Computation Energy Efficiency Maximization for NOMA-Based and Wireless-Powered Mobile Edge Computing with Backscatter Communication

Junhui Du, Huaming Wu, *Senior Member, IEEE*, Minxian Xu, *Member, IEEE*, and
Rajkumar Buyya, *Fellow, IEEE*

Abstract—In the Internet of Things (IoT) environment, a wide variety of mobile devices (MDs) have become part of it, leading to a dramatic increase in the amount of task data. However, due to the limited battery capacity and computing resources of MDs, a lot of effort is required to be taken on how to process more data with less energy. In this paper, we take into account the low utilization of spectrum resources and the short battery life of the equipment, and a backscatter communication-mobile edge computing (BC-MEC) network system based on Non-orthogonal multiple access (NOMA) communication mode is proposed. In order to maximize the computation energy efficiency (CEE) of the system, we jointly optimize the backscatter coefficient of each MD, the backscatter communication duration, the direct offloading duration, the MEC server processing time, the local processing time, the direct offloading power of each MD, the calculation frequency of the MEC server, and the local calculation frequency of each MD. We then formulate it as a joint fractional optimization problem, which is a non-convex optimization problem that is difficult to solve by heuristic algorithms with high computational complexity. To this end, we transform such a problem into a convex problem and apply the Lagrangian dual method to solve it efficiently. Furthermore, in order to meet different user requirements, two effective iterative Dinkelbach algorithms based on Backscatter Coefficient Updates (DBCU) are proposed to solve this problem. Extensive simulation results demonstrate the superiority of our proposed approach, which improves the system CEE by at least 10% compared to state-of-the-art methods.

Index Terms—Mobile edge computing, Backscatter communications, Partial offloading, Computation energy efficiency

✦

1 INTRODUCTION

THE rapid proliferation of the Internet of Things (IoT) paradigm and the advent of the era of big data have facilitated a significant increase in the interconnection of mobile devices (MDs) with the core network, resulting in the generation of vast volumes of data. Consequently, the need to process data generated from these devices in a real-time and accurate manner has become increasingly imperative. However, driven by considerations of production cost and market demand, the trend toward miniaturization in mobile device development has inadvertently imposed constraints on battery life and computing resources, thereby exacerbating the challenges associated with data processing. In the contemporary epoch of data explosion, how to meet the diverse requirements of users while concurrently minimizing energy consumption is of great significance.

- J. Du, and H. Wu are with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China. E-mail: {dujunhui_0325, whming}@tju.edu.cn.
- M. Xu is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. E-mail: mx.xu@siat.ac.cn.
- R. Buyya is with Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia. E-mail: rbuyya@unimelb.edu.au.

(Corresponding author: Huaming Wu)

To address the aforementioned challenges, two distinct technologies, namely, wireless power transmission (WPT) [1] and mobile edge computing (MEC) [2], have emerged as potential solutions that can be applied individually or synergistically. On the one hand, WPT can improve the battery life of mobile devices to a certain extent. The core principle involves the deployment of power beacons (PB) or energy towers that broadcast energy signals to the surrounding environment, thereby facilitating mobile devices to supplement energy reserves by receiving and harnessing these transmitted energy signals. On the other hand, MEC enables users to fulfill the requirements of users within a specified time by offloading computationally intensive tasks, beyond the processing capabilities of mobile devices within designated timeframes, to specialized MEC servers. In addition, MEC optimizes the utilization of the radio access network to deliver essential services and computational resources, establishing a low-latency and high-bandwidth environment that effectively alleviates the predicament of insufficient local computing resources on mobile devices.

This paper introduces the integration of backscatter communication (BC) technology into MEC as a means to substantially augment the battery life of MDs. This novel approach significantly deviates from the traditional implementation of WPT. In conventional WPT scenarios, MDs are subject to wireless charging first, following which they directly offload compute-intensive tasks to the MEC server. In

contrast, BC scenarios enable MDs to modulate the received energy signal, allowing them to carry a portion of the task data to be offloaded onto the MEC server. This innovative process further enhances the endurance and operational longevity of the MDs. Furthermore, this paper leverages the non-orthogonal multiple access (NOMA) communication method, and its crucial aspect lies in its capacity to enable MDs employing distinct channels to concurrently transmit data with the MEC server, all within the same spectrum resource. Inspired by the above facts, this paper posits that the integration of these two technologies can effectively prolong the endurance of MDs while concurrently optimizing the spectrum utilization.

Conceptually, optimization objectives within MEC scenarios can be decomposed into two key performance indicators, namely, energy consumption minimization and computing rate maximization. Existing research endeavors predominantly concentrate on addressing one of these aspects individually or by assigning different weights to both indicators for optimization purposes. Nevertheless, in order to facilitate a more comprehensive evaluation of the trade-off between energy consumption and computational efficiency, an alternative metric, namely, computation energy efficiency (CEE) [3], [4], proves to be an effective performance measure in MEC systems and has garnered widespread adoption. CEE is quantitatively defined as the ratio of computing bits to energy consumption, i.e., the physical meaning of the amount of data that can be processed per unit of energy consumed. For example, Mao *et al.* [3] jointly optimized offloading decisions and resource allocation to maximize the minimum CEE among edge users in wirelessly powered MEC systems. Ji *et al.* [4] studied the maximization of system CEE in wirelessly powered MEC networks.

In this paper, we tackle the CEE problem from the perspective of the entire MEC system, rather than optimizing from the perspective of each MD, while considering the limited computing resources and processing time of the MEC server simultaneously, and the energy consumption of PB is also taken into account. Specifically, this work aims to maximize the system CEE while strictly meeting the constraints of latency, energy and computing/communication resources of MDs and MEC. The main contributions of this paper can be summarized as follows:

- We incorporate the backscatter communication technology into MEC to effectively minimize the energy consumption of MDs. Moreover, we leverage NOMA to enhance backscatter communication, thereby optimizing the utilization of spectrum resources. Furthermore, we adopt a nonlinear model for the WPT scheme, aligning it more closely with the practical characteristics of energy harvesting circuits.
- To address the CEE problem across the entire system, we jointly optimize the backscatter coefficient of each MD, the backscatter communication duration, the direct offloading duration, the MEC server processing time, the local processing time, the direct offloading power of each MD, the calculation frequency of the MEC server, and the local calculation frequency of each MD. The problem is modeled as a joint fractional optimization problem, and we transform such

a non-convex optimization problem into a convex problem to facilitate a feasible solution. To the best of our knowledge, this is the first work dedicated to maximizing the system CEE of a NOMA-based wireless-powered BC-MEC network.

- To solve the problem, this work designs two effective iterative <u>D</u>inkelbach algorithms based on <u>B</u>ackscatter <u>C</u>oefficient <u>U</u>pdates (DBCU). A comprehensive simulation is conducted under MEC scenarios. We can see that both DBCU variants proposed in this paper are superior to the comparison schemes in terms of system CEE, and the performance of these two methods consistently surpasses their respective counterparts by a substantial margin of at least 10%.

The remainder of this paper is organized as follows: Section 2 investigates the related work. The system model is presented in Section 3. Section 4 formulated the system CEE maximization problem. In Section 5, according to different user requirements, two novel Dinkelbach algorithms based on backscattering coefficient updates are proposed to obtain the optimal solution. Simulation results are provided in Section 6. This paper is concluded in Section 7.

2 RELATED WORK

This section discusses relevant works in the context of the deployment of energy optimization techniques in the MEC environment from three aspects, namely, typical MEC, WPT-Aided MEC, and Backscatter-Aided MEC.

2.1 Typical MEC

Mao *et al.* [2] introduced a lot of standardization work on MEC and some typical MEC application scenarios. Huang *et al.* [5] studied communication systems assisted by Intelligent Reflective Surfaces (IRS). In this system, they jointly optimize the phase shift coefficient in different time slots and the transmit power of mobile devices to minimize the long-term energy consumption of all mobile devices, while applying Lyapunov's theory to ensure queue stability. Huang *et al.* [6] studied MEC networks and adopted a full binary offloading strategy. They jointly offload decision and bandwidth allocation, and propose a distributed deep learning-based offloading (DDLO) algorithm for MEC networks.

Communication is an indispensable part of the MEC system, many research works use Time Division Multiple Access (TDMA) [13] or Frequency Division Multiple Access (FDMA) [14] communication methods. You *et al.* [7] studied the resource allocation problem of multi-user MEC systems based on TDMA and orthogonal frequency division multiple access (OFDMA). However, the above communication methods are communicated with low spectrum utilization.

2.2 WPT-Aided MEC

Bi *et al.* [8] considered a multi-user MEC network powered by WPT, where each user follows a binary computation offloading strategy, i.e., a set of tasks must be executed locally or on the MEC server as a whole through task offloading. System transfer time is also allocated to maximize the weighted sum of computation speed for all users in the

TABLE 1: The qualitative comparison of the current literature. The symbol "✓" indicates that this factor is taken into account, and the symbol "✗" indicates that this factor is not considered.

Model	Optimize computation rate	Reduce energy consumption	Enhance communication	Wireless energy transfer	Resource allocation
Huang *et al.* [5]	✗	✓	✓	✗	✓
huang *et al.* [6]	✗	✓	✗	✗	✓
You *et al.* [7]	✗	✓	✗	✗	✓
Bi *et al.* [8]	✓	✗	✗	✓	✓
Zeng *et al.* [9]	✓	✗	✗	✗	✓
Huang *et al.* [10]	✓	✗	✗	✓	✗
Shi *et al.* [11]	✓	✗	✓	✓	✓
Xie *et al.* [12]	✓	✗	✓	✗	✓
This work	✓	✓	✓	✓	✓

network. Zeng *et al.* [9] utilized wirelessly powered MEC networks, where each user gets wireless energy and follows a binary computation offloading strategy. In addition, they introduced NOMA communication for data transmission and maximized the total computing rate for all users by jointly optimizing individual computing mode selection (local computing or offloading), time allocation for energy transfer and data transmission, and local computing speed or transmission power level. Huang *et al.* [10] presented a WPT-MEC network with a binary offloading strategy, where each WD's computational tasks are either executed locally or completely offloaded to the MEC server. They proposed a Deep Reinforcement learning-based Online Offloading (DROO) framework, which can optimize task offloading decisions and wireless resource allocation according to time-varying wireless channel conditions.

2.3 Backscatter-Aided MEC

Backscatter-aided MEC networks are gaining increasing popularity these days [15]–[18].

Shi *et al.* [11] investigated backscatter-assisted wirelessly powered MEC networks, in which each edge user (EU) collects energy before backscattering and offloading, using a partial offloading scheme, in which they maximize the weighting and computation bits of all EU by jointly optimizing the backscatter reflectance coefficient and time, effective transmission power and time, local calculation frequency, and execution time in each EU. Xie *et al.* [12] proposed a new hybrid data offloading scheme that allows each device to offload data via conventional RF communication or low-power backscatter communication. Deep reinforcement learning (DRL) was used to learn the best offloading strategy from past experience [18]. By interacting with the network environment, they optimized each user's energy harvesting time and workload distribution between different offload scenarios.

2.4 A Qualitative Comparison

Table 1 identifies and compares key elements of related works with ours in terms of computation rate, energy consumption, communication, wireless energy transfer, and resource allocation.

Unfortunately, several prior studies in the aforementioned literature tend to presume that MEC servers possess very powerful processing capabilities, usually ignoring both the data processing time and the energy consumption associated with MEC servers [11]. Furthermore, in the context of WPT-assisted MEC, many works directly adopt a linear

energy harvesting model [8], [10], despite being inconsistent with the inherent nonlinearity of energy harvesting circuits.

Motivated by the above facts, this work aims to design a nonlinear energy harvesting model, which can match the performance of real-world energy harvesting circuits. We consider the computing resources of the MEC server and optimize the performance from the perspective of the whole system, rather than from the perspective of each MD [5], [7]. Unlike previous approaches that adopt deep reinforcement learning-based methods [6], [10], [12] to solve the complex non-convex optimization problem, we transform it into a convex optimization problem for solution. To this end, we propose to use reduced-complexity iterative algorithms based on the Dinkelbach algorithm to obtain the optimal solution to the problem. In order to further enhance the overall performance of the MEC system and improve the utilization of the spectrum, in this paper, we use NOMA [13], [19]–[22] communication methods, which can improve the spectrum utilization of the transmission link. To the best of our knowledge, this is the first work that jointly optimizes computation rate, energy consumption, data communication, wireless energy transfer, and resource allocation in the NOMA-Based BC-MEC system.

3 SYSTEM MODEL

As illustrated in Fig. 1, we consider a NOMA-based BC-MEC network, wherein wireless power supply technology is incorporated to enhance the endurance of user devices. Here, g_k denotes the channel gain from PB to the k_{th} MD and h_k denotes the channel gain from the k_{th} MD to the MEC server.
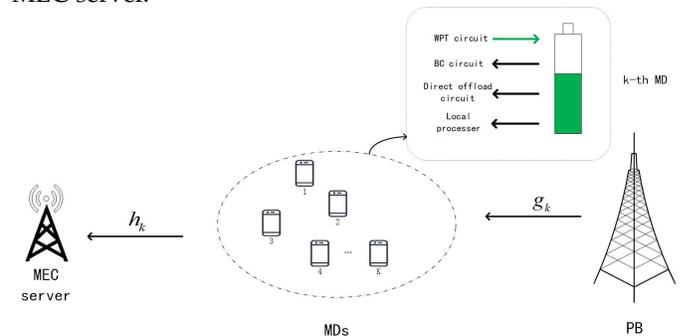

Fig. 1: A NOMA-based BC-MEC network model

In order to avoid issues such as round-trip path loss and self-interference commonly encountered in traditional backscatter communication systems [23], a bistatic backscatter communication system (BBCS) [24], [25] is employed. In

the BBCS setup, the receiver and transmitter are spatially separated, thereby enhancing the overall system performance. The network configuration comprises a PB, K MDs, and a MEC server. Each MD is equipped with a wirelessly rechargeable battery, a backscatter communication circuit, a direct offloading circuit (distinguished from backscatter communication offloading), and a local processor. We assume that each circuit operates independently [26]. Therefore, each MD possesses the ability to backscatter a portion of the task data, directly offload another portion of the task data, and process a further portion of the task data locally within the MD. Owing to the independence of the various circuits, each MD can effectively offload and process the task data concurrently.
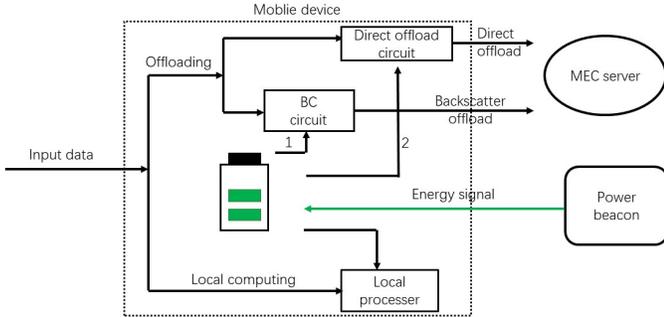

Fig. 2: Specific process, where 1 indicates that backscatter offloading occurs first and 2 means that the direct offloading takes place after 1

As shown in Fig. 2, we adopt a partial offloading strategy while assuming that each task adheres to bit-wise independence [27]. Each MD first engages in backscattering and wireless charging, subsequently transitioning to direct offloading. Specifically, during this process, every MD initially harnesses energy from the transmitted PB signal. Furthermore, it performs backscattering of a segment of task data to the MEC server through the PB transmission signal. Following this, each MD proceeds to offload a portion of the task data directly to the MEC server. All channel models employed in this paper adhere to a quasi-stationary behavior, meaning they remain constant throughout each time slot but can vary between different time slots. The main notations used in this paper are summarized in Table 2.
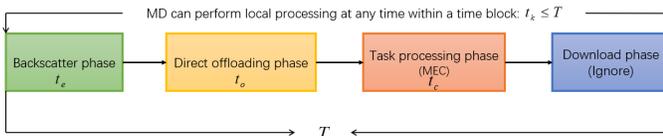

Fig. 3: Time allocation of the considered network

As shown in Fig. 3, we use TDMA to divide the running time of the entire system into time slots, each time slot has a duration of T, and for each time slot, we divide it into four different phases, namely, backscatter communication, direct offloading, task processing, and result download.

- In the first phase, due to the independence of each MD's backscatter communication circuit and wireless charging circuit, MD can perform backscattering and wireless charging simultaneously. Guided by

TABLE 2: Notations and Their Definitions

Notitations	Definitions
P_t	The signal transmission power of PB
h_k	The k_{th} MD to MEC link channels
g_k	The PB to the k_{th} MD link channels
t_e	The energy collection time of MD
t_o	The offloading time of MD
t_c	The processing time of MEC server
t_k	The local processing time of the k_{th} MD
f_m	The CPU frequency on the MEC server
f_k	The CPU frequency on the k_{th} MD
P_k	The offloading power of the k_{th} MD
T	The entire time block
B	The communication bandwidth
K	The number of MDs
C_{cpu}^m	The number of CPU cycles for one bit data (MEC)
C_{cpu}^k	The number of CPU cycles for one bit data (k_{th} MD)
ε_k	The ECC for the k_{th} MD
ε_m	The ECC for the MEC server
f_{max}^k	The maximum CPU frequency of the k_{th} MD
f_{max}	The maximum CPU frequency of the MEC server
L_{min}	The minimum amount of computational data

the backscattering coefficient $\alpha_k (0 \leq \alpha_k \leq 1)$ [28], the k_{th} MD divides the received PB energy signal into two distinct allocations. A portion of the energy signal is employed for the purpose of backscattering a fraction of the task data to the MEC server through the uplink NOMA technique. Meanwhile, the remaining energy signal is dedicated to charging the MD and replenishing its energy reserves.

- In the second phase, the PB stops working and goes into a dormant state to save energy, while each MD again offloads a part of the task data directly to the MEC server by means of uplink NOMA.
- In the third phase, the MEC server performs the processing of task data.
- The fourth phase is to return the processing results of each MD's task data, which is ignored due to the small magnitude of the returned result.

Given that each MD is equipped with an individual local processor, and due to the inherent independence among the different circuits, it becomes feasible for the MD to conduct partial processing of the task data locally within each time slot [29].

3.1 Backscatter Communication Phase

In this phase, the PB broadcasts the energy signal [30], and then all MDs offload part of their task data to the MEC server through a backscatter communication circuit via uplink NOMA. The MEC server acquires task data for each MD through successive interference cancelation (SIC) [31]. According to the communication principle of NOMA, the MEC server will first decode the data from the MD of the optimal channel condition, then subtract the message that has been decoded from the received composite signal, and then continue to decode the data of the MD of the sub-optimal channel condition, and so on [19]. According to the backscatter communication model, the channel conditions between MD and MEC servers are determined not only by the direct connection channel between MD and MEC servers, but also by the channel conditions between PB and

MD. Similar to [11], we assume that g_k, h_k are in descending order, i.e., $g_1 \geq g_2 \geq \cdots \geq g_K$ and $h_1 \geq h_2 \geq \cdots \geq h_K$, then the data throughput that the k_{th} MD can achieve is:

$$D_k^B = t_e B \log_2 \left(1 + \frac{\zeta \alpha_k P_t g_k h_k}{\sum_{i=k+1}^{K} \zeta \alpha_i P_t g_i h_i + B\sigma^2} \right), \quad (1)$$

where B is the communication bandwidth, P_t is the signal transmission power of PB, ζ is the performance gap between the backscatter communication circuit and the direct offloading circuit [32], and σ^2 is the thermal noise power spectral density. According to (1), we can calculate that:

$$D_K^B + D_{K-1}^B = t_e B \log_2 \left(1 + \frac{\zeta \alpha_K P_t g_K h_K}{B\sigma^2} \right)$$
$$+ t_e B \log_2 \left(1 + \frac{\zeta \alpha_{k-1} P_t g_{k-1} h_{k-1}}{\zeta \alpha_K P_t g_K h_K + B\sigma^2} \right)$$
$$= t_e B \log_2 \left(1 + \frac{\zeta \alpha_K P_t g_K h_K + \zeta \alpha_{k-1} P_t g_{k-1} h_{k-1}}{B\sigma^2} \right). \quad (2)$$

On the basis of (2), we can easily get the total throughput of all MDs in the first phase as:

$$\sum_k^K D_k^B = \sum_k^K t_e B \log_2 \left(1 + \frac{\zeta \alpha_k P_t g_k h_k}{\sum_{i=k+1}^{K} \zeta \alpha_i P_t g_i h_i + B\sigma^2} \right)$$
$$= t_e B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k P_t g_k h_k}{B\sigma^2} \right). \quad (3)$$

For the energy harvesting link in this phase, in the past research, a lot of work is directly using the linear energy harvesting model, that is, the energy receiving power is proportional to the energy transmission power, but because the linear energy harvesting model does not match the nonlinear behavior of the energy harvesting circuit, it will lead to a significant performance loss of the circuit, so in this paper, we consider a nonlinear energy harvesting model [33]. In this phase, each MD can collect the energy as:

$$E_k^B = t_e \left(\frac{c_k (1 - \alpha_k) P_t g_k + d_k}{(1 - \alpha_k) P_t g_k + v_k} - \frac{d_k}{v_k} \right), \quad (4)$$

where c_k, d_k and v_k represent the parameters characterizing the nonlinear energy harvesting model of the k_{th} MD. During this phase, the constant circuit consumption of the k_{th} MD is $p_{c,k} t_e$, where $p_{c,k}$ is the inherent power consumption of the backscatter communication circuit. Even though the MEC server does not process the task data during the phase, it still needs to decode the information and eliminate interference from the PB signal. As a result, the energy consumed by the MEC server in this phase is $P_s t_e$, where P_s is the inherent power consumption related to the MEC server's decoding and interference elimination processes.

3.2 Direct Offloading Phase

In this phase, all MDs also use the uplink NOMA method to directly offload part of the task data, since h_k is in descending order, that is, $h_1 \geq h_2 \geq h_3 \geq \cdots \geq h_K$, then the data throughput that k_{th} MD can achieve is:

$$D_k^o = t_o B \log_2 \left(1 + \frac{p_k h_k}{B\sigma^2 + \sum_{i=k+1}^{K} p_i h_i} \right), \quad (5)$$

where p_k is the offloading power of k_{th} MD. According to the calculation process of the total system throughput in the first phase, we can obtain the total system throughput in this phase as:

$$\sum_k^K D_k^o = \sum_k^K t_o B \log_2 \left(1 + \frac{p_k h_k}{B\sigma^2 + \sum_{i=k+1}^{K} p_i h_i} \right)$$
$$= t_o B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k h_k}{B\sigma^2} \right). \quad (6)$$

The system energy consumption in this phase is mainly composed of the offloading energy consumption per MD, and the inherent energy consumption of the direct offloading circuit on each MD, which can be formulated as:

$$E_k^o = p_k t_o + p_{a,k} t_o, \quad (7)$$

where $p_{a,k}$ is the inherent circuit power consumption of the offloading circuit of the k_{th} MD.

3.3 Task Processing Phase

In this phase, the amount of data that the MEC server can process in a given time is:

$$D_M = \frac{t_c f_m}{C_{cpu}^m}, \quad (8)$$

where f_m denotes the CPU frequency of the MEC server, and C_{cpu}^m is the number of CPU cycles required for the MEC server to compute 1-bit data. Since we define not only the processing time, but also the backscatter communication and offloading duration, the total amount of data that the MEC server can process in the processing phase is determined not only by the total amount of offloaded data of all MDs, but also by the amount of data that the MEC server can process, so we can get the total amount of data that the MEC server can process in this phase:

$$D_p = \min \left\{ \sum_{k=1}^{K} D_k^B + \sum_{k=1}^{K} D_k^o, \frac{t_c f_m}{C_{cpu}^m} \right\}. \quad (9)$$

The system energy consumption of this phase mainly includes the processing energy consumption of the MEC server, as well as the constant circuit consumption of the decoding and successive interference cancelation of the MEC server, so the total system energy consumption at this stage can be expressed as [34]:

$$E_m = \varepsilon_m f_m^3 t_c + P_s t_c, \quad (10)$$

where P_s is the power inherent consumption of the MEC server circuit for decoding and eliminating interference, and ε_m represents the ECC of the MEC server processor chip.

3.4 Local Processing Phase

Because each MD is configured with an independent local processor, each MD can process a part of the task data at any time within each time slot. The amount of data that k_{th} MD can process locally during the local processing time is:

$$D_k = \frac{t_k f_k}{C_{cpu}^k}, \quad (11)$$

where f_k denotes the CPU frequency of the k_{th} MD, C_{cpu}^k is denotes the number of CPU cycles required for the k_{th} MD local processor to compute 1-bit data. During this phase, the energy consumption of the k_{th} MD can be mathematically described as $\varepsilon_k f_k^3 t_k$, where ε_k denotes the ECC of the local processor chip for the k_{th} MD.

3.5 System CEE

The total amount of data that the system can process in a time slot consists of two parts: one part is the total amount of data that the MEC server can process, which is $\min\left\{\sum_{k=1}^{K} D_k^B + \sum_{k=1}^{K} D_k^o, \frac{t_c f_m}{C_{cpu}^m}\right\}$, and the other part is the total amount of data processed locally for all MDs, i.e., $\sum_{k=1}^{K} \frac{t_k f_k}{C_{cpu}^k}$.

The total energy consumed by the system in a time slot is divided into three components: i) The energy consumed by PB in the first phase, i.e., $P_t t_e$; ii) The total energy consumption of the MEC server, i.e., $\varepsilon_m f_m^3 t_c + P_s (t_c + t_e)$; iii) The energy consumption for each MD to offload data and process task data locally, i.e., $\sum_{k=1}^{K} p_{c,k} t_e + \sum_{k=1}^{K} (p_k t_o + p_{a,k} t_o) + \sum_{k=1}^{K} \varepsilon_k f_k^3 t_k$. In addition to energy consumption, another part is used by MD for charging, and this part of the energy needs to be deducted from the total energy consumption, which can be expressed as $\sum_{k=1}^{K} E_k^B$.

In this paper, we consider maximizing the CEE of the entire system, which is defined as the ratio between the total amount of data that can be processed by the system and the total energy consumption within a given time slot. The system CEE of the network model can be expressed as:

$$q(\{\alpha_k\}_{k=1}^{K}, t_e, t_o, t_c, \{t_k\}_{k=1}^{K}, \{f_k\}, f_m, \{p_k\}_{k=1}^{K})$$
$$= \frac{\min\left\{\sum_{k=1}^{K} D_k^B + \sum_{k=1}^{K} D_k^o, \frac{t_c f_m}{C_{cpu}^m}\right\} + \sum_{k=1}^{K} \frac{t_k f_k}{C_{cpu}^k}}{Energy}, \quad (12)$$

where $Energy$ represents the total energy consumption of the system, which is $Energy = P_t t_e + \sum_{k=1}^{K} p_{c,k} t_e + \sum_{k=1}^{K} (p_k t_o + p_{a,k} t_o) + \varepsilon_m f_m^3 t_c + P_s (t_c + t_e) + \sum_{k=1}^{K} \varepsilon_k f_k^3 t_k - \sum_{k=1}^{K} E_k^B$.

4 COMPUTATION ENERGY EFFICIENCY MAXIMIZATION

4.1 Problem Formulation

In this paper, we maximize the CEE of the system by jointly optimizing the backscatter coefficient of each MD, the backscatter communication duration, the direct offloading duration, the MEC server processing time, the local processing time, the direct offloading power of each MD, the calculation frequency of the MEC server, and the local calculation frequency of each MD. Thus, the optimization

problem $\mathcal{P}_0$ is formulated as follows:

$$\mathcal{P}_0 : \max_{\left\{\{\alpha_k\}_{k=1}^K, t_e, t_o, t_c, \{t_k\}_{k=1}^K, \{f_k\}_{k=1}^K, f_m, \{p_k\}_{k=1}^K\right\}} q$$

$$s.t. \quad \min\left\{\sum_{k=1}^{K} D_k^B + \sum_{k=1}^{K} D_k^o, \frac{t_c f_m}{C_{cpu}^m}\right\} + \sum_{k=1}^{K} \frac{t_k f_k}{C_{cpu}^k} \geq L_{\min} \quad (13a)$$

$$p_{c,k} t_e + p_k t_o + p_{a,k} t_o + \varepsilon_k f_k^3 t_k \leq E_k^B, \forall k \quad (13b)$$

$$t_e + t_o + t_c \leq T \quad (13c)$$

$$0 \leq t_k \leq T, \forall k \quad (13d)$$

$$0 \leq f_k \leq f_{\max}^k, \forall k \quad (13e)$$

$$0 \leq f_m \leq f_{\max}^m \quad (13f)$$

$$p_k > 0, \forall k \quad (13g)$$

$$t_e, t_o, t_c \geq 0 \quad (13h)$$

$$0 \leq \alpha_k \leq 1, \forall k \quad (13i)$$

where $L_{\min}$ is the minimum amount of computational task data that must be completed within each time slot, $f_{\max}^k$ and $f_{\max}^m$ denote the maximum CPU frequencies of the k_{th} MD's local processor and the MEC server, respectively. Constraint (13a) ensures the fulfillment of the minimum required computation bits for each time slot. Constraint (13b) ensures that the energy consumed by the k_{th} MD in each time slot remains within the bounds of the energy it harvests during the same time slot, allowing for potential energy storage. Constraint (13c) guarantees the processing of all backscattered or offloaded data within the current time slot. Constraint (13d) limits the local processing time of each MD to the duration of the present time slot. Constraints (13e) and (13f) establish upper bounds for the maximum computation frequency for both MDs and MEC servers.Constraint (13g) constrains the transmit power of each MD. Constraint (13h) ensures non-negative durations for all phases. Constraint (13i) confines the possible values of the backscatter coefficient α_k for each MD within a specific range.

Due to the coupling relationships between the various parameter variables and the non-concave nature of the objective function itself, we can see that the formulated problem $\mathcal{P}_0$ is a complex non-convex optimization problem and thus non-trivial. Hence, successive convex optimization tools or meta-heuristic algorithms invariably face difficulties in exploring multidimensional search spaces and adapting to all possible state changes, which significantly increases the computational complexity.

4.2 Solution and Iterative Algorithm

Next, we will analyze the problem step by step, making it an easy-to-solve optimization constraint problem. To deal with the coupling relationship between α_k and t_e, we have the numerator denominator of the objective function in $\mathcal{P}_0$ divide by t_e at the same time, and we make ${t_o}/{t_e} = \tau_o$, ${t_c}/{t_e} = \tau_c$, ${t_k}/{t_e} = \tau_k$, ${1}/{t_e} = \tau_e$, then the optimization problem $\mathcal{P}_0$ can be rewritten as $\mathcal{P}_1$:

$$\mathcal{P}_1 : \max_{\left\{ \begin{array}{c} \{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{\tau_k\}_{k=1}^K, \\ \{f_k\}_{k=1}^K, f_m, \{p_k\}_{k=1}^K \end{array} \right\}} q = \frac{Q + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k}}{E_1}$$

$$s.t. \quad Q + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k} \geq L_{\min} \tau_e \tag{14a}$$

$$p_{c,k} + p_k \tau_o + p_{a,k} \tau_o + \varepsilon_k f_k^3 \tau_k$$
$$\leq \frac{c_k (1 - \alpha_k) P_t g_k + d_k}{(1 - \alpha_k) P_t g_k + v_k} - \frac{d_k}{v_k}, \forall k \tag{14b}$$

$$1 + \tau_o + \tau_c \leq T \tau_e \tag{14c}$$

$$0 \leq \tau_k \leq T \tau_e, \forall k \tag{14d}$$

$$(13e), (13f), (13g), (13i)$$

$$\tau_e, \tau_o, \tau_c \geq 0 \tag{14e}$$

In $\mathcal{P}_1$, $Data = B\log_2 \left(1 + \sum_{k=1}^K \frac{\zeta \alpha_k P_t g_k h_k}{B\sigma^2} \right) + \tau_o B\log_2 \left(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2} \right), Q = \min \left\{ Data, \frac{\tau_c f_m}{C_{cpu}^m} \right\}, E_1 = P_t + \sum_{k=1}^K p_{c,k} + \sum_{k=1}^K (p_k \tau_o + p_{a,k} \tau_o) + \varepsilon_m f_m^3 \tau_c + P_s (\tau_c + \tau_e) + \sum_{k=1}^K \varepsilon_k f_k^3 \tau_k - \sum_{k=1}^K \left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k} \right)$.

To further simplify $\mathcal{P}_1$ and remove the influence of the min function in the objective function, we introduce relaxation factor λ ($\lambda \geq 0$), where $\lambda = \min \left\{ Data, \frac{\tau_c f_m}{C_{cpu}^m} \right\}$. So $\mathcal{P}_1$ can be simplified to $\mathcal{P}_2$:

$$\mathcal{P}_2 : \max_{\left\{ \begin{array}{c} \{\alpha_k\}_{k=1}^K, t_e, t_o, t_c, \{t_k\}_{k=1}^K, \\ \{f_k\}_{k=1}^K, f_m, \{p_k\}_{k=1}^K, \lambda \end{array} \right\}} q = \frac{\lambda + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k}}{E_1}$$

$$s.t. \quad \lambda + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k} \geq L_{\min} \tau_e \tag{15a}$$

$$(13f), (13g), (13i), (13e), (14b), (14c), (14d), (14e)$$

$$B\log_2 \left(1 + \sum_{k=1}^K \frac{\zeta \alpha_k P_t g_k h_k}{B\sigma^2} \right) + \tau_o B\log_2 \left(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2} \right) \geq \lambda \tag{15b}$$

$$\frac{\tau_c f_m}{C_{cpu}^m} \geq \lambda \tag{15c}$$

It can be seen that $\mathcal{P}_2$ is still a non-convex optimization problem, and then we apply Dinkelbach's method [35] and Lagrange dual decomposition [36] to transform the problem into a sequence of convexly constrained optimization problems. By doing this, the original problem can be solved at low complexity.

Theorem 1. *If* $\{\alpha_k^*\}_{k=1}^K$, τ_e^*, τ_o^*, τ_c^*, $\{\tau_k^*\}_{k=1}^K$, $\{f_k^*\}_{k=1}^K$, f_m^*, $\{p_k^*\}_{k=1}^K$, λ^* *is the optimal solution to the optimization problem* $\mathcal{P}_2$ *and* q^* *is the maximum CEE of the system, then we must get*

the following equation:

$$\max_{\left\{ \begin{array}{c} \{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{\tau_k\}_{k=1}^K, \\ \{f_k\}_{k=1}^K, f_m, \{p_k\}_{k=1}^K, \lambda \end{array} \right\}} \lambda + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k} - q\left(P_t + \sum_{k=1}^K p_{c,k} \right)$$

$$+ \sum_{k=1}^K (p_k \tau_o + p_{a,k} \tau_o) + \varepsilon_m f_m^3 \tau_c + P_s (\tau_c + \tau_e) + \sum_{k=1}^K \varepsilon_k f_k^3 \tau_k$$

$$- \sum_{k=1}^K \left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k} \right)$$

$$= \lambda^* + \sum_{k=1}^K \frac{\tau_k^* f_k^*}{C_{cpu}^k} - q\left(P_t + \sum_{k=1}^K p_{c,k} + \sum_{k=1}^K (p_k^* \tau_o^* + p_{a,k} \tau_o^*) \right)$$

$$+ \varepsilon_m (f_m^*)^3 \tau_c^* + P_s (\tau_c^* + \tau_e^*) + \sum_{k=1}^K \varepsilon_k (f_k^*)^3 \tau_k^*$$

$$- \sum_{k=1}^K \left(\frac{c_k(1-\alpha_k^*)P_t g_k + d_k}{(1-\alpha_k^*)P_t g_k + v_k} - \frac{d_k}{v_k} \right) = 0 \tag{16}$$

Proof. **Theorem 1** can be proved according to the generalized distributed programming theory, similar to [35], to save space, we give a short proof.

To provide a proof of the Dinkelbach transformation, we refer to [37]. In **Theorem 1**, it is worth noting that the optimal parameter is identical in both $\mathcal{P}_1$ and $\mathcal{P}_2$ under the condition that the target function on the molecule exhibits concavity, while the target function in the denominator demonstrates convexity. In order to establish the concavity of the objective function in the molecular denominator, we will provide a rigorous proof by performing variable substitutions involving various parameters. □

According to **Theorem 1**, we can convert optimization problem $\mathcal{P}_2$ into an easy-to-solve optimization problem $\mathcal{P}_3$, and design an iterative algorithm based on Dinkelbach to obtain the optimal solution of the optimization problem. The algorithmic process is as demonstrated in **Alg. 1**.

Algorithm 1 Dinkelbach iterative algorithm for $\mathcal{P}_3$

1: Set $q = 0$;
2: Set the maximum error tolerance ε;
3: **while** true **do**
4:    Calculate the optimal value of $f_k, f_m, \tau_c, \tau_o, \tau_e, \tau_k, p_k$ with q value;
5:    Calculate a new CEE q^+;
6:    **if** $|q - q^+| < \varepsilon$ **then**
7:       The current individual argument variables are already the optimal solution to problem P_3,
8:    **else**
9:       Let $q = q^+$, and continue iterating
10:   **end if**
11: **end while**

$$\mathcal{P}_3 : \max_{\left\{ \begin{array}{c} \{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{\tau_k\}_{k=1}^K, \\ \{f_k\}_{k=1}^K, f_m, \{p_k\}_{k=1}^K, \lambda \end{array} \right\}} \lambda + \sum_{k=1}^K \frac{\tau_k f_k}{C_{cpu}^k} - q\left(P_t + \sum_{k=1}^K p_{c,k} \right)$$

$$+ \sum_{k=1}^K (p_k \tau_o + p_{a,k} \tau_o) + \varepsilon_m f_m^3 \tau_c + P_s (\tau_c + \tau_e) + \sum_{k=1}^K \varepsilon_k f_k^3 \tau_k$$

$$- \sum_{k=1}^K \left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k} \right). \tag{17}$$

$$s.t. \quad (13f), (13g), (13i), (13e), (14b), (14c), (14d), (14e), (15a), (15b), (15c)$$

where q is a given parameter in each iteration, and in the case of a given q value, we solve the optimal solution of each parameter variable corresponding to problem $\mathcal{P}_3$, and then calculate the new system CEE as q^+, which is compared with the original q value until the termination condition is met. But we can see that for $\mathcal{P}_3$, there is still a coupling relationship between different variables, and the problem is still a non-convex optimization problem, so we set the following variables to handle the coupling relationships between different variables, we make $x_k = \tau_k f_k, y_k = \tau_k f_k^3, x_m = \tau_c f_m, y_m = \tau_c f_m^3$, then we can get $\mathcal{P}_4$:

$$\mathcal{P}_4 : \max_{\substack{\left\{\{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{x_k\}_{k=1}^K,\right\} \\ \left\{y_k\}_{k=1}^K, x_m, y_m, \{p_k\}_{k=1}^K, \lambda\right\}}} \lambda + \sum_{k=1}^K \frac{x_k}{C_{cpu}^k} - q\left(P_t + \sum_{k=1}^K p_{c,k}\right)$$

$$+ \sum_{k=1}^K (p_k \tau_o + p_{a,k}\tau_o) + \varepsilon_m y_m + P_s(\tau_c + \tau_e) + \sum_{k=1}^K \varepsilon_k y_k$$

$$- \sum_{k=1}^K \left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)\Bigg)$$

$$s.t.\ \lambda + \sum_{k=1}^K \frac{x_k}{C_{cpu}^k} \geq L_{\min}\tau_e \tag{18a}$$

$$p_{c,k} + p_k\tau_o + p_{a,k}\tau_o + \varepsilon_k y_k \leq \frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}, \forall k \tag{18b}$$

$$1 + \tau_o + \sqrt{\frac{x_m^3}{y_m}} \leq T\tau_e \tag{18c}$$

$$0 \leq \sqrt{\frac{x_k^3}{y_k}} \leq T\tau_e, \forall k \tag{18d}$$

$$0 \leq y_k \leq x_k\left(f_{\max}^k\right)^2, \forall k \tag{18e}$$

$$0 \leq y_m \leq x_m(f_{\max}^m)^2 \tag{18f}$$

$$p_k > 0 \tag{18g}$$

$$t_e, t_o, t_c \geq 0 \tag{18h}$$

$$0 \leq \alpha_k \leq 1, \forall k \tag{18i}$$

$$B\log_2\left(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right) + \tau_o B\log_2\left(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2}\right) \geq \lambda \tag{18j}$$

$$x_m \geq \lambda C_{cpu}^m \tag{18k}$$

Theorem 2. *$\mathcal{P}_4$ is a strictly convex constraint optimization problem.*

Proof. First, we need to prove that the objective function of problem P_4 is concave, and for the convexity of the objective function, we only need to explain the convexity of $q\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)$.

Similar to [11], we define the function $F(x) = \frac{c_k(1-x)P_t g_k + d_k}{(1-x)P_t g_k + v_k} - \frac{d_k}{v_k}$, where $0 \leq x \leq 1$. We calculate the second-order derivative of $F(x)$, yielding the following expression: $\frac{\partial^2 F}{\partial x^2} = \frac{2P_t^2 g_k^2(d_k - v_k c_k)}{((1-x)P_t g_k + v_k)^3}$, according to the general conclusion of the nonlinear energy harvesting model, i.e., with the increase of PB-level transmit power, the receive power of MD also increases until convergence.

Let us introduce the variable $y = (1-\alpha_k)P_t g_k$, which represents the transmit power of the PB, so we can get that the function $F(x)$ is a monotonic function for y. Consequently, the first derivative of $F(x)$ in terms of y is greater than or equal to zero. This leads us to the expression

$\frac{\partial^2 F}{\partial y^2} = \frac{c_k v_k - d_k}{(y+v_k)^2} \geq 0$. As a result, we derive the inequality $c_k v_k - d_k \geq 0$. For function $F(x)$, even when the transmit power of PB approaches infinity, the power received by the MD must consistently remain greater than 0. When $y \to \infty$, we arrive at $\lim_{y\to\infty} F(x) = \frac{c_k y + d_k}{y + v_k} - \frac{d_k}{v_k} = c_k - \frac{d_k}{v_k} = \frac{c_k v_k - d_k}{v_k} \geq 0$, which becomes possible due to the fact that $c_k v_k - d_k \geq 0$, so we can get $v_k > 0$. Furthermore, it is possible to establish that $\frac{\partial^2 F}{\partial x^2} = \frac{2P_t^2 g_k^2(d_k - v_k c_k)}{((1-x)P_t g_k + v_k)^3} \leq 0$, i.e., the function $F(x)$ is inherently concave. Our endeavor now involves proving that $q\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)$ is a concave function. Drawing upon the proven concavity of the function $F(x)$, we can get that when $q \geq 0$, $F(x)$ is a concave function. As a result, under the condition of $q \geq 0$, the objective function of problem P_4 can be recognized as a concave function.

Then for the constraints of this optimization problem, since (18e), (18f), (18g), (18h), (18i), (18k) are all linear affine conditions, whether $\mathcal{P}_4$ is a convex optimization problem depends on the convexity of (18a), (18b), (18c), (18d), (18j). For (18a), since all but $B\log_2\left(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right)$ in (18a) are linear functions, we only need to prove that $B\log_2\left(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right)$ is a concave function. Because $\sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}$ is a linear function, and $\log_2(1 + x)$ is a concave function about x, and its extension function is a non-decreasing function, so according to the convexity of the composite function, we get that $B\log_2\left(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right)$ is a concave function. Regarding (18b), as the left-hand side of the inequality forms a linear function, our focus shifts to demonstrating the concavity of $\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}$ on the right-hand side. Through the proof of concavity and convexity concerning the objective function, we deduce that $\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}$ is indeed a concave function. For (18c) and (18d), we introduce the function $H(x,y) = \sqrt{x^3/y}$. To establish the convexity of the function $H(x,y)$, we calculate the second-order partial derivative of $H(x,y)$ with respect to x and y. The resulting Hessian matrix is as follows:

$$\begin{bmatrix} \frac{3}{4}\frac{1}{\sqrt{xy}} & -\frac{3}{4}\sqrt{\frac{x}{y^3}} \\ -\frac{3}{4}\sqrt{\frac{x}{y^3}} & \frac{3}{4}\sqrt{\frac{x^3}{y^5}} \end{bmatrix}$$

Indeed, the Hessian matrix of function $H(x,y)$ is demonstrably semi-positive definite, confirming the convexity of the function $H(x,y)$. Therefore, both constraints (18c) and (18d) can be recognized as convex constraints. For (18j), the key lies in establishing the convexity of the functions $B\log_2(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2})$ and $\tau_o B\log_2(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2})$. In particular, $\sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}$ is a linear function, while $\log_2(1 + x)$ is a concave function, further supported by its non-decreasing extension behavior. Leveraging the principles of convexity for composite functions, we get that $B\log_2\left(1 + \sum_{k=1}^K \frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right)$ is a concave function. Regarding the convexity of $\tau_o B\log_2(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2})$, considering that $\sum_{k=1}^K p_k$ is treated as a variable in the experimental process, we extend this perspective when discussing the convexity of $\tau_o B\log_2(1 + \sum_{k=1}^K \frac{p_k h_k}{B\sigma^2})$. We define the function $F(x,y) = x\log_2(1 + y)$, thereby obtaining the second-order partial derivatives of $F(x,y)$ with respect to x, y. The

$$\mathcal{L}(\mathrm{E}) = \lambda + \sum_{k=1}^{K} \frac{x_k}{C_{cpu}^k} - q \left(P_t + \sum_{k=1}^{K} p_{c,k} + \sum_{k=1}^{K} (p_k \tau_o + p_{a,k} \tau_o) + \varepsilon_m y_m + P_s (\tau_c + \tau_e) + \sum_{k=1}^{K} \varepsilon_k y_k - \sum_{k=1}^{K} \left(\frac{c_k (1 - \alpha_k) P_t g_k + d_k}{(1 - \alpha_k) P_t g_k + v_k} - \frac{d_k}{v_k} \right) \right)$$

$$+ A_0 \left(\lambda + \sum_{k=1}^{K} \frac{x_k}{C_{cpu}^k} - L_{\min} \tau_e \right) + \sum_{k=1}^{K} A_k \left(\frac{c_k (1 - \alpha_k) P_t g_k + d_k}{(1 - \alpha_k) P_t g_k + v_k} - \frac{d_k}{v_k} - p_{c,k} - p_k \tau_o - p_{a,k} \tau_o - \varepsilon_k y_k \right) + \beta_0 \left(T \tau_e - 1 - \tau_o - \sqrt{\frac{x_m^3}{y_m}} \right)$$

$$+ \gamma_0 \left(x_m (f_{\max}^m)^2 - y_m \right) + \sum_{k=1}^{K} \gamma_k \left(T \tau_e - \sqrt{\frac{x_k^3}{y_k}} \right) + \sum_{k=1}^{K} \beta_k \left(x_k \left(f_{\max}^k \right)^2 - y_k \right) + N_1 \left(x_m - \lambda C_{cpu}^m \right)$$

$$+ N_2 \left(B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k P_t g_k h_k}{B \sigma^2} \right) + \tau_o B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k h_k}{B \sigma^2} \right) - \lambda \right). \qquad (19)$$

resulting Hessian matrix is: $\begin{bmatrix} 0 & \frac{1}{1+y} \\ \frac{1}{1+y} & \frac{x}{-(1+y)^2} \end{bmatrix}$. We can easily get that the Hessian matrix of the function $F(x, y)$ is a semi-negative definite matrix, so $F(x, y)$ is a concave function. To sum up, we can get that (18j) is a convex constraint.

By proving the convexity of the objective function and the constraints, we can get that $\mathcal{P}_4$ is a convex optimization problem under the condition of $q \geq 0$. Here, $q \geq 0$ is an implicit constraint of the model in this paper, which is necessarily true, so this theorem is verified. $\square$

5 SOLUTION TO THE OPTIMIZATION PROBLEM

In this section, we apply the Lagrangian dual method [38] to solve the convex optimization problem $\mathcal{P}_4$.

5.1 Proposed Solution for Computation Energy Efficiency)

We analyze the Karush–Kuhn–Tucker (KKT) conditions of $\mathcal{P}_4$, so as to obtain the optimal solution with q as the condition. The non-negative Lagrangian factors of $\mathcal{P}_4$ can be given by $A = (A_0, A_1, \cdots, A_K)$, $\beta = (\beta_0, \beta_1, \cdots, \beta_K)$, $\gamma = (\gamma_0, \gamma_1, \cdots, \gamma_K)$, N_1, N_2. Using the Lagrange duality method, we can get the Lagrange function $\mathcal{L}(\mathrm{E})$ by (19), where $\mathrm{E} = (\{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{x_k\}_{k=1}^K, \{y_k\}_{k=1}^K, x_m, y_m, \{p_k\}_{k=1}^K, \lambda, A, \beta, \gamma, N_1, N_2)$. The dual function of $\mathcal{P}_4$ is defined as:

$$g(A, \beta, \gamma, N_1, N_2) = \max_{\substack{\left\{ \{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{x_k\}_{k=1}^K, \right\} \\ \left\{ \{y_k\}_{k=1}^K, x_m, y_m, \{p_k\}_{k=1}^K, \lambda \right\}}} \mathcal{L}(\mathrm{E}), \qquad (20)$$

where the duality problem for $\mathcal{P}_4$ is as follows:

$$\mathcal{P}_5 : \min_{\{A, \beta, \gamma, N_1, N_2\}} g(A, \beta, \gamma, N_1, N_2). \qquad (21)$$

Since $\mathcal{P}_4$ is a convex problem and satisfies the Slater's condition [38], the optimal solution of $\mathcal{P}_4$ is equal to the solution of $\mathcal{P}_5$. This means that the optimal solution of $\mathcal{P}_4$ with fixed q can be obtained by iteratively solving two optimization problems: the primary variable optimization that maximizes $\mathcal{L}(\mathrm{E})$ on $(\{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{x_k\}_{k=1}^K, \{y_k\}_{k=1}^K, x_m, y_m, \{p_k\}_{k=1}^K, \lambda)$ and the dual variable optimization that minimizes $\mathcal{L}(\mathrm{E})$ on $(A, \beta, \gamma, N_1, N_2)$.

By calculating the partial derivative of the Lagrangian function to each parameter variable and making it equal to zero, we can obtain the optimal values of the partial parameter variables as follows:

$$f_k^* = \left[\frac{3\gamma_k}{2 \left(\frac{1+A_0}{C_{cpu}^k} + \beta_k (f_{\max}^k)^2 \right)} \right]^+ = \left[\left(\frac{\gamma_k}{2 ((q + A_k) \varepsilon_k + \beta_k)} \right)^{\frac{1}{3}} \right]^+, \qquad (22)$$

$$f_m^* = \left[\frac{3\beta_0}{2 \left(\gamma_0 (f_{\max}^m)^2 + N_1 \right)} \right]^+ = \left[\left(\frac{\beta_0}{2 (q\varepsilon_m + \gamma_0)} \right)^{\frac{1}{3}} \right]^+. \qquad (23)$$

where $[x]^+ = \max \{x, 0\}$. For the purpose of initializing the Lagrange factor as little as possible, the following observations can be derived from the above equations:

$$\gamma_k = \left[\frac{\left(\frac{2}{3} \left(\frac{1+A_0}{C_{cpu}^k} + \beta_k (f_{\max}^k)^2 \right) \right)^3}{2 ((q + A_k) \varepsilon_k + \beta_k)} \right]^{\frac{1}{2}}, \qquad (24)$$

$$\beta_0 = \left[\frac{\left(\frac{2}{3} \left(\gamma_0 (f_{\max}^m)^2 + N_1 \right) \right)^3}{2 (q\varepsilon_m + \gamma_0)} \right]^{\frac{1}{2}}. \qquad (25)$$

Meanwhile, we can also get an expression about the optimal offload power:

$$\sum_{k=1}^{K} p_k^* h_k = \frac{N_2 B h_k}{(q + A_k) \ln 2} - B\sigma^2. \qquad (26)$$

From the above formula, we can observe that the MD will opt to directly offload the task data only when the channel condition between the MD and the MEC server satisfies the condition $N_2 h_k - (q + A_k) \sigma^2 \ln 2 > 0$. In other words, the MD will choose direct offloading only when the channel conditions are favorable, indicating that the quality of the channel plays a critical role in the decision-making process. Next, we compute the partial derivative of the function $\mathcal{L}(\mathrm{E})$ with respect to λ, yielding the following conclusions:

$$A_0 = N_1 C_{cpu}^m + N_2 - 1. \qquad (27)$$

In this paper, we regard $\sum_{k=1}^{K} p_k$ as a variable. In doing so, there are two advantages: i) It can reduce the complexity of solving the power optimal value; ii) Since in the end we give the optimal solution of power sum, that is to say, in reality, due to the heterogeneity of each device node, in the case of giving the optimal power sum, it can be adjusted according to the own resources of each device node, resource-rich equipment nodes can increase some power, and nodes with less own resources can appropriately reduce power, only need to ensure the optimization of power sum, and in

order to be able to find the optimal value of $\sum_{k=1}^{K} p_k$, we find the partial derivative of $\mathcal{L}(\mathrm{E})$ about τ_o, as follows:

$$
\frac{\partial L}{\partial \tau_o} = -q \sum_{k=1}^{K} (p_k + p_{a,k}) - \sum_{k=1}^{K} A_k (p_k + p_{a,k}) - \beta_0
$$
$$
+ N_2 B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k h_k}{\sigma^2}\right) = 0. \tag{28}
$$

In order to find the optimal value of $\sum_{k=1}^{K} p_k$, and due to the randomness of A_k. We set the values of A_k $(k = \{1, 2, \cdots, K\})$ to be equal, denoted as $A_k = A_1$, then the optimal value of $\sum_{k=1}^{K} p_k$ is as follows:

$$
\sum_{k=1}^{K} p_k^* = \frac{N_2 B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k^* h_k}{B \sigma^2}\right) - \beta_0 - (q + A_1) \sum_{k=1}^{K} p_{a,k}}{q + A_1}. \tag{29}
$$

When both β_0 and γ_k greater than zero, through the complementary slackness condition, we can obtain:

$$
\begin{cases} t_k^* = T & (30) \\ t_e^* + t_o^* + t_c^* = T & (31) \end{cases}
$$

Remark: Through (22) and (23), we can get that the system CEE is inversely proportional to the calculation frequency of the MD and MEC servers. Therefore, we can improve the system CEE by appropriately reducing the calculation frequency of both. Through (29), we can get the inverse ratio of system CEE to the sum of offloading powers of all MDs. Similarly, we can also enhance the system CEE by appropriately reducing the offloading power of the MD. Through (30), we can get that, within each slot, each MD spends the entire slot for local processing to maximize the system CEE.

Theorem 3. *When the two items represented by λ are equal, the system CEE can take the maximum value, i.e., for MD backscattering and direct unloading of task data, the MEC server should complete the processing of these task data within a given period of time.*

Proof. If **Theorem 3** does not hold, then it means that when the system CEE takes the maximum, the two terms represented by λ can not be equal, then we assume that $\{\alpha_k^*\}_{k=1}^{K}, t_e^*, t_o^*, t_c^*, \{t_k^*\}_{k=1}^{K}, \{f_k^*\}_{k=1}^{K}, f_m^*, \{p_k^*\}_{k=1}^{K}$ is the optimal solution to the optimization problem P_2, and q^* is the maximum CEE of the system, so we can get $B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k^* P_t g_k h_k}{B \sigma^2}\right) + \tau_o^* B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k^* h_k}{B \sigma^2}\right) \neq \frac{\tau_c^* f_m^*}{C_{cpu}^m}$.

We can establish another set of solutions $\{\alpha_k^*\}_{k=1}^{K}, t_e^*, t_o^*, t_c^*, \{t_k^*\}_{k=1}^{K}, \{f_k^*\}_{k=1}^{K}, f_m^\#, \{p_k^*\}_{k=1}^{K}$, which satisfy $B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k^* P_t g_k h_k}{B \sigma^2}\right) + \tau_o^* B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k^* h_k}{B \sigma^2}\right) = \frac{\tau_c^* f_m^\#}{C_{cpu}^m}$, and we make the system CEE of this set of solutions to $q^\#$. We can see that $\frac{\tau_c^* f_m^\#}{C_{cpu}^m} = B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k^* P_t g_k h_k}{B \sigma^2}\right) + \tau_o^* B \log_2 \left(1 + \sum_{k=1}^{K} \frac{p_k^* h_k}{B \sigma^2}\right) < \frac{\tau_c^* f_m^*}{C_{cpu}^m}$, i.e., $f_m^\# < f_m^*$.

It can be seen from (23) that the system CEE is inversely proportional to the calculation frequency of the MEC server, so we can get $q^\# > q^*$, which contradicts the assumption that q^* is the maximum system CEE, so the theorem is verified. $\square$

According to **Theorem 3**, we can obtain the relationship between the backscatter communication duration, the direct offloading duration, and the MEC processing duration, as follows:

$$
t_e^* G + t_o^* H = \frac{t_c^* f_m^*}{C_{cpu}^m}, \tag{32}
$$

where $G = B \log_2 \left(1 + \sum_{k=1}^{K} \frac{\zeta \alpha_k P_t g_k h_k}{B \sigma^2}\right)$ and $H = B \log_2 \left(1 + \frac{1}{B \sigma^2} \sum_{k=1}^{K} p_k^* h_k\right)$. On the basis of (32), we can obtain the relationship between t_o^* and t_c^* as follows:

$$
t_c^* = \frac{TG + t_o^* (H - G)}{G + \frac{f_m^*}{C_{cpu}^m}}. \tag{33}
$$

Given that α_k is embedded within a log function, obtaining an explicit optimal expression for α_k presents a challenge. Nonetheless, it becomes evident that the objective function exhibits concavity concerning α_k with respect to α_k. Thus, our approach involves the initialization of α_k followed by iterative updates through gradient adjustments, enabling convergence towards the optimal value in each iteration. Apart from updating α_k through gradients, achieving the optimal solution for the $\mathcal{P}_4$ problem necessitates gradient-based updates for the associated Lagrangian factor. However, we remain unable to derive an optimal formulation for the durations associated with different phases.

5.2 Dinkelbach Algorithm based on Backscatter Coefficient Update (DBCU)

In order to solve the optimal duration of each phase, according to different user requirements, we design the following two iterative methods by employing the Dinkelbach algorithm based on backscatter coefficient update.

5.2.1 DBCU-I

This method is designed to meet minimum processing data requirements.

We attempt to reduce the energy consumption of the system, while satisfying the user's request for minimal amount of task data, which is formulated as follows:

$$
\lambda + \sum_{k=1}^{K} \frac{x_k}{C_{cpu}^k} = L_{\min} \tau_e. \tag{34}
$$

With variable substitution, we get:

$$
L_{\min} = t_e G + t_o H + \sum_{k=1}^{K} \frac{t_k f_k}{C_{cpu}^k}. \tag{35}
$$

And from (30) and (31), we can obtain:

$$
t_c G = TG + t_o (H - G) + \sum_{k=1}^{K} \frac{t_k f_k}{C_{cpu}^k} - L_{\min}. \tag{36}
$$

$$\mathcal{L}\left(\mathrm{E}^2\right) = \lambda + \sum_{k=1}^{K} \frac{x_k}{C_{cpu}^k} - q\left(P_t + \sum_{k=1}^{K} p_{c,k} + \sum_{k=1}^{K}\left(p_k\tau_o + p_{a,k}\tau_o\right) + \varepsilon_m y_m + P_s\left(\tau_c + \tau_e\right) + \sum_{k=1}^{K}\varepsilon_k y_k - \sum_{k=1}^{K}\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)\right)$$

$$+ \sum_{k=1}^{K} A_k\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k} - p_{c,k} - p_k\tau_o - p_{a,k}\tau_o - \varepsilon_k y_k\right) + \beta_0\left(T\tau_e - 1 - \tau_o - \sqrt{\frac{x_m^3}{y_m}}\right) + \gamma_0\left(x_m(f_{\max}^m)^2 - y_m\right)$$

$$+ \sum_{k=1}^{K}\gamma_k\left(T\tau_e - \sqrt{\frac{x_k^3}{y_k}}\right) + \sum_{k=1}^{K}\beta_k\left(x_k\left(f_{\max}^k\right)^2 - y_k\right) + N_1\left(x_m - \lambda C_{cpu}^m\right) + N_2\left(B\log_2\left(1 + \sum_{k=1}^{K}\frac{\zeta\alpha_k P_t g_k h_k}{B\sigma^2}\right) + \tau_o B\log_2\left(1 + \sum_{k=1}^{K}\frac{p_k h_k}{B\sigma^2}\right) - \lambda\right).$$

$$(40)$$

Therefore, we can find the optimal duration of each phase as follows:

$$t_c^* = \frac{\left(L_{\min} - \sum_{k=1}^{K}\frac{t_k^* f_k^*}{C_{cpu}^k}\right)C_{cpu}^m}{f_m^*}, \tag{37}$$

$$t_o^* = \frac{t_c^*\left(G + \frac{f_m^*}{C_{cpu}^m}\right) - TG}{H - G}, \tag{38}$$

$$t_e^* = T - t_o^* - t_c^*. \tag{39}$$

5.2.2 DBCU-II

This method is designed to maximize the throughput of the system without considering energy consumption.

In this case, each MD will be exactly exhausted in each time slot to maximize the data throughput of the system, then the minimum completion of the calculation data per round constraint should be lifted, that is, the Lagrange function of the optimization constraint problem in this case is $\mathcal{L}\left(\mathrm{E}^2\right)$, where $\mathrm{E}^2 = (\{\alpha_k\}_{k=1}^K, \tau_e, \tau_o, \tau_c, \{x_k\}_{k=1}^K, \{y_k\}_{k=1}^K, x_m, y_m, \{p_k\}_{k=1}^K, \lambda, A(k \neq 0), \beta, \gamma, N_1, N_2)$.

In contrast to the original Lagrangian function $\mathcal{L}(\mathrm{E})$, under this model we take $A_0 = 0$, by calculation we can get the expression of the remaining variables unchanged, through MD to use up the energy collected in each time slot, and we can get the following conclusions (for each MD):

$$\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k} = p_{c,k} + p_k\tau_o + p_{a,k}\tau_o + \varepsilon_k y_k. \tag{41}$$

And because we make $A_k = A_1$, we can come to the formula as follows:

$$\sum_{k=1}^{K}\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)$$
$$= \sum_{k=1}^{K} p_{c,k} + \sum_{k=1}^{K}\left(p_k\tau_o + p_{a,k}\tau_o\right) + \sum_{k=1}^{K}\varepsilon_k y_k. \tag{42}$$

With variable substitution, we can get:

$$t_e\sum_{k=1}^{K}\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)$$
$$= t_e\sum_{k=1}^{K} p_{c,k} + \sum_{k=1}^{K}\left(p_k t_o + p_{a,k} t_o\right) + \sum_{k=1}^{K}\varepsilon_k(f_k)^3 t_k. \tag{43}$$

And from (30) and (31), we can further obtain:

$$t_o^* = \frac{T\left(C - \sum_{k=1}^{K} p_{c,k}\right) - \sum_{k=1}^{K}\varepsilon_k\left(f_k^*\right)^3 t_k^* - \frac{TG\left(C-\sum_{k=1}^{K} p_{c,k}\right)}{G + \frac{f_m^*}{C_{cpu}^m}}}{\left(C - \sum_{k=1}^{K} p_{c,k}\right)\left(1 + \frac{H-G}{G + \frac{f_m^*}{C_{cpu}^m}}\right) + \left(\sum_{k=1}^{K} p_k^* + \sum_{k=1}^{K} p_{a,k}\right)}, \tag{44}$$

where $C = \sum_{k=1}^{K}\left(\frac{c_k(1-\alpha_k)P_t g_k + d_k}{(1-\alpha_k)P_t g_k + v_k} - \frac{d_k}{v_k}\right)$.

5.2.3 Computational Complexity

Within the internal iteration, the computational complexities of updating f_k, τ_k, and p_k are linear with respect to the number of MDs, denoted as K. Additionally, the complexity of updating the variables using the gradient method is $O\left(K^2\right)$, considering a total of $3K + 4$ variables. If we have N external iterations required for convergence, the overall complexity of **Alg. 2** amounts to $O\left(NK^3\right)$.

Algorithm 2 Dinkelbach algorithm based on backscatter coefficient update for $\mathcal{P}_4$

1: Set $q = 0$, and initialize individual Lagrangian factors;
2: **while** true **do**
3:     Use DBCU-I or DBCU-II to calculate the optimal values of $f_k, f_m, \tau_c, \tau_o, \tau_e, \tau_k, p_k$ for a given q value
4:     Calculate a new CEE q^+;
5:     **if** $q^+ > q$ **then**
6:         Set the loss function $= |q^+ - q|$, and update α_k by $\alpha_k = \alpha_k - \eta\frac{\partial loss}{\partial \alpha_k}$; Let $q = q^+$.
7:     **else**
8:         $A = A - \eta_1\frac{\partial\mathcal{L}(\mathrm{E})}{\partial A}; \beta = \beta - \eta_1\frac{\partial L(\mathrm{E})}{\partial \beta}$;
9:         $\gamma = \gamma - \eta_1\frac{\partial\mathcal{L}(\mathrm{E})}{\partial \gamma}; N_1 = N_1 - \eta_1\frac{\partial\mathcal{L}(\mathrm{E})}{\partial N_1}$;
10:        $N_2 = N_2 - \eta_1\frac{\partial\mathcal{L}(\mathrm{E})}{\partial N_2}$;
11:     **end if**
12: **end while**

---

6 PERFORMANCE EVALUATION

In this section, we conduct simulation-based experiments to evaluate the performance of the proposed algorithm and investigate its effectiveness and superiority.

6.1 Parameter Setting

Unless explicitly specified, the basic parameters utilized for simulations are outlined in Table 3. In our approach, the standard power loss propagation model is adopted to represent the channel gains of two distinct links: the PB to the k-th MD link and the k-th MD to the MEC server link. Specifically, the channel gains are defined as $g_k = g_k' d_{0k}^{-\beta}$

and $h_k = h'_k d_{1k}^{-\beta}$, where g'_k and h'_k denote the corresponding small-scale fading, d_{0k} and d_{1k} represent the distances from the k-th MD to the PB and the MEC server, respectively, and β is the path loss exponent. Herein, we set $\beta = 3$, $d_{0k} = 5$ m, and $d_{1k} = 50$ m. Furthermore, we set $c_k = 2.463$, $d_k = 1.635$ and $v_k = 0.836$. Additionally, the performance gap denoted as ζ, between the backscatter circuit and the direct transmission circuit, is established at -15 dB.

TABLE 3: Parameter Settings

Notation	Parameters	Value
T	The entire time block	1 second
B	The communication bandwidth	1 MHz
P_t	The PB's maximum transmit power	3.0 W
K	No. of MDs	4
η	Learning rate of α_k	0.1
η_1	Learning rate of Lagrangian factor	0.01
ε_k	The ECC (k_{th} MD)	10^{-26}
ε_m	The ECC (MEC)	10^{-28}
$f_{\max}^k$	Maximum CPU frequency (k_{th} MD)	10^8 Hz
$f_{\max}$	Maximum CPU frequency (MEC)	10^9 Hz
$L_{\min}$	The minimum computation bits	4.8×10^5 bit
C_{cpu}^m	No. of CPU cycles for 1 bit data (MEC)	1000
C_{cpu}^k	No. of CPU cycles for 1 bit data (k_{th} MD)	1000
$p_{a,k}$	The power of offloading circuit (k_{th} MD)	0.0001 W
$p_{c,k}$	The power of backscatter circuit (k_{th} MD)	0.0001 W
P_s	The power to decode and eliminate (MEC)	0.001 W

6.2 Baselines

We compare the proposed DBCU-I and DBCU-II with four baselines in terms of system CEE, while keeping other conditions consistent, which are defined as follows:

- *0.5*T local computing*: In this strategy, we take only half of the entire time slot for the duration of the local calculation.
- *tc random computing*: In this strategy, we employ a random sampling technique for the selection of t_c in each time slot.
- *to random computing*: In this strategy, we employ a random sampling technique for the selection of t_o in each time slot.
- *te random computing*: In this strategy, we employ a random sampling technique for the selection of t_e in each time slot.

There are two different parameter update orders for different backscatter coefficients α_k and Lagrangian factors, which are defined as follows:

- **Simultaneously**: A gradient update to the backscatter coefficient α_k is followed by gradient descent to the Lagrangian factor.
- **Asynchronously**: A non-simultaneous update way, i.e., after updating the gradient of the backscatter coefficient α_k and reaching a good solution to the dual problem, we update the Lagrangian factor and gradually approach the optimal solution of the original problem.

6.3 Performance Comparison

Through comparative experimental results as shown in Fig. 4, we can see that taking the first way of updating asynchronously can obtain better system performance. When the

original problem adheres to the strong duality principle, its optimal solution coincides with that of the dual problem, i.e., the saddle point of the dual problem [39]. The general saddle point search is generally the case of fixing one or several dimensions, optimizing the other dimensions, and then optimizing the fixed dimensions to realize the saddle point. Therefore, choosing the asynchronous update method is in line with the process of finding the saddle point, and choosing the synchronous update method will be more difficult to find the saddle point.

For simultaneous update methods, the entire optimization process is full of uncertainty. As shown in Fig. 4b, in the case of DBCU-II, the simultaneous update method does not result in greater system throughput. However, by comparing the changes in the total energy consumption of the system in Fig. 4c, we can see that the DBCU-II under simultaneous update brings an increase in the total energy consumption of the system compared with the DBCU-II under the asynchronous update. In summary, it can be seen from Fig. 4 that both models can obtain better system performance by using the asynchronous update method. Therefore, this paper adopts the asynchronous update method to find the optimal solution to the original problem.

As shown in Fig. 5, we study the trend of system CEE under different numbers of DBCU-I and DBCU-II, and the experimental results show that as the number of MD increases, the CEE of the system also increases.

As shown in Fig. 6, we study the trend of system CEE under different $\varepsilon_k/\varepsilon_m$ ratios for DBCU-I and DBCU-II. Through the analysis facilitated by two distinct models, we can draw the same conclusion: a rise in the ratio of $\varepsilon_k/\varepsilon_m$ results in a reduction in the system CEE. This is because as the ratio increases, the energy consumed by the local computation also increases. Meanwhile, ε_m remains unaltered, which in turn leads to an increase in the total energy consumption of the system, and eventually leads to a decrease in the system CEE.

As illustrated in Fig. 7, we investigate the trend of system CEE for DBCU-I and DBCU-II under different MD-BS distances. Through different models, we can conclude that as the MD-PB distance decreases, the system CEE also experiences a decline. This phenomenon finds a preliminary explanation through Eq.(26), where the size of the MD-PB distance directly affects the channel state between MD and PB. Eq. (26) further demonstrates that under equivalent conditions, the relationship between system CEE and MD-PB channel state is inversely proportional. Specifically, as shown in $h_k = h'_k d_{1k}^{-\beta}$, it becomes evident that a reduction in MD-PB distance leads to amplified channel gain between MD and PB, which in turn leads to in a decrease in the overall system CEE.

In Fig. 8, we delve into a comparison between the achievable data throughput using backscatter communication offloading and direct offloading within a single time slot. It becomes evident that the direct offloading mode yields higher data throughput. This discrepancy arises because the backscatter communication offloading method involves local storage of a portion of the received energy signal within the MD through WPT. Therefore, the data throughput achievable via backscatter communication offloading remains lower than that achieved through direct offloading.

(a) CEE   (b) Data throughput   (c) Energy consumption
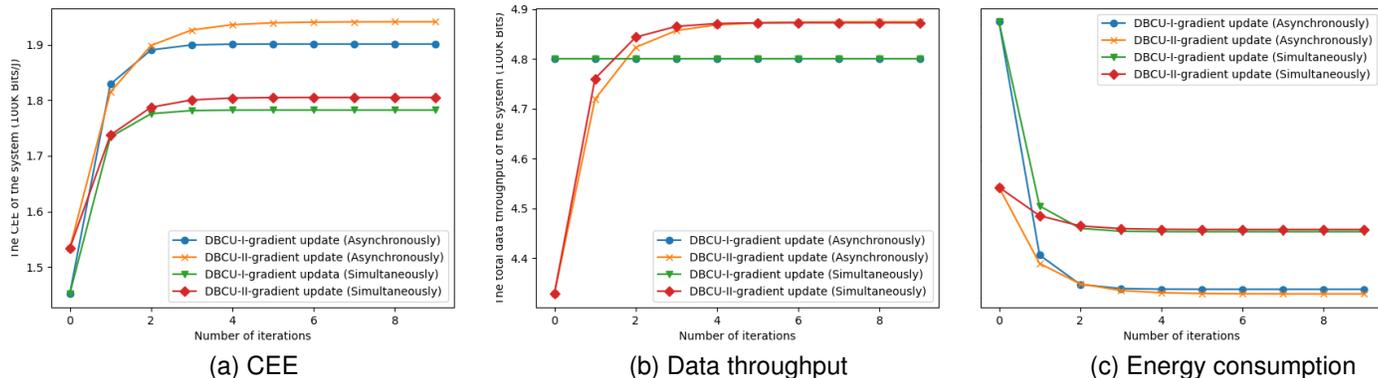
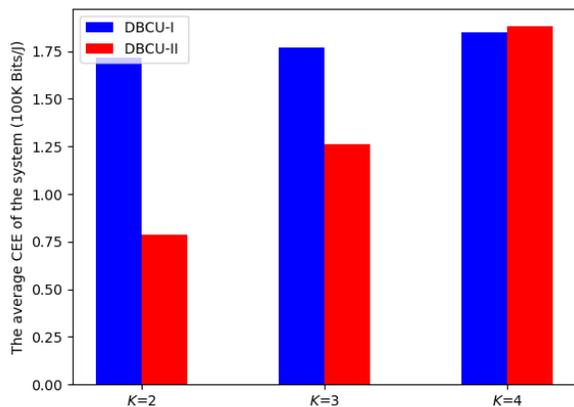Fig. 4: Performance comparison of two models with different update times

Fig. 5: System CEE under different numbers of MD

Additionally, a comparison of two distinct models reveals an intriguing insight: DBCU-II, which releases the stored energy within each time slot, outperforms DBCU-I in terms of data throughput potential.

As shown in Fig. 9, we compare the trend of CEE of the system of DBCU-I and DBCU-II under the communication model OMA. We can clearly see that superior system performance can be achieved through the implementation of DBCU-I and DBCU-II within the NOMA communication model.

As shown in Fig. 10, we attempt to explore the effect of $L_{\min}$ on the system CEE. We exclusively vary the value of $L_{\min}$ to discern its individual impact on the system CEE, maintaining all other conditions unchanged. This figure clearly reveals the trend of DBCU-I: with the increase of $L_{\min}$, there is a corresponding increase in the system CEE. This observation demonstrates a direct and proportional relationship between $L_{\min}$ and the system CEE, within the specified constraints.

To demonstrate the superiority of our proposed DBCU-I and DBCU-II approaches, we conduct a comparative analysis against the four aforementioned algorithms in terms of system CEE under the same initial conditions. As shown in Fig. 11, we can see that the system performance obtained by the algorithm of using only half of the time slot length for local calculation is far inferior to the algorithm that directly uses the entire time slot as the local calculation time. This observation serves to corroborate the rationale behind our

direct utilization of the entire time slot duration for MD local processing. Moreover, the results highlight the enhanced system performance achieved by both DBCU-I and DBCU-II in contrast to the alternative algorithms under investigation.

It is important to note that while DBCU-I accounts for the constraint that the energy consumption of each MD in a given time slot does not exceed the energy harvested during that same time slot, it does not encompass the utilization of any residual energy from the current round into the resource allocation process for the next time slot. As such, there exists room for further enhancement in the DBCU-I approach. Therefore, on the basis of the error gradient descent update method adopted by both models, the following conclusions can be drawn: i) For users whose primary goal is to reduce energy consumption, DBCU-I can be used for resource allocation and offloading decisions; ii) For users whose primary goal is to increase the overall throughput of the system, DBCU-II can be used for resource allocation and offloading decisions.

All in all, from the closed solutions of the various parameters, we can get many favorable insights as follows:

- The system CEE increases with the decrease of the calculation frequency of the MD local processor and the MEC server.
- In order to obtain the maximum system CEE, the total amount of backscattering and direct offloading of all MDs should be equal to the maximum amount of data that the MEC server can process in the MEC server processing phase, and the MD should process the task data in the entire time slot.
- For the selection of offloading strategies, only when the communication channel state is better, the MD will choose to directly offload the task data.
- We can increase the CEE of the system in the case of DBCU-I by appropriately increasing L_{min}.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we investigate the problem of maximizing the system CEE of the wireless-powered BC-MEC network based on NOMA. In view of the current local energy shortage of MD, we introduce WPT technology to improve the endurance of MD. We jointly optimize the backscatter coefficient of each MD, the backscatter communication duration,
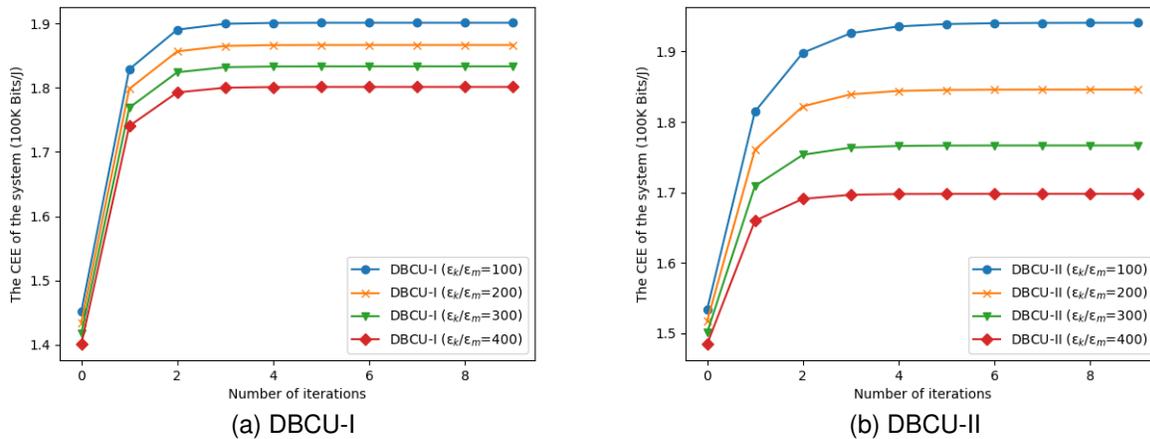
(a) DBCU-I

(b) DBCU-II

Fig. 6: System CEE undet different $\varepsilon_k/\varepsilon_m$ ratios

(a) DBCU-I

(b) DBCU-II
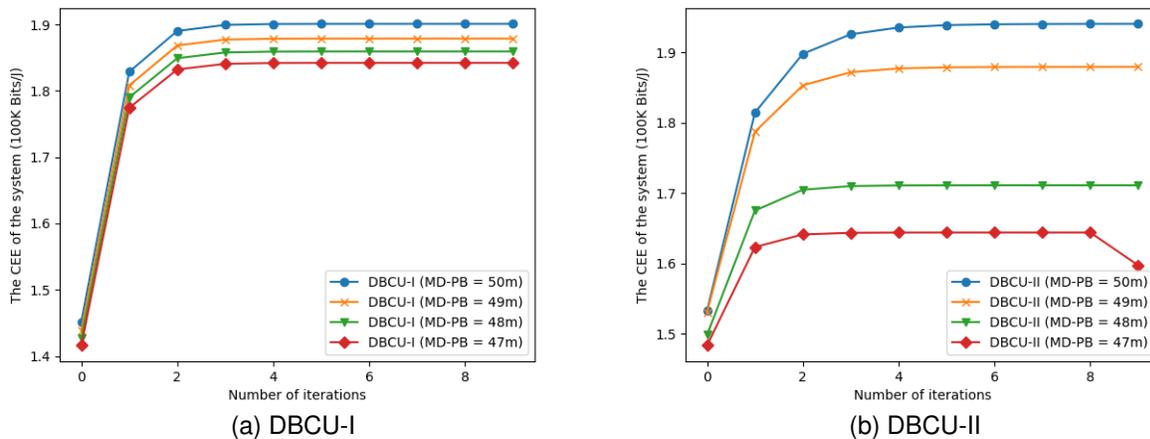
Fig. 7: System CEE at different MD-PB distances
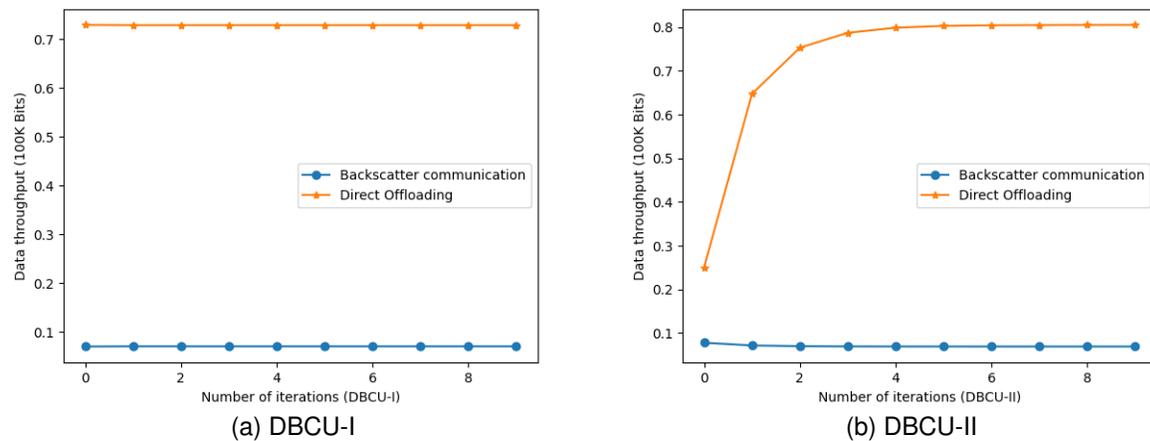

(a) DBCU-I

(b) DBCU-II

Fig. 8: The amount of offloaded data for different offloading methods

the direct offloading duration, the MEC server processing time, the local processing time, the direct offloading power of each MD, the calculation frequency of the MEC server, and the local calculation frequency of each MD to maximize the system CEE. In order to solve the joint optimization fraction problem, we design the Dinkelbach algorithm based on the backscatter coefficient update, making it more suitable for the application scenarios in this paper. Meanwhile, we propose DBCU-I and DBCU-II, aimed at finding closed-form

solutions for optimizing individual parameter variables based on different system requirements. Compared with other algorithms, the performance of these two methods is at least 10% higher under the same conditions, which reflects the superiority of our approach.

Furthermore, as we mentioned in DBCU-I, the energy that is not consumed in one time slot can be stored in the MD's battery, and the MD can optimize the duration selection of the next time slot according to the battery power
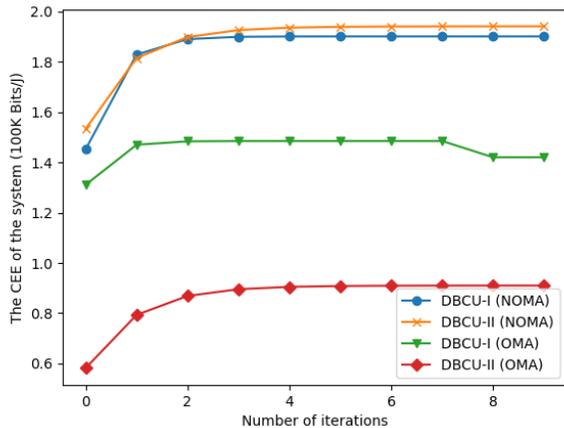
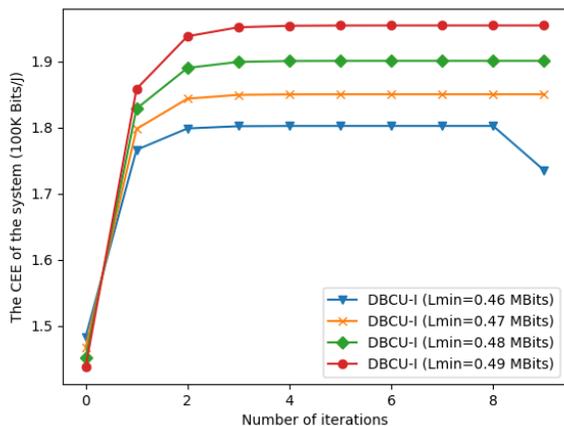Fig. 9: System CEE under different communication models

Fig. 10: System CEE under different $L_{\min}$ (DBCU-I)

situation, as well as the power consumption strategy, and then boost the performance of the entire system, which can be reflected in future work.

ACKNOWLEDGMENT

REFERENCES

[1] K. W. Choi, L. Ginting, A. A. Aziz, D. Setiawan, J. H. Park, S. I. Hwang, D. S. Kang, M. Y. Chung, and D. I. Kim, "Toward realization of long-range wireless-powered sensor networks," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 184–192, 2019.

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

[3] S. Mao, S. Leng, K. Yang, X. Huang, and Q. Zhao, "Fair energy-efficient scheduling in wireless powered full-duplex mobile-edge computing systems," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[4] L. Ji and S. Guo, "Energy-efficient cooperative resource allocation in wireless powered mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4744–4754, 2019.

[5] P.-Q. Huang, Y. Wang, K. Wang, and Q. Zhang, "Combining lyapunov optimization with evolutionary transfer optimization for long-term energy minimization in irs-aided communications," *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2647–2657, 2022.

[6] L. Huang, X. Feng, A. Feng, Y. Huang, and L. P. Qian, "Distributed deep learning-based offloading for mobile edge computing networks," *Mobile networks and applications*, pp. 1–8, 2018.

[7] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.

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

[9] M. Zeng, R. Du, V. Fodor, and C. Fischione, "Computation rate maximization for wireless powered mobile edge computing with noma," in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–9.

[10] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2020.

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

[12] Y. Xie, Z. Xu, Y. Zhong, J. Xu, S. Gong, and Y. Wang, "Backscatter-assisted computation offloading for energy harvesting iot devices via policy-based deep reinforcement learning," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, 2019, pp. 65–70.

[13] G. Chen, Q. Wu, W. Chen, D. W. K. Ng, and L. Hanzo, "Irs-aided wireless powered mec systems: Tdma or noma for computation offloading?" *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1201–1218, 2022.

[14] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier fdma for uplink wireless transmission," *IEEE vehicular technology magazine*, vol. 1, no. 3, pp. 30–38, 2006.

[15] Y. Ye, L. Shi, X. Chu, R. Q. Hu, and G. Lu, "Resource allocation in backscatter-assisted wireless powered mec networks with limited mec computation capacity," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 678–10 694, 2022.

[16] Y. Xu, B. Gu, R. Q. Hu, D. Li, and H. Zhang, "Joint computation offloading and radio resource allocation in mec-based wireless-powered backscatter communication networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6200–6205, 2021.

[17] C. Zheng, W. Zhou, and X. Lu, "Energy efficiency maximization in the wireless-powered backscatter communication networks with df relaying," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[18] 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 Network*, vol. 34, no. 5, pp. 106–113, 2020.

[19] Y. Ye, R. Q. Hu, G. Lu, and L. Shi, "Enhance latency-constrained computation in mec networks using uplink noma," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2409–2425, 2020.

[20] M. Sheng, Y. Dai, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Delay-aware computation offloading in noma mec under differentiated uploading delay," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2813–2826, 2020.

[21] X. Pang, J. Tang, N. Zhao, X. Zhang, and Y. Qian, "Energy-efficient design for mmwave-enabled noma-uav networks," *Science China Information Sciences*, vol. 64, pp. 1–14, 2021.

[22] N. Zhao, Y. Li, S. Zhang, Y. Chen, W. Lu, J. Wang, and X. Wang, "Security enhancement for noma-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3994–4005, 2020.

[23] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1091–1104, 2014.

[24] X. Lu, D. Niyato, H. Jiang, E. Hossain, and P. Wang, "Ambient backscatter-assisted wireless-powered relaying," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 4, pp. 1087–1105, 2019.

[25] N. Van Huynh, D. T. Hoang, X. Lu, D. Niyato, P. Wang, and D. I. Kim, "Ambient backscatter communications: A contemporary
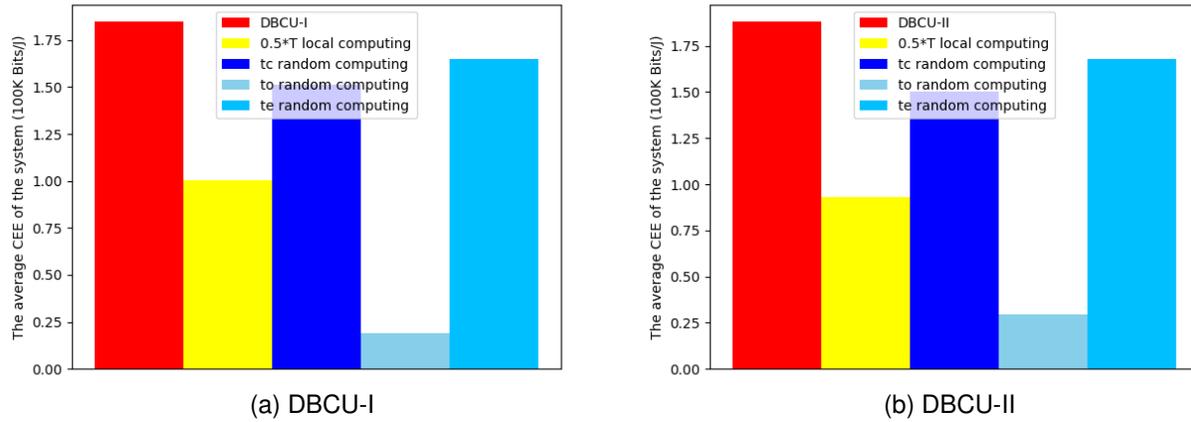
(a) DBCU-I

(b) DBCU-II

Fig. 11: System CEE under different algorithms

survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2889–2922, 2018.

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

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

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

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

[30] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[31] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji, and Z. Shi, "Optimal sic ordering and computation resource allocation in mec-aware noma nb-iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2806–2816, 2019.

[32] D. Li and Y.-C. Liang, "Price-based bandwidth allocation for backscatter communication with bandwidth constraints," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5170–5180, 2019.

[33] Y. Chen, N. Zhao, and M.-S. Alouini, "Wireless energy harvesting using signals from multiple fading channels," *IEEE Transactions on Communications*, vol. 65, no. 11, pp. 5027–5039, 2017.

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

[35] W. Dinkelbach, "On nonlinear fractional programming," *Management science*, vol. 13, no. 7, pp. 492–498, 1967.

[36] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[37] K. Shen and W. Yu, "Fractional programming for communication systems—part ii: Uplink scheduling via matching," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2631–2644, 2018.

[38] S. Zargari, C. Tellambura, and S. Herath, "Energy-efficient hybrid offloading for backscatter-assisted wirelessly powered mec with reconfigurable intelligent surfaces," *IEEE Transactions on Mobile Computing*, 2022.

[39] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Junhui Du received the BSc degree in mathematics from the Nanjing University of Information Science Technology, China, in 2021. He is currently working toward the master's degree in mathematics with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include Internet of Things, deep learning, and mobile edge computing.

Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees from Harbin Institute of Technology, China in 2009 and 2011, respectively, both in electrical engineering. He received the Ph.D. degree with the highest honor in computer science at Freie Universität Berlin, Germany in 2015. He is currently an associate professor at the Center for Applied Mathematics, Tianjin University, China. His research interests include mobile cloud computing, edge computing, internet of things, deep learning, complex network, and DNA storage.

Minxian Xu (Member, IEEE) is currently an Associate Professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. He received the BSc degree in 2012 and the MSc degree in 2015, both in software engineering from University of Electronic Science and Technology of China. He obtained his PhD degree from the University of Melbourne in 2019. His research interests include resource scheduling and optimization in cloud computing. He has co-authored 40+ peer-reviewed papers published in prominent international journals and conferences, such as ACM CSUR, ACM TOIT, IEEE TSUSC, IEEE TCC, IEEE TASE, IEEE TGCN, JPDC, JSS and ICSOC. His Ph.D. Thesis was awarded the 2019 IEEE TCSC Outstanding Ph.D. Dissertation Award. More information can be found at: minxianxu.info.

Rajkumar Buyya (Fellow, IEEE) is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He served as a Future Fellow of the Australian Research Council during 2012-2016. He received the Ph.D. degree in Computer Science and Software Engineering from Monash University, Melbourne, Australia, in 2002. He has authored over 850 publications and seven text books. He is one of the highly cited authors in computer science and software engineering worldwide (h-index=155, gindex=334, 126,300+ citations). He is recognized as a "Web of Science Highly Cited Researcher" during 2016-2021 by Thomson Reuters.