\mathbf{P}_8$ and obtain the following proposition.

Proposition 4: The system computation EE is maximized when the MEC server uses the maximum allowable time for task computing, i.e., $t_\mathrm{c}^+ = T - \left(t_\mathrm{e}^+ + \sum_{k=1}^K \left(t_k^{\mathrm{b}+} + t_k^{\mathrm{a}+}\right)\right)$.

Proof. Please see Appendix C. ■

Algorithm 1 Dinkelbach-based Iterative Algorithm for $\mathbf{P}_6$

1: Set the maximum error tolerance ϵ;
2: Set the iteration index $l = 1$ and $q = 0$;
3: **repeat**
4: Solve $\mathbf{P}_8$ with a given q, and obtain the optimal solution, denoted by $\left\{t_\mathrm{e}^\diamond, \mathbf{t}^{\mathrm{b}\diamond}, \mathbf{t}^{\mathrm{a}\diamond}, \boldsymbol{\rho}^\diamond, \mathbf{p}^\diamond, \mathbf{f}^\diamond, t_\mathrm{c}^\diamond, f_\mathrm{m}^\diamond, \lambda^\diamond\right\}$;
5: Compute the computation EE of the system as $q^\diamond = \frac{\lambda^\diamond + \sum_{k=1}^K \frac{f_k^\diamond T}{C_{\mathrm{cpu},k}}}{E_\mathrm{total}\left(P_\mathrm{t},t_\mathrm{e}^\diamond,\mathbf{t}^{\mathrm{b}\diamond},\mathbf{t}^{\mathrm{a}\diamond},\boldsymbol{\rho}^\diamond,\mathbf{p}^\diamond,\mathbf{f}^\diamond,T,t_\mathrm{c}^\diamond,f_\mathrm{m}^\diamond\right)}$;
6: **if** $|q^\diamond - q| \leq \epsilon$ **then**
7: Set $q^+ = q$, Flag $= 1$ and the obtained solution is the optimal solution to $\mathbf{P}_6$;
8: **else**
9: Set $q = q^\diamond$, Flag $= 0$ and $l = l + 1$;
10: **end if**
11: **until** Flag $= 1$.

Algorithm 2 Bisection-based Iterative Algorithm for $P_{\min}$

1: Initialize the maximum tolerance ϵ;
2: Set $P_{\rm low}=0$, $P_{\rm up}=P_{\max}$;
3: **repeat**
4: Set $P_{\rm t}=(P_{\rm low}+P_{\rm up})/2$;
5: Solve (20) with a given $P_{\rm t}$, and obtain $\mathcal{A}$;
6: **if** $\mathcal{A}$ is an nonempty set **then**
7: Set $P_{\rm up}=P_{\rm t}$;
8: **else**
9: Set $P_{\rm low}=P_{\rm t}$;
10: **end if**
11: **until** $P_{\rm up}-P_{\rm low}<\epsilon$;
12: $P_{\min}$ is given by $\frac{P_{\rm up}+P_{\rm low}}{2}$.

2) Optimal solution to $\mathbf{P}_5$*:* After $\mathbf{P}_6$ is solved in its feasible range $[P_{\min},P_{\max}]$ via Algorithm 1, the optimal solution to $\mathbf{P}_5$ can be obtained by identifying the $\mathbf{P}_6$ solution that returns the largest objective function value of $\mathbf{P}_5$ among all the solutions obtained by Algorithm 1 together with the corresponding value of P_t. This indicates that limiting the search scope of $P_{\rm t}$ can reduce the complexity of solving $\mathbf{P}_5$. Due to the energy-causality and QoS constraints in (13)-(16), there exists a lower bound of $P_{\rm t}$, denoted by $P_{\min}$, which can be determined by minimizing $P_{\rm t}$ while satisfying all the constraints of $\mathbf{P}_5$. The obtained $P_{\min}$ can be used to limit the search scope of $P_{\rm t}$. Let $[P_{\min},P_{\max}]$ denote the range of $P_{\rm t}$ that makes $\mathbf{P}_5$ feasible, and $P_{\min}$ can be obtained by solving

$$\begin{aligned}\mathbf{P}_9:\ &\min_{P_{\rm t},t_{\rm e},\mathbf{t}^{\rm b},\mathbf{t}^{\rm a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_{\rm c},f_{\rm m},\boldsymbol{\beta}} P_{\rm t}\\ &{\rm s.t.C1}^{\dagger\dagger},{\rm C2}^{\dagger\dagger},{\rm C3}^{\dagger},{\rm C4}^{\dagger},{\rm C6},{\rm C7}^{\dagger},{\rm C8}.\end{aligned}$$

$\mathbf{P}_9$ is a non-convex problem due to the non-convex constraints ${\rm C1}^{\dagger\dagger}$ and ${\rm C2}^{\dagger\dagger}$, where each IoT node has unique ρ_k and $t_k^{\rm b}$ but they share the same $P_{\rm t}$. Fortunately, we find that the objective function of $\mathbf{P}_9$ is a monotonous function of $P_{\rm t}$ and when $P_{\rm t}$ is fixed, $\mathbf{P}_9$ reduces to

$$\begin{aligned}&{\rm find}\ \mathcal{A}=\left\{t_{\rm e},\mathbf{t}^{\rm b},\mathbf{t}^{\rm a},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_{\rm c},f_{\rm m},\boldsymbol{\beta}\right\}\\ &{\rm s.t.C1}^{\dagger\dagger},{\rm C2}^{\dagger\dagger},{\rm C3}^{\dagger},{\rm C4}^{\dagger},{\rm C7}^{\dagger},{\rm C8},\end{aligned}\tag{20}$$

which is convex and can be solved using the existing convex tools. Based on (20), the minimum $P_{\rm t}$ that satisfies all the constraints of $\mathbf{P}_9$ can be obtained by the bisection method. Hence, we propose a bisection-based iterative algorithm in Algorithm 2 to search for $P_{\min}$. For a certain error tolerance ϵ, the bisection method in Algorithm 2 is guaranteed to converge.

Based on Algorithm 1 and Algorithm 2, a two-layer iterative algorithm is devised in Algorithm 3 to obtain the optimal solution to $\mathbf{P}_5$. In the outer loop (lines 7, 9 and 10), the value of $P_{\rm t}$ increases by a sufficiently small step size ϵ_0 from $P_{\min}$ to $P_{\max}$ and $\mathbf{P}_5$'s objective function values obtained from the current and previous runs of the inner loop are compared to identify the largest objective function value (and the corresponding solution to $\mathbf{P}_6$ and $P_{\rm t}$ value); while in the inner loop (line 8), the optimal solution of $\mathbf{P}_6$ for a given $P_{\rm t}$ is obtained.

Remark 1: It is worth noting that our proposed two-layer iterative algorithm in Algorithm 3 can serve the following two purposes. Firstly, compared with the exhaustive search method, Algorithm 3 enjoys a much lower complexity while achieving the optimal resource allocation because the optimal local computation time of each IoT node has been analytically derived and Algorithm 2 narrows the searching range for $P_{\rm t}$ by identifying its lower bound $P_{\min}$. Secondly, in addition to the considered hybrid BackCom-AT and partial computation offloading, Algorithm 3 can also be used to achieve the optimal resource allocation under the complete offloading mode (i.e., all the IoT nodes offload all their task bits to the MEC server via hybrid BackCom-AT) or the pure BackCom mode, (i.e., all the IoT nodes only use BackCom to offload their task bits) by letting $f_k=0,\forall k$ or $p_k=0,t_k^{\rm a}=0,\forall k$, respectively.

Algorithm 3 Two-layer Iterative Algorithm for $\mathbf{P}_5$

1: Initialize a sufficiently small step size ϵ_0;
2: Obtain $P_{\min}$ by using Algorithm 2;
3: Set $P_{\rm t}=P_{\min}$;
4: Solve $\mathbf{P}_6$ with a given $P_{\rm t}$ via Algorithm 1;
5: Compute the computation EE of the system q^* and obtain the corresponding solution, denoted by $\left\{P_{\rm t},t_{\rm e}^*,\mathbf{t}^{\rm b*},\mathbf{t}^{\rm a*},\boldsymbol{\rho}^*,\mathbf{p}^*,\mathbf{f}^*,t_{\rm c}^*,f_{\rm m}^*,\lambda^*\right\}$;
6: **repeat**
7: Set $P_{\rm t}=P_{\rm t}+\epsilon_0$;
8: Solve $\mathbf{P}_6$ with a given $P_{\rm t}$ to obtain the system computation EE q^+ and the corresponding solution using Algorithm 1;
9: If $q^+>q^*$
10: Set $q^*=q^+$ and update the optimal solution;
11: End
12: **until** $P_{\rm t}>P_{\max}$.

Note that in the fully local computing mode or the pure AT mode where all the IoT nodes perform local computing only or only use AT to offload their task bits, respectively, the proposed Dinkelbach-based iterative algorithm in Algorithm 1 can be applicable by letting making a few variable substitutions because in these two modes, $P_{\rm t}$ is only coupled with $t_{\rm e}$.

3) A reduced-complexity iterative algorithm for solving $\mathbf{P}_5$*:* The computational complexity of Algorithm 3 may be high for some special cases, e.g., the case with a large value of $P_{\max}-P_{\min}$, because the number of feasible values of $P_{\rm t}$ and thus the times of calling Algorithm 1 can be vary large. Instead, we devise a suboptimal, reduced-complexity iterative algorithm to solve $\mathbf{P}_5$ via BCD.

When $P_{\rm t}$ is fixed, $\mathbf{P}_5$ reduces to $\mathbf{P}_6$ and Algorithm 1 can be used to achieve the optimal solution to $\mathbf{P}_6$, which can be solved optimally by Algorithm 1.

When all the other optimization variables except for $P_{\rm t}$ are given, $\mathbf{P}_5$ reduces to

$$\begin{aligned}\mathbf{P}_{10}:\ &\max_{P_{\rm t}}\frac{\min\left\{\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2(1+C_kP_{\rm t})+R_k^{\rm a}\right),R_{\rm m}\right\}+D_0}{P_{\rm t}A_0+B_0}\\ &{\rm s.t.C1}^{\dagger},{\rm C2}^{\dagger},{\rm C6},\end{aligned}$$

where $A_0=t_{\rm e}+\sum_{k=1}^{K}t_k^{\rm b}-\eta\sum_{k=1}^{K}\left(g_k\left(t_{\rm e}+\sum_{i=1}^{K}t_i^{\rm b}\right)-t_k^{\rm b}\rho_kg_k\right)$, $B_0=$

$\sum_{k=1}^{K}\left(P_{\rm c}t_k^{\rm b}+\left(p_k+p_{\rm c}\right)t_k^{\rm a}+\varepsilon_k f_k^3\tau_k\right)+E_{\rm m}^{\rm e}+P_{\rm sc}\left(t_{\rm e}+\sum_{k=1}^{K}t_k^{\rm b}\right)$, $C_k=\frac{\xi\rho_k g_k h_k}{B\sigma^2}$ and $D_0=\sum_{k=1}^{K}R_{\rm m}^{\rm e}$.

Combining $\mathrm{C1}^{\dagger}, \mathrm{C2}^{\dagger}$ and C6, we can obtain the range of $P_{\rm t}$ as $P_{\rm L}\le P_{\rm t}\le P_{\max}$, where

$$P_{\rm L}=\max\left\{0,\frac{2^{\frac{\beta_k L_{\min,k}-R_k^{\rm a}}{t_k^{\rm b}B}}-1}{C_k},\right.$$
$$\left.\frac{P_{\rm c}t_k^{\rm b}+\left(p_k+p_{\rm c}\right)t_k^{\rm a}+\varepsilon_k f_k^3 T}{\eta g_k\left(t_{\rm e}+\sum_{i=1}^{K}t_i^{\rm b}\right)-t_k^{\rm b}\eta\rho_k g_k},\forall k\right\}. \quad (21)$$

In order to handle the *min* function in the objective function of $\mathbf{P}_{10}$, we consider the following two cases, which are **Case I**: $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)\ge R_{\rm m}$, and **Case II**: $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)\le R_{\rm m}$.

In Case I, $\mathbf{P}_{10}$ can be rewritten as

$$\mathbf{P}_{11}:\ \max_{P_{\rm t}}\ \frac{R_{\rm m}+D_0}{P_{\rm t}A_0+B_0}$$
$$\text{s.t.}\ \max\left(0,P_{\rm L},P_{\rm L2}\right)\le P_{\rm t}\le P_{\max},$$

where $P_{\rm L2}$ is the unique solution[5] to $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)=R_{\rm m}$ with respect to $P_{\rm t}$. Since the objective function is a monotonically decreasing function of $P_{\rm t}$, the optimal solution to $\mathbf{P}_{11}$ is given by $\max\left(0,P_{\rm L},P_{\rm L2}\right)$.

In Case II, $\mathbf{P}_{10}$ can be rewritten as

$$\mathbf{P}_{12}:\ \max_{P_{\rm t}}\ \frac{\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)+D_0}{P_{\rm t}A_0+B_0}$$
$$\text{s.t.}\ \max\left(0,P_{\rm L}\right)\le P_{\rm t}\le\min\left(P_{\rm L2},P_{\max}\right).$$

Proposition 5: Let $P_{\rm t}^{\star}$ denote the optimal solution to $\mathbf{P}_{12}$. $P_{\rm t}^{\star}$ can be computed as

$$P_{\rm t}^{\star}=\begin{cases}\min\left(P_{\rm L2},P_{\max}\right),\text{if}\ \sum_{k=1}^{K}t_k^{\rm b}Bf_k\left(\min\left(P_{\rm L2},P_{\max}\right)\right)\\ \qquad\qquad\ge A_0\left(\sum_{k=1}^{K}R_k^{\rm a}+D_0\right),\\ \max\left(0,P_{\rm L}\right),\text{if}\ \sum_{k=1}^{K}t_k^{\rm b}Bf_k\left(\max\left(0,P_{\rm L}\right)\right)\\ \qquad\qquad\le A_0\left(\sum_{k=1}^{K}R_k^{\rm a}+D_0\right),\\ P_0,\text{otherwise},\end{cases} \quad (22)$$

where $f_k\left(P_{\rm t}\right)=\frac{C_k\left(P_{\rm t}A_0+B_0\right)}{\left(1+C_kP_{\rm t}\right)\ln 2}-A_0\sum_{k=1}^{K}\log_2\left(1+C_kP_{\rm t}\right)$ and P_0 is the unique solution to $\sum_{k=1}^{K}t_k^{\rm b}Bf_k\left(P_{\rm t}\right)-A_0\left(\sum_{k=1}^{K}R_k^{\rm a}+D_0\right)=0$.

Proof. Please see Appendix D. ■

Combining the above two cases, the optimal solution to $\mathbf{P}_{10}$ is given by $\max\left(0,P_{\rm L},P_{\rm L2}\right)$ or $P_{\rm t}^{\star}$ depending on which of them achieves a higher system computation EE.

By exploiting the BCD technique based on Algorithm 1 and the above obtained optimal solution to $\mathbf{P}_{10}$, we propose a reduced-complexity iterative algorithm for solving $\mathbf{P}_5$ is shown in Algorithm 4.

Note that in Algorithm 4, the value of $P_{\rm L2}$ in the ii-th iteration is determined by the PB's transmit power obtained in the ii-th iteration, namely $P_{\rm t}^{ii}$. The reasons are as follows. On the one hand, the following Lemma indicates

[5]In line 7 of Algorithm 4, we provide a way to determine $P_{\rm L2}$ and the detailed reasons are provided in the last three paragraphs of this subsection.

Algorithm 4 Reduced-complexity Iterative Algorithm for $\mathbf{P}_5$

1: Initialize the maximum iterations $I_{\max}$ and the maximum tolerance ϵ;
2: Obtain $P_{\min}$ by using Algorithm 2;
3: Set $P_{\rm t}^1=\frac{P_{\min}+P_{\max}}{2}$ and iteration index $ii=1$;
4: **repeat**
5: Solve $\mathbf{P}_6$ for a given $P_{\rm t}^{ii}$ via Algorithm 1 and obtain $\left\{t_{\rm e}^{ii+1},\left\{t_k^{{\rm b}(ii+1)}\right\}_{k=1}^{K},\left\{t_k^{{\rm a}(ii+1)}\right\}_{k=1}^{K},\left\{\rho_k^{ii+1}\right\}_{k=1}^{K},t_{\rm c}^{ii+1},f_{\rm m}^{ii+1},\left\{p_k^{ii+1}\right\}_{k=1}^{K},\left\{f_k^{ii+1}\right\}_{k=1}^{K}\right\}$;
6: Compute the corresponding computation EE, denoted by q_1^{ii+1};
7: Compute $P_{\rm L}$ based on (21) and set $P_{\rm L2}=P_{\rm t}^{ii}$;
8: Obtain P_0 by means of the bisection method and determine $P_{\rm t}^{\star}$ based on Proposition 5;
9: **if** $\frac{R_{\rm m}+D_0}{\max(0,P_{\rm L},P_{\rm L2})A_0+B_0}\ge\frac{\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2(1+C_kP_{\rm t}^{\star})+R_k^{\rm a}\right)+D_0}{P_{\rm t}^{\star}A_0+B_0}$ **then**
10: Set $P_{\rm t}^{ii+1}=\max\left(0,P_{\rm L},P_{\rm L2}\right)$ and $q_2^{ii+1}=\frac{R_{\rm m}+D_0}{\max(0,P_{\rm L},P_{\rm L2})A_0+B_0}$;
11: **else**
12: Set $P_{\rm t}^{ii+1}=P_{\rm t}^{\star}$ and $q_2^{ii+1}=\frac{\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2(1+C_kP_{\rm t}^{\star})+R_k^{\rm a}\right)+D_0}{P_{\rm t}^{\star}A_0+B_0}$;
13: **end if**
14: **if** $|q_2^{ii+1}-q_1^{ii+1}|<\epsilon$ **then**
15: Set $P_{\rm t}^{*}=P_{\rm t}^{ii+1},t_{\rm e}^{*}=t_{\rm e}^{ii+1},t_{\rm c}^{*}=t_{\rm c}^{ii+1},f_{\rm m}^{*}=f_{\rm m}^{ii+1},p_k^{*}=p_k^{ii+1},f_k^{*}=f_k^{ii+1},\rho_k^{*}=\rho_k^{ii+1},t_k^{{\rm b}*}=t_k^{{\rm b}(ii+1)},t_k^{{\rm a}*}=t_k^{{\rm a}(ii+1)},\forall k$, Flag $=1$ and return;
16: **else**
17: Set $ii=ii+1$ and Flag $=0$;
18: **end if**
19: **until** Flag $=1$ or $ii=I_{\max}$

that by using the obtained solution of $\mathbf{P}_6$ in each iteration, the given PB's transmit power can make the equation $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)=R_{\rm m}$ hold.

Lemma 2. When $\mathbf{P}_6$ is optimally solved to a given $P_{\rm t}$, the total task bits offloaded by all the IoT nodes should equal the MEC server's computation capacity during the task execution phase, i.e., $\sum_{k=1}^{K}\left(t_k^{\rm b+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\rm t}g_kh_k}{B\sigma^2}\right)+t_k^{\rm a+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)=\frac{t_{\rm c}^{+}f_{\rm m}^{+}}{C_{\rm cpu}}$, and $P_{\rm L2}$ is equal the given $P_{\rm t}$.

Proof. Please see Appendix E. ■

On the other hand, according to the definition of $P_{\rm L2}$ below $\mathbf{P}_{11}$, $P_{\rm L2}$ is also a solution for the equation $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)=R_{\rm m}$ with respect to $P_{\rm t}$. Substituting the obtained solution of $\mathbf{P}_6$ into the equation $\sum_{k=1}^{K}\left(t_k^{\rm b}B\log_2\left(1+C_kP_{\rm t}\right)+R_k^{\rm a}\right)=R_{\rm m}$, we know that the left side of this equation is an increasing function with respect to $P_{\rm t}$ and $R_{\rm m}$ is a constant. As only one solution exists for the above equation, $P_{\rm L2}$ equals the given $P_{\rm t}$.

V. Numerical Results

In this section, we verify the effectiveness and the superiority of the proposed schemes via computer simulations.

Tabel 1: Key Simulation Settings

Parameters	Notation	Value
The entire time block	T	1 Second
The communication bandwidth	B	100 kHz
The number of cycles for one bit at the k-th IoT node	$C_{\mathrm{cpu},k}$	1000 Cycles/bit
The number of cycles for one bit at the MEC server	C_{cpu}	1000 Cycles/bit
The constant circuit power consumption for BackCom at the k-th IoT node	P_{c}	10 μW
The constant circuit power consumption for AT at the k-th IoT node	p_{c}	1 mW
The constant circuit power consumption at the PB	P_{sc}	10 mW
The maximum transmit power at the PB	$P_{\max}$	1 W
The fixed energy conversion efficiency at each IoT node	η	0.7
The performance gap reflecting the real modulation for BackCom	ξ	-15dB
The noise power spectral density	σ^2	-120dBm/Hz
The number of IoT nodes	K	4
The effective capacitance coefficient of the k-th IoT node	ε_k	10^{-26}
The effective capacitance coefficient of the MEC server	ε_{m}	10^{-28}
The maximum CPU frequency at the k-th IoT node	$f_k^{\max}$	5×10^8 Hz
The maximum CPU frequency at the MEC server	$f_{\max}$	10^{10} Hz
The minimum computation bits of the k-th IoT node	$L_{\min,k}$	10 kbits

Unless otherwise specified, the basic simulation parameters are provided as shown in Table I [5], [11], as shown at the top of the next page. We consider a standard channel fading model, where the channel gain between the PB and the k-th IoT node is modeled by $g_k = g_k' d_{0k}^{-\alpha}$ with the small-scale fading g_k', distance d_{0k}, and path loss exponent α. The channel gain from the MEC server to the k-th IoT node is given by $h_k = h_k' d_{1k}^{-\alpha}$ with the small-scale fading h_k' and distance d_{1k}. Here we let $\alpha = 3$, $d_{01} = 12$m, $d_{02} = 10$m, $d_{03} = 15$m, $d_{04} = 13$m, $d_{11} = 60$m, $d_{12} = 65$m, $d_{13} = 50$m and $d_{14} = 55$m.

In order to illustrate the advantages of the proposed computation bits/EE maximization scheme, we compare the performance under the proposed schemes with the following four benchmark schemes:

- **Complete offloading scheme**: All the tasks at all the IoT nodes can only be offloaded to the MEC server for computation and each IoT node can choose BackCom, AT, or hybrid BackCom and AT to transmit its task bits. For maximizing the total computation bits of all the IoT nodes, this scheme is obtained by solving $\mathbf{P}_3$ with $f_k = 0$, while for maximizing the system computation EE, this scheme is achieved based on Remark 1.
- **Fully local computing scheme**: All the IoT nodes can only compute their tasks locally. For maximizing the total computation bits of all the IoT nodes, this scheme is optimized under the same constraints as $\mathbf{P}_3$ with $t_k^{\mathrm{b}} = 0$, $t_k^{\mathrm{a}} = 0$, $x_k = 0$, $y_k = 0$, and $t_{\mathrm{c}} = 0$. This scheme with the aim of maximizing the system computation EE is obtained by means of the proposed Dinkelbach-based iterative algorithm through a few changes, i.e., letting $P_{\mathrm{t}} = \frac{x}{t_{\mathrm{e}}}$.
- **Pure BackCom assisted MEC networks**: This scheme is optimized for pure BackCom assisted MEC networks, where each IoT node can offload its partial task bits to the MEC server for computation and the offloaded task bits can only be transmitted via BackCom. Likewise, for maximizing the total computation bits of all the IoT nodes, this scheme is achieved by solving $\mathbf{P}_3$ with $t_k^{\mathrm{a}} = 0$ and $y_k = 0$. Remark 1 is used to obtain the scheme for maximizing the system computation EE in pure BackCom assisted MEC networks.

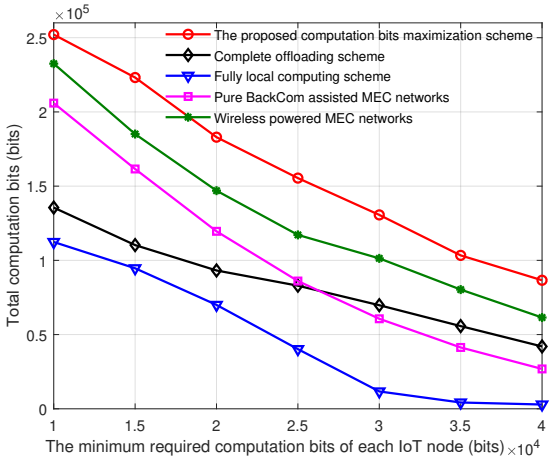

Fig. 2. R_{total} versus the minimum required computation bits.

- **Wireless powered MEC networks**: This scheme is optimized for the wireless powered MEC networks, where each IoT node harvests energy from RF signals first and then uses the harvested energy to offload its partial task bits to the MEC server for computation. Accordingly, this scheme for maximizing the total computation bits of all the IoT nodes is optimized by solving $\mathbf{P}_3$ with $t_k^{\mathrm{b}} = 0$ and $x_k = 0$, while this scheme for maximizing the system computation EE is achieved by using the proposed Dinkelbach-based iterative algorithm with $P_{\mathrm{t}} = \frac{x}{t_{\mathrm{e}}}$.

A. Performance analysis for the proposed computation bits maximization scheme

Fig. 2 demonstrates the total computation bits of all the IoT nodes R_{total} versus the minimum required computation bits at each IoT node. For convenience, we let $L_{\min,1} = L_{\min,2} = L_{\min,3} = L_{\min,4} = L_{\min}$ and $L_{\min}$ is ranged from 10 kbits to 40 kbits. In order to illustrate the superiority of the proposed computation bits maximization scheme in terms of R_{total}, we compare R_{total} achieved by the proposed scheme with those obtained by the complete offloading scheme, the fully local computing scheme, pure BackCom assisted MEC networks and wireless powered MEC networks. From this figure, we

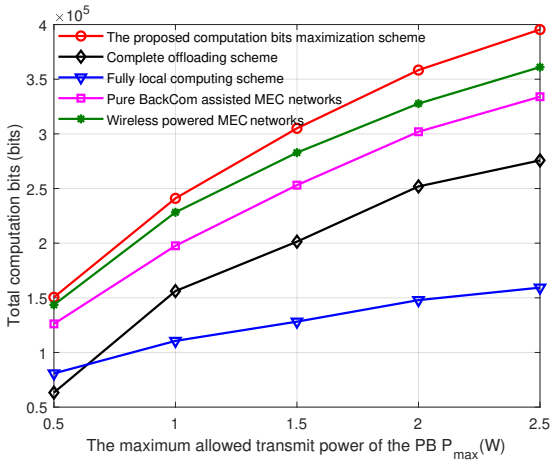

Fig. 3. Total computation bits of all the IoT nodes $R_{\rm total}$ versus $P_{\max}$.

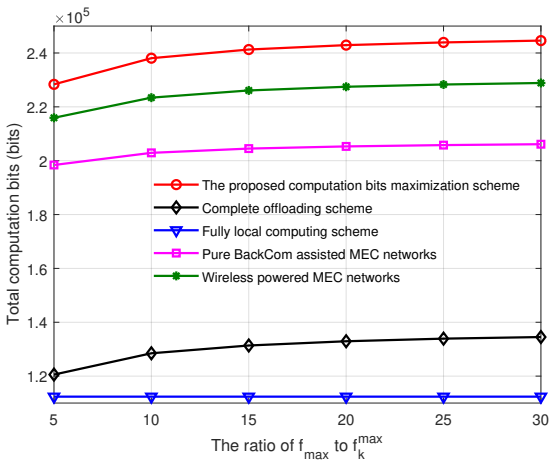

Fig. 4. Total computation bits v.s. the ratio of $f_{\max}$ to $f_k^{\max}$.

can observe that $R_{\rm total}$ under all the schemes decreases with the increase of $L_{\min}$. This is because the increase of $L_{\min}$ denotes a higher computation bits requirement for each IoT node and more resources need to be allocated to the IoT nodes with worse channels, leading to a reduction to the total computation bits. Besides, we can find that the proposed computation bits maximization scheme always outperforms pure BackCom assisted MEC networks and wireless powered MEC networks in terms of the total computation bits, which verifies the advantages of the combination of BackCom and the wireless powered MEC networks. By comparisons, we can also see that the proposed computation bits maximization scheme can achieve the highest $R_{\rm total}$ since the other schemes can be regarded as special cases for the proposed scheme and have less flexibility to utilize resources for maximizing the total computation bits.

Fig. 3 shows the total computation bits of all the IoT nodes versus the maximum allowed transmit power at the PB, $P_{\max}$, under different schemes. Here $P_{\max}$ varies from 0.5 W to 2.5 W. It can be observed that the total computation bits of all the IoT nodes under all the schemes increases with the increase of $P_{\max}$. The reasons are as follows. Based on Proposition 1, the optimal transmit power of the PB is determined by $P_{\max}$ and a higher transmit power allows IoT nodes to harvest more energy for task offloading and computing. Besides, a higher transmit power also improves the power of the backscattered signal, increasing the task bits offloaded at each IoT node via BackCom. Likewise, the advantages of the combination of BackCom and the wireless powered MEC networks are also illustrated by comparing the proposed scheme with pure BackCom assisted MEC networks and wireless powered MEC networks. Besides, we also observe that the total computation bits under the proposed scheme are the highest among these five schemes, which also demonstrates the superiority of the proposed scheme in terms of total computation bits. The IoT nodes choose to perform task offloading only when the harvested energy (or the transmit power of the PB) is large enough; otherwise, IoT nodes prefer to compute locally for achieving more computation bits.

Fig. 4 depicts the impact of the computation capacity of the MEC server on the total computation bits of all the IoT nodes, where the ratio of the maximum computation frequency of the MEC server $f_{\max}$ to that of the k-th IoT node $f_k^{\max}$ varies from 5 to 30 and $f_k^{\max}$ is fixed as 500 MHz. It can be observed that the total computation bits under all the schemes except for the fully local computing scheme increase when the MEC server's computation capacity improves, while the total computation bits under the fully local computing scheme basically keeps unchanged. This is because a higher computation capacity of the MEC server allows more task bits to be offloaded and computed, increasing the total computation bits. Whereas the MEC server's computation capacity does not influence the total computation bits under the fully local computing scheme. It can also be seen that the proposed computation bits maximization scheme is superior to the other schemes in terms of total computation bits for all the considered values of $\frac{f_{\max}}{f_k^{\max}}$.

B. Convergence analysis for the proposed iterative algorithms

Fig. 5 demonstrates the convergence analysis of the proposed iterative algorithms, such as, Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. Specifically, Fig. 5(a) shows the convergence of Algorithm 1 under different settings of $P_{\rm t}$ and $L_{\min}$, the convergence of Algorithm 2 with different values of $L_{\min}$ is illustrated in Fig. 5(b) and in Fig. 5(c), the convergence analysis and performance comparisons between Algorithm 3 and Algorithm 4 are plotted. From Fig. 5(a), it can be observed that less than 4 iterations are required for the proposed Dinkelbach-based iterative algorithm in Algorithm 1 to converge to the maximum system computation EE, which illustrates that the proposed Dinkelbach-based iterative algorithm is computationally efficient. The proposed bisection-based iterative algorithm in Algorithm 2 is used to obtain the maximum value of $P_{\rm t}$, namely $P_{\min}$, and it can be observed from Fig. 5(b) that it takes about 9 iterations to achieve the value of $P_{\min}$, which verifies the convergence of the proposed bisection-based iterative algorithm. Besides, we can also observe that $P_{\min}$ increases when $L_{\min}$ increases since with a larger $L_{\min}$, the energy consumption at each IoT node

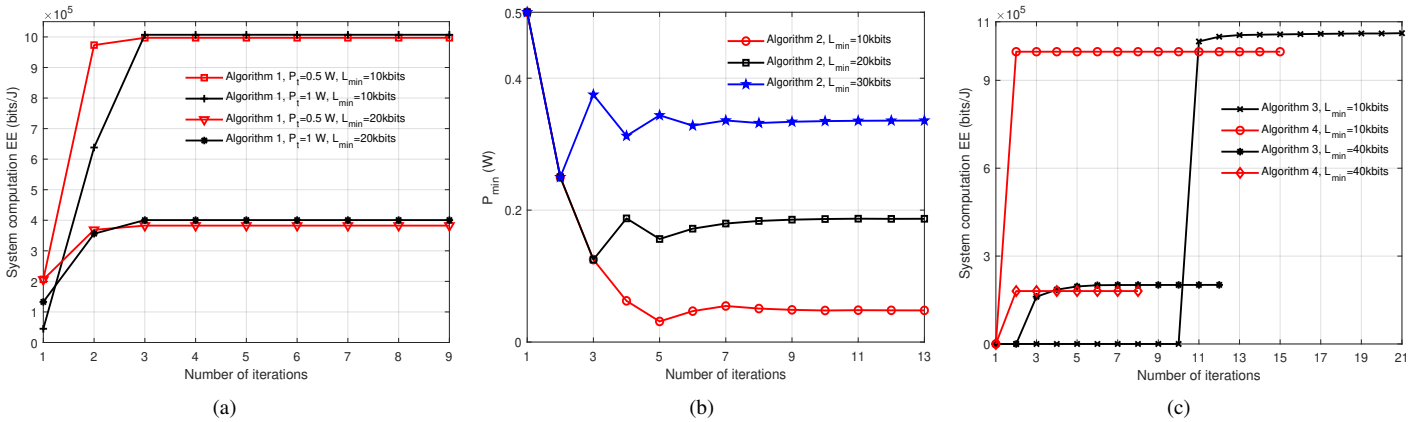

Fig. 5. The convergence of the proposed iterative algorithms: (a) the convergence of Algorithm 1; (b) the convergence of Algorithm 2; (c) the convergence analysis and comparisons between Algorithm 3 and Algorithm 4.

for task offloading and computing may increases, leading to an improvement for $P_{\rm t}$.

From Fig. 5(c), we can observe that both the proposed two-layer iterative algorithm in Algorithm 3 and the proposed reduced-complexity iterative algorithm in Algorithm 4 are convergent. Specifically, the number of iterations required for the convergence of the proposed reduced-complexity iterative algorithm is much less than that of the proposed two-layer iterative algorithm, illustrating the low complexity of the proposed reduced-complexity iterative algorithm. By comparisons, we can also see that the system computation EE achieved by the proposed reduced-complexity iterative algorithm is always close to that of the proposed two-layer iterative algorithm, which further verifies the suboptimal performance of the proposed reduced-complexity iterative algorithm in terms of system computation EE. That is, the low complexity of the proposed reduced-complexity iterative algorithm is achieved at the cost of a slight reduction to the system computation EE. Moreover, we also find that the required iterations for the proposed two-layer iterative algorithm under the set of $L_{\min} = 40$kbits are less than those under the set of $L_{\min} = 10$kbits. This is because the improvement of $L_{\min}$ increases the value of $P_{\min}$, resulting in a narrow searching range of $P_{\rm t}$.

C. Performance analysis for the proposed system computation EE maximization schemes

Fig. 6 shows the system computation EE under different schemes versus $L_{\min}$. Here we consider two schemes, which are the proposed two-layer iterative algorithm in Algorithm 3 and the proposed reduced-complexity iterative algorithm shown in Algorithm 4, to maximize the system computation EE. Besides, for better comparisons, we also include the performance under the proposed computation bits maximization scheme. As shown in this figure, it can be observed that the system computation EE under all the schemes decreases when $L_{\min}$ increases. The reasons are as follows. With the increase of $L_{\min}$, the energy consumption at each IoT node for task offloading and computing may increase. Besides, the total computation bits offloaded by all the IoT nodes may also increase, leading to an improvement for the energy consumption of the MEC server. That is, the total system energy

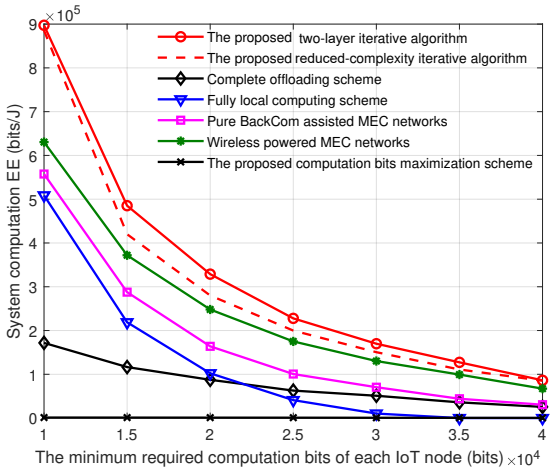

Fig. 6. System computation EE versus the minimum required computation bits at each IoT node.

consumption increases with $L_{\min}$. Since the energy consumption grows faster than the growth of the computation bits, the system computation EE decreases with the increase of $L_{\min}$. By comparisons, we also find that the system computation EE under the proposed reduced-complexity iterative algorithm is close to that under the proposed two-layer iterative algorithm and always outperforms other schemes, namely, the complete offloading scheme, the fully local computing scheme, pure BackCom assisted MEC networks, and wireless powered MEC networks, verifying the suboptimal performance of Algorithm 4 and the superiority of the combination of BackCom and the wireless powered MEC networks. Moreover, it can be observed that the proposed computation bits maximization scheme achieves the worst performance among these schemes in terms of system computation EE, showing that the proposed computation bits maximization scheme has compromise on energy-efficiency performance.

Fig. 7 shows the effect of the system computation EE on $P_{\max}$ under different schemes. It can be observed that the system computation EE under all the schemes increases with the increase of $P_{\max}$ when $P_{\max}$ is small and when $P_{\max}$ is large enough, the system computation EE converges. This is

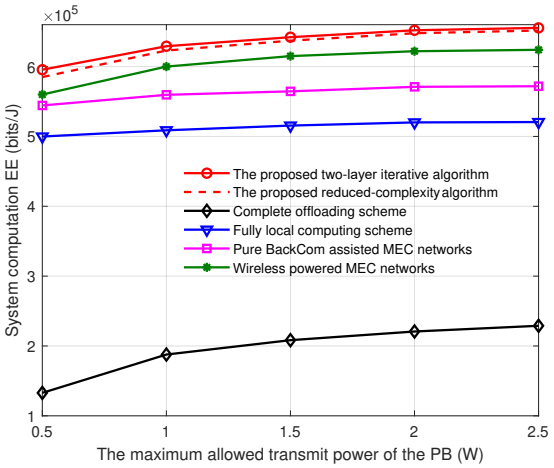

Fig. 7. System computation EE versus the maximum allowed transmit power at the PB.

because when $P_{\max}$ is small, the optimal transmit power of the PB is determined by $P_{\max}$, while when $P_{\max}$ is large enough, the optimal transmit power of the PB is fixed and its value is independent of $P_{\max}$. By comparisons, we can also see that the proposed reduced-complexity iterative algorithm can achieve the suboptimal performance in terms of system computation EE and the proposed system computation EE maximization scheme always outperforms the other schemes.

VI. Conclusions

In this paper, we studied the total computation bits and system computation EE maximizations for a hybrid BackCom-AT WPMEC network, subject to the MEC server's computation capacity constraint, the energy causality and QoS constraints of each IoT node. The total computation bits maximization problem was transformed into a convex one and solved by the CVX, while the system computation EE maximization problem was optimally solved by our proposed two-layer iterative algorithm. Subsequently, we devised a reduced-complexity iterative algorithm to obtain a suboptimal solution to the system computation EE maximization problem. Simulations results validated the superiority of the proposed total-computation-bits maximization scheme and system-computation-EE maximization algorithms over several baseline schemes. Besides, we obtained the following two insights. First, when the MEC server's computation capacity is finite, the complete offloading scheme is not the optimal choice for maximizing the total computation bits or system computation EE. Second, the total computation bits are always maximized when the MEC server adopts the maximum allowed computing frequency, which, however, does not maximize the system computation EE.

Appendix A
Proof of Proposition 1

Here we prove Proposition 1 by means of contradiction.

A. Proofs for $P_{\rm t}^*=P_{\max}$ and $f_{\rm m}^*=f_{\max}$

Assume that $\left\{P_{\rm t}^*, t_{\rm e}^*, \{t_k^{\rm b*}\}_{k=1}^K, \{t_k^{\rm a*}\}_{k=1}^K, \{\rho_k^*\}_{k=1}^K, t_{\rm c}^*, f_{\rm m}^*, \{p_k^*\}_{k=1}^K, \{f_k^*\}_{k=1}^K, \{\tau_k^*\}_{k=1}^K, \lambda^*\right\}$ is the optimal solution to $\mathbf{P_1}$, where $P_{\rm t}^*<P_{\max}$ and $\lambda^*=\min\Big\{\sum_{k=1}^K\Big(t_k^{\rm b*}B\log_2\Big(1+\frac{\xi\rho_k^*P_{\rm t}^*g_kh_k}{B\sigma^2}\Big)+t_k^{\rm a*}B\log_2\Big(1+\frac{p_k^*h_k}{B\sigma^2}\Big)\Big), \frac{t_{\rm c}^*f_{\rm m}^*}{C_{\rm cpu}}\Big\}$. Accordingly, the maximum number of the computation bits of the considered system $R_{\rm total}^*$ is given by $\lambda^*+\sum_{k=1}^K\frac{\tau_k^*f_k^*}{C_{{\rm cpu},k}}$. Then we can construct another solution satisfying $P_{\rm t}^+=P_{\max}>P_{\rm t}^*, t_{\rm e}^+=t_{\rm e}^*, t_{\rm c}^+=t_{\rm c}^*, f_{\rm m}^+=f_{\rm m}^*, t_k^{\rm b+}=t_k^{\rm b*}, t_k^{\rm a+}=t_k^{\rm a*}, \rho_k^+=\rho_k^*, p_k^+=p_k^*, f_k^+=f_k^*, \tau_k^+=\tau_k^*$ and $\lambda^+=\min\Big\{\sum_{k=1}^K\Big(t_k^{\rm b+}B\log_2\Big(1+\frac{\xi\rho_k^+P_{\rm t}^+g_kh_k}{B\sigma^2}\Big)+t_k^{\rm a+}B\log_2\Big(1+\frac{p_k^+h_k}{B\sigma^2}\Big)\Big), \frac{t_{\rm c}^+f_{\rm m}^+}{C_{\rm cpu}}\Big\}$. It is quite evident that the constructed solution is a feasible solution which satisfies all the constraints of $\mathbf{P_1}$. Let $R_{\rm total}^+$ be the achievable computation bits under the constructed solution. Then we have $R_{\rm total}^+=\lambda^++\sum_{k=1}^K\frac{\tau_k^+f_k^+}{C_{{\rm cpu},k}}\geq R_{\rm total}^*$ since $\lambda^+\geq\lambda^*$, which contradicts the above assumption. Thus, $P_{\rm t}^*=P_{\max}$ holds when the maximum computation bits of the considered system are achieved. The same method can also be applied to prove that $f_{\rm m}^*=f_{\max}$ is satisfied for maximizing the total computation bits by making only a few changes and the detailed process is omitted here for brevity.

B. Proof for $\tau_k^*=T$

When $P_{\rm t}, t_{\rm e}, \mathbf{t}^{\rm b}, \mathbf{t}^{\rm a}, \boldsymbol{\rho}, \mathbf{p}, t_{\rm c}, f_{\rm m}, \lambda$ and $\{f_i,\tau_i\}_{i=\{1,2,\ldots,K\}\backslash k}$ are fixed, we jointly optimize f_k and τ_k to maximize the total achievable computation bits of the system. Assume that the optimal computation frequency f_k^* and time $\tau_k^*<T$ satisfy all the constraints of of $\mathbf{P_1}$ with other parameters fixed. Then the maximum computation bits of the considered system $R_{\rm total}^*$ is given by $\lambda+\sum_{i=1,i\neq k}^K\frac{\tau_if_i}{C_{{\rm cpu},k}}+\frac{\tau_k^*f_k^*}{C_{{\rm cpu},k}}$. Another feasible solution can be constructed as $\{f_k^+,\tau_k^+\}$ with $\tau_k^+=T$ and $\tau_k^+f_k^+(f_k^+)^2=\tau_k^*f_k^*(f_k^*)^2$. Accordingly, the achievable computation bits under the constructed solution are given by $R_{\rm total}^+=\lambda+\sum_{i=1,i\neq k}^K\frac{\tau_if_i}{C_{{\rm cpu},k}}+\frac{\tau_k^+f_k^+}{C_{{\rm cpu},k}}$. Based on $\tau_k^+f_k^+(f_k^+)^2=\tau_k^*f_k^*(f_k^*)^2$ and $\tau_k^+=T>\tau^*$, we have $f_k^+<f_k^*$ and $\tau_k^+f_k^+>\tau_k^*f_k^*$, leading to $R_{\rm total}^+>R_{\rm total}^*$, which contradicts the above assumption that $\tau_k^*<T$. Thus, $\tau_k^*=T$ holds when the maximum computation bits of the considered system are achieved.

Based on the above analysis, Proposition 1 is obtained.

Appendix B
Proof of Proposition 2

After carefully analyzing $\mathbf{P}_3$, it is not hard to conclude that the objective function and all the constraints except C1″ and C9″ are linear. Thus, $\mathbf{P}_3$ is convex if and only if constraints C1″ and C9″ are convex. For constraint C1″, we only need to prove that for given k, $t_k^{\rm b}B\log_2\Big(1+\frac{\xi x_kP_{\max}g_kh_k}{t_k^{\rm b}B\sigma^2}\Big)+t_k^{\rm a}B\log_2\Big(1+\frac{y_kh_k}{t_k^{\rm a}B\sigma^2}\Big)\geq\beta_kL_{\min,k}$ is convex. Using the fact that the perspective function can preserve convexity, we know that the convexity of the above constraint is the same as that of $B\log_2\Big(1+\frac{\xi x_kP_{\max}g_kh_k}{B\sigma^2}\Big)+B\log_2\Big(1+\frac{y_kh_k}{B\sigma^2}\Big)\geq\beta_kL_{\min,k}$, which is convex. Similarly, we can prove that C9″ is also convex. The proof is complete.

APPENDIX C
PROOF OF PROPOSITION 4

Let $\boldsymbol{\alpha}=(\alpha_0,\alpha_1,\cdots,\alpha_6)$, $\boldsymbol{\theta}=(\theta_0,\theta_1,\theta_2,\cdots,\theta_K)$, $\boldsymbol{\varpi}=(\varpi_0,\varpi_1,\varpi_2,\cdots,\varpi_K)$, $\boldsymbol{\omega}=(\omega_0,\omega_1,\omega_2,\cdots,\omega_K)$, and $\boldsymbol{\mu}=(\mu_0,\mu_1,\mu_2,\cdots,\mu_K)$ be the non-negative Lagrange multipliers with respect to all the constraints for $\mathbf{P}_8$. Then the Lagrangian function of $\mathbf{P}_8$ is given by (C.1), as shown at the top of the next page.

By taking the partial derivative of $\mathcal{L}$ with respect to φ and ϕ, we have

$$\frac{\partial\mathcal{L}}{\partial\varphi}=\frac{\mu_0\sqrt{\phi^3}}{2\sqrt{\varphi^3}}-q\varepsilon_{\mathrm{m}}-\omega_0, \tag{C.2}$$

$$\frac{\partial L}{\partial\phi}=\frac{\alpha_0}{C_{\mathrm{cpu}}}+\omega_0 f_{\max}^2-\frac{3\mu_0}{2}\sqrt{\frac{\phi}{\varphi}}. \tag{C.3}$$

By letting $\frac{\partial\mathcal{L}}{\partial\varphi}=\frac{\partial L}{\partial\phi}=0$, we can obtain

$$f_{\mathrm{m}}^{+}=\left[\frac{3\mu_0 C_{\mathrm{cpu}}}{2\left(\alpha_0+\omega_0 f_{\max}^2 C_{\mathrm{cpu}}\right)}\right]^{+}=\left[\sqrt[3]{\frac{\mu_0}{2\left(q\varepsilon_{\mathrm{m}}+\omega_0\right)}}\right]^{+}, \tag{C.4}$$

where $[x]^{+}=\max\{x,0\}$.

Based on (C.4), it can be observed that when there are tasks to be computed at the MEC server, namely $f_{\mathrm{m}}^{+}>0$, $\mu_0>0$ must be satisfied. Then according to the Karush-Kuhn-Tucker (KKT) conditions, the equation $\mu_0\left(T-t_{\mathrm{e}}^{+}-\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}+t_k^{\mathrm{a}+}\right)-\sqrt{\frac{(\phi^{+})^3}{\varphi^{+}}}\right)=0$ should always hold. Substituting $\mu_0>0$ into the above equation, we have $T-t_{\mathrm{e}}^{+}-\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}+t_k^{\mathrm{a}+}\right)-\sqrt{\frac{(\phi^{+})^3}{\varphi^{+}}}=T-t_{\mathrm{e}}^{+}-\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}+t_k^{\mathrm{a}+}\right)-t_{\mathrm{c}}^{+}=0$. That is, if $f_{\mathrm{m}}^{+}>0$, the MEC server always uses the maximum allowed time to compute tasks. Note that in the case of $f_{\mathrm{m}}^{+}=0$, the MEC server can not provide computation service for the IoT nodes. Then the value of t_{c}^{+} does not influence the system computation EE of the investigated network and we can also let $t_{\mathrm{c}}^{+}=T-t_{\mathrm{e}}^{+}-\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}+t_k^{\mathrm{a}+}\right)$ for convenience.

APPENDIX D
PROOF OF PROPOSITION 5

Let $f\left(P_{\mathrm{t}}\right)$ denote the objective function of $\mathbf{P}_{12}$, given by $f\left(P_{\mathrm{t}}\right)=\frac{\sum_{k=1}^{K}t_k^{\mathrm{b}}B\log_2(1+C_kP_{\mathrm{t}})+\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0}{P_{\mathrm{t}}A_0+B_0}$. Taking the first-order derivative of $f\left(P_{\mathrm{t}}\right)$ with respect to P_{t}, we have

$$\frac{\partial f\left(P_{\mathrm{t}}\right)}{\partial P_{\mathrm{t}}}=\frac{\sum\limits_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)-A_0\left(\sum\limits_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)}{\left(P_{\mathrm{t}}A_0+B_0\right)^2}, \tag{D.1}$$

where $f_k\left(P_{\mathrm{t}}\right)=\frac{C_k(P_{\mathrm{t}}A_0+B_0)}{(1+C_kP_{\mathrm{t}})\ln 2}-A_0\log_2\left(1+C_kP_{\mathrm{t}}\right)$. In order to tell the monotonicity of $\frac{\partial f(P_{\mathrm{t}})}{\partial P_{\mathrm{t}}}$, we first calculate the first-order derivative of $f_k\left(P_{\mathrm{t}}\right)$ as

$$\frac{\partial f_k\left(P_{\mathrm{t}}\right)}{\partial P_{\mathrm{t}}}=\frac{-C_k^2\left(B_0+A_0P_{\mathrm{t}}\right)}{\left(1+C_kP_{\mathrm{t}}\right)^2\ln 2}<0. \tag{D.2}$$

Then the function $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)$ decreases with the increase of P_{t}. That is, there is only one solution making $\frac{\partial f(P_{\mathrm{t}})}{\partial P_{\mathrm{t}}}=0$ hold. Let P_0 be the solution to $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)-A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)=0$ and the value of P_0 can be obtained by means of the bisection method since the function $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)$ is a monotonic decrease function. Then we find that when $0<P_{\mathrm{t}}<P_0$, $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)-A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)>0$ and $f\left(P_{\mathrm{t}}\right)$ increases with P_{t}. Likewise, when $P_{\mathrm{t}}>P_0$, $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(P_{\mathrm{t}}\right)-A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)<0$ and $f\left(P_{\mathrm{t}}\right)$ shows a downward trend.

Combining the range of P_{t} with $\max\left(0,P_{\mathrm{L}}\right)\le P_{\mathrm{t}}\le\min\left(P_{\mathrm{L2}},P_{\max}\right)$, the optimal solution to $\mathbf{P}_{12}$, denoted by $P_{\mathrm{t}}^{\star}$, is determined by $\max\left(0,P_{\mathrm{L}}\right)$, P_0, or $\min\left(P_{\mathrm{L2}},P_{\max}\right)$. Specifically, if $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(\max\left(0,P_{\mathrm{L}}\right)\right)\le A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)$ is satisfied, then $f\left(P_{\mathrm{t}}\right)$ monotonically decreases with $P_{\mathrm{t}}\in\left[\max\left(0,P_{\mathrm{L}}\right),\min\left(P_{\mathrm{L2}},P_{\max}\right)\right]$ and $P_{\mathrm{t}}^{\star}$ is given by $\max\left(0,P_{\mathrm{L}}\right)$. If $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(\min\left(P_{\mathrm{L2}},P_{\max}\right)\right)\ge A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)$ holds, $f\left(P_{\mathrm{t}}\right)$ monotonically increases at the condition of $P_{\mathrm{t}}\in\left[\max\left(0,P_{\mathrm{L}}\right),\min\left(P_{\mathrm{L2}},P_{\max}\right)\right]$ and $P_{\mathrm{t}}^{\star}$ is determined by $\min\left(P_{\mathrm{L2}},P_{\max}\right)$. If $\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(\min\left(P_{\mathrm{L2}},P_{\max}\right)\right)<A_0\left(\sum_{k=1}^{K}R_k^{\mathrm{a}}+D_0\right)<\sum_{k=1}^{K}t_k^{\mathrm{b}}Bf_k\left(\max\left(0,P_{\mathrm{L}}\right)\right)$ is obtained, then we have $\max\left(0,P_{\mathrm{L}}\right)\le P_0\le\min\left(P_{\mathrm{L2}},P_{\max}\right)$ and $P_{\mathrm{t}}^{\star}=P_0$.

Therefore, the optimal solution to $\mathbf{P}_{12}$ is summarized as Proposition 5.

APPENDIX E
PROOF OF LEMMA 2

Here we prove Lemma 2 by means of contradiction. In particular, when $\mathbf{P}_6$ is optimally solved, $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\mathrm{t}}g_kh_k}{B\sigma^2}\right)+t_k^{\mathrm{a}+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)\neq\frac{t_{\mathrm{c}}^{+}f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$ is assumed to be satisfied. That means that either $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\mathrm{t}}g_kh_k}{B\sigma^2}\right)+t_k^{\mathrm{a}+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)>\frac{t_{\mathrm{c}}^{+}f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$ or $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\mathrm{t}}g_kh_k}{B\sigma^2}B\sigma^2\right)+t_k^{\mathrm{a}+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)<\frac{t_{\mathrm{c}}^{+}f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$ holds. If $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\mathrm{t}}g_kh_k}{B\sigma^2}\right)+t_k^{\mathrm{a}+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)<\frac{t_{\mathrm{c}}^{+}f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$ holds, then we can construct another solution which satisfies $p_k^{o}=p_k^{+}$, $f_k^{o}=f_k^{+}$, $\rho_k^{o}=\rho_k^{+}$, $t_k^{\mathrm{b}o}=t_k^{\mathrm{b}+}$, $t_k^{\mathrm{a}o}=t_k^{\mathrm{a}+}$, $t_{\mathrm{e}}^{o}=t_{\mathrm{e}}^{+}$, $t_{\mathrm{c}}^{o}=t_{\mathrm{c}}^{+}$ and $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}o}B\log_2\left(1+\frac{\xi\rho_k^{o}P_{\mathrm{t}}g_kh_k}{B\sigma^2}\right)+t_k^{\mathrm{a}o}B\log_2\left(1+\frac{p_k^{o}h_k}{B\sigma^2}\right)\right)=\frac{t_{\mathrm{c}}^{o}f_{\mathrm{m}}^{o}}{C_{\mathrm{cpu}}}<\frac{t_{\mathrm{c}}^{+}f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$. It can be observed that the constructed solution satisfies all the constraints of $\mathbf{P}_6$ and the constructed solution can achieve the same computation bits as the optimal one while consuming less energy due to $f_{\mathrm{m}}^{o}<f_{\mathrm{m}}^{+}$. This means that the system computation EE under the constructed solution is higher than that under the optimal solution, which contradicts the fact that $\{t_{\mathrm{e}}^{+},\{t_k^{\mathrm{b}+}\}_{k=1}^{K},\{t_k^{\mathrm{a}+}\}_{k=1}^{K},\{\rho_k^{+}\}_{k=1}^{K},\{p_k^{+}\}_{k=1}^{K},\{f_k^{+}\}_{k=1}^{K},t_{\mathrm{c}}^{+},f_{\mathrm{m}}^{+}\}$ is the optimal solution to $\mathbf{P}_6$. The same way can also be applied to the case with the case $\sum_{k=1}^{K}\left(t_k^{\mathrm{b}+}B\log_2\left(1+\frac{\xi\rho_k^{+}P_{\mathrm{t}}g_kh_k}{B\sigma^2}\right)+t_k^{\mathrm{a}+}B\log_2\left(1+\frac{p_k^{+}h_k}{B\sigma^2}\right)\right)>$

$$\mathcal{L} = \lambda + \sum_{k=1}^{K} \frac{f_k T}{C_{\mathrm{cpu},k}} - q \left(\left(t_{\mathrm{e}} + \sum_{k=1}^{K} t_k^{\mathrm{b}} \right) \left(P_{\mathrm{t}} + P_{\mathrm{sc}} - \eta P_{\mathrm{t}} \sum_{k=1}^{K} g_k \right) + \eta P_{\mathrm{t}} \sum_{k=1}^{K} x_k g_k + \varepsilon_{\mathrm{m}} \varphi + \sum_{k=1}^{K} \left(P_{\mathrm{c}} t_k^{\mathrm{b}} + y_k + p_{\mathrm{c}} t_k^{\mathrm{a}} + \varepsilon_k f_k^3 T \right) \right)$$
$$- \sum_{k=1}^{K} \alpha_k \left(t_k^{\mathrm{b}} B \log_2 \left(1 + \frac{\xi x_k P_{\mathrm{t}} g_k h_k}{t_k^{\mathrm{b}} B \sigma^2} \right) + t_k^{\mathrm{a}} B \log_2 \left(1 + \frac{y_k h_k}{t_k^{\mathrm{a}} B \sigma^2} \right) - \beta_k L_{\min,k} \right) + \alpha_0 \left(\frac{\phi}{C_{\mathrm{cpu}}} - \sum_{k=1}^{K} \beta_k L_{\min,k} \right) + \omega_0 \left(\phi f_{\max}^2 - \varphi \right)$$
$$+ \sum_{k=1}^{K} \theta_k \left(\frac{f_k T}{C_{\mathrm{cpu},k}} - (1 - \beta_k) L_{\min,k} \right) + \sum_{k=1}^{K} \varpi_k \left(\eta P_{\mathrm{t}} g_k \left(t_{\mathrm{e}} + \sum_{i=1}^{K} t_i^{\mathrm{b}} \right) - \eta x_k P_{\mathrm{t}} g_k - P_{\mathrm{c}} t_k^{\mathrm{b}} - y_k - p_{\mathrm{c}} t_k^{\mathrm{a}} - \varepsilon_k f_k^3 T \right)$$
$$+ \sum_{k=1}^{K} \omega_k \left(f_k^{\max} - f_k \right) + \mu_0 \left(T - t_{\mathrm{e}} - \sum_{k=1}^{K} \left(t_k^{\mathrm{b}} + t_k^{\mathrm{a}} \right) - \sqrt{\frac{\phi^3}{\varphi}} \right) + \sum_{k=1}^{K} \mu_k \left(t_k^{\mathrm{b}} - x_k \right) + \varpi_0 \left(\frac{\phi}{C_{\mathrm{cpu}}} - \lambda \right)$$
$$+ \theta_0 \left(\sum_{k=1}^{K} \left(t_k^{\mathrm{b}} B \log_2 \left(1 + \frac{\xi x_k P_{\mathrm{t}} g_k h_k}{t_k^{\mathrm{b}} B \sigma^2} \right) + t_k^{\mathrm{a}} B \log_2 \left(1 + \frac{y_k h_k}{t_k^{\mathrm{a}} B \sigma^2} \right) \right) - \lambda \right). \tag{C.1}$$

$\frac{t_{\mathrm{c}}^{+} f_{\mathrm{m}}^{+}}{C_{\mathrm{cpu}}}$ and the detailed process is omitted here for brevity. On this basis, Lemma 2 is obtained.

REFERENCES

[1] J. Li, M. Dai, and Z. Su, "Energy-aware task offloading in the internet of things," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 112–117, 2020.

[2] C. R. Valenta and G. D. Durgin, "Harvesting wireless power: Survey of energy-harvester conversion efficiency in far-field, wireless power transfer systems," *IEEE Microw. Mag.*, vol. 15, no. 4, pp. 108–120, 2014.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[4] X. Cao, B. Yang, C. Huang, C. Yuen, Y. Zhang, D. Niyato, and Z. Han, "Converged reconfigurable intelligent surface and mobile edge computing for space information networks," *IEEE Network*, vol. 35, no. 4, pp. 42–48, 2021.

[5] Y. Ye, R. Q. Hu, G. Lu, and L. Shi, "Enhance latency-constrained computation in MEC networks using uplink NOMA," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2409–2425, 2020.

[6] 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, 2016.

[7] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, 2018.

[8] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, pp. 1–1, 2019.

[9] F. Wang, "Computation rate maximization for wireless powered mobile edge computing," in *Proc. 23rd Asia-Pacific Conf. Commun.*, 2017, pp. 1–6.

[10] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, 2018.

[11] F. Zhou and R. Q. Hu, "Computation efficiency maximization in wireless-powered mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3170–3184, 2020.

[12] X. Hu, K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, 2018.

[13] Y. Ye, L. Shi, R. Q. Hu, and G. Lu, "Energy-efficient resource allocation for wirelessly powered backscatter communications," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1418–1422, 2019.

[14] D. Darsena, G. Gelli, and F. Verde, "Modeling and performance analysis of wireless networks with ambient backscatter devices," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1797–1814, 2017.

[15] Y. Ye, L. Shi, X. Chu, and G. Lu, "Throughput fairness guarantee in wireless powered backscatter communications with HTT," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 449–453, 2021.

[16] S. H. Kim and D. I. Kim, "Hybrid backscatter communication for wireless-powered heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6557–6570, 2017.

[17] S. Gong, Y. Xie, J. Xu, D. Niyato, and Y.-C. Liang, "Deep reinforcement learning for backscatter-aided data offloading in mobile edge computing," *IEEE Netw.*, vol. 34, no. 5, pp. 106–113, 2020.

[18] Y. Zou, J. Xu, S. Gong, Y. Guo, D. Niyato, and W. Cheng, "Backscatter-aided hybrid data offloading for wireless powered edge sensor networks," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[19] L. Shi, Y. Ye, X. Chu, and G. Lu, "Computation bits maximization in a backscatter assisted wirelessly powered MEC network," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 528–532, 2021.

[20] L. Shi, Y. Ye, G. Zheng, and G. Lu, "Computational EE fairness in backscatter-assisted wireless powered MEC networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 1088–1092, 2021.

[21] J. Lu, P. Wu, and M. Xia, "Computation-efficient hybrid offloading for backscatter-assisted wirelessly powered MEC," in *Proc. IEEE VTC-Spring*, 2021, pp. 1–6.

[22] Y. Ye, L. Shi, X. Chu, D. Li, and G. Lu, "Delay minimization in wireless powered mobile edge computing with hybrid backcom and AT," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1532–1536, 2021.

[23] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, 2019.

[24] B. Lyu, H. Guo, Z. Yang, and G. Gui, "Throughput maximization for hybrid backscatter assisted cognitive wireless powered radio networks," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2015–2024, 2018.

[25] L. Shi, R. Q. Hu, J. Gunther, Y. Ye, and H. Zhang, "Energy efficiency for RF-powered backscatter networks using HTT protocol," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 932–13 936, 2020.

[26] H. Yang, Y. Ye, X. Chu, and S. Sun, "Energy efficiency maximization for UAV-enabled hybrid backscatter-harvest-then-transmit communications," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.

[27] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, 2003.

[28] Z. B. Zawawi, Y. Huang, and B. Clerckx, "Multiuser wirelessly powered backscatter communications: Nonlinearity, waveform design, and sinr-energy tradeoff," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 241–253, 2019.

[29] L. Shi, Y. Ye, R. Q. Hu, and H. Zhang, "Energy efficiency maximization for SWIPT enabled two-way DF relaying," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 755–759, 2019.

[30] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, 2016.

[31] Y. Ye, L. Shi, H. Sun, R. Q. Hu, and G. Lu, "System-centric computation energy efficiency for distributed noma-based mec networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8938–8948, 2020.

[32] L. Shi, Y. Ye, X. Chu, and G. Lu, "Computation energy efficiency maximization for a NOMA-based WPT-MEC network," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 731–10 744, 2021.

[33] W. Dinkelbach, "On nonlinear fractional programming," *Manag. Sci.*, vol. 13, no. 7, pp. 494–494, 1967.