Zheng and Zhou *J Wireless Com Network* (2022) 2022:23
https://doi.org/10.1186/s13638-022-02097-4

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Computation bits maximization in backscatter-assisted wireless-powered NOMA-MEC networks

Chuangming Zheng* and Wengang Zhou

*Correspondence:
zhengcmconnect@126.com
School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China

Abstract

In the field of Internet-of-Things (IoT), how to enhance the computation capacity and prolong the life span of massive smart IoT nodes is becoming a more practically important but challenging problem. In this paper, we propose a backscatter-assisted wireless-powered non-orthogonal multiple access (NOMA)-mobile edge computing (MEC) network, where IoT nodes harvest energy from signals transmitted by a dedicated energy source and then offload data to the MEC server via backscattering and uplink NOMA. To exploit its potential performance, we formulate a non-convex problem to maximize the total computation bits of all the IoT nodes by jointly optimizing the communication resource of each IoT node and the computation resource of both the IoT nodes and the MEC server. By introducing a slack variable and a series of auxiliary variables as well as using the contradiction and inequality transformation approaches, we transform the original non-convex problem into a convex one and devise an iterative algorithm to optimally solve it. Simulation results are provided to demonstrate the quick convergence of the proposed iterative algorithm and to confirm the superiority of the proposed scheme over the existing benchmark schemes.

Keywords: Backscatter communication, Computation bits, Mobile edge computing, Wireless power transfer

1 Introduction

With the development of Internet of Things (IoT), an increasing number of IoT nodes will be deployed to monitor, sense, collect, and analyze enormous data timely for intelligent services [1]. Due to the limitations of both the production cost and size, most of IoT nodes are typically energy- and computation-constraint, which imposes high requirements on energy and computation resources [2, 3]. In order to address the above challenges, wireless-powered mobile edge computing (MEC) has been proposed by leveraging the ability of wireless power transfer [4] and MEC [5]. To be specific, in wireless-powered MEC, IoT nodes harvest energy from a dedicated energy source (ES) on demand and then use the harvested energy to support their local computation and data offloading. Accordingly, how to schedule communication and computational resources

for energy harvesting (EH), data offloading and computation is vital to the achievable performance of wireless-powered MEC.

Until now, there are many investigations regarding binary or partial offloading-based resource allocation in wireless-powered MEC networks. In the binary offloading, the task bits of an IoT node are either locally computed or offloaded to the MEC server, while in the partial offloading, the task bits of an IoT node are split into two parts: one for local computation and the remaining for data offloading [3]. The authors of [6] studied the successful computation probability maximization problem in a wireless-powered MEC network with a single IoT node by jointly optimizing the binary offloading decision and the IoT node's computation frequency and local computation time. The single IoT user scenario [6] was extended into multi-user wireless-powered MEC, where the authors maximized the weighted sum computation bits under the time-block [7] and time-varying [8] channels, respectively. Considering the partial offloading, the authors in [9] maximized the weighted sum computation bits of all the IoT nodes by jointly optimizing the beamforming of the ES, the IoT nodes' computation frequencies and offloading time and the task bits for data offloading and local computation. By optimizing the same variables as [9], the authors in [10] studied the energy minimization problem while satisfying the data computation requirement of all IoT nodes.

The above works [6–10] assumed that all IoT nodes offload data bits to the MEC server via orthogonal multiple access (OMA). Recall that non-orthogonal multiple access (NOMA) achieves a better transmission performance than OMA in terms of spectrum and energy efficiencies. Recent works have integrated NOMA for data offloading into wireless-powered multi-user MEC and validated its superior performance compared to wireless-powered OMA-MEC. The author in [11] maximized the sum computation bits in a wireless-powered multi-user NOMA-MEC by jointly optimizing the binary offloading decision, local computation frequency, transmit power, EH and data offloading time of each IoT node. Considering the partial offloading in wireless-powered NOMA-MEC, the energy efficiency was maximized from the system centric [12] perspective. The authors in [13] considered a user cooperation enabled wireless-powered NOMA-MEC network and minimized the energy consumed by all the IoT nodes via jointly optimizing the IoT node's EH time and computation frequency, the IoT node's transmit power, and the tasks bits for local computation and offloading.

In the above works [6–13], the data offloading was achieved following the harvest-then-transmit protocol via active radios (ARs) that require power-hungry components such as oscillators [14, 15]. Due to the energy-causality constraint for each IoT node, it has to allocate more resources for EH in each transmission block, leaving a short time for AR based data offloading and limiting task offloading performance of each IoT node. In contrary to AR, backscatter communication (BackCom) is receiving much attention due to its extra-low power consumption and thus, has been considered for data offloading in wireless-powered MEC [16]. The key idea of BackCom is to allow an IoT node modulating and backscattering its information on the incident signal, resulting in an extra-low power consumption but with a lower transmission rate than AR [15]. To exploit the different tradeoffs between power consumption and transmission rate for AR and BackCom, hybrid AR-BackCom has been used for data offloading in wireless-powered MEC networks [17, 18]. The authors in [18]

considered a hybrid AR-BackCom-based wireless-powered MEC and minimized the energy consumption of the ES by jointly optimizing the EH time, AR, and BackCom under the complete offloading where all the IoT nodes' task bits are offloaded to the MEC server. Under the same scenario, the computation energy efficiency of all the IoT nodes was maximized [19]. Considering the partial offloading, the authors in [20] jointly optimized the time for AR, BackCom and EH, the IoTs' transmit power for AR-based offloading, and the portions of task bits for local computation and offloading to maximize the weighted sum computation bits of all IoT nodes. Considering the same network and optimization variables as [20], the max-min computation energy efficiency among IoT nodes and the total delay for data offloading and computing of all IoT nodes were studied in [21] and [22], respectively.

These works [17–22] have validated that leveraging hybrid AR-BackCom for offloading is capable of improving the computational performance of wireless-powered MEC networks. We note that the previous works [6–10] have confirmed that employing NOMA for offloading achieves a better performance than OMA. However, such a advantage has not been exploited in hybrid AR-BackCom wireless-powered MEC since [17, 18, 20–22] only considered OMA for AR-based data offloading, and this motivates this work. Besides, this work also considers the limited computation capacity of the MEC server, which removes the ideal assumption that the MEC server's computation capacity is unlimited in existing works and makes this work more practical.

In this paper, we propose a backscattered-assisted wireless-powered NOMA-MEC network that consists of one ES, multiple IoT nodes and one MEC server, and aim to develop a resource allocation scheme to maximize the total computation bits of all IoT nodes while considering the computation resource allocation at the MEC server. The main contributions are as listed below.

- We formulate an optimization problem to maximize the total computation bits of all IoT nodes by jointly optimizing the EH time, BackCom time, and uplink NOMA time of the IoT node, the power reflection coefficient and the transmit power of each IoT node, the computing frequencies and time at both the MEC server and the IoT nodes. The formulated problem is non-convex and challenging to solve due to the following reasons. First, the optimization of the MEC server's computation time and frequency introduces a min function in the objective function and more coupled optimization variables in both the objective function and constraints. Second, the optimization of the IoT node's transmit power during uplink NOMA results in a difference of convex (DC) structure.
- We first transform the original problem into an equivalent convex one and then propose an iterative algorithm to obtain the optimal solution. To be specific, a slack variable and the contradiction approach are introduced to remove the min function from the objective function and determine partial optimal solutions, respectively. Then, we employ an inequality transformation approach to address the DC structure and obtain a sub-problem with given parameters that are from the used inequality transformation approach. By doing so, the optimal solution can be obtained by solving the sub-problem and update the given parameters iter-

atively. Further, the sub-problem is transformed into a convex one by constructing a series of auxiliary variables to decouple the coupled variables. On this basis, an efficient iterative algorithm is developed for obtaining the optimal solution.

The remainder of the whole paper is organized as follows. The system model is clarified in Sect. 2. In Sect. 3, a system computation bits maximization problem for the backscattered-assisted wireless-powered NOMA-MEC network is formulated by jointly optimizing the computing frequencies and time of the MEC server and the IoT nodes, the EH time, the BackCom time, the uplink NOMA time, etc., and then, based on the contradiction approach, the inequality transformation approach and variable replacement, an efficient iterative algorithm is proposed to obtain the optimal solutions. Simulation results are presented in Sect. 4. The conclusion of this work can be found in Sect. 5.

2 Method

2.1 System model

In this work, we consider a backscatter-assisted wireless-powered NOMA-MEC network as shown in Fig. 1, where K IoT nodes harvest energy from energy signals transmitted by a nearly deployed ES and upload their tasks to the MEC server for computation as well as perform local computation. Each IoT node has separate EH, offloading and computing circuits so that the IoT node can compute tasks locally when offloading tasks

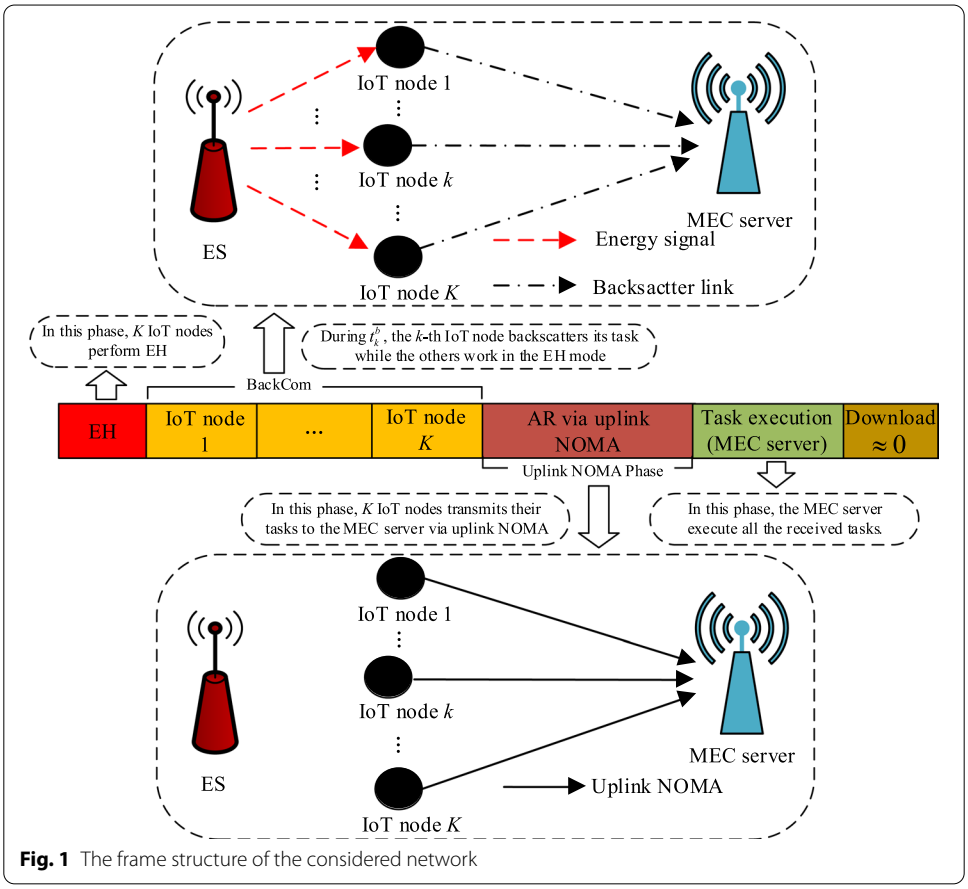

Fig. 1 The frame structure of the considered network

or harvesting energy. Following [4–6], we assume that the consumed energy at each IoT node for task offloading and local computing is less than or equal to its harvested energy so as to avoid drawing energy from its battery. Following [7], we assume that each IoT node can adjust its computing frequency using the dynamic voltage scaling (DVS) technology. We assume that the data bits of each task are bit-wise independent so that the partial offloading scheme is employed at each IoT node to determine how many tasks should be processed locally/offloaded to the MEC server. Assume that the channels from the IoT nodes to the ES or the MEC server are quasi-static fading and remain static within one transmission block but vary among different blocks. Perfect channel state information (CSI) is assumed and available at the MEC server. On this basis, the MEC server can determine the optimal resource allocation scheme and feed it back to all the IoT nodes and the ES at the beginning of each transmission block. Following [6–8], all the devices in the considered network are assumed to be time synchronized.

Let T denote the time duration of the whole transmission block. According to the behaviors of the IoT nodes, the transmission block can be divided into five phases. The first phase is the EH phase, where the ES broadcasts energy signals and all the IoT nodes operate in the EH mode. The second phase is called as the BackCom phase, in which all the IoT nodes take turn to offload their tasks via BackCom and the MEC server can receive the offloading tasks accordingly. The third phase is the uplink NOMA phase, where the ES keeps silent and all the IoT nodes offload tasks to the MEC server simultaneously via uplink NOMA. The fourth phase is the task execution phase. In this phase, all the IoT nodes stop offloading tasks, and the MEC server executes all the received computation tasks to obtain the computation results. These results will be broadcasted to all the IoT nodes in the fifth phase, namely the downloading phase. Following [17, 18, 20–22], we assume that the downloading time of the MEC server is negligible compared with the offloading time and the computing time since the computation results of the MEC server are usually just a few bits in some scenarios, and the transmit power of the MEC server is much larger than IoT nodes. Therefore, the downloading phase of each transmission block will be ignored hereafter.

2.2 EH phase

Let τ_{e} $(0 \leq \tau_{\mathrm{e}} \leq T)$ and P_{s} denote the duration of the EH phase and the ES's transmit power, respectively. This work considers a linear EH model by following the existing works, i.e., [7], where η $(0 < \eta < 1)$ is the energy conversion efficiency and fixed as a constant. Then, the k-th IoT node's harvested energy in this phase can be calculated as $E_k^{\mathrm{h}} = \tau_{\mathrm{e}} \eta P_{\mathrm{s}} g_k$, where g_k is the channel power gain between the ES and the k-th IoT node, and $k \in \mathcal{K} = \{1, 2, ..., K\}$.

2.3 BackCom phase

In order to avoid the mutual interference among different IoT nodes, the BackCom phase will be divided into K sub-phases. In each sub-phase, a specific IoT node performs task offloading via BackCom while the other IoT nodes perform EH. Let ρ_k $(0 \leq \rho_k \leq 1)$ be the power reflection coefficient of the k-th IoT node based on which the received signals at the k-th IoT node can be divided into two parts. In particular, a ρ_k portion of the received signals at the k-th IoT node will be used as the vehicle that carries the k-th IoT

node's information via advanced modulation schemes and then be backscattered to the MEC server for task offloading, while the others are flowed into the EH circuit for EH.

For backscattering, let t_k^{b} denote the BackCom time of the k-th IoT node. Then, the offloading task bits at the k-th IoT node can be computed as

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

where W denotes the channel bandwidth, ξ represents the performance gap reflecting the real modulation [23, 24], h_k is the channel power gain between the MEC server and the k-th IoT node, and σ^2 denotes the noise power spectral density. In this work, the ES serves only as a RF power source, and the RF signal from the ES can be predetermined and known to the MEC server. By using the existing channel estimation methods, the MEC server can obtain the CSI of the ES-to-the MEC server link. Accordingly, the MEC server can remove the interference from the ES by performing successive interference cancellation (SIC).

For EH, both the backscatter IoT node and non-backscatter IoT nodes can perform EH during each sub-phase. Specifically, in 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 are given by $E_k^{\mathrm{b}} = t_k^{\mathrm{b}} \eta (1 - \rho_k) P_{\mathrm{s}} g_k$ and $E_{k,i}^{\mathrm{b}} = t_k^{\mathrm{b}} \eta P_{\mathrm{s}} g_i$, respectively, where η is the energy harvesting efficiency [4]. On this basis, we can calculate the total harvested energy of the k-th IoT node at the end of the BackCom phase as

$$\begin{aligned} E_k^{\mathrm{tot}} &= E_k^{\mathrm{h}} + E_k^{\mathrm{b}} + \sum_{i \in \mathcal{K}, i \neq k} E_{i,k}^{\mathrm{b}} \\ &= \eta P_{\mathrm{s}} g_k \left(\tau_{\mathrm{e}} + \sum_{i=1}^{K} t_i^{\mathrm{b}}\right) - t_k^{\mathrm{b}} \eta \rho_k P_{\mathrm{s}} g_k. \end{aligned} \tag{2}$$

As for the energy consumption in this phase, we assume a constant circuit power consumption for BackCom by following [20] and [21], where the energy consumption of BackCom is the product of the power consumption rate and the BackCom time. In particularly, let $P_{\mathrm{c},k}$ denote the circuit power consumption for BackCom. Then, the energy consumption of the k-th IoT node in this phase is determined by $P_{\mathrm{c},k} t_k^{\mathrm{b}}$.

2.4 Uplink NOMA phase

Let t_{a} denote the duration of the uplink NOMA phase. Following [12], we assume that the channel gains between the IoT nodes and the MEC server are ranked in an descending order, i.e., $h_1 \geq h_2 \geq \cdots \geq h_K$ and the MEC server adopts a fixed decoding order to decode the offloading tasks of each IoT node in this phase by means of SIC technology. Specifically, the MEC server decodes the information of the i-th IoT node first, subtracts it from the received composite signal and then, continues to decode the information from the $(i+1)$-th IoT node, until the information from all the IoT nodes is decoded. Note that such a decoding order allows decoding the weakest IoT node's information without interference, thus, maximizing the sum uplink offloading task bits. Including the decoding order in the joint optimization may further improve the system computation bits, while it is outside the scope of this work and will be studied in our future work. Denote p_k as the transmit power

of the k-th IoT node in the uplink NOMA phase. Then, the achievable computation bits of the k-th IoT node in this phase is given by

$$R_k^{\mathrm{a}} = \begin{cases} W t_{\mathrm{a}} \log_2\left(1 + \frac{p_k h_k}{\sum_{i=k+1}^{K} p_i h_i + W\sigma^2}\right), & \text{if } k \le K-1, \\ W t_{\mathrm{a}} \log_2\left(1 + \frac{p_K h_K}{W\sigma^2}\right), & \text{if } k = K. \end{cases} \tag{3}$$

Accordingly, at the end of the uplink NOMA phase, the total achievable task bits offloaded by all the IoT nodes can be computed as

$$\begin{aligned} C_{\mathrm{off}} &= \sum_{k=1}^{K} \left(R_k^{\mathrm{b}} + R_k^{\mathrm{a}}\right) = W t_{\mathrm{a}} \log_2\left(1 + \frac{\sum_{k=1}^{K} p_k h_k}{W\sigma^2}\right) \\ &\quad + \sum_{k=1}^{K} t_k^{\mathrm{b}} W \log_2\left(1 + \frac{\xi \rho_k P_{\mathrm{s}} g_k h_k}{W\sigma^2}\right). \end{aligned} \tag{4}$$

2.5 Task execution phase

In this phase, the MEC server will execute all the received computation tasks. Here, we consider a limited computation capacity at the MEC server. That is to say, all the received tasks may not be computed within the task execution phase and the effective computation bits at the MEC server depends on both its computation capacity and received computation tasks. Let f_{m} and t_{c} denote the computing frequency of the MEC server and the duration of the task execution phase, respectively. Then, we can compute the MEC server's computation capacity as $C_{\mathrm{cc}} = \frac{t_{\mathrm{c}} f_{\mathrm{m}}}{C_{\mathrm{cpu}}}$, where C_{cpu} is the number of CPU cycles required for computing one bit at the MEC server. On this basis, the effective computation bits at the MEC server, which are decided by the minimum between the received tasks and the computation capacity, are given by

$$\begin{aligned} C_{\mathrm{eff}} &= \min\{C_{\mathrm{off}}, C_{\mathrm{cc}}\} \\ &= \min\left\{ W t_{\mathrm{a}} \log_2\left(1 + \frac{\sum_{k=1}^{K} p_k h_k}{W\sigma^2}\right) \right. \\ &\quad \left. + \sum_{k=1}^{K} t_k^{\mathrm{b}} W \log_2\left(1 + \frac{\xi \rho_k P_{\mathrm{s}} g_k h_k}{W\sigma^2}\right), \frac{t_{\mathrm{c}} f_{\mathrm{m}}}{C_{\mathrm{cpu}}} \right\}. \end{aligned} \tag{5}$$

2.6 Local computation

For local computing, let τ_k and f_k denote the computing frequency and time of the k-th IoT node, respectively. Then, the effective computation bits at the k-th IoT node can be computed as

$$C_k^{\mathrm{loc}} = \frac{\tau_k f_k}{C_{\mathrm{cpu},k}}, \tag{6}$$

where $C_{\mathrm{cpu},k}$ denotes the number of CPU cycles required for computing one bit at the k-th IoT node. For each IoT node, its harvested energy is used to support the local computation. Let ε_k denote the effective capacitance coefficient of the processor's chip of the

k-th IoT node. Then, the energy consumption for local computing at the k-th IoT node is given by [7]

$$E_k^{\mathrm{loc}} = \varepsilon_k f_k^3 \tau_k. \tag{7}$$

3 Computation bits maximization

In this section, we aim to design an optimal resource allocation scheme to maximize the total computation bits for the backscatter-assisted wireless-powered NOMA-MEC network with the MEC server's limited computation capacity considered. In particular, we first formulate a computation bits maximization problem by jointly optimizing the time allocation among EH, BackCom, uplink NOMA and task execution, the power reflection coefficient and the transmit power of each IoT node, the computing frequencies and time at both the MEC server and the IoT nodes, subject to multiple constraints, i.e., quality-of-service (QoS), energy causality, latency, etc. Then, we transform the formulated non-convex problem into a convex one by means of many convex tools and develop an efficient iterative algorithm to solve it.

3.1 Problem formulation

In order to formulate the optimization problem, we should create the optimization objective and several constraints (such as QoS and energy causality constraints) first. The optimization objective is to maximize the total computation bits of the considered network and thus, includes the computation bits of both the MEC server and IoT nodes. Accordingly, the optimization objective can be written as

$$C_{\mathrm{tot}}\left(\tau_{\mathrm{e}}, \mathbf{t}^{\mathrm{b}}, t_{\mathrm{a}}, \boldsymbol{\rho}, \mathbf{p}, \mathbf{f}, \boldsymbol{\tau}, t_{\mathrm{c}}, f_{\mathrm{m}}\right) = C_{\mathrm{eff}} + \sum_{k=1}^{K} C_k^{\mathrm{loc}}, \tag{8}$$

where $\mathbf{t}^{\mathrm{b}} = \left[t_1^{\mathrm{b}}, \ldots, t_K^{\mathrm{b}}\right]$, $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_K]$, $\mathbf{p} = [p_1, \ldots, p_K]$, $\mathbf{f} = [f_1, \ldots, f_K]$ and $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_K]$.

3.1.1 QoS constraint

This constraint is used to guarantee the minimum required computation task bits of each IoT node, which can be expressed as the computation bits of the k-th IoT node are not less than its minimum required computation task bits, denoted by $L_{\min,k}$. Due to the limitation of the computation capacity at the MEC sever, it is hard to characterize how many offloading tasks of each IoT node are executed by the MEC server. To solve this problem, we introduce an auxiliary variable α_k ($0 \le \alpha_k \le 1$) and use $\alpha_k L_{\min,k}$ to denote how many required computation bits should be offloaded and executed at the MEC server. On this basis, we can denote the QoS constraint of the k-th IoT node as the following three inequalities, i.e.,

$$R_k^{\mathrm{b}} + R_k^{\mathrm{a}} \ge \alpha_k L_{\min,k}, \quad \forall k, \tag{9}$$

$$\sum\nolimits_{k=1}^{K} \alpha_k L_{\min,k} \le C_{\mathrm{cc}}, \tag{10}$$

$$C_k^{\mathrm{loc}} \geq (1-\alpha_k)L_{\min,k}, \quad \forall k, \tag{11}$$

where (9) and (10) jointly ensure that the required computation bits offloaded by the k-th IoT node, $\alpha_k L_{\min,k}$, can be computed successfully at the MEC server, while (11) is used to ensure that $(1-\alpha_k)L_{\min,k}$ can be computed locally. Combining (9), (10) and (11), the QoS constraint of the k-th IoT can be guaranteed.

3.1.2 Energy-causality constraint

In order to avoid using the battery power of each IoT node and prolong their operation time, we consider the energy-causality constraint, which ensures that the energy consumption of each IoT node for offloading and computing is no more than its harvested energy within each transmission block. Accordingly, the energy-causality constraint of the k-th IoT node is given by

$$P_{\mathrm{c},k}t_k^{\mathrm{b}} + \left(p_k + p_{\mathrm{c},k}\right)t_{\mathrm{a}} + \varepsilon_k f_k^3 \tau_k \leq E_k^{\mathrm{tot}}, \forall k, \tag{12}$$

where $p_{\mathrm{c},k}$ denotes the circuit energy consumption of the k-th IoT node during the uplink NOMA phase.

3.1.3 Computation bits maximization problem formulation

Based on (8), (9), (10), (11) and (12), we can formulate the computation bits maximization problem as

$$\begin{aligned}
\mathbf{P}_1: \quad & \max_{\tau_{\mathrm{e}},\mathbf{t}^{\mathrm{b}},t_{\mathrm{a}},\boldsymbol{\rho},\mathbf{p},\mathbf{f},\boldsymbol{\tau},t_{\mathrm{c}},f_{\mathrm{m}},\boldsymbol{\alpha}} C_{\mathrm{tot}}\left(\tau_{\mathrm{e}},\mathbf{t}^{\mathrm{b}},t_{\mathrm{a}},\boldsymbol{\rho},\mathbf{p},\mathbf{f},\boldsymbol{\tau},t_{\mathrm{c}},f_{\mathrm{m}}\right) \\
\text{s.t. } & \mathrm{C1}: (9)-(11), \\
& \mathrm{C2}: (12), \\
& \mathrm{C3}: \ \tau_{\mathrm{e}}+\textstyle\sum_{k=1}^{K} t_k^{\mathrm{b}}+ t_{\mathrm{a}} + t_{\mathrm{c}} \leq T, \tau_{\mathrm{e}} t_k^{\mathrm{b}}, t_{\mathrm{a}}, t_{\mathrm{c}} \geq 0, \forall k, \\
& \mathrm{C4}: 0 \leq \tau_k \leq T, \ \forall k, \\
& \mathrm{C5}: \ 0 \leq f_{\mathrm{m}} \leq f_{\max}, 0 \leq f_k \leq f_k^{\max}, \ \forall k \\
& \mathrm{C6}: 0 \leq \rho_k \leq 1, \forall k, \\
& \mathrm{C7}: 0 \leq \alpha_k \leq 1, \forall k, \\
& \mathrm{C8}: p_k \geq 0, \forall k,
\end{aligned}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$, and $f_k^{\max}$ and $f_{\max}$ are the maximum computing frequencies at the k-th IoT node and the MEC server, respectively.

In $\mathbf{P_1}$, C1 and C2 are the QoS constraint and the energy-causality constraint for each IoT node. C3 and C4 are the latency constraints which guarantee that all the computation tasks should be executed within the transmission block. C5 demonstrates the limitation of the computation capacity at both the MEC server and the IoT nodes. From $\mathbf{P_1}$, we can observe that $\mathbf{P_1}$ is a highly non-convex problem due to the following three reasons. Firstly, the objective function is very complicated due to the min function brought by the MEC server's limited computation capacity and coupled variables. Secondly, C1 is highly non-convex since the mutual interference among different IoT nodes in the uplink NOMA phase exists, leading to coupled relationships among several variables. Note that these coupled relationships cannot be tackled by using existing convex tools, i.e., variable substitution, etc., resulting in a new challenge to solve $\mathbf{P_1}$. Thirdly, C2 is a non-convex constraint due to the existence of coupled variables, i.e., f_k and τ_k, ρ_k and t_k^{b}, etc.

In order to solve $\mathbf{P_1}$, we will explore the closed-form expressions for parts of the optimal solutions first to simplify the original non-convex problem. Then, we transform the simplified but still non-convex problem into a more traceable problem and propose an efficient iterative algorithm to obtain the solutions.

3.2 Problem transformation and solution

To simplify the objective function in $\mathbf{P_1}$, we introduce a slack variable λ to remove the min function by letting $\lambda = \min\{C_{\text{off}}, C_{\text{cc}}\}$. Then, $\mathbf{P_1}$ can be transformed as

$$\begin{aligned}\mathbf{P_2}: \max_{\tau_{\text{e}},\mathbf{t}^{\text{b}},t_{\text{a}},\boldsymbol{\rho},\mathbf{p},\mathbf{f},\boldsymbol{\tau},t_{\text{c}},f_{\text{m}},\boldsymbol{\alpha},\lambda} \ & \lambda + \sum_{k=1}^{K} C_k^{\text{loc}} \\ \text{s.t. } & \text{C1} - \text{C8}, \\ & \text{C9}: C_{\text{off}} \geq \lambda, \\ & \text{C10}: C_{\text{cc}} \geq \lambda.\end{aligned}$$

In order further simplify $\mathbf{P_2}$, the following proposition is provided to determine the optimal computing frequency of the MEC server f_{m}^* and the optimal computing time of the k-th IoT node τ_k^*.

Proposition 1 *In order to achieve the maximum computation bits of the considered network, the MEC server should use its maximum allowed computing frequency to execute the received tasks and each IoT node will perform local computing during the whole transmission block, namely* $f_{\text{m}}^* = f_{\max}$ *and* $\tau_k^* = T$.

Proof Please see "Appendix A". □

Based on Proposition 1, we substitute $f_{\text{m}} = f_{\max}$ and $\tau_k = T$ into $\mathbf{P_2}$ and $\mathbf{P_2}$ can be rewritten as

$$\begin{aligned}\mathbf{P}_3: \max_{\tau_{\text{e}},\mathbf{t}^{\text{b}},t_{\text{a}},\boldsymbol{\rho},\mathbf{p},\mathbf{f},t_{\text{c}},\boldsymbol{\alpha},\lambda} \ & \lambda + \textstyle\sum_{k=1}^{K} \frac{f_k T}{C_{\text{cpu},k}} \\ \text{s.t.C1}': & \text{ (9)}, \textstyle\sum_{k=1}^{K} \alpha_k L_{\min,k} \leq \frac{t_{\text{c}} f_{\max}}{C_{\text{cpu}}}, \\ & \tfrac{f_k T}{C_{\text{cpu},k}} \geq (1-\alpha_k) L_{\min,k}, \forall k, \\ & \text{C2}': P_{\text{c},k} t_k^{\text{b}} + (p_k + p_{\text{c},k}) t_{\text{a}} + \varepsilon_k f_k^3 T \leq E_k^{\text{tot}}, \forall k, \\ & \text{C3, C5}': 0 \leq f_k \leq f_k^{\max}, \ \forall k, \\ & \text{C6} - \text{C9, C10}': \tfrac{t_{\text{c}} f_{\max}}{C_{\text{cpu}}} \geq \lambda.\end{aligned}$$

Although $\mathbf{P_3}$ is simplified, it is still non-convex due to the non-convex constraints caused by the coupled variables. To tackle this problem, we introduce the following auxiliary variables, i.e., $\phi_k = \rho_k t_k^{\text{b}}$ and $P_k = t_{\text{a}} p_k$, into $\mathbf{P}_3$ and then, we have $\mathbf{P}_4$, given as

$$
\begin{aligned}
\mathbf{P}_4: & \max_{\tau_e,\mathbf{t}^b,t_a,\boldsymbol{\phi},\mathbf{P},\mathbf{f},t_c,\alpha,\lambda} \lambda+\sum_{k=1}^{K}\frac{f_kT}{C_{\mathrm{cpu},k}} \\
\text{s.t.C1}'': & \begin{cases} F_k\left(t_k^b,\phi_k\right)+F_k^N(\mathbf{P},t_a)\geq\alpha_kL_{\min,k}, \text{if } k\leq K-1, \\ F_K\left(t_K^b,\phi_K\right)+F_K^N(P_K,t_a)\geq\alpha_KL_{\min,K}, \quad \text{if } k=K \end{cases} \\
& \sum_{k=1}^{K}\alpha_kL_{\min,k}\leq\frac{t_cf_{\max}}{C_{\mathrm{cpu}}},\frac{f_kT}{C_{\mathrm{cpu},k}}\geq(1-\alpha_k)L_{\min,k},\forall k, \\
& \mathrm{C2}'':P_{c,k}t_k^b+\left(P_k+p_{c,k}t_a\right)+\varepsilon_kf_k^3T\leq\eta P_sg_k\left(\tau_e+\sum_{i=1}^{K}t_i^b\right)-\eta\phi_kP_sg_k,\forall k, \\
& \mathrm{C3},\mathrm{C5}',\mathrm{C7},\mathrm{C10}',\mathrm{C6}':0\leq\phi_k\leq t_k^b,\forall k,\mathrm{C8}':P_k\geq0,\forall k, \\
& \mathrm{C9}':Wt_a\log_2\left(1+\frac{\sum_{k=1}^{K}P_kh_k}{t_aW\sigma^2}\right)+\sum_{k=1}^{K}F_k\left(t_k^b,\phi_k\right)\geq\lambda,
\end{aligned}
$$

where $\boldsymbol{\phi}=[\phi_1,\ldots,\phi_K]$, $\mathbf{P}=[P_1,\ldots,P_K]$, $F_k\left(t_k^b,\phi_k\right)=t_k^bW\log_2\left(1+\frac{\xi\phi_kP_sg_kh_k}{t_k^bW\sigma^2}\right)$ with $k\leq K$, $F_k^N(\mathbf{P},t_a)=Wt_a\log_2\left(1+\frac{P_kh_k}{\sum_{i=k+1}^{K}P_ih_i+t_aW\sigma^2}\right)$ with $k\leq K-1$ and $F_K^N(P_K,t_a)=Wt_a\log_2\left(1+\frac{P_Kh_K}{t_aW\sigma^2}\right)$.

Proposition 2 *In* $\mathbf{P}_4$*, both the objective function and constraints except* C1″ *are convex.*

Proof Please see "Appendix B". □

According to Proposition 2, it can be observed that $\mathbf{P}_4$ is not convex due to the fact that C1″ is non-convex. By observing C1″, we find that the co-channel interference caused by the uplink NOMA makes $F_k^N(\mathbf{P},t_a)$ jointly non-concave with respect to $\{P_k\}_{k=1}^{K-1}$ and t_a, leading to a non-convex C1″.

In the following part, we aim to deal with the non-convexity of C1″. From $F_k^N(\mathbf{P},t_a)$, we can see that the main difficulty comes from the DC structure. To address this problem, we introduce a specific inequality to relax $F_k^N(\mathbf{P},t_a)$. Specifically, we first replace $F_k^N(\mathbf{P},t_a)$ with its low bound based on the above inequality and then, obtain a sub-problem with given parameters which are from the used inequality. Compared to the problem $\mathbf{P}_4$, the sub-problem is more traceable and can be solved by means of the existing convex tools. On this basis, we can develop an efficient iterative algorithm to solve $\mathbf{P}_4$, where in each iteration, we just need to solve the sub-problem with given parameters instead of $\mathbf{P}_4$. When we achieve the optimal solution to the sub-problem with given parameters, we can update the parameters of the inequality based on the achieved solution and repeat the above steps until the lower bound is tight.

The following theorem [25] provides an inequality to relax the function with the DC structure.

Theorem 1 *For* $\theta,\tilde{\theta}>0$*, we have the following inequality:* $\log_2(1+\theta)\geq\varpi\log_2(\theta)+\mu$ *with the constant parameters* $\varpi=\frac{\tilde{\theta}}{1+\tilde{\theta}}$ *and* $\mu=\log_2\left(1+\tilde{\theta}\right)-\varpi\log_2\left(\tilde{\theta}\right)$*, where the lower bound is tight at* $\tilde{\theta}=\theta$*.*

According to Theorem 1, we have the following inequality

$$
\begin{aligned}
F_k^{\mathrm{N}}(\mathbf{P}, t_{\mathrm{a}}) &= W t_{\mathrm{a}} \log_2\left(1+\frac{P_k h_k}{\sum_{i=k+1}^{K} P_i h_i + t_{\mathrm{a}} W \sigma^2}\right) \\
&\geq W t_{\mathrm{a}}\left(\varpi_k \log_2\left(\frac{P_k h_k}{\sum_{i=k+1}^{K} P_i h_i + t_{\mathrm{a}} W \sigma^2}\right)+\mu_k\right), k \leq K,
\end{aligned} \tag{13}
$$

where ϖ_k and μ_k ($k \leq K$) are given parameters in one iteration and decided by the solution obtained in the previous iteration. For example, let $\left\{\tau_{\mathrm{e}}^{(i)}, \left\{t_k^{\mathrm{b}(i)}\right\}_{k=1}^{K}, t_{\mathrm{a}}^{(i)}, \left\{\phi_k^{(i)}\right\}_{k=1}^{K}, \left\{P_k^{(i)}\right\}_{k=1}^{K}, \left\{f_k^{(i)}\right\}_{k=1}^{K}, t_{\mathrm{c}}^{(i)}, \left\{\alpha_k^{(i)}\right\}_{k=1}^{K}, \lambda^{(i)}\right\}$ denote the obtained solution in the i-th iteration. Then, in the $i+1$-th iteration, the parameters at the k-th IoT node, $\varpi_k^{(i+1)}$ and $\mu_k^{(i+1)}$, can be updated as

$$
\varpi_k^{(i+1)} = \frac{P_k^{(i)} h_k}{\sum_{j=k}^{K} P_j^{(i)} h_j + t_{\mathrm{a}}^{(i)} W \sigma^2}, \tag{14}
$$

$$
\begin{aligned}
\mu_k^{(i+1)} &= \log_2\left(1+\frac{P_k^{(i)} h_k}{\sum_{j=k+1}^{K} P_j^{(i)} h_j + t_{\mathrm{a}}^{(i)} W \sigma^2}\right) \\
&\quad - \varpi_k^{(i)} \log_2\left(\frac{P_k^{(i)} h_k}{\sum_{j=k+1}^{K} P_j^{(i)} h_j + t_{\mathrm{a}}^{(i)} W \sigma^2}\right).
\end{aligned} \tag{15}
$$

Based on (13), (18) and (19), $\mathbf{P}_4$ can be relaxed as the sub-problem in the $i+1$-th iteration, given by

$$
\begin{aligned}
\mathbf{P}_5: \quad & \max_{\substack{\tau_{\mathrm{e}}^{(i+1)}, \left\{t_k^{\mathrm{b}(i+1)}\right\}_{k=1}^{K}, t_{\mathrm{a}}^{(i+1)}, \left\{\phi_k^{(i+1)}\right\}_{k=1}^{K}, \\ \left\{P_k^{(i+1)}\right\}_{k=1}^{K}, \left\{f_k^{(i+1)}\right\}_{k=1}^{K}, t_{\mathrm{c}}^{(i+1)}, \left\{\alpha_k^{(i+1)}\right\}_{k=1}^{K}, \lambda^{(i+1)}}} \quad \lambda^{(i+1)} + \sum_{k=1}^{K} \frac{f_k^{(i+1)} T}{C_{\mathrm{cpu},k}} \\
\text{s.t.} \; & \mathrm{C1}''': \begin{cases} F_k\left(t_k^{\mathrm{b}(i+1)}, \phi_k^{(i+1)}\right) + W t_{\mathrm{a}}\left(\varpi_k^{(i+1)} \log_2\left(\frac{P_k^{(i+1)} h_k}{\sum_{j=k+1}^{K} P_j^{(i+1)} h_j + t_{\mathrm{a}}^{(i+1)} W \sigma^2}\right) + \mu_k^{(i+1)}\right) \\ \geq \alpha_k^{(i+1)} L_{\min,k}, \text{if } k \leq K-1, \\ F_K\left(t_K^{\mathrm{b}(i+1)}, \phi_K^{(i+1)}\right) + F_K^{\mathrm{N}}\left(P_K^{(i+1)}, t_{\mathrm{a}}^{(i+1)}\right) \geq \alpha_K^{(i+1)} L_{\min,K}, \quad \text{if } k = K \end{cases} \\
& \sum_{k=1}^{K} \alpha_k^{(i+1)} L_{\min,k} \leq \frac{t_{\mathrm{c}}^{(i+1)} f_{\max}}{C_{\mathrm{cpu}}}, \frac{f_k^{(i+1)} T}{C_{\mathrm{cpu},k}} \geq \left(1-\alpha_k^{(i+1)}\right) L_{\min,k}, \forall k, \\
& \mathrm{C2}''', \mathrm{C3}', \mathrm{C5}'', \mathrm{C6}'', \mathrm{C7}', \mathrm{C8}'', \mathrm{C9}'', \mathrm{C10}'',
\end{aligned}
$$

where $\mathrm{C2}''', \mathrm{C3}', \mathrm{C5}'', \mathrm{C6}'', \mathrm{C7}', \mathrm{C8}'', \mathrm{C9}''$ and $\mathrm{C10}''$ are transformed from $\mathrm{C2}'', \mathrm{C3}$, $\mathrm{C5}', \mathrm{C6}', \mathrm{C7}, \mathrm{C8}', \mathrm{C9}'$ and $\mathrm{C10}'$ by replacing $\left\{\tau_{\mathrm{e}}, \mathbf{t}^{\mathrm{b}}, t_{\mathrm{a}}, \boldsymbol{\phi}, \mathbf{P}, \mathbf{f}, t_{\mathrm{c}}, \alpha, \lambda\right\}$ with the following term, given by

$$
\begin{aligned}
&\{\tau_{\mathrm{e}}^{(i+1)}, \left\{t_k^{\mathrm{b}(i+1)}\right\}_{k=1}^{K}, t_{\mathrm{a}}^{(i+1)}, \left\{\phi_k^{(i+1)}\right\}_{k=1}^{K}, \left\{P_k^{(i+1)}\right\}_{k=1}^{K} \\
&, \left\{f_k^{(i+1)}\right\}_{k=1}^{K}, t_{\mathrm{c}}^{(i+1)}, \left\{\alpha_k^{(i+1)}\right\}_{k=1}^{K}, \lambda^{(i+1)}\}.
\end{aligned}
$$

As for $\mathbf{P}_5$, it is still non-convex due to the following non-concave function, given as

$$W t_{\mathrm{a}}\left(\varpi_k^{(i+1)}\log_2\left(\frac{P_k^{(i+1)}h_k}{\sum_{j=k+1}^{K}P_j^{(i+1)}h_j+t_{\mathrm{a}}^{(i+1)}W\sigma^2}\right)+\mu_k^{(i+1)}\right).$$

Thus, we further introduce the following auxiliary variables $x_k^{(i+1)}=t_{\mathrm{a}}^{(i+1)}\log_2\left(\frac{P_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}\right)$, $k\le K$ into $\mathbf{P}_5$, and we have $P_k^{(i+1)}=t_{\mathrm{a}}^{(i+1)}2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}},k\le K$. Substituting $P_k^{(i+1)}=t_{\mathrm{a}}^{(i+1)}2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}},k\le K$ into $\mathbf{P}_5$, $\mathbf{P}_5$ can be transformed into

$$\begin{aligned}
\mathbf{P}_6:\ &\max_{\substack{\tau_{\mathrm{e}}^{(i+1)},\left\{t_k^{\mathrm{b}(i+1)}\right\}_{k=1}^{K},t_{\mathrm{a}}^{(i+1)},\left\{\phi_k^{(i+1)}\right\}_{k=1}^{K},\\ \left\{x_k^{(i+1)}\right\}_{k=1}^{K},\left\{f_k^{(i+1)}\right\}_{k=1}^{K},t_{\mathrm{c}}^{(i+1)},\left\{\alpha_k^{(i+1)}\right\}_{k=1}^{K},\lambda^{(i+1)}}} \lambda^{(i+1)}+\sum_{k=1}^{K}\frac{f_k^{(i+1)}T}{C_{\mathrm{cpu},k}}\\
\text{s.t.C1}'''':\ &\begin{cases}F_k\left(t_k^{\mathrm{b}(i+1)},\phi_k^{(i+1)}\right)+F_k^{\mathrm{T}}\left(t_{\mathrm{a}}^{(i+1)},\{x_{ii}^{(i+1)}\}_{ii=k}^{K}\right)\ge\alpha_k^{(i+1)}L_{\min,k},\text{if } k\le K-1,\\ F_K\left(t_K^{\mathrm{b}(i+1)},\phi_K^{(i+1)}\right)+F_K^{\mathrm{NT}}\left(x_K^{(i+1)},t_{\mathrm{a}}^{(i+1)}\right)\ge\alpha_K^{(i+1)}L_{\min,K},\ \text{if } k=K\end{cases}\\
&\sum_{k=1}^{K}\alpha_k^{(i+1)}L_{\min,k}\le\frac{t_{\mathrm{c}}^{(i+1)}f_{\max}}{C_{\mathrm{cpu}}},\frac{f_k^{(i+1)}T}{C_{\mathrm{cpu},k}}\ge\left(1-\alpha_k^{(i+1)}\right)L_{\min,k},\forall k,\\
\text{C2}'''':\ &P_{\mathrm{c},k}t_k^{\mathrm{b}(i+1)}+\left(t_{\mathrm{a}}^{(i+1)}2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}+p_{\mathrm{c},k}t_{\mathrm{a}}^{(i+1)}\right)+\varepsilon_k\left(f_k^{(i+1)}\right)^3T\\
&\le\eta P_{\mathrm{s}}g_k\left(\tau_{\mathrm{e}}^{(i+1)}+\sum_{i=1}^{K}t_i^{\mathrm{b}(i+1)}\right)-\eta\phi_k^{(i+1)}P_{\mathrm{s}}g_k,\forall k,\\
&\text{C3}',\text{C5}'',\text{C6}'',\text{C7}',\text{C10}'',\\
\text{C9}''':\ &\sum_{k=1}^{K-1}F_k^{\mathrm{T}}\left(t_{\mathrm{a}}^{(i+1)},x_k^{(i+1)}\right)+F_K^{\mathrm{NT}}\left(x_K^{(i+1)},t_{\mathrm{a}}^{(i+1)}\right)+\sum_{k=1}^{K}F_k\left(t_i^{\mathrm{b}(i+1)},\phi_k^{(i+1)}\right)\ge\lambda^{(i+1)},
\end{aligned}$$

where

$$\begin{aligned}
F_k^{\mathrm{T}}\left(t_{\mathrm{a}}^{(i+1)},\{x_{ii}^{(i+1)}\}_{ii=k}^{K}\right)=&\left[t_{\mathrm{a}}^{(i+1)}\log_2h_k+x_k^{(i+1)}-t_{\mathrm{a}}^{(i+1)}\log_2\left(\sum_{j=k+1}^{K}2^{\frac{x_j^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}h_j+W\sigma^2\right)\right]\\
&\times W\varpi_k^{(i+1)}+Wt_{\mathrm{a}}^{(i+1)}\mu_k^{(i+1)}
\end{aligned}$$

with $k\le K-1$ and

$$F_K^{\mathrm{NT}}\left(x_K^{(i+1)},t_{\mathrm{a}}^{(i+1)}\right)=\varpi_K^{(i+1)}W\left(t_{\mathrm{a}}^{(i+1)}\log_2h_K-t_{\mathrm{a}}^{(i+1)}\log_2\left(W\sigma^2\right)+x_K^{(i+1)}\right)+\mu_K^{(i+1)}Wt_{\mathrm{a}}^{(i+1)}$$

with $k=K$.

Proposition 3 $\mathbf{P}_6$ *is a convex problem, which can be solved by the existing convex tools.*

Proof Please see "Appendix C". □

3.3 Design of the iterative algorithm

In this subsection, we provide an efficient iterative algorithm to solve $\mathbf{P}_4$, which is shown in Algorithm 1 at the top of the next page. Specifically, in the $i+1$-th iteration, we should

solve $\mathbf{P}_6$ with given $\varpi_k^{(i+1)}$ and $\mu_k^{(i+1)}$, $\forall k$ and obtain its optimal solution, denoted by $\left\{\tau_{\mathrm{e}}^{(i+1)}, \left\{t_k^{\mathrm{b}(i+1)}\right\}_{k=1}^K, t_{\mathrm{a}}^{(i+1)}, \left\{\phi_k^{(i+1)}\right\}_{k=1}^K, \left\{x_k^{(i+1)}\right\}_{k=1}^K, \left\{f_k^{(i+1)}\right\}_{k=1}^K, t_{\mathrm{c}}^{(i+1)}, \left\{\alpha_k^{(i+1)}\right\}_{k=1}^K, \lambda^{(i+1)}\right\}$. On this basis, the IoT nodes's optimal transmit powers $\{p_k^*\}_{k=1}^K$, reflection coefficients $\{\rho_k^*\}_{k=1}^K$ and computing frequencies $\left\{f_k^*\right\}_{k=1}^K$ as well as the optimal time allocation at both IoT nodes and the MEC server $\tau_{\mathrm{e}}^*, \left\{t_k^{\mathrm{b}*}\right\}_{k=1}^K, t_{\mathrm{a}}^*, t_{\mathrm{c}}^*$ are obtained. Then, we can compute the maximum computation bits of the system $C_{\mathrm{tot}}^* = \lambda^{(i+1)} + \sum_{k=1}^K \frac{f_k^{(i+1)} T}{C_{\mathrm{cpu},k}}$. If the stop condition is satisfied, then the obtained solution $\{\tau_{\mathrm{e}}^*, \left\{t_k^{\mathrm{b}*}\right\}_{k=1}^K, t_{\mathrm{a}}^*, t_{\mathrm{c}}^*, \{f_k^*\}_{k=1}^K, \{p_k^*\}_{k=1}^K, \{\rho_k^*\}_{k=1}^K\}$ is the solution to $\mathbf{P}_4$ and the maximum computation bits of the system are given by C_{tot}^*. Otherwise, we should update $\varpi_k^{(i+1)}$ and $\mu_k^{(i+1)}$, $\forall k$ based on the obtained solution by following (18) and (19) and repeat the above steps. Please note that the proposed iterative algorithm can always converge since the iteration to update ϖ_k and μ_k always converges and the detailed proof can be refer to [25].

Algorithm 1 Efficient Iterative Algorithm for Solving $\mathbf{P}_4$

1: Set the maximum tolerance ε;
2: Set the iteration index $i = 0$, $\mathrm{Flag} = 0$ and $\theta_k^{(i)} = 0.01$ for $k \leq K$;
3: Set $\varpi_k^{(i+1)} = \frac{\theta_k^{(i)}}{1+\theta_k^{(i)}}$ and $\mu_k^{(i+1)} = \log_2\left(1+\theta_k{}^{(i)}\right) - \varpi_k^{(i+1)}\log_2\left(\theta^{(i)}\right)$ for $k \leq K$;
4: **repeat**
5: Solve the optimization problem $\mathbf{P_6}$ with given $\varpi_k^{(i+1)}$ and $\mu_k^{(i+1)}$, $\forall k$, to achieve the optimal solution, denoted by $\left\{\tau_{\mathrm{e}}^{(i+1)}, \left\{t_k^{\mathrm{b}(i+1)}\right\}_{k=1}^K, t_{\mathrm{a}}^{(i+1)}, \left\{\phi_k^{(i+1)}\right\}_{k=1}^K, \left\{x_k^{(i+1)}\right\}_{k=1}^K, \left\{f_k^{(i+1)}\right\}_{k=1}^K, t_{\mathrm{c}}^{(i+1)}, \left\{\alpha_k^{(i+1)}\right\}_{k=1}^K, \lambda^{(i+1)}\right\}$;
6: Obtain the optimal transmit powers $\{p_k^*\}_{k=1}^K$ and reflection coefficients $\{\rho_k^*\}_{k=1}^K$ at IoT nodes based on the above obtained solutions;
7: Obtain the optimal time allocation at both IoT nodes and the MEC server and computing frequencies of IoT nodes, namely $\tau_{\mathrm{e}}^*, \{t_k^{\mathrm{b}*}\}_{k=1}^K, t_{\mathrm{a}}^*, t_{\mathrm{c}}^*$ and $\{f_k^*\}_{k=1}^K$;
8: Compute the maximum computation bits of the system, denoted by C_{tot}^* as $\lambda^{(i+1)} + \sum_{k=1}^K \frac{f_k^{(i+1)} T}{C_{\mathrm{cpu},k}}$;
9: Compute $\theta_k^{(i+1)} = \frac{p_k^* h_k}{\sum_{j=k+1}^K p_j^* h_j + W\sigma^2}$;
10: **if** $\left|\theta_k^{(i+1)} - \theta_k^{(i)}\right| \geq \varepsilon$, $\forall k$, **then**
11: Set $i = i + 1$;
12: Update $\varpi_k^{(i+1)}$ and $\mu_k^{(i+1)}$, $\forall k$, based on (18) and (19);
13: **else**
14: The obtained solution $\{\tau_{\mathrm{e}}^*, \{t_k^{\mathrm{b}*}\}_{k=1}^K, t_{\mathrm{a}}^*, t_{\mathrm{c}}^*, \{f_k^*\}_{k=1}^K, \{p_k^*\}_{k=1}^K, \{\rho_k^*\}_{k=1}^K\}$ is the solution to $\mathbf{P}_4$ and the maximum computation bits of the system is given by C_{tot}^*;
15: Set $\mathrm{Flag} = 1$;
16: **end if**
17: **until** $\mathrm{Flag} = 1$.

4 Experiments and results discussion

4.1 Experiments

This section is provided to verify the effectiveness and the superiority of the proposed scheme by means of MATLAB. Unless otherwise specified, we set the basic simulation parameters as follows. Specifically, we set $T = 1$ s, $W = 100$ kHz, $P_{\mathrm{s}} = 30$ dBm, $K = 4$, $\sigma^2 = -120$ dBm/Hz, $f_{\max} = 10$ GHz, $\xi = -15$ dB, $\eta = 0.7$, $C_{\mathrm{cpu}} = 1000$ Cycles/bit, $P_{\mathrm{c},1} = P_{\mathrm{c},2} = P_{\mathrm{c},3} = P_{\mathrm{c},4} = 10$ μW, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 10^{-26}$,

$f_1^{\max} = f_2^{\max} = f_3^{\max} = f_4^{\max} = 500$ MHz, $C_{\text{cpu},1} = C_{\text{cpu},2} = C_{\text{cpu},3} = C_{\text{cpu},4} = 1000$ Cycles/bit and $L_{\min,1} = L_{\min,2} = L_{\min,3} = L_{\min,4} = L_{\min} = 100$ kbits. Here, we consider the standard channel fading model, where the channel gain of each channel is modeled by the product of the small-scale fading and the large-scale fading. Let g'_k and h'_k, $\forall k$, denote the small-scale fadings from the k-th IoT node to the ES and the MEC server, respectively. Likewise, denote $D_{\text{s},k}$ and $D_{\text{e},k}$ as the distance from the k-th IoT node to the ES and the MEC server. The channel gains of the ES-the k-th IoT node link and the MEC server-the k-th IoT node link are given by $g_k = g'_k D_{\text{s},k}^{-\beta}$ and $h_k = h'_k D_{\text{e},k}^{-\beta}$, where β denotes the path loss exponent. In this work, we set $\beta = 2.7$, $D_{\text{s},1} = 13$ m, $D_{\text{s},2} = 10$ m, $D_{\text{s},3} = 11$ m, $D_{\text{s},4} = 10$ m, $D_{\text{e},1} = 50$ m, $D_{\text{e},2} = 55$ m, $D_{\text{e},3} = 50$ m and $D_{\text{e},4} = 51$ m.

4.2 Results discussion

Figure 2 illustrates the convergence of Algorithm 1 under different settings of P_{s} and $L_{\min}$. Specifically, we set P_{s} as 25 dBm or 30 dBm. $L_{\min}$ is set as 30 kbits, 50 kbits, 70 kbits, 90 kbits or 110 kbits. From this figure, it can be observed that the proposed iterative algorithm can always converge to a certain value within a few iterations, e.g., 4 iterations. That is, the proposed iterative algorithm is convergent and computationally efficient. By comparing the achievable computation bits under different settings of P_{s} and $L_{\min}$, we also see that with P_{s} (or $L_{\min}$) fixed, a larger $L_{\min}$ (or P_{s}) may decrease (or increase) the achievable computation bits.

Figure 3 shows the system computation bits versus the transmit power of the ES P_{s}, where P_{s} varies from 0.5 W to 2.5 W. In order to demonstrate the superiority of the proposed scheme, we compare the performance under the proposed scheme with that under the four other benchmark schemes, which are the complete offloading scheme, the fully local computing scheme, the backscatter-assisted MEC and the wireless-powered NOMA-MEC, respectively. In the complete offloading scheme, each IoT node only performs task offloading and both BackCom and AT can be used to offload tasks. In the fully local computing scheme, all the IoT nodes can only compute their tasks locally.

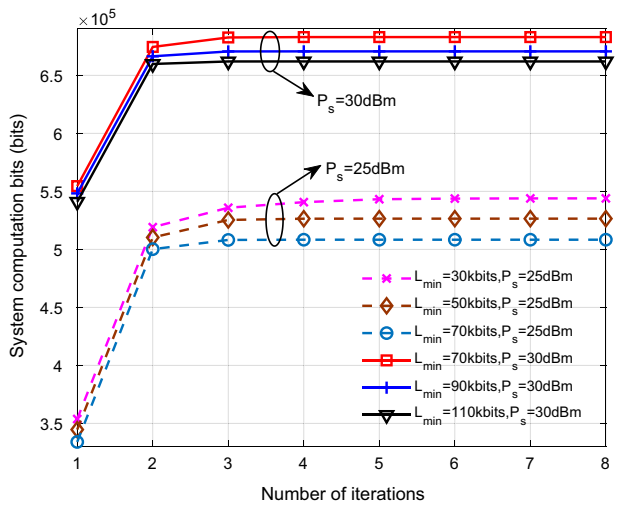

Fig. 2 The convergence analysis of Algorithm 1

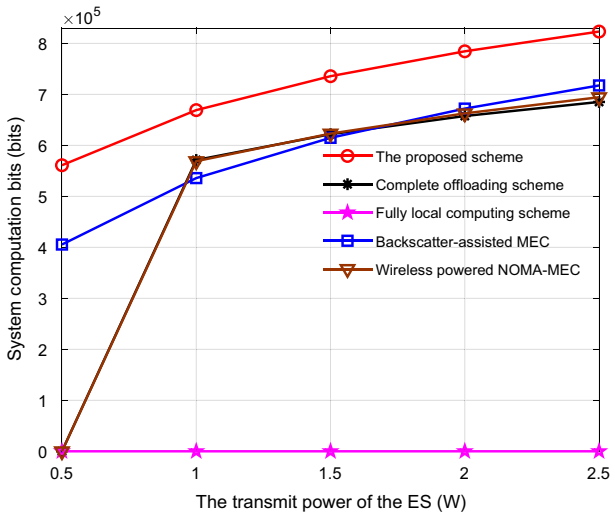

Fig. 3 System computation bits versus the transmit power at the ES

For the backscatter-assisted MEC, each IoT node can not only perform local computing but also offload its tasks to the MEC server and only BackCom can be adopted for task offloading. For the wireless-powered NOMA-MEC, each IoT node can only choose the NOMA technology to offload parts of tasks and the rest tasks can be computed locally. Please note that the above four schemes are obtained by means of some methods used in this work. Specifically, the complete offloading scheme is obtained by using the proposed iterative algorithm after a few changes, e.g., solving $\mathbf{P}_6$ with $f_k = 0, \forall k$. The fully local computing scheme is optimized by solving $\mathbf{P}_6$ with $t_\mathrm{a} = 0, t_\mathrm{c} = 0$ and $t_k^\mathrm{b} = 0, \forall k$. The results under the backscatter-assisted MEC can be obtained by solving $\mathbf{P}_4$ with $t_\mathrm{a} = 0$ and $P_k = 0, \forall k$. The results of the wireless-powered NOMA-MEC are achieved by means of Algorithm 1 by solving $\mathbf{P}_6$ with $t_k^\mathrm{b} = 0, \rho_k = 0, \forall k$. It is not hard to prove that the above problems, $\mathbf{P}_6$ with $f_k = 0, \forall k$, $\mathbf{P}_6$ with $t_\mathrm{a} = 0, t_\mathrm{c} = 0$ and $t_k^\mathrm{b} = 0, \forall k$, $\mathbf{P}_4$ with $t_\mathrm{a} = 0$ and $P_k = 0, \forall k$ and $\mathbf{P}_6$ with $t_k^\mathrm{b} = 0, \rho_k = 0, \forall k$, are convex, which can be solved by using the existing convex tools.

As shown in this figure, it can be observed that all the schemes except the fully local computing scheme show an upward trend while the results under the fully local computing scheme are always 0. The reasons are as follows. With the increasing of P_s, both the total harvested energy and the achievable BackCom rate at each IoT node increase, resulting in an improvement to the system computation bits, while the fully local computing scheme may not satisfy the QoS constraint, etc., which also demonstrates the advantage of MEC. By comparisons, we can also see that the proposed scheme can achieve the highest computation bits among the five schemes. This is because the proposed scheme combines the advantages of BackCom and AT as well as the partial offloading scheme, providing more flexibilities to utilize resources efficiently.

Figure 4 plots the system computation bits versus the minimum required computation bits at each IoT node, where L_min is ranged from 50 to 150 kbits. Similarly, we compare the performance under the proposed scheme with that under the complete offloading scheme, the fully local computing scheme, the backscatter-assisted MEC

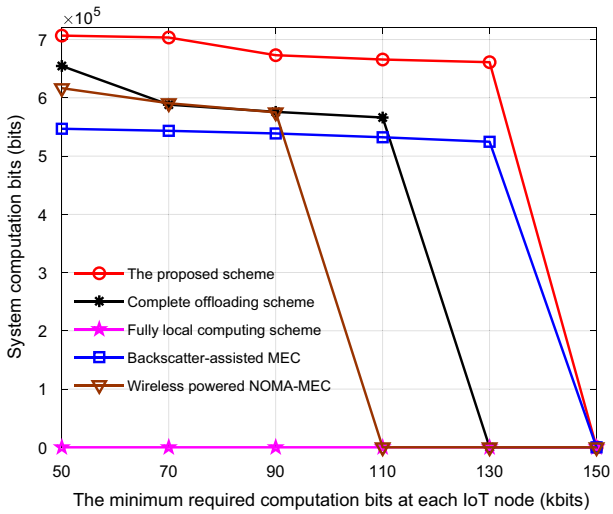

Fig. 4 System computation bits versus the minimum required computation bits at each IoT node

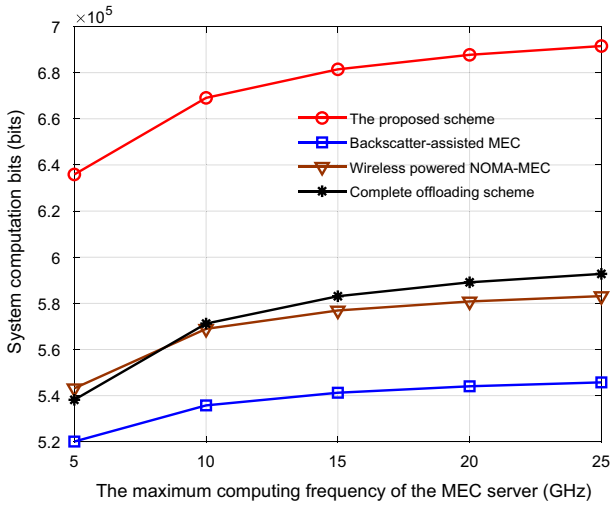

Fig. 5 System computation EE versus the maximum computing frequency at the MEC server

and the wireless-powered NOMA-MEC in order to demonstrate the superiority of the proposed scheme. It can be observed that the system computation bits under all the schemes except the fully local computing scheme decrease when $L_{\min}$ increases since a larger $L_{\min}$ means a higher QoS requirement for each IoT node and more resources will be allocated to the IoT nodes with worse channels, which reduces the system computation bits. Besides, we can also see that the proposed scheme outperforms the other schemes in terms of system computation bits, which further illustrates the advantage of the proposed scheme.

Figure 5 depicts the effect of the computation capacity of the MEC server on the system computation bits, where $f_{\max}$ varies from 5 GHz to 25 GHz. It can be observed that the system computation bits under the proposed scheme, the backscatter-assisted MEC, the wireless-powered NOMA-MEC and the complete offloading

scheme increase with the increasing of $f_{\max}$. This is because the improvement of the MEC server's computation capacity allows more task bits to be offloaded and executed, resulting in the increasing system computation bits. Note that the fully local computing scheme is not included in this figure since its performance is not influenced by $f_{\max}$. By comparisons, we still can see that the proposed scheme is superior to the other schemes in terms of system computation bits.

5 Conclusions

In this paper, we have studied the computation bits maximization for the backscatter-assisted wireless-powered NOMA-MEC network while considering the limited computation capacity and the computation resource allocation at the MEC server. In particular, we have formulated a computation bits maximization problem by jointly optimizing the EH time, the BackCom time and reflection coefficients of the IoT nodes, the IoT nodes' transmit power and time in the uplink NOMA phase, as well as the computing frequencies and time of both the IoT nodes and the MEC server. In order to solve the formulated non-convex problem, we first introduced a slack variable and the contradiction approach to remove the min function from the objective function and determine partial optimal solutions, respectively. Then, based on the inequality transformation approach, an efficient iterative algorithm has been proposed to obtain the optimal solutions. Simulations results have verified the quick convergence of the proposed iterative algorithm and demonstrated the superiority of the proposed scheme in terms of the computation bits over the existing benchmark schemes.

Appendix A

Here, we prove Proposition 1 by means of contradiction.

Proof for $f_{\mathrm{m}}^* = f_{\max}$ Suppose that $\left\{\tau_{\mathrm{e}}^*, \left\{t_k^{\mathrm{b}*}\right\}_{k=1}^K, t_{\mathrm{a}}^*, \left\{\rho_k^*\right\}_{k=1}^K, \left\{p_k^*\right\}_{k=1}^K, \left\{f_k^*\right\}_{k=1}^K, \left\{\tau_k^*\right\}_{k=1}^K, t_{\mathrm{c}}^*, f_{\mathrm{m}}^*, \left\{\alpha_k^*\right\}_{k=1}^K, \lambda^*\right\}$ is the optimal solution to $\mathbf{P_2}$, where $f_{\mathrm{m}}^* < f_{\max}$ and $\lambda^* = \min\left\{\sum_{k=1}^K t_k^{\mathrm{b}*} W \log_2\left(1+\frac{\xi\rho_k^* P_{\mathrm{s}} g_k h_k}{W\sigma^2}\right) + W t_{\mathrm{a}}^* \log_2\left(1+\frac{\sum_{k=1}^K p_k^* h_k}{W\sigma^2}\right), \frac{t_{\mathrm{c}}^* f_{\mathrm{m}}^*}{C_{\mathrm{cpu}}}\right\}$. Then, the maximum computation bits of the system can be computed as $C_{\mathrm{tot}}^* = \lambda^* + \sum_{k=1}^K \frac{f_k^* \tau_k^*}{C_{\mathrm{cpu},k}}$. Further, we can also construct another solution satisfying $f_{\mathrm{m}}^+ = f_{\max} > f_{\mathrm{m}}^*, \tau_{\mathrm{e}}^+ = \tau_{\mathrm{e}}^*, t_k^{\mathrm{b}+} = t_k^{\mathrm{b}*}, t_{\mathrm{a}}^+ = t_{\mathrm{a}}^*, \rho_k^+ = \rho_k^*, f_k^+ = f_k^*, \tau_k^+ = \tau_k^*, t_{\mathrm{c}}^+ = t_{\mathrm{c}}^*$ and $\lambda^+ = \min\left\{\sum_{k=1}^K t_k^{\mathrm{b}+} W \log_2\left(1+\frac{\xi\rho_k^+ P_{\mathrm{s}} g_k h_k}{W\sigma^2}\right) + W t_{\mathrm{a}}^+ \log_2\left(1+\frac{\sum_{k=1}^K p_k^+ h_k}{W\sigma^2}\right), \frac{t_{\mathrm{c}}^+ f_{\mathrm{m}}^+}{C_{\mathrm{cpu}}}\right\}$. Obviously, the constructed solution is a feasible solution to $\mathbf{P_2}$ which satisfies all the constraints of $\mathbf{P_2}$. Accordingly, we can calculate the computation bits of the system under this constructed solution as $C_{\mathrm{tot}}^+ = \lambda^+ + \sum_{k=1}^K \frac{f_k^+ \tau_k^+}{C_{\mathrm{cpu},k}}$. Since $f_{\mathrm{m}}^+ = f_{\max} > f_{\mathrm{m}}^*$ holds, we have $\frac{t_{\mathrm{c}}^+ f_{\mathrm{m}}^+}{C_{\mathrm{cpu}}} > \frac{t_{\mathrm{c}}^* f_{\mathrm{m}}^*}{C_{\mathrm{cpu}}}$, resulting in $\lambda^+ \geq \lambda^*$. Based on the expression of C_{tot}, we have $C_{\mathrm{tot}}^+ \geq C_{\mathrm{tot}}^*$ due to the fact that $\lambda^+ \geq \lambda^*$, $f_k^+ = f_k^*$ and $\tau_k^+ = \tau_k^*$. However, the fact that $C_{\mathrm{tot}}^+ \geq C_{\mathrm{tot}}^*$ contradicts the above assumption that the solution $\left\{\tau_{\mathrm{e}}^*, \left\{t_k^{\mathrm{b}*}\right\}_{k=1}^K, t_{\mathrm{a}}^*, \left\{\rho_k^*\right\}_{k=1}^K, \left\{p_k^*\right\}_{k=1}^K, \left\{f_k^*\right\}_{k=1}^K, \left\{\tau_k^*\right\}_{k=1}^K, t_{\mathrm{c}}^*, f_{\mathrm{m}}^*, \left\{\alpha_k^*\right\}_{k=1}^K, \lambda^*\right\}$ is optimal. Therefore, in order to maximize the computation bits of the system, the MEC server should compute its received tasks with its maximum allowed computing frequency, namely $f_{\mathrm{m}}^* = f_{\max}$.

Proof for $\tau_k^* = T$

When $\tau_e, \mathbf{t}^b, t_a, \boldsymbol{\rho}, \mathbf{p}, t_c, f_m, \boldsymbol{\alpha}, \lambda$ and $\{f_i, \tau_i\}_{i=\{1,2,\ldots,K\}\backslash k}$ are fixed, we should jointly optimize f_k and τ_k to achieve the maximum computation bits of the system. Assume that the k-th IoT node's optimal computing frequency f_k^* and time $\tau_k^* < T$ satisfy all the constraints of of $\mathbf{P_2}$ with other parameters fixed. Then, the maximum computation bits of the considered system C_{tot}^* are computed as $\lambda + \sum_{i=1,i\neq k}^{K} \frac{\tau_i f_i}{C_{\text{cpu},k}} + \frac{\tau_k^* f_k^*}{C_{\text{cpu},k}}$. On this basis, we construct another solution as $\{f_k^+, \tau_k^+\}$ with $\tau_k^+ = T$ and $\tau_k^+ f_k^+ (f_k^+)^2 = \tau_k^* f_k^* (f_k^*)^2$. Note that the constructed solution is feasible. Correspondingly, we can calculate the maximum computation bits of the system under the constructed solution as $C_{\text{tot}}^+ = \lambda + \sum_{i=1,i\neq k}^{K} \frac{\tau_i f_i}{C_{\text{cpu},k}} + \frac{\tau_k^+ f_k^+}{C_{\text{cpu},k}}$. Since both $\tau_k^+ f_k^+ (f_k^+)^2 = \tau_k^* f_k^* (f_k^*)^2$ and $\tau_k^+ = T > \tau_k^*$ hold, we have $f_k^+ < f_k^*$ and $\tau_k^+ f_k^+ > \tau_k^* f_k^*$. Combining the expression of C_{tot}, we can obtain $C_{\text{tot}}^+ > C_{\text{tot}}^*$, which contradicts the above assumption. Thus, $\tau_k^* = T$ holds when the maximum computation bits of the system are achieved.

Based on the above analysis, Proposition 1 is obtained.

Appendix B

From $\mathbf{P}_4$, we can observe that both the objective function and the constraints C3, C5′, C6′, C7, C8′ and C10′ are linear, making both the objective function and these constraints convex. For the rest constraints C1″, C2″ and C9′, we can prove that both C2″ and C9′ are convex constraints as follows. For constraint C2″, its convexity depends on the function f_k^3. Since the second derivative of the function f_k^3 is $6f_k$, which is larger than or equal to 0, the function f_k^3 is convex and C2″ is a convex constraint. For constraint C9′, we need to prove both $F_k\left(t_k^b, \phi_k\right) = t_k^b W \log_2\left(1 + \frac{\xi \phi_k P_s g_k h_k}{t_k^b W \sigma^2}\right)$ and $F_{\sum}\left(\{P_k\}_{k=1}^K, t_a\right) = W t_a \log_2\left(1 + \frac{\sum_{k=1}^K P_k h_k}{t_a W \sigma^2}\right)$ are concave when it is convex. Since the perspective function can preserve convexity, we find that the convexities of $F_k\left(t_k^b, \phi_k\right)$ and $F_{\sum}\left(\{P_k\}_{k=1}^K, t_a\right)$ are same as those of $W \log_2\left(1 + \frac{\xi \phi_k P_s g_k h_k}{W \sigma^2}\right)$ and $W \log_2\left(1 + \frac{\sum_{k=1}^K P_k h_k}{W \sigma^2}\right)$, which are concave. Thus, the constraint C9′ is also convex. The proof is complete.

Appendix C

After carefully analyzing $\mathbf{P}_6$, it is not hard to conclude that the objective function and all constraints except C1‴, C2‴ and C9‴ are linear. Thus, $\mathbf{P}_3$ is convex if and only if constraints C1‴, C2‴ and C9‴ are convex.

On the convexity of C1‴: We only need to prove that $-t_a^{(i+1)} \log_2\left(\sum_{j=k+1}^{K} 2^{\frac{x_j^{(i+1)}}{t_a^{(i+1)}}} h_j + W\sigma^2\right)$ is convex. Using the perspective function, we know that the convexity of $\log_2\left(\sum_{j=k+1}^{K} 2^{x_j^{(i+1)}} h_j + W\sigma^2\right)$ is the same as that of $t_a^{(i+1)} \log_2\left(\sum_{j=k+1}^{K} 2^{\frac{x_j^{(i+1)}}{t_a^{(i+1)}}} h_j + W\sigma^2\right)$. Since $\log_2\left(\sum_{j=k+1}^{K} 2^{x_j^{(i+1)}} h_j + W\sigma^2\right)$ is a log-

sum-exp function which is proved to be convex, $\log_2\left(\sum_{j=k+1}^{K} 2^{x_j^{(i+1)}} h_j + W\sigma^2\right)$ is convex and C1″ is also a convex constraint.

On the convexity of C2‴: The convexity of C2″ is decided by the function $t_{\mathrm{a}}^{(i+1)} 2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}$. If $t_{\mathrm{a}}^{(i+1)} 2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}$ is convex, then the constraint C2″ is convex. Since $t_{\mathrm{a}}^{(i+1)} 2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}$ is the perspective function of $2^{x_k^{(i+1)}}$ that is convex, $t_{\mathrm{a}}^{(i+1)} 2^{\frac{x_k^{(i+1)}}{t_{\mathrm{a}}^{(i+1)}}}$ is convex and the constraint C2″ is convex.

On the convexity of C9‴: Since $\log_2\left(\sum_{j=k+1}^{K} 2^{x_j^{(i+1)}} h_j + W\sigma^2\right)$ is convex, we have $F_k^{\mathrm{T}}\left(t_{\mathrm{a}}^{(i+1)}, x_k^{(i+1)}\right)$ is concave. Combining the results in Appendix B, we can conclude that C9‴ is also convex. The proof is complete.

Abbreviations

IoT: Internet of Things; MEC: Mobile edge computing; OMA: Orthogonal multiple access; NOMA: Non-orthogonal multiple access; AR: Active radios; BackCom: Backscatter communication; DC: Difference of convex; CSI: Channel state information; QoS: Quality-of-service; EH: Energy harvesting.

Acknowledgements

No applicable.

Authors' contributions

ZCM conceived of the study, designed and modeled, carried out experiments, participated in the performance analysis and drafted the manuscript. ZWG participated in the design and helped to draft the manuscript. All authors read and approved the final manuscript.

Authors' Information

Chuangming Zheng received B.S. and M.S. degrees from South China University of Technology, Guangzhou, China, in 1997 and 2003, respectively. During 2005–2012, he was a Researcher in wireless communications in Huawei Technologies Co., Ltd. He has drafted over 20 patents, with over 10 of them being granted in China, the USA, and Europe. He was with Zhoukou Normal University from 2013 to 2016. He received the Ph.D. degree from the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. He is working in Zhoukou Normal University. His current research interests include wireless energy harvesting, mobile-edge computing and radio resource management of LTE wireless networks.

Wengang Zhou received the B.Eng. degree in computational mathematics from Henan Normal University, in 1997, and the M.Eng. degree in computer application technology from the North China University of Technology, in 2007. He is currently pursuing the Ph.D. degree with the City University of Macau. He is currently a Full Professor with the School of Computer Science and Technology, Zhoukou Normal University. His research interests include RF energy harvesting-enabled wireless communications and intelligent algorithm design for wireless communications.

Funding

This work was supported in part by the Key Scientific and Technological Research Projects in Henan Province under Grant 212102210572, in part by Key Scientific Research Projects of Colleges and Universities in Henan Province under Grant 22A510011.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 24 November 2021 Accepted: 19 February 2022
Published online: 25 March 2022

References

1. D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, H.V. Poor, 6G internet of things: a comprehensive survey. IEEE Internet Things J. (2021). https://doi.org/10.1109/JIOT.2021.3103320
2. J. Li, M. Dai, Z. Su, Energy-aware task offloading in the internet of things. IEEE Wirel. Commun. **27**(5), 112–117 (2020). https://doi.org/10.1109/MWC.001.1900495

3. Y. Ye, L. Shi, H. Sun, R.Q. Hu, G. Lu, System-centric computation energy efficiency for distributed NOMA-based MEC networks. IEEE Trans. Veh. Technol. **69**(8), 8938–8948 (2020). https://doi.org/10.1109/TVT.2020.2997673
4. Y. Xu, H. Sun, Y. Ye, Distributed resource allocation for SWIPT-based cognitive ad-hoc networks. IEEE Trans. Cognit. Commun. Netw. (2021). https://doi.org/10.1109/TCCN.2021.3068396
5. H. Wu, H. Tian, G. Nie, P. Zhao, Wireless powered mobile edge computing for industrial internet of things systems. IEEE Access **8**, 101539–101549 (2020). https://doi.org/10.1109/ACCESS.2020.2995649
6. C. You, K. Huang, H. Chae, Energy efficient mobile cloud computing powered by wireless energy transfer. IEEE J. Sel. Areas Commun. **34**(5), 1757–1771 (2016)
7. S. Bi, Y.J. Zhang, Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. IEEE Trans. Wirel. Commun. **17**(6), 4177–4190 (2018)
8. L. Huang, S. Bi, Y.J. Zhang, Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Trans. Mobile Comput. **19**(11), 2581–2593 (2019)
9. F. Wang, Computation rate maximization for wireless powered mobile edge computing, in *2017 23rd Asia-Pacific Conference on Communications (APCC)*, pp. 1–6 (2017). https://doi.org/10.23919/APCC.2017.8304010
10. F. Wang, J. Xu, X. Wang, S. Cui, Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans. Wirel. Commun. **17**(3), 1784–1797 (2018)
11. M. Zeng, R. Du, V. Fodor, C. Fischione, Computation rate maximization for wireless powered mobile edge computing with NOMA, in *Proceedings on IEEE WoWMoM*, pp. 1–9 (2019)
12. L. Shi, Y. Ye, X. Chu, G. Lu, Computation energy efficiency maximization for a NOMA-based WPT-MEC network. IEEE Internet Things J. **8**(13), 10731–10744 (2021). https://doi.org/10.1109/JIOT.2020.3048937
13. B. Li, F. Si, W. Zhao, H. Zhang, Wireless powered mobile edge computing with NOMA and user cooperation. IEEE Trans. Veh. Technol. **70**(2), 1957–1961 (2021). https://doi.org/10.1109/TVT.2021.3051651
14. F. Rezaei, C. Tellambura, S. Herath, Large-scale wireless-powered networks with backscatter communications—a comprehensive survey. IEEE Open J. Commun. Soc. **1**, 1100–1130 (2020). https://doi.org/10.1109/OJCOMS.2020.3012466
15. H. Yang, Y. Ye, X. Chu, S. Sun, Energy efficiency maximization for UAV-enabled hybrid backscatter-harvest-then-transmit communications. IEEE Trans. Wirel. Commun. (2021). https://doi.org/10.1109/TWC.2021.3116509
16. Y. Xu, B. Gu, R.Q. Hu, D. Li, H. Zhang, Joint computation offloading and radio resource allocation in MEC-based wireless-powered backscatter communication networks. IEEE Trans. Veh. Technol. **70**(6), 6200–6205 (2021). https://doi.org/10.1109/TVT.2021.3077094
17. S. Gong, Y. Xie, J. Xu, D. Niyato, Y.-C. Liang, Deep reinforcement learning for backscatter-aided data offloading in mobile edge computing. IEEE Netw. **34**(5), 106–113 (2020). https://doi.org/10.1109/MNET.001.1900561
18. Y. Zou, J. Xu, S. Gong, Y. Guo, D. Niyato, W. Cheng, Backscatter-aided hybrid data offloading for wireless powered edge sensor networks, in *Proceedings on IEEE GLOBECOM*, pp. 1–6 (2019). https://doi.org/10.1109/GLOBECOM38437.2019.9014101
19. L. Shi, R.Q. Hu, J. Gunther, Y. Ye, H. Zhang, Energy efficiency for RF-powered backscatter networks using HTT protocol. IEEE Trans. Veh. Technol. **69**(11), 13932–13936 (2020). https://doi.org/10.1109/TVT.2020.3014500
20. L. Shi, Y. Ye, X. Chu, G. Lu, Computation bits maximization in a backscatter assisted wirelessly powered MEC network. IEEE Commun. Lett. **25**(2), 528–532 (2021). https://doi.org/10.1109/LCOMM.2020.3027294
21. L. Shi, Y. Ye, G. Zheng, G. Lu, Computational EE fairness in backscatter-assisted wireless powered MEC networks. IEEE Wirel. Commun. Lett. **10**(5), 1088–1092 (2021). https://doi.org/10.1109/LWC.2021.3058295
22. Y. Ye, L. Shi, X. Chu, D. Li, G. Lu, Delay minimization in wireless powered mobile edge computing with hybrid BackCom and AT. IEEE Wirel. Commun. Lett. **10**(7), 1532–1536 (2021). https://doi.org/10.1109/LWC.2021.3073406
23. Y. Ye, L. Shi, X. Chu, G. Lu, Throughput fairness guarantee in wireless powered backscatter communications with HTT. IEEE Wirel. Commun. Lett. **10**(3), 449–453 (2021). https://doi.org/10.1109/LWC.2020.3014740
24. S.H. Kim, D.I. Kim, Hybrid backscatter communication for wireless-powered heterogeneous networks. IEEE Trans. Wirel. Commun. **16**(10), 6557–6570 (2017). https://doi.org/10.1109/TWC.2017.2725829
25. J. Papandriopoulos, J.S. Evans, SCALE: a low-complexity distributed protocol for spectrum balancing in multiuser DSL networks. IEEE Trans. Inf. Theory **55**(8), 3711–3724 (2009). https://doi.org/10.1109/TIT.2009.2023751

Publisher's Note