Abstract
This paper focuses on a mobile edge-computing-enabled heterogeneous network. A battery level-aware task-scheduling framework is proposed to improve the energy efficiency and prolong the operating hours of battery-powered mobile devices. The formulated optimization problem is a typical mixed-integer nonlinear programming problem. To solve this non-deterministic polynomial (NP)-hard problem, a decomposition-based task-scheduling algorithm is proposed. Using an alternating optimization technology, the original problem is divided into three subproblems. In the outer loop, task offloading decisions are yielded using a pruning search algorithm for the task offloading subproblem. In the inner loop, closed-form solutions for computational resource allocation subproblems are derived using the Lagrangian multiplier method. Then, it is proven that the transmitted power-allocation subproblem is a unimodal problem; this subproblem is solved using a gradient-based bisection search algorithm. The simulation results demonstrate that the proposed framework achieves better energy efficiency than other frameworks. Additionally, the impact of the battery level-aware scheme on the operating hours of battery-powered mobile devices is also investigated.

KEYWORDS
battery level aware, energy efficiency, heterogeneous network, mobile edge computing, task scheduling

1 | INTRODUCTION

The evolution of information and communication technology, besides the proliferation of mobile devices, has also expanded the boundaries of the Internet of Things (IoT). The sustained development of IoT is a challenging prospect due to the advances in numerous emerging applications, such as remote medical surgery, environmental monitoring [1], transportation safety [2], and smart industrial manufacturing. Furthermore, the limited computational resources of battery-powered mobile devices cannot deal with the problem of increasing latency in emerging applications. Nowadays, mobile cloud computing is considered as a potential solution that meets the strict requirements of emerging IoT applications due to its specific advantages, such as convenient access, high processing capacity, and competitive system scalability [3,4]. However, mobile cloud
computing cannot address several issues, such as unpredictable network latency, expensive bandwidth, and single point of failure [5,6].

According to predictions, the number of IoT devices connected to core networks will reach 29.3 billion by 2023, up from 18.4 billion in 2018, where the number of machine-to-machine (M2M) connections (which is also referred to as IoT) will reach 14.7 billion [7]. Due to the ever-increasing sensing, computing, and communication capabilities of mobile devices, 40% of data generated by IoT devices will be handled by edge devices. However, processing this huge amount of data is an overwhelming challenge due to the high heterogeneity of IoT-based applications. The bandwidth of mobile communications has been greatly improved by the expansion of 5G network technologies, which exploit the millimeter wave spectrum and the advantages of multiple antennas. 5G networks are committed to meeting the needs if emerging application scenarios, in which ultrareliable, low latency communications (uRLLC) are mainly used for latency-sensitive services.

The 5G communication network is more decentralized than other networks. By deploying small-scale or portable data centers at the edge of heterogeneous networks, edge processing of terminal requests (which is called mobile edge computing [MEC]) is performed to meet the ultra low latency requirements of uRLLC. MEC has recently become a prevailing computing paradigm. With the help of computational resources at the edge of wireless networks, the required computational tasks can be processed in situ [8]. In proximity to mobile devices, edge-computing servers can process nearby computational tasks promptly, resulting in reduced task times and increased transfer rates. By utilizing edge computational resources, MEC can greatly avoid the long delay and heavy overhead encountered in core networks [9]. In next-generation wireless networks, access points with edge-computing servers can provide data storage, data processing, and other services [10].

MEC is an effective way to satisfy the computationally intensive scheduling of latency-sensitive tasks. MEC exhibits the following three main advantages: (1) Compared with local processing, the offloading of tasks to edge-computing servers can overcome the restrictions of limited computational resources available on mobile devices; (2) compared with cloud computing, MEC can avoid the transmission delay caused by mobile devices located away from cloud centers; (3) in battery-powered mobile devices, because the energy consumption required for transmission is usually less than that required for task processing, the offloading of tasks can reduce the energy consumption and prolong the operating hours of mobile devices. However, it is difficult to schedule tasks effectively because of the decentralized characteristics of heterogeneous networks. As an emerging technology, the software-defined network (SDN) technology (which benefits from the network function visualization technology [11]) enables programmable, dynamic, and efficient network configuration in next generation wireless networks. SDN technology provides logically centralized control and global scheduling by separating the control plane (routing processing) from the data plane (network packet forwarding processing) and putting centralizing intelligence in one network component [12,13]. Because of this centralized control, an optimal task-scheduling decision for MEC can be achieved.

In recent years, several studies have addressed task-scheduling in the MEC scenario. Liu and others [14] proposed a stochastic computation task-scheduling policy to achieve a minimum average delay by incorporating different timescales in the data processing. By exploiting the SDN technology, Chen and Hao [4] investigated the task offloading problem to minimize the task duration in ultradense networks under power constraints. By utilizing the device-to-device (D2D) communication, Xie and others [15] proposed a peer-to-peer (P2P) enhanced task offloading and resource allocation framework to minimize the computational delay in MEC networks. However, different application environments have different quality of service (QoS) requirements. Thus, all application scenarios cannot be implemented by focusing only on the task duration.

Based on the demands of mobile devices in different application scenarios, the joint optimization of task duration and energy consumption in mobile devices has been investigated [16,17]. Tran and Pompili [17] investigated the task-scheduling problem in multi-server MEC networks to maximize a user’s task offloading gain, which is defined as a weighted sum of the reductions in the task duration and energy consumption. Zhang and others [16] considered the energy harvesting technique as an extra energy supply to mobile devices and proposed a Lyapunov optimization method to investigate the trade-off between the energy consumption and the task duration in MEC systems.

With the development of green communications, the energy consumption in mobile devices has been an increasing concern. Labidi and others [18] provided a joint optimization framework to minimize the average energy consumption at user terminals in small cell long-term evolution (LTE)-based networks. Furthermore, Zhang and others [19] exploited the multiple access features of 5G heterogeneous networks to formulate an energy consumption minimization problem for MEC in 5G heterogeneous networks. Subsequently, they solved this problem using a three-stage offloading scheme. Chen
and others [20] investigated a more complex scenario of multiuser computation offloading in a multichannel wireless contention environment and formulated a multiuser computation offloading game. They also designed a distributed scheduling algorithm to achieve Nash equilibrium. However, the remaining battery level in mobile devices (especially for energy efficiency maximization in densely deployed multi-small cell heterogeneous networks) has received less attention in the literature.

Different from previous studies, in this paper, we focus on a densely deployed multi-small cell heterogeneous network, considering the remaining battery level in mobile devices, the demand for energy by mobile devices, and the limited resources of edge-computing servers. The joint optimization of task offloading, power control, and computational resource allocation in edge-computing servers is a large-scale mixed-integer nonlinear programming (MINLP) problem, which is nondeterministic polynomial (NP)-hard. To solve this optimization problem, a battery level-aware task-scheduling framework is proposed. The main contributions of this paper are summarized as follows:

1. To improve the energy efficiency and prolong the operating hours of mobile devices in an MEC-enabled heterogeneous network, a battery level-aware task-scheduling framework is proposed. In this framework, the task offloading decision, the transmitted power control, and the computational resource allocation are jointly optimized.

2. The energy efficiency maximization problem is formulated as a large-scale MINLP problem and is NP hard. Because it is nonconvex and challenging to solve, the original problem is decoupled into a task offloading master problem, a computational resource allocation subproblem, and a power-allocation subproblem.

3. The task offloading master problem is converted into a typical knapsack problem, and the optimal decision is obtained using a pruning search-based algorithm.

4. Regarding the resource allocation subproblems at mobile devices and MEC servers, a closed-form solution is derived using the Lagrangian method.

5. It is proven that the power-allocation subproblem is unimodal. Then, an approximate optimal solution can be obtained using a gradient-based bisection search algorithm.

The rest of the paper is organized as follows. The system model of the SDN- and MEC-enabled heterogeneous network is constructed in Section 2. The problem formulation is described in Section 3. Then, a task-scheduling algorithm for energy efficiency maximization is proposed in Section 4. The simulation results are presented in Section 5. Finally, Section 6 concludes this paper.

2 | SYSTEM MODEL

In this section, the system architecture is described to provide the necessary background. Then, the related models adopted in this paper (including the local processing model, the communication model, the edge-computing model, and the battery level-aware model) are presented.

2.1 | System architecture

As shown in Figure 1, an SDN- and MEC-enabled heterogeneous network architecture, which consists of two parts (i.e., the control plane and the data plane) is considered. In the control plane, there is an SDN controller deployed in the macro base station. Because the SDN controller has global state information about the wireless network, it is capable of scheduling and allocating all communication and computation resources. In the data plane, it is assumed that the macro-base station is located at the center of the system. There are $N$ small cells at the edge of the heterogeneous network, where $s_n$ denotes the $n$th small cell ($n \in \{1, 2, \ldots, N\}$). All small base stations in a small cell are connected to the macro-base station via wired links. An edge-computing server is deployed at the
macro-base station. Assume that $M_n$ denotes the number of mobile devices in the $n$th small cell, and $M_{n,i}$ is the $i$th mobile device ($i \in \{1, 2, ..., M_n\}$) in the $n$th small cell. Each mobile device can be represented as a 2-tuple $m_{n,i} = (F_{n,i}, P_{n,i})$, where $F_{n,i}$ is the maximum computational frequency and $P_{n,i}$ is the maximum transmitted power.

Assume that each mobile device is assigned to handle several computationally intensive and latency-sensitive tasks. These tasks are independent of each other and cannot be split. A computational task in mobile device $m_{n,i}$ is modeled as $q_{n,i} = \{d_{n,i}, r_{n,i}\}$, where $d_{n,i}$ is the data size of the computational task, and $r_{n,i}$ denotes the maximum tolerable delay of the computational task. The task can be performed either by the mobile device itself, or it can be scheduled to an edge-computing server. Assume that $a_{n,i} \in A$ denotes the scheduling decision of computational task $q_{n,i}$. When $q_{n,i}$ is processed locally on the mobile device $m_{n,i}$, $a_{n,i} = 0$, otherwise, $a_{n,i} = 1$.

### 2.2 Local processing model

The maximum computational ability of mobile device $m_{n,i}$ is denoted as $F_{n,i}$. Because the CPU frequency can be dynamically adjusted using dynamic voltage and frequency scaling technology [21], the computational resource assigned to computational task $q_{n,i}$ is $f_{n,i}$. Assume that $c_{n,i} = \lambda d_{n,i}$ denotes the total CPU cycles required to accomplish task $q_{n,i}$, where $\lambda$ is the computational coefficient, which depends on the type of the task. When a computational task is processed by the mobile device itself, the computational delay $T_{n,i}$ is

$$T_{n,i} = \frac{c_{n,i}}{f_{n,i}}$$

Then, the processing rate of task $q_{n,i}$ is

$$R_{n,i} = \frac{d_{n,i}}{f_{n,i}} = \frac{f_{n,i}}{\lambda}$$

The power required to perform a computational task can be calculated as

$$P_{n,i} = \kappa \left( \frac{d_{n,i}}{\gamma_{n,i} F^c} \right)^{\frac{2}{3}}$$

where $\kappa$ is a coefficient depending on the chip architecture. According to the practical measurement in [22], $\kappa$ is set to $10^{-26}$.

### 2.3 Communication model

To make the most of the available computational resources on an edge-computing server, the computational tasks are transmitted to the server via wireless links. If mobile device $m_{n,i}$ transmits its computational task to the base station, the channel coefficient from mobile device $m_{n,i}$ to the base station can be written as

$$h_{n,i} = L_{n,i} g_{n,i},$$

where $L_{n,i} = l_{n,i}^{-\beta}$ is the path loss and $l_{n,i}$ is the distance between mobile device $m_{n,i}$ and the base station. $\beta$ is the path loss coefficient, and $g_{n,i}$ is the Rayleigh fading channel gain. Then, the signal-to-interference-plus-noise ratio from mobile device $m_{n,i}$ to the base station can be expressed as

$$\Gamma_{n,i} = \frac{p_{n,i} h_{n,i}}{\sigma^2 + I_{n,i}}$$

where $I_{n,i} = \sum_{k \in M_n \setminus i} P_{n,k} h_{n,k}$ is the interference caused by other mobile devices and can be approximated by $\mathbb{E}(I_{n,i})$ [23]. By convention, $\sigma^2$ is the additive white Gaussian noise power.

According to Shannon’s formula, the achievable uplink transmission rate can be obtained as

$$R_{n,i} = B \log_2 (1 + \Gamma_{n,i}),$$

where $B$ is the bandwidth, $p_{n,i}$ is the transmitted power, and $h_{n,i}$ is the channel gain between mobile device $m_{n,i}$ and the base station.

### 2.4 Edge-computing model

Assume that $F^c$ denotes the total CPU frequency of an edge-computing server, and $\gamma_{n,i}$ denotes the percentage of computational resource assigned to computational task $q_{n,i}$. The total delay of the computational task includes the transmission delay via a wireless link and the computational delay in the edge-computing server. The total delay can be written as

$$T_{n,i} = \frac{d_{n,i}}{R_{n,i}} + \frac{\lambda d_{n,i}}{\gamma_{n,i} F^c}$$

Then, the processing rate of $q_{n,i}$ can be obtained as
maximum energy efficiency. According to the processing rate and power consumption, the utility function of local processing can be calculated as

$$\eta_{\text{local}} = \frac{\varepsilon_n R_n}{P_n} = \frac{\varepsilon_n}{\lambda \kappa (f^j_{i,n})}. \quad (12)$$

Similarly, the utility function of edge computing is obtained as

$$\eta_{\text{mec}} = \frac{\varepsilon_n R_{n,j}}{P_{n,j}} = \frac{\varepsilon_n}{\lambda (f^j_{i,n})}. \quad (13)$$

Generally, the overall energy efficiency of the system can be written as

$$\eta = \eta_{\text{local}} + \eta_{\text{mec}} = \sum_{n=1}^{N} \sum_{i=1}^{M} (1-a_{n,i}) - \frac{\varepsilon_n}{\lambda \kappa (f^j_{i,n})}$$

$$+ \sum_{n=1}^{N} \sum_{i=1}^{M} a_{n,i} \frac{\varepsilon_n}{\lambda \kappa (f^j_{i,n})} \left(\frac{P_n + P_c}{P_{n,j}}\right). \quad (14)$$

The overall energy efficiency includes both the energy efficiency of mobile devices and edge-computing services, so that all the participants in the network are involved in achieving the optimum energy efficiency of the entire system.

P1: \[ \max_{a_{n,i}} \eta \]

s.t. \[ C1: a_{n,i} \in \{0,1\}, \]

\[ C2: 0 \leq P_{n,i} \leq P_{n,i}^\text{th}, \]

\[ C3: 0 \leq f^j_{i,n} \leq F_{n,i}, \]

\[ C4: I_{n,i} \leq I_{n,i}^\text{th}, \]

\[ C5: T_{n,i} \leq \tau_{n,i}, \]

\[ C6: \sum_{n=1}^{N} \gamma_{n,i} \leq 1, \forall n,i, \]

where \( I_{n,i} \) is the interference experienced by mobile device \( m_{n,i} \) and \( P_{n,i} \) indicates the maximum transmitted power. Constraint C1 shows that the decision variable \( a_{n,i} \) is binary. Constraint C2 indicates that the power transmitted by a mobile device cannot exceed its maximum transmitted power. Constraint C3 shows that the allocated computational resource is limited by the maximum computational resource. Constraint C4 indicates that the interference cannot exceed a threshold value. Constraint C5 represents the maximum tolerance delay

2.5 Battery level-aware model

Both the processing delay and energy consumption are vital metrics during the processing of computational tasks that determine the QoS and battery level limitation of mobile devices. Therefore, a weighting factor \( \omega_{n,i} \) is required to weight the energy consumption and processing time. Reducing the processing time while prolonging the battery life of mobile devices can be achieved by increasing the residual energy rate \( \omega_{n,i} \) of the battery; \( \omega_{n,i} \) is defined as

$$\omega_{n,i} = \frac{E_{n,i}}{E_{n,i}^0}, \quad (10)$$

where \( E_{n,i} \) is the residual battery level of mobile device \( m_{n,i} \) and \( E_{n,i}^0 \) is the total battery level in Joules [24]. \( \omega_{n,i} \) is a definite value that reflects the real-time battery level of mobile devices.

To describe as close as possible an actual case scenario, we define

$$\xi_{n,i} = \sqrt{\omega_{n,i} (2 - \omega_{n,i})}. \quad (11)$$

According to (11), the above formula reflects the fact that the handle willingness will drop sharply when the battery is at a low level, whereas it will drop slowly when the battery is at a high level.

3 PROBLEM FORMULATION

In this section, the problem of finding the system’s maximum energy efficiency is formulated. The utility function is formulated as follows to facilitate the investigation of
of computational task \( q_{n,i} \) that must be met, where \( T_{n,i} = (1 - a_{n,i})T_{n,i}^{h} + a_{n,i}T_{n,i}^{c} \). The optimization problem is a typical MINLP problem.

4 | PROPOSED ALGORITHM

In this section, a decomposition-based algorithm is proposed to solve the optimization problem.

4.1 | Optimal local computational resource allocation

When the integer variable \( a_{n,i} \) is fixed (\( a_{n,i} = 0 \)), the optimization problem P1 can be rewritten as

\[
P2: \max \sum_{n=1}^{N} \sum_{i=1}^{M_{n}} \xi_{n,i} \frac{\partial}{\partial f_{n,i}} \frac{1}{\lambda_{k}(f_{n,i})}
\]

\[
s.t. \quad C1: 0 \leq f_{n,i}^{l} \leq F_{n,i},
C2: T_{n,i}^{l} \leq T_{n,i}, \forall n, i.
\]

The objective function is a strictly monotonically decreasing function with respect to \( f_{n,i}^{l} \). Constrained by the maximum tolerated latency of the computational task and the computing power of the mobile device, the optimal computational resource allocation can be easily obtained as

\[
f_{n,i}^{l} \geq \frac{\lambda_{d_{n,i}}}{\tau_{n,i}} \triangleq f_{n,i}^{l,\text{min}}.
\]

According to constraint C1 in (16) and (17), the closed-form expression of the optimal \( f_{n,i} \) can be written as

\[
(f_{n,i}^{l})^* = \begin{cases} 
\frac{\lambda_{d_{n,i}}}{\tau_{n,i}}, & f_{n,i}^{l,\text{min}} \leq F_{n,i}, \\
F_{n,i}, & f_{n,i}^{l,\text{min}} > F_{n,i}.
\end{cases}
\]

4.2 | Optimal edge computational resource allocation

Assuming that the integer variable \( a_{n,i} \) is fixed (\( a_{n,i} = 1 \)), the subproblem is formulated as

\[
P3: \max_{p \in [0, 1]} \sum_{n=1}^{N} \sum_{i=1}^{M_{n}} \frac{P_{n,i}}{P_{n,i}^{h}}
\]

\[
s.t. \quad C1: 0 \leq P_{n,i} \leq P_{n,i}^{h},
C2: I_{n,i} \leq I_{n,i}^{h},
C3: T_{n,i}^{c} \leq T_{n,i},
C4: \sum_{n=1}^{N} T_{n,i} \leq 1, \forall n, i.
\]

With a fixed set of transmitted power levels, the optimization problem P3 can be rewritten as

\[
P4: \min_{\gamma} \frac{\sum_{n=1}^{N} \sum_{i=1}^{M_{n}} P_{n,i} + P_{c}}{R_{n,i}} + \gamma \frac{(p_{n,i} + p_{c})}{\gamma_{n,i}}
\]

\[
s.t. \quad \sum_{n=1}^{N} \gamma_{n,i} \leq 1, \forall n, i.
\]

Based on Lemma 1, it can be proved that the optimization problem P4 is a convex optimization problem.

**Lemma 1.** Given a fixed set of transmitted power levels, the original optimization problem in P4 with respect to \( \gamma \) is convex.

**Proof.** First, P4’s Hessian matrix can be obtained as

\[
H = \begin{bmatrix}
\frac{\partial^{2}f}{\partial \gamma_{1,1}} & \ldots & \frac{\partial^{2}f}{\partial \gamma_{1,M_{n}}}
\end{bmatrix}
\]

where each element

\[
\frac{\partial^{2}f}{\partial \gamma_{n,i} \partial \gamma_{m,j}} = \begin{cases} 
\frac{2\lambda}{(\gamma_{n,i})^{2}}, & \text{if } n = m, i = j, \\
0, & \text{otherwise}.
\end{cases}
\]

Obviously, the parameters reported in (21) are all positive numbers; consequently, it can be derived that \( \frac{\partial^{2}f}{\partial \gamma_{n,i} \partial \gamma_{m,j}} \geq 0 \). Thus, it can be concluded that every eigenvalue of Hessian matrix \( H \) is positive, and \( H \) is an asymmetrical positive definite matrix. Because the constraints are linear, the optimization problem P4 is a convex optimization problem.

The proof is now complete.

Based on Lemma 1 and using the Karush–Kuhn–Tucker (KKT) conditions, Theorem 1 can be obtained.
Theorem 1. In the optimization problem P4, the optimal value $\gamma$ can be obtained.

Proof. In terms of the optimization problem in (20) and the constraint in (20), the Lagrangian function is obtained by

$$L(\gamma, \mu) = \sum_{n=1}^{N} \sum_{i=1}^{M_n} \left( \frac{p_{n,i} + p_c}{R_{n,i}} + \frac{\lambda(p_{n,i} + p_c)}{\gamma_{n,i} F^c} \right) + \mu \left( \sum_{n=1}^{N} \sum_{i=1}^{M_n} \gamma_{n,i} - 1 \right),$$

(22)

where $\mu$ is the Lagrange multiplier. The problem in (22) is solved using the KKT conditions.

$$\nabla f(\bar{\gamma}_{1,1}, \ldots, \bar{\gamma}_{n,i}) + \bar{\mu} \nabla \left( \sum_{n=1}^{N} \sum_{i=1}^{M_n} \gamma_{n,i} - 1 \right) = 0$$

where $\bar{\gamma}$ and $\bar{\mu}$ are the points that satisfy the KKT conditions. According to above formulas, the optimal $(\gamma_{n,i})^*$ is obtained as

$$\gamma_{n,i}^* = \frac{\sqrt{\lambda(p_{n,i} + p_c)}}{\sum_{n=1}^{N} \sum_{i=1}^{M_n} \sqrt{\lambda(p_{n,i} + p_c)}}.$$

(23)

The proof is completed.

After obtaining the optimal $\gamma$, the optimization problem P3 can be solved by solving for the optimal energy efficiency of each mobile device $m_{n,i}$. Then, the optimal subproblem can be rewritten as

$$\text{P5:}$$

$$\begin{align*}
\text{min}_{p} & \quad \frac{p_{n,i} + p_c}{R_{n,i}} + \frac{\lambda(p_{n,i} + p_c)}{\gamma_{n,i}^* F^c} \\
\text{s.t.} & \quad C1: 0 \leq p_{n,i} \leq P_{n,i}, \\
& \quad C2: I_{n,i} \leq I_{n,i}^h, \\
& \quad C3: T_{n,i}^c \leq t_{n,i}, \forall n, i.
\end{align*}$$

(24)

Lemma 2. Function

$$f(p_{n,i}) = \frac{p_{n,i} + p_c}{B \log_2(1 + p_{n,i} z)} + \frac{\lambda(p_{n,i} + p_c)}{\gamma_{n,i}^* F^c}$$

is unimodal, where $z = h_{n,i}/(\sigma^2 + I_{n,i})$.

Proof. Let $x = 1/B \log_2(1 + p_{n,i} z)$; consequently, it can be derived that $p_{n,i} = (2^{1/x} - 1)/z$. Then, the original function can be rewritten as

$$f(x) = \left(x + \frac{\lambda}{\gamma_{n,i}^* F^c} \right) \left(\frac{2^{1/x} - 1}{z} + p_c \right).$$

Since the second derivative of $f(x)$ with respect to $x$ is greater than 0, function $f(x)$ is a convex function with respect to $x (x \geq 0)$. Hence, function $f(x)$ has a minimum, which is unique. Due to the procedure followed, the relationship between $x$ and $p_{n,i}$ is a one-to-one relationship. Therefore, $f(p_{n,i})$ has a minimum value, which is unique. In other words, function $f(p_{n,i})$ is unimodal.

$$\begin{align*}
\frac{\partial f(x)}{\partial x} &= p_c + \frac{2^{1/x} - 1}{z} + \frac{2^{3/2} \log(2) \left(x + \frac{1}{2} p_c \right)}{B z x^2}, \\
\frac{\partial^2 f(x)}{\partial x^2} &= \frac{2^{1/x} \log(2) \left(\frac{1}{2} \log(2) + Bex \right)}{B^2 z^2 \gamma_{n,i}^* F^c x^4} \left(1 + \gamma_{n,i}^* F^c x \log(2) \right).
\end{align*}$$

The proof is completed.

Based on Lemma 2, it can be proved that $f(p_{n,i})$ is a strict unimodal function with a unique maximum that tends to initially increase and then decrease, as $p_{n,i}$ increases. According to the nature of the first derivative of $f(p_{n,i})$ with respect to $p_{n,i}$, the unique optimal solution $p_{n,i}^*$ should satisfy

$$\frac{\partial f(p_{n,i})}{\partial p_{n,i}} \bigg|_{p_{n,i}=p_{n,i}^*} = 0.$$

(25)
Then, the optimal power $p^*_n$ can be obtained using the gradient-based bisection search algorithm. The detailed algorithm is shown in Algorithm 1.

**Algorithm 1 Bisection search algorithm based on gradient**

1. if $\frac{\partial f(p)}{\partial p} |_{p=P_{\text{min}}} > 0$, $\frac{\partial f(p)}{\partial p} |_{p=P_{\text{max}}} > 0$ then
2. $p^* = P_{\text{max}}$
3. else if $\frac{\partial f(p)}{\partial p} |_{p=P_{\text{min}}} < 0$, $\frac{\partial f(p)}{\partial p} |_{p=P_{\text{max}}} < 0$ then
4. $p^* = P_{\text{min}}$
5. else
6. Set $p_1 = \frac{P_{\text{max}} + P_{\text{min}}}{2}$, $g_1 = \frac{\partial f(p)}{\partial p} |_{p=p_1}$
7. Set the multiplier $\alpha (\alpha > 1)$
8. if $g_1 < 0$ then
9. repeat
10. $p_2 = \frac{p_1}{\alpha}$
11. Calculate $g_2 = \frac{\partial f(p)}{\partial p} |_{p=p_2}$
12. until $g_2 \geq 0$
13. else
14. repeat
15. Set $p_2 = p_1$, $p_1 = \alpha p_1$
16. Calculate $g_1 = \frac{\partial f(p)}{\partial p} |_{p=p_1}$
17. until $g_1 \leq 0$
18. end if
19. while $|p_1 - p_2| > \varepsilon$ do
20. $p^* = \frac{p_1 + p_2}{2}$, $g^* = \frac{\partial f(p)}{\partial p} |_{p=p^*}$
21. if $g^* < 0$ then
22. $p_1 = p^*$
23. else
24. $p_2 = p^*$
25. end if
26. end while
27. end if

Then, the optimal power $p^*_n$ can be obtained using the gradient-based bisection search algorithm. The detailed algorithm is shown in Algorithm 1.

4.3 | Pruning search-based task-scheduling algorithm

After allocating the computational resources and optimizing the transmitted power, the task-scheduling problem is converted into an integer programming problem. This problem can be solved by converting the task-scheduling problem into the following form:

$$P6: \max_a \eta(a,p^*,f^*,g^*)$$
$$\text{s.t. } C1: a_{n,i} \in \{0,1\},$$
$$C2: T_{n,i} \leq r_{n,i} \ \forall n,i.$$  (26)

Thus, the original mixed-integer programming problem is transformed into a pure-integer programming problem. This optimization problem is extremely tough to be solved in polynomial time. The most efficient approach to solve the problem is to use exhaustive search. However, the total number of task-scheduling decisions would be $2^x$, where $x = \sum_{n=1}^{N} M_n$. Consequently, it is practically impossible to solve this problem using exhaustive search. To overcome this difficulty, the pruning search-based task-scheduling algorithm is adopted.

The pruning search-based task-scheduling algorithm ensures the optimal global solution by enumerating only a small subset of the entire solution space. During the branching process, the solution space is iteratively divided into multiple smaller subspaces and then maximizes P6 on these smaller subspaces. By performing branching, all the candidate solutions can be enumerated. However, enumerating all the candidate solutions is an extremely challenging task. The proposed algorithm updates the lower bound of the maximum and uses this lower bound to prune the subspace. In this way, candidate solutions that are lower than the lower bound are eliminated. The pruning search algorithm is proposed as a centralized way to find all the candidate solutions. The proposed algorithm can also be developed in a distributed way, such as that reported in [25,26].

Finally, the flow of the decomposition-based task-scheduling algorithm is presented in Figure 2.
In this section, simulation results are presented to evaluate the proposed battery level-aware task-scheduling algorithm. In an SDN- and MEC-enabled heterogeneous network, the edge-computing server is located at the small base station, and its total computational frequency is 10 GHz. The data size of tasks is randomly generated from 0.5 MB to 0.5 MB. The tolerable duration of tasks is randomly distributed between 0.5 and 1.5 s. The maximum CPU frequency of mobile devices ranges from 0.5 to 1.5 GHz. Furthermore, the maximum transmitted power of mobile devices is 30 dBm. The path loss coefficient $\beta$ of the system is 2.5. The bandwidth of each subchannel is 1 MHz. The proposed algorithm is compared with two different baseline algorithms. “NO” represents the non-offloading scheme, where all computational tasks are computed by mobile devices, and “SDTO” represents the software-defined task offloading scheme reported in [4].

Figure 3 shows that the proposed algorithm achieves better energy efficiency than the NO scheme and the SDTO scheme. In the proposed algorithm, power control, task offloading and computational resource allocation are jointly used to improve the energy efficiency of mobile devices. Compared with the NO scheme (which operates at the maximum CPU frequency), the NO scheme (which operates at the optimal CPU frequency) achieves better energy efficiency. It can also be observed that the proposed algorithm (which operates at a higher battery level) achieves better energy efficiency. This is because mobile devices operating at high battery levels do not need to save energy by reducing their energy efficiency.

Figure 4 shows the impact of the average data size of computational tasks on the energy consumption under different battery levels and different CPU frequencies. The NO scheme (which operates at the maximum CPU frequency) achieves the largest energy consumption because all computational tasks are performed locally with the maximum CPU frequency. The SDTO scheme with its lower latency consumes less energy because part of computational tasks is offloaded to edge-computing servers. Constrained by the delay, after the CPU frequency allocation, the NO scheme (which operates at the optimal CPU frequency) consumes less energy. Furthermore, in the proposed algorithm, where power control, task offloading, and computational resource allocation are jointly used, achieves the lowest energy consumption. In the battery level-aware algorithm, mobile devices with a low battery level can prolong their operating hours by reducing a small amount of their energy efficiency.

Figure 5 shows the impact of the average tolerable delay of computational tasks on the energy efficiency under different battery levels. It can be clearly observed that the energy efficiency of the network increases as the average tolerable delay increases. Compared with the NO scheme, the proposed algorithm achieves higher energy efficiency by controlling the transmitted power and scheduling the computational tasks. For different battery levels, the higher the battery level of the mobile device, the higher the energy efficiency the system can achieve. This is because, as the battery level of mobile devices decreases, to prolong their operating hours, the mobile devices give a high priority to offload their computational tasks.

The impact of the number of small cells on the energy efficiency under different transmitted power levels is presented in Figure 6. Two fixed transmitted power levels
(i.e., 15 dBm to 15 dBm) are selected to perform the simulation verification. Figure 6 shows that the energy efficiency increases with the number of small cells. This is mainly because the number of edge-computing servers increases, as the number of small cells increases. Therefore, the computational tasks have more opportunities to be scheduled to the edge-computing servers to improve the energy efficiency.

Figure 7 shows the impact of the number of small cells on the energy consumption under different transmitted power levels. Similar to the energy efficiency, the proposed algorithm achieves less energy as the number of small cells increases. Moreover, it can be observed that the proposed algorithm achieves less energy consumption than the fixed transmitted power for most of the time. Also, mobile devices with low battery power levels consume less energy.

The impact of the number of mobile devices on the energy consumption is presented in Figure 8. It can be observed that the proposed algorithm achieves less energy consumption when the number of mobile devices is less than 17. This is because all computational tasks have been offloaded to the edge computing server. Furthermore, compared with mobile devices with high battery levels, mobile devices with low battery levels achieve lower energy consumption. This is because mobile devices with lower battery levels are more willing to offload their computational tasks, instead of performing them locally.

In the proposed algorithm, the battery level is introduced into the definition of the weighting factor. Figure 9 shows the impact of the battery level on the residual
energy of mobile devices. In this figure, the NO scheme, the proposed algorithm without the battery level-aware scheme, and the proposed algorithm with the battery level-aware scheme (xi = 0.5 and xi = 1) are selected and compared. First, it can be observed that the proposed algorithm achieves less energy consumption than the NO scheme. Then, with the continuous energy consumption from the battery, it can be observed that the proposed algorithm with the battery level-aware scheme can save more battery power than the algorithm without the battery level-aware scheme. The reason is that the weighting coefficient ξ decreases, as the battery level decreases. Then, the computational tasks of the low battery level mobile devices are preferentially offloaded to the edge-computing server. In fact, the mobile devices with lower battery levels care more about energy consumption than other mobile devices.

6 | CONCLUSION

The objective of this paper was to provide energy-efficient MEC services in heterogeneous networks and to prolong the operating hours of mobile devices. For this purpose, the battery level-aware concept was introduced. Specifically, a battery level-aware task-scheduling framework was proposed to provide energy-efficient task-scheduling services. The formulated optimization problem is a typical mixed-integer nonlinear programming problem. To solve this nonconvex problem, a battery level-aware task-scheduling algorithm based on the problem decomposition principle was designed. The optimization problem was decomposed into a task offloading master subproblem, a computational resource allocation subproblem, and a power control subproblem. It was proven that the computational resource allocation subproblem is convex, and a closed-form solution was obtained. The power control subproblem is a unimodal problem, which was solved using a bisection search algorithm based on gradients. For the task offloading master subproblem, the optimal solution was obtained using a pruning search-based task-scheduling algorithm. Numerical results verified that the proposed algorithm achieves better energy efficiency than conventional algorithms. Additionally, the proposed framework is able to effectively address the energy puzzle issue. Furthermore, because the proposed algorithm is based on a battery level-aware scheme, the operating time of mobile devices can be prolonged.

ACKNOWLEDGMENTS

This research work was supported by the National Nature Science Foundation of China under Grants No. 61473066 and No. 61601109, the Fundamental Research Funds for Central Universities under Grant No. N152305001, and the Natural Science Foundation of Hebei Province under Grant No. F2021501020.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ORCID

Zhigang Xie https://orcid.org/0000-0002-0992-8900
Xin Song https://orcid.org/0000-0001-6700-8670
Siyang Xu https://orcid.org/0000-0002-3838-2345

REFERENCES

1. S. Fang, L. D. Xu, Y. Zhu, J. Ahati, H. Pei, J. Yan, and Z. Liu, An integrated system for regional environmental monitoring and management based on Internet of Things, IEEE Trans. Ind. Inform. 10 (2014), no. 2, 1596–1605.
2. X. Yang, X. Wang, Y. Wu, L. P. Qian, W. Lu, and H. Zhou, Small-cell assisted secure traffic offloading for narrowband Internet of Thing (NB-IoT) systems, IEEE Internet Things J. 5 (2018), no. 3, 1516–1526.
3. S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing, IEEE Trans. Mob. Comput. 18 (2019), no. 2, 319–333.
4. M. Chen and Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network, IEEE J. Sel. Areas Commun. 36 (2018), no. 3, 587–597.
5. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (2016), no. 5, 637–646.
6. R. Roman, J. Lopez, and M. Mambo, Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges, Future Generation Computer systems, Int. J. Sci. 78 (2018), no. 2, 680–698.
7. Cisco public, Cisco Annual Internet Report (2018-2023), 2020. White Paper. 1–35.
8. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. 19 (2017), no. 4, 2322–2358.
9. X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, Selective offloading in mobile edge computing for the green Internet of Things, IEEE Netw. 32 (2018), no. 1, 54–60.
10. C. Gao, Y. Li, Y. Zhao, and S. Chen, A two-level game theory approach for joint relay selection and resource allocation in network coding assisted D2D communications, IEEE Trans. Mob. Comput. 16 (2017), no. 10, 2697–2711.
11. J. Pei, P. Hong, K. Xue, and D. Li, Resource aware routing for service function chains in SDN and NFV-enabled network, IEEE Trans. Serv. Comput. 1 (2018), 1–1.
12. A. Ksentini, M. Bagaa, and T. Taleb, On using SDN in 5G: The controller placement problem, 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6.
13. Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, Mobility support for fog computing: An SDN approach, IEEE Commun. Mag. 56 (2018), no. 5, 53–59.
14. J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, Delay-optimal computation task scheduling for mobile-edge computing systems, 2016 IEEE International Symposium on Information Theory (ISIT), 2016, pp. 1451–1455.
15. Z. Xie, X. Song, and S. Xu, Peer-to-peer enhanced task scheduling for D2D enabled MEC network, IEEE Access 8 (2020), 138236–138250.
16. G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices, IEEE Trans. Ind. Inform. 14 (2018), no. 10, 4642–4655.
17. T. X. Tran and D. Pompili, Joint task offloading and resource allocation for multi-server mobile-edge computing networks, IEEE Trans. Veh. Technol. 68 (2019), no. 1, 856–868.
18. W. Labidi, M. Sarkiss, and M. Kamoun, Energy-optimal resource scheduling and computation offloading in small cell networks, 2015 22nd International Conference on Telecommunications (ICT), 2015, pp. 313–318.
19. K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks, IEEE Access 4 (2016), 5896–5907.
20. X. Chen, L. Jiao, W. Li, and X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Trans. Netw. 24 (2016), no. 5, 2795–2808.
21. Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, Mobile-edge computing: Partial computation offloading using dynamic voltage scaling, IEEE Trans. Commun. 64 (2016), no. 10, 4268–4282.
22. A. Miettinen and J. Nurminen, Energy efficiency of mobile clients in cloud computing, Hot Cloud 10 (2010), 4.
23. M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, Computation offloading and content caching in wireless blockchain networks with mobile edge computing, IEEE Trans. Veh. Technol. 67 (2018), no. 11, 11008–11021.
24. J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks, IEEE Internet Things J. 5 (2018), no. 4, 2633–2645.
25. L. Munguia, O. Oxberry, and D. Rajan, PIPS-SBB: A parallel distributed-memory branch-and-bound algorithm for stochastic mixed-integer programs, 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2016, pp. 730–739.
26. S. Kouki, M. Jenni, and T. Ladhari, Scalable distributed branch and bound for the permutation flow shop problem, 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2013, pp. 503–508.

AUTHOR BIOGRAPHIES

Zhizhang Xie received his BE degree in communications engineering from Northeastern University at Qinhuangdao, Qinhuangdao, China in 2012 and his MS degree in Software Engineering from Northeastern University, Shenyang, China in 2016. He is currently working toward his PhD degree in Communication and Information Systems at Northeastern University, Shenyang, China. His current research interests include Internet of Things, mobile computing, radio access networks, and green communications.

Xin Song was born in Jilin, China, in 1978. She received her PhD degree in Communication and Information Systems from Northeastern University, China, in 2008. She is now a teacher working in Northeastern University at Qinhuangdao, China. Her research interests are in the area of robust adaptive beamforming and wireless communications.

Jing Cao received her BS degree from Hebei University of Economics and Business, Hebei, China and her MS degree from Yanshan University, Hebei, China. She is an Associate Professor with the Hebei Normal University of Science and Technology in Qinhuangdao, China. She is currently pursuing her PhD degree in Communication and Information Systems at Northeastern University, Shenyang, China. Her research interests include D2D communications, green communications, and wireless resource allocation.
Siyang Xu received his BS degree in Electronics and Information Engineering from University of Science and Technology Liaoning, China, 2016. He is currently a PhD candidate in the Computer Science and Engineering Department, Northeastern University, China. His current research interests include energy harvesting and relay cooperative communications.

How to cite this article: Z. Xie, X. Song, J. Cao, and S. Xu, Energy efficiency task scheduling for battery level-aware mobile edge computing in heterogeneous networks, ETRI Journal 44 (2022), 746–758. https://doi.org/10.4218/etrij.2021-0312