Li *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2022) 11:17
https://doi.org/10.1186/s13677-022-00290-w

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH

Open Access

Joint task offloading and resource allocation in mobile edge computing with energy harvesting

Shichao Li[1,2], Ning Zhang[2], Ruihong Jiang[3*], Zou Zhou[1], Fei Zheng[1] and Guiqin Yang[4]

Abstract

Mobile edge computing (MEC) is considered to be a promising technique to enhance the computation capability and reduce the energy consumption of smart mobile devices (SMDs) in the sixth-generation (6G) networks. With the huge increase of SMDs, many applications of SMDs can be interrupted due to the limited energy supply. Combining MEC and energy harvesting (EH) can help solve this issue, where computation-intensive tasks can be offloaded to edge servers and the SMDs can also be charged during the offloading. In this work, we aim to minimize the total energy consumption subject to the service latency requirement by jointly optimizing the task offloading ratio and resource allocation (including time switching (TS) factor, uplink transmission power of SMDs, downlink transmission power of eNodeB, computation resources of SMDs and MEC server). Compared with the previous studies, the task uplink transmission time, MEC computation time and the computation results downloading time are all considered in this problem. Since the problem is non-convex, we first reformulate it, and then decompose it into two subproblems, i.e., joint uplink and downlink transmission time optimization subproblem (JUDTT-OP) and joint task offloading ratio and TS factor optimization subproblem (JTORTSF-OP). By solving the two subproblems, a joint task offloading and resource allocation with EH (JTORAEH) algorithm is proposed to solve the considered problem. Simulation results show that compared with other benchmark methods, the proposed JTORAEH algorithm can achieve a better performance in terms of the total energy consumption.

Keywords: Sixth-generation (6G) networks, Mobile edge computing (MEC), Downloading time, Energy harvesting (EH), Joint task offloading and resource allocation

Introduction

Artificial intelligence (AI), virtual reality (VR), Internet of things (IoT), and the new generation of wireless communication technology have promoted a new round of technological revolution in the world. Although compared with the fourth-generation (4G) networks, the fifth-generation (5G) networks can provide a higher information transmission rate, for the computation-hungry and delay-sensitive mobile applications, such as autonomous driving and online gaming, 5G still cannot guarantee the quality of service (QoS) of these applications [1, 2]. In order to solve this problem, the sixth-generation (6G) networks emerge as the times require. Mobile edge computing (MEC) is one of the enabling technologies of 6G [3, 4]. By deploying the computation resources at the edge of networks, it can address the issues of insufficient computation capability and large service delay of mobile devices.

Energy efficiency (EE) is an important performance of 6G, which is required to be 10 $\sim$100 times that of 5G. Although MEC can reduce energy consumption by deploying the computation resources at the edge of networks, some applications may still be interrupted due to the limited energy of smart mobile devices (SMDs) batter-

*Correspondence: rhjiang@bjtu.edu.cn
[3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
Full list of author information is available at the end of the article

ies [5, 6]. Using large batteries or recharging the batteries can be used to mitigate this issue. However, considering the size and cost of hardware, the battery capacity is finite, which cannot provide a long-term stable power supply for the SMDs. Meanwhile, it is impossible to recharge the batteries in some special scenarios, such as the large-scale deployment of IoT and outdoor wireless sensor networks (WSN) [7]. To achieve sustainable operation, energy harvesting (EH) allows the SMDs to harvest renewable energy from the environment (e.g., solar radiation, wind, and mechanical energy) or data communications and task processing. EH technology becomes very important for the green communication and durable operation of SMDs [8].

In the literature, a flurry of studies on task offloading with EH in MEC networks has been reported, which can be classified into two categories based on different objectives. The first is to minimize the MEC system power consumption [9–13]. In [9], the authors investigated the problem of power consumption for multiuser MEC system with EH, where an online algorithm was proposed based on the Lyapunov optimization method. In [10], the energy minimization problem is studied for MEC system considering the channel fluctuations and dynamic task arrivals over time, where a well-structured optimal solution is achieved. In [11], the total energy consumption was minimized for MEC system considering the independent variation of the wireless channel conditions and computing tasks, where a mixed-timescale joint computation offloading and wireless resource allocation algorithm is presented. In [12], the access point power consumption was minimized for MEC system with consideration of the doubly near-far effect for the farther SMDs, where a two-phase task offloading method is proposed. In [13], the total energy consumption of access points was minimized for wireless powered multiuser MEC system subject to the users individual computation latency, where an optimal resource allocation scheme was designed. The second is to maximize the EE performance of MEC system [14–18]. In [14], the system minimum EE was maximized for a MEC system, where a cooperative scheme among users was presented. In [15], a partial offloading scheme was proposed to improve the EE for a wireless powered MEC system. In [16], the EE in partial offloading and local computing scenarios for a MEC system was maximized, where a two-phase resource allocation method was proposed. In [17], the EE was maximized in a MEC-based heterogeneous system, where a quantum-behaved particle swarm optimization algorithm was involved. In [18], the EE was also maximized for a wireless powered MEC system with dynamic task arrivals, where the Lagrange duality method was used.

Nevertheless, the computation results downloading transmission time is not considered in previous studies above [9–18]. When the computation outcome with large sizes, the downloading time cannot be ignored, such as augmented reality and multi-media transformation [19]. In addition, when considering the computation results downloading transmission, the SMDs with EH components can harvest energy while receiving information, the EH-aware decision making should be considered. Therefore, the resource allocation schemes should be re-investigated when considering the downloading time and energy harvesting in MEC system, which motivated this work.

The motivations of this paper can be summarized as follows.

- In most previous work [9–18], the downloading transmission time of the computation results was ignored. However, for computation outcome with large sizes, the downloading transmission time is an important part of task offloading delay, such as augmented reality and multi-media transformation, which cannot be ignored for simplify.
- Considering the computation results downloading transmission, the SMDs with EH components can harvest energy while receiving information, i.e., simultaneous wireless information and power transfer (SWIPT) [20]. Therefore, how to make full use of the renewable energy resources to improve system performance is a challenging issue in the downlink SWIPT system.

Therefore, in this paper, we mainly focus on the total energy consumption minimization problem. Since the problem is a non-convex problem, we reformulate the problem firstly, and then decompose it into two subproblems, which are solved by Lagrangian dual method. Based on the results of two subproblems, a joint task offloading and resource allocation with energy harvesting (JTORAEH) algorithm is proposed. Simulation results demonstrate the effectiveness of the proposed JTORAEH algorithm. Our main contributions of the paper are summarized as follows.

- To realize green MEC design for 6G networks, the total energy consumption minimization problem is formulated, which jointly optimizes the task offloading ratio and resource allocation with considering the energy harvesting and computation results downloading time.
- Due to the coupling of optimization variables and the nonconvexity of the formulated problem, we decompose the primal problem into two subproblems, which can be solved by Lagrangian dual method. Based on the results of two subproblems, we propose a JTORAEH algorithm to minimize the total energy consumption of the system.

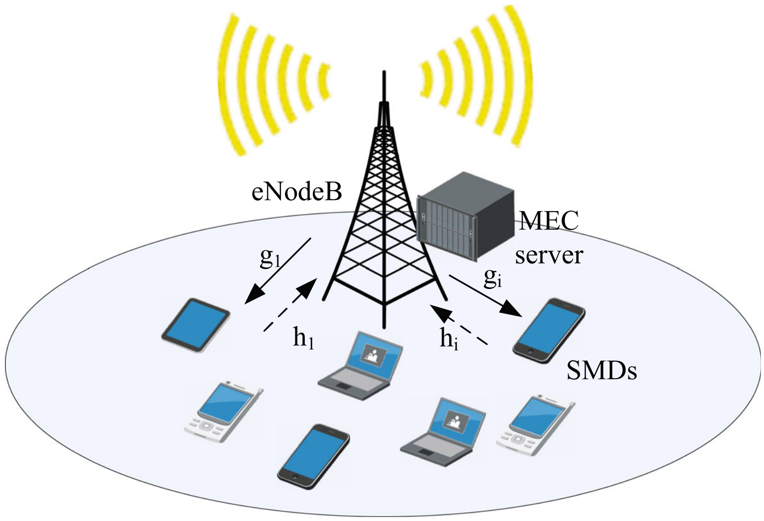

Fig. 1 System Model

- Simulation results are provided, which demonstrate the accuracy and effectiveness of our proposed JTORAEH algorithm.

The rest of this paper is organized as follows. In "System model" section, the system model is described. In "Problem formulation and reformulation" section, the total energy consumption minimization problem is formulated and reformulated. In "Joint task offloading and resource allocation with eH algorithm: jTORAEH" section, the JTORAEH algorithm is proposed to solve the considered problem. Simulation results are provided in "Simulation results" section. Finally, "Conclusions" section concludes this paper.

System model

In this section, we first describe the system model of MEC, including network model, uplink transmission model, computation model, downlink transmission model, and EH model.

Network model

A multiple EH-enabled users MEC system is considered, as shown in Fig. 1, which consists of an eNodeB equipped with a MEC server and I single-antenna SMDs. In order to achieve sustainable operation, each SMD has EH function, which can harvest energy from radio frequency (RF) signals. Besides, in the given time period, each SMD has one computation-intensive task to be executed. Let $\mathcal{I} = \{1, ..., I\}$ denote the set of SMD indices, which are also the set of task indices. Using a 4-tuple $\{c_i, s_i, o_i, t_i^{\max}\}$ to represent the task i, where c_i is the number of CPU-cycle required to accomplish task i. s_i is the input computation file size of task i, o_i is the output computation result size of task i, and $t_i^{\max}$ is the maximal delay requirement of task i. The output results size is proportional to the input data size, i.e., $o_i = \beta_i s_i$, $\beta_i \in (0, 1]$.

The partial task offloading is considered in this work. Therefore, we denote λ_i as the offloading ratio variable of task i, which means that $\lambda_i s_i$ bits of task i are offloaded to the MEC server, and $(1 - \lambda_i)s_i$ bits of task i are executed by SMDs locally. Let T be the total EH and task offloading time, as shown in Fig. 2. In the EH phase with the interval of αT, each SMD harvests energy from RF signals transmitted by the eNodeB, where α is the time switching (TS) factor. In the task offloading phase with the interval of $(1 - \alpha)T$, each SMD offloads partial task to the MEC server or executes partial task locally. Particularly, when the partial task is offloaded to the MEC server, the

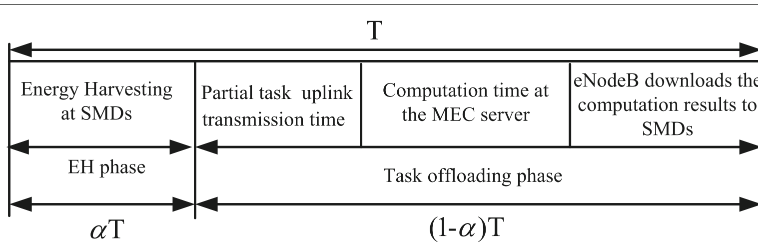

Fig. 2 The process of EH and task offloading

phase is divided into three parts. The first part is used for uplink transmission in which a partial task is offloaded to the eNodeB. The second part is used for computation in which a partial task is executed by the MEC server. The third part is used for downlink transmission in which the eNodeB downloads the computation results to SMDs. The notations used in this paper are summarized in Table 1.

Table 1 Summary of key notations

Notation	Description
I	number of SMDs
$\mathcal{I}$	set of SMDs
c_i	number of CPU-cycle required to accomplish task i
s_i	input computation file size of task i
o_i	output computation result size of task i
$t_i^{\max}$	maximal delay requirement of task i
λ_i	offloading ratio variable of task i
T	total EH and task offloading time
α	time switching factor
b_i	bandwidth between eNodeB and SMD i
h_i	uplink channel gain from SMD i to eNodeB
p_i^u	uplink transmission power of SMD i to eNodeB
r_i^u	uplink transmission rate of task i
t_i^u	uplink transmission time of the partial task i
E_i^u	uplink transmission energy consumption of partial task i
f_i^{loc}	CPU-cycle frequency of SMD i allocated to the task
$F_i^{\max}$	maximal CPU-cycle frequency of SMD i
t_i^{loc}	local computation time
E_i^{loc}	local computation energy consumption of SMD i
κ_i^{loc}	conversion coefficient of SMD i
$F_{\mathrm{mec}}^{\max}$	maximal CPU-cycle frequency of MEC server
f_i^{mec}	CPU-cycle frequency of MEC server allocated to the task i
t_i^{mec}	MEC server computation time of task i
E_i^{mec}	MEC server computation energy consumption of task i
κ^{mec}	conversion coefficient of MEC server
p_i^d	downlink transmission power of eNodeB to SMD i
r_i^d	downlink transmission rate from eNodeB to SMD i
t_i^d	downlink transmission time from eNodeB to SMD i
E_i^d	downlink transmission energy consumption of eNodeB to SMD i
E_i^{har}	total energy harvested by SMD i
η_i	energy conversion efficiency

Uplink transmission model

Let b_i be the bandwidth between eNodeB and SMD i, and the bandwidth is allocated to each SMD orthogonally in this system. Thus, there is no interference among the SMDs. The uplink channel gain from SMD i to eNodeB is denoted as h_i. The uplink transmission power of SMD i to eNodeB is p_i^u. Therefore, the uplink transmission rate of task i can be given by

$$r_i^u = b_i \log\left(1 + \frac{h_i p_i^u}{N_0}\right), \tag{1}$$

where N_0 is the noise power. Since $\lambda_i s_i$ bits of task i are offloaded to the MEC server, the uplink transmission time of the partial task i can be expressed as

$$t_i^u = \frac{\lambda_i s_i}{b_i \log\left(1 + \frac{h_i p_i^u}{N_0}\right)}. \tag{2}$$

Further, the uplink transmission energy consumption of partial task i is given by

$$E_i^u = p_i^u \frac{\lambda_i s_i}{b_i \log\left(1 + \frac{h_i p_i^u}{N_0}\right)}. \tag{3}$$

Computation model

Local computation

Denote $F_i^{\max}$ as the maximal CPU-cycle frequency of SMD i, and f_i^{loc} as the CPU-cycle frequency of SMD i allocated to the task. Therefore, the local computation time can be given by

$$t_i^{\mathrm{loc}} = \frac{(1-\lambda_i)c_i}{f_i^{\mathrm{loc}}}. \tag{4}$$

While, the local computation time, i.e., t_i^{loc}, should satisfy the task delay requirement. That is,

$$t_i^{\mathrm{loc}} \le \min\left(t_i^{\max}, (1-\alpha)T\right). \tag{5}$$

Following with (2) in [21], the local computation energy consumption of SMD i can be given by

$$E_i^{\mathrm{loc}} = \kappa_i^{\mathrm{loc}} \left(f_i^{\mathrm{loc}}\right)^2 (1-\lambda_i)c_i, \tag{6}$$

where κ_i^{loc} is the conversion coefficient of SMD i, which is determined by the SMD i CPU chip architecture. Based on the practical measurement, we set $\kappa_i^{\mathrm{loc}} = 10^{-27}$ in the sequel [22].

MEC Server Computation

Denote $F_{\mathrm{mec}}^{\max}$ as the maximal CPU-cycle frequency of MEC server, and f_i^{mec} as the CPU-cycle frequency of MEC server allocated to the task i. Therefore, the MEC server computation time can be given by

$$t_i^{\mathrm{mec}} = \frac{\lambda_i c_i}{f_i^{\mathrm{mec}}}. \tag{7}$$

Considering the leakage power can take more than 30% of the total power of the modern CPU. We need to consider the leakage power of MEC server. According to [23, 24], the leakage power can be expressed as

$$P_{\mathrm{staic}} = V_{\mathrm{dd}}(I_{\mathrm{leak}} + I_{\mathrm{gate}}), \tag{8}$$

where V_{dd} is bias voltage, I_{leak} is sub-threshold leakage current, I_{gate} is current due to gate tunneling effect.

Following with (6), the MEC server computation energy consumption of task i is expressed as

$$E_i^{\mathrm{mec}} = \kappa^{\mathrm{mec}}(f_i^{\mathrm{mec}})^2\lambda_i c_i + P_{\mathrm{staic}} t_i^{\mathrm{mec}}, \tag{9}$$

where κ^{mec} is the conversion coefficient of MEC server, which is determined by the MEC server CPU chip architecture. Similar to [22], we set $\kappa^{\mathrm{mec}} = 10^{-28}$ in the paper.

Downlink transmission model

After finishing the MEC server computation, the eNodeB downloads the computation results to SMDs. Let g_i be the downlink channel gain from eNodeB to SMD i, and the downlink transmission power of eNodeB to SMD i be p_i^d. Therefore, the downlink transmission rate from eNodeB to SMD i can be given by

$$r_i^d = b_i \log\left(1 + \frac{g_i p_i^d}{N_0}\right), \tag{10}$$

and the downlink transmission time from eNodeB to SMD i can be expressed as

$$t_i^d = \frac{\beta_i \lambda_i s_i}{b_i \log\left(1 + \frac{g_i p_i^d}{N_0}\right)}. \tag{11}$$

Meanwhile, the downlink transmission energy consumption of eNodeB to SMD i is given by

$$E_i^d = p_i^d \frac{\beta_i \lambda_i s_i}{b_i \log\left(1 + \frac{g_i p_i^d}{N_0}\right)}. \tag{12}$$

Moreover, when the partial task i is offloaded to the MEC server, the task offloading delay should satisfy the task delay requirement

$$t_i^u + t_i^{\mathrm{mec}} + t_i^d \le \min(t_i^{\max}, (1-\alpha)T). \tag{13}$$

EH model

The energy harvested by SMD i includes two parts. In the first part, SMD i harvests energy from RF signals transmitted by eNodeB over αT, which is given by

$$E_{i,1}^{\mathrm{har}} = \alpha T \eta_i g_i p_i^d, \tag{14}$$

where $\eta_i \in (0,1)$ is the energy conversion efficiency [25]. The second part is that SMD i collects energy from the eNodeB downlink transmission of the computation results with SWIPT. That is,

$$E_{i,2}^{\mathrm{har}} = \eta_i t_i^d g_i p_i^d, \tag{15}$$

Therefore, the total energy harvested by SMD i can be given by

$$E_i^{\mathrm{har}} = E_{i,1}^{\mathrm{har}} + E_{i,2}^{\mathrm{har}}. \tag{16}$$

As a result, the total energy consumption can be given by

$$E_{tot} = \sum_{i=1}^{I}\left(E_i^{\mathrm{loc}} + E_i^u + E_i^{\mathrm{mec}} + E_i^d\right). \tag{17}$$

Problem formulation and reformulation

In this section, a total energy consumption minimization problem is formulated firstly. And then, due to the nonconvexity of the primal problem, we reformulate the problem.

Problem formulation

Based on the system model, the total energy consumption minimization problem can be formulated as

$$
\begin{aligned}
(\mathbf{P1}) \quad & \min_{\alpha,\boldsymbol{\lambda},\boldsymbol{f}^{\mathrm{loc}},\boldsymbol{f}^{\mathrm{mec}},\boldsymbol{p}^u,\boldsymbol{p}^d} \quad E_{tot} && \text{(18a)}\\
\text{s.t. } \quad & t_i^{\mathrm{loc}} \le \min\left(t_i^{\max}, (1-\alpha)T\right), \forall i \in \mathcal{I}, && \text{(18b)}\\
& t_i^u + t_i^{\mathrm{mec}} + t_i^d \le \min\left(t_i^{\max}, (1-\alpha)T\right), \forall i \in \mathcal{I}, && \text{(18c)}\\
& 0 \le f_i^{\mathrm{loc}} \le F_i^{\max}, \quad \forall i \in \mathcal{I}, && \text{(18d)}\\
& \sum_{i=1}^{I} f_i^{\mathrm{mec}} \le F_{\mathrm{mec}}^{\max}, && \text{(18e)}\\
& p_i^u \le P_i^{\max}, \quad \forall i \in \mathcal{I}, && \text{(18f)}\\
& \sum_{i=1}^{I} p_i^d \le P_e^{\max}, && \text{(18g)}\\
& E_i^{\mathrm{loc}} + E_i^u \le E_i^{\mathrm{har}} + E_i^o, \quad \forall i \in \mathcal{I}, && \text{(18h)}\\
& 0 < \alpha < 1, && \text{(18i)}\\
& 0 \le \lambda_i \le 1, \forall i \in \mathcal{I}, && \text{(18j)}
\end{aligned}
$$

where $\boldsymbol{\lambda} = \{\lambda_i, i \in \mathcal{I}\}$, $\boldsymbol{f}^{\mathrm{loc}} = \{f_i^{\mathrm{loc}}, i \in \mathcal{I}\}$, $\boldsymbol{f}^{\mathrm{mec}} = \{f_i^{\mathrm{mec}}, i \in \mathcal{I}\}$, $\boldsymbol{p}^u = \{p_i^u, i \in \mathcal{I}\}$, $\boldsymbol{p}^d = \{p_i^d, i \in \mathcal{I}\}$ are the vectors of task offloading ratio, SMDs CPU-cycle frequency allocation, MEC server CPU-cycle frequency allocation, SMDs uplink power allocation, and eNodeB downlink power allocation, respectively. $P_i^{\max}$ is the maximal transmission power of SMD i. $P_e^{\max}$ is the maximal downlink transmission power of eNodeB. E_i^o is the initial energy of SMD i. Constraints (18b) and (18c) are the task offloading delay requirement of local computing and MEC computing, respectively. Constraints (18d) and (18e) are the SMD and MEC computation resource constraints, respectively. Constraints (18f) and (18g) are the transmission power constraints of SMD and eNodeB, respectively. Constraint (18h) means that the total energy consumption of SMDs should be no more than the harvested energy. Due to the product relationship between λ_i and p_i^u as well as p_i^d, problem (**P1**) is non-convex, which is difficult to solve.

Problem reformulation

In order to solve problem (**P1**), we need to reformulate the problem. For (6), given α and $\boldsymbol{\lambda}$, the local computation energy consumption of SMD is monotonously increasing with f_i^{loc}. This means that the smaller f_i^{loc} is, the less local computation energy consumption is. For constraint (18b), when the optimal solution is obtained, $t_i^{\text{loc}} = \min(t_i^{\max}, (1-\alpha)T)$ should be held. Therefore, the optimal local CPU-cycle frequency of SMD i can be given by

$$f_i^{\text{loc}*} = \frac{(1-\lambda_i)c_i}{\min(t_i^{\max}, (1-\alpha)T)}. \tag{19}$$

Similarly, for (9), given α and $\boldsymbol{\lambda}$, the MEC server computation energy consumption of SMD is monotonously increasing with f_i^{mec}. For constraint (18c), when the optimal solution is obtained, $t_i^{\text{loc}} = \min(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d)$ should be held. Therefore, the optimal MEC server CPU-cycle frequency to SMD i is given by

$$f_i^{\text{mec}*} = \frac{\lambda_i c_i}{\min(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d)}. \tag{20}$$

Then, constraint (18e) can be transformed as

$$\sum_{i=1}^{I} \frac{\lambda_i c_i}{\min(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d)} \le F_{\text{mec}}^{\max}. \tag{21}$$

Following (3) and (12), let $f(x) = x\frac{1}{\log_2(1+x)}$, where $f(x)$ is monotonously increasing with $x \ge 0$. That is, given $\boldsymbol{\lambda}$, the smaller $\boldsymbol{p}^u$ and $\boldsymbol{p}^d$ are, the less uplink and downlink transmission energy consumption are. With (2) and (11), the optimal uplink transmission power of SMD i, and downlink transmission power to SMD i are given by

$$p_i^{u*} = \frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right), \tag{22}$$

and

$$p_i^{d*} = \frac{N_0}{g_i}\left(2^{\frac{\beta_i \lambda_i s_i}{b_i t_i^d}} - 1\right), \tag{23}$$

respectively.

Therefore, the problem (**P1**) can be reformulated as

$$(\mathbf{P2}) \quad \min_{\alpha, \boldsymbol{\lambda}, \boldsymbol{t}^u, \boldsymbol{t}^d} \quad \varphi(\alpha, \lambda_i, t_i^u, t_i^d) \tag{24a}$$

$$\text{s.t.} \quad \frac{(1-\lambda_i)c_i}{\min(t_i^{\max}, (1-\alpha)T)} \le F_i^{\max} \quad \forall i \in \mathcal{I}, \tag{24b}$$

$$\sum_{i=1}^{I} \theta_i(\alpha, \lambda_i, t_i^u, t_i^d) \le F_{\text{mec}}^{\max}, \tag{24c}$$

$$\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right) \le P_i^{\max}, \quad \forall i \in \mathcal{I}, \tag{24d}$$

$$\sum_{i=1}^{I} \frac{N_0}{g_i}\left(2^{\frac{\beta_i \lambda_i s_i}{b_i t_i^d}} - 1\right) \le P_e^{\max}, \tag{24e}$$

$$\begin{aligned}&\kappa_i^{\text{loc}} \frac{(1-\lambda_i)^3 c_i^3}{(\min(t_i^{\max}, (1-\alpha)T))^2} + \frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right) t_i^u \\ &\le \eta_i N_0 \left(2^{\frac{\beta_i \lambda_i s_i}{b_i t_i^d}} - 1\right)(\alpha T + t_i^d) + E_i^o, \quad \forall i \in \mathcal{I},\end{aligned} \tag{24f}$$

$$0 < \alpha < 1, \tag{24g}$$

$$0 \le \lambda_i \le 1, \forall i \in \mathcal{I}, \tag{24h}$$

where $\boldsymbol{t}^u = \{t_i^u, i \in \mathcal{I}\}$ and $\boldsymbol{t}^d = \{t_i^d, i \in \mathcal{I}\}$ are the vectors of uplink and downlink transmission time, respectively,

$$\theta_i\left(\alpha, \lambda_i, t_i^u, t_i^d\right) = \frac{\lambda_i c_i}{\min(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d)},$$

and

$$\begin{aligned}\varphi(\alpha, \lambda_i, t_i^u, t_i^d) =& \kappa_i^{\text{loc}} \frac{(1-\lambda_i)^3 c_i^3}{(\min(t_i^{\max}, (1-\alpha)T))^2} \\ &+ \frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right) t_i^u + \kappa^{\text{mec}} \theta_i^2 \lambda_i c_i + P_{\text{staic}} \frac{\lambda_i c_i}{\theta_i} \\ &+ \frac{N_0}{g_i}\left(2^{\frac{\beta_i \lambda_i s_i}{b_i t_i^d}} - 1\right) t_i^d.\end{aligned}$$

However, since the optimal variables are coupled, problem (**P2**) is still non-convex, which is difficult to solve.

Joint task offloading and resource allocation with eH algorithm: jTORAEH

In this section, we first decompose the problem (**P2**) into two subproblems: joint uplink and downlink transmission time optimization subproblem (JUDT-OP) and joint task offloading ratio and TS factor optimization subproblem (JTORTSF-OP). Then, the Lagrangian dual method is used to solve the two subproblems. By doing so, an iterative algorithm is proposed to the primal problem.

Joint uplink and downlink transmission time optimization subproblem: jUDTT-OP

By fixing α and $\boldsymbol{\lambda}$, the JUDTT-OP can be given by

$$(\mathbf{P3})\ \min_{\boldsymbol{t}^u,\boldsymbol{t}^d}\ \varphi\left(t_i^u,t_i^d\right) \tag{25a}$$

$$\text{s.t.}\ \sum_{i=1}^{I}\theta_i\left(t_i^u,t_i^d\right)\le F_{\text{mec}}^{\max}, \tag{25b}$$

$$\sum_{i=1}^{I}\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\le P_e^{\max}, \tag{25c}$$

$$\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)\le P_i^{\max},\ \forall i\in\mathcal{I}, \tag{25d}$$

$$\kappa_i^{\text{loc}}\frac{(1-\lambda_i)^3c_i^3}{\left(\min\left(t_i^{\max},(1-\alpha)T\right)\right)^2}+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u \le \eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\left(\alpha T+t_i^d\right)+E_i^o,\ \forall i\in\mathcal{I}. \tag{25e}$$

Due to the objective function and constraints in problem (**P3**) are convex, problem (**P3**) is a convex problem. Therefore, the Lagrangian dual method can be used to solve the JUDTT-OP [26].

Then, the Lagrangian function of problem (**P3**) is expressed as

$$\begin{aligned}\mathcal{L}_1=&\mathcal{L}\left(\boldsymbol{t}^u,\boldsymbol{t}^d,\mu_1,\mu_2,\nu_i,\gamma_i\right)\\=&\varphi\left(t_i^u,t_i^d\right)+\mu_1\left(\sum_{i=1}^{I}\theta_i\left(t_i^u,t_i^d\right)-F_{\text{mec}}^{\max}\right)\\&+\mu_2\left(\sum_{i=1}^{I}\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)-P_e^{\max}\right)\\&+\nu_i\left(\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)-P_i^{\max}\right)\\&+\gamma_i\Bigg(\kappa_i^{\text{loc}}\frac{(1-\lambda_i)^3c_i^3}{\left(\min\left(t_i^{\max},(1-\alpha)T\right)\right)^2}\\&+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u\\&-\eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\left(\alpha T+t_i^d\right)-E_i^o\Bigg),\end{aligned} \tag{26}$$

where μ_1, μ_2, ν_i, and γ_i are the Lagrange multipliers.

Since the Lagrange function, i.e., $\mathcal{L}_1$, is differentiable, the gradients of the Lagrange multipliers can be respectively given by

$$\frac{\partial\mathcal{L}_1}{\partial\mu_1}=\sum_{i=1}^{I}\theta_i\left(t_i^u,t_i^d\right)-F_{\text{mec}}^{\max}, \tag{27}$$

$$\frac{\partial\mathcal{L}_1}{\partial\mu_2}=\sum_{i=1}^{I}\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)-P_e^{\max}, \tag{28}$$

$$\frac{\partial\mathcal{L}_1}{\partial\nu_i}=\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)-P_i^{\max}, \tag{29}$$

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial\gamma_i}=&\kappa_i^{\text{loc}}\frac{(1-\lambda_i)^3c_i^3}{\left(\min\left(t_i^{\max},(1-\alpha)T\right)\right)^2}+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u\\&-\eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\left(\alpha T+t_i^d\right)-E_i^o,\end{aligned} \tag{30}$$

Then, the Lagrange multipliers can be obtained by utilizing the gradient method. That is,

$$\mu_1(t+1)=\left[\mu_1(t)+\tau_1\frac{\partial\mathcal{L}}{\partial\mu_1}\right]^+, \tag{31}$$

$$\mu_2(t+1)=\left[\mu_2(t)+\tau_2\frac{\partial\mathcal{L}}{\partial\mu_2}\right]^+, \tag{32}$$

$$\nu_i(t+1)=\left[\nu_i(t)+\tau_3\frac{\partial\mathcal{L}}{\partial\nu_i}\right]^+, \tag{33}$$

$$\gamma_i(t+1)=\left[\gamma_i(t)+\tau_4\frac{\partial\mathcal{L}}{\partial\gamma_i}\right]^+, \tag{34}$$

where τ_1, τ_2, τ_3, and τ_4 are the iteration steps, t is the iteration number, and $[\cdot]^+$ means $\max(0,\cdot)$.

Taking the derivative of $\mathcal{L}_1$ w.r.t t_i^u, t_i^d to be zero respectively, we have

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial t_i^u}=&(1+\gamma_i)\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)\\&-(1+\gamma_i)\frac{N_0}{h_i}2^{\frac{\lambda_i s_i}{b_i t_i^u}}\frac{\lambda_i s_i}{b_i t_i^u}\ln 2\\&+\left(\mu_1+2\kappa^{\text{mec}}\theta_i\lambda_i c_i-P_{\text{staic}}\frac{\lambda_i c_i}{\theta_i^2}\right)\\&\frac{\lambda_i c_i}{\left(\min\left(t_i^{\max}-t_i^u-t_i^d,(1-\alpha)T-t_i^u-t_i^d\right)\right)^2}\\&-\nu_i\frac{N_0}{h_i}2^{\frac{\lambda_i s_i}{b_i t_i^u}}\frac{\lambda_i s_i}{b_i\left(t_i^u\right)^2}\ln 2\\&=0,\end{aligned} \tag{35}$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}_1}{\partial t_i^d} =&(1-\gamma_i\eta_i g_i)\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\\
&-(1-\gamma_i\eta_i g_i)\frac{N_0}{g_i}2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}\frac{\beta_i\lambda_i s_i}{b_i t_i^d}\ln 2\\
&+\left(\mu_1+2\kappa^{\mathrm{mec}}\theta_i\lambda_i c_i-P_{\mathrm{staic}}\frac{\lambda_i c_i}{\theta_i^2}\right)\\
&\frac{\lambda_i c_i}{\left(\min\left(t_i^{\max}-t_i^u-t_i^d,(1-\alpha)T-t_i^u-t_i^d\right)\right)^2}\\
&-(\mu_2-\gamma_i\eta_i g_i\alpha T)\frac{N_0}{g_i}2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}\frac{\beta_i\lambda_i s_i}{b_i\left(t_i^d\right)^2}\ln 2\\
&=0.
\end{aligned}\tag{36}
$$

Thus, t_i^u and t_i^d can be obtained by (35)) and (36), respectively.

Joint task offloading ratio and tS factor optimization subproblem: jTORTSF-OP

By fixing $\boldsymbol{t}^u$ and $\boldsymbol{t}^d$, the JTORTSF-OP is given by

$$
\begin{aligned}
(\mathbf{P4})\ &\min_{\alpha,\boldsymbol{\lambda}}\ \ \varphi(\alpha,\lambda_i) &&(37a)\\
\text{s.t.}\ &\frac{(1-\lambda_i)c_i}{\min\left(t_i^{\max},(1-\alpha)T\right)}\le F_i^{\max}\ \ \forall i\in\mathcal{I}, &&(37b)\\
&\sum_{i=1}^{I}\theta_i(\alpha,\lambda_i)\le F_{\mathrm{mec}}^{\max}, &&(37c)\\
&\sum_{i=1}^{I}\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\le P_e^{\max}, &&(37d)\\
&\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)\le P_i^{\max},\ \ \forall i\in\mathcal{I}, &&(37e)\\
&\kappa_i^{\mathrm{loc}}\frac{(1-\lambda_i)^3c_i^3}{\left(\min\left(t_i^{\max},(1-\alpha)T\right)\right)^2}+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u\\
&\le\eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\left(\alpha T+t_i^d\right)+E_i^o,\ \ \forall i\in\mathcal{I}, &&(37f)\\
&0<\alpha<1, &&(37g)\\
&0\le\lambda_i\le 1,\forall i\in\mathcal{I}. &&(37h)
\end{aligned}
$$

However, since the optimal variables are coupled, problem (**P4**) is a non-convex problem. In order to solve this problem, we first fix α to calculate $\boldsymbol{\lambda}$. And then, we fix $\boldsymbol{\lambda}$ to calculate α, and repeat this process until convergence.

Optimization of task offloading ratio

When the TS factor α is fixed, the task offloading ratio problem is formulated as

$$
\begin{aligned}
(\mathbf{P5})\ &\min_{\boldsymbol{\lambda}}\ \ \varphi(\lambda_i) &&(38a)\\
\text{s.t.}\ &\frac{(1-\lambda_i)c_i}{\min(t_i^{\max},(1-\alpha)T)}\le F_i^{\max},\ \ \forall i\in\mathcal{I}, &&(38b)\\
&\sum_{i=1}^{I}\theta_i(\lambda_i)\le F_{mec}^{\max}, &&(38c)\\
&\sum_{i=1}^{I}\frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)\le P_e^{\max}, &&(38d)\\
&\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)\le P_i^{\max},\ \ \forall i\in\mathcal{I}, &&(38e)\\
&\kappa_i^{\mathrm{loc}}\frac{(1-\lambda_i)^3c_i^3}{(\min(t_i^{\max},(1-\alpha)T))^2}+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u\\
&\le\eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)(\alpha T+t_i^d)+E_i^o,\ \ \forall i\in\mathcal{I}, &&(38f)\\
&0\le\lambda_i\le 1,\forall i\in\mathcal{I}. &&(38g)
\end{aligned}
$$

Problem (**P5**) is a convex problem, the Lagrangian dual method is used to solve it. Then, the Lagrangian function of problem (**P5**) is given by

$$
\begin{aligned}
\mathcal{L}_2 =&\mathcal{L}(\boldsymbol{\lambda},\varepsilon_i,\mu_1,\mu_2,\nu_i,\gamma_i,\xi_i)\\
=&\varphi(\lambda_i)+\varepsilon_i\left(\frac{(1-\lambda_i)c_i}{\min(t_i^{\max},(1-\alpha)T)}-F_i^{\max}\right)\\
&+\mu_1\left(\sum_{i=1}^{I}\theta_i(\lambda_i)-F_{\mathrm{mec}}^{\max}\right)\\
&+\mu_2\left(\sum_{i=1}^{I}\frac{N_0}{g_i}(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1)-P_e^{\max}\right)\\
&+\nu_i\left(\frac{N_0}{h_i}(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1)-P_i^{\max}\right)\\
&+\gamma_i(\kappa_i^{\mathrm{loc}}\frac{(1-\lambda_i)^3c_i^3}{(\min(t_i^{\max},(1-\alpha)T))^2}\\
&+\frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}}-1\right)t_i^u\\
&-\eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}}-1\right)(\alpha T+t_i^d)-E_i^o)\\
&+\xi_i(\lambda_i-1),
\end{aligned}\tag{39}
$$

where ε_i, μ_1, μ_2, ν_i, γ_i, and ξ_i are the Lagrange multipliers. Since the Lagrange function is differentiable, the gradients of the Lagrange multipliers can be respectively given by

$$\frac{\partial \mathcal{L}_2}{\partial \varepsilon_i} = \frac{(1-\lambda_i)c_i}{\min(t_i^{\max},(1-\alpha)T)} - F_i^{\max}, \quad (40)$$

$$\frac{\partial \mathcal{L}_2}{\partial \mu_1} = \sum_{i=1}^{I} \theta_i(\lambda_i) - F_{\text{mec}}^{\max}, \quad (41)$$

$$\frac{\partial \mathcal{L}_2}{\partial \mu_2} = \sum_{i=1}^{I} \frac{N_0}{g_i}\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}} - 1\right) - P_e^{\max}, \quad (42)$$

$$\frac{\partial \mathcal{L}_2}{\partial \nu_i} = \frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right) - P_i^{\max}, \quad (43)$$

$$\begin{aligned}\frac{\partial \mathcal{L}_2}{\partial \gamma_i} =& \kappa_i^{\text{loc}} \frac{(1-\lambda_i)^3 c_i^3}{(\min(t_i^{\max},(1-\alpha)T))^2} + \frac{N_0}{h_i}\left(2^{\frac{\lambda_i s_i}{b_i t_i^u}} - 1\right) t_i^u \\ & - \eta_i N_0\left(2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}} - 1\right)(\alpha T + t_i^d) - E_i^o, \end{aligned} \quad (44)$$

$$\frac{\partial \mathcal{L}_2}{\partial \xi_i} = \lambda_i - 1. \quad (45)$$

The Lagrange multipliers can be obtained by utilizing the gradient method, i.e.,

$$\varepsilon_i(t+1) = \left[\varepsilon_i(t) + \rho_1 \frac{\partial \mathcal{L}}{\partial \varepsilon_i}\right]^+, \quad (46)$$

$$\mu_1(t+1) = \left[\mu_1(t) + \rho_2 \frac{\partial \mathcal{L}}{\partial \mu_1}\right]^+, \quad (47)$$

$$\mu_2(t+1) = \left[\mu_2(t) + \rho_3 \frac{\partial \mathcal{L}}{\partial \mu_2}\right]^+, \quad (48)$$

$$\nu_i(t+1) = \left[\nu_i(t) + \rho_4 \frac{\partial \mathcal{L}}{\partial \nu_i}\right]^+, \quad (49)$$

$$\gamma_i(t+1) = \left[\gamma_i(t) + \rho_5 \frac{\partial \mathcal{L}}{\partial \gamma_i}\right]^+, \quad (50)$$

$$\xi_i(t+1) = \left[\xi_i(t) + \rho_6 \frac{\partial \mathcal{L}}{\partial \xi_i}\right]^+, \quad (51)$$

where ρ_1, ρ_2, ρ_3, ρ_4, ρ_5 and ρ_6 are the iteration steps, and t is the iteration number.

Taking the derivative of $\mathcal{L}_2$ w.r.t λ_i to be zero, we have

$$\begin{aligned}\frac{\partial \mathcal{L}_2}{\partial \lambda_i} =& (1+\gamma_i)\frac{-\kappa_i^{\text{loc}} 3(1-\lambda_i)^2 c_i^3}{\left(\min\left(t_i^{\max},(1-\alpha)T\right)\right)^2} \\ & + \frac{N_0}{h_i}\frac{s_i}{b_i t_i^u}(t_i^u + \gamma_i t_i^u + \nu_i) 2^{\frac{\lambda_i s_i}{b_i t_i^u}} \ln 2 \\ & - \varepsilon_i \frac{c_i}{\min(t_i^{\max},(1-\alpha)T)} \\ & + \kappa^{\text{mec}} \frac{3\lambda_i^2 c_i^3}{\left(\min\left(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d\right)\right)^2} \\ & + \mu_1 \frac{c_i}{\min(t_i^{\max} - t_i^u - t_i^d, (1-\alpha)T - t_i^u - t_i^d)} \\ & + N_0 \frac{\beta_i s_i}{b_i t_i^d} 2^{\frac{\beta_i\lambda_i s_i}{b_i t_i^d}} \left(\frac{t_i^d + \mu_2}{g_i} - \gamma_i\eta_i(\alpha T + t_i^d)\right)\ln 2 \\ & + \xi_i = 0. \end{aligned} \quad (52)$$

Therefore, λ_i can be achieved by (52).

Optimization of tS factor

The TS factor is not only related to EH of SMDs, but related to the task execution time. In (14), it is observed that the larger α is, SMDs can harvest more energy. Meanwhile, less time is available for task execution. By fixing $\boldsymbol{\lambda}$, we set the initial α as

$$\alpha = 1 - \frac{\min_{i\in\mathcal{I}}\{t_i^{\max}\}}{T}. \quad (53)$$

By substituting (53) into problem (**P1**), after several iterations, such as M iterations, some task offloading delay requirements cannot be satisfied due to the time reserved is not enough, α can be updated by

$$\alpha = \alpha - \Delta, \quad (54)$$

where $\Delta > 0$. Therefore, we can use the offline experiments to determine the iteration numbers to guarantee each task offloading delay.

JTORAEH algorithm

To solve the primal problem, a JTORAEH algorithm is proposed, as shown in Algorithm 1.

Simulation results

In this section, some simulation results are provided to discuss for the system performance with the proposed JTORAEH algorithm. Unless otherwise stated, the number of eNodeB and SMDs is set to be 1 and 10, respectively. SMDs are uniformly distributed over a cell. The bandwidth allocated to each SMD is 200 KHz. The noise power is 10^{-10} W. The wireless channel h_i and g_i are respectively obtained by $h_i = \zeta d_i^{-3}\bar{h}_i$ and $g_i = \zeta d_i^{-3}\bar{g}_i$, where $\bar{h}_i \sim \mathcal{CN}(\mathbf{0},\boldsymbol{I})$, $\bar{g}_i \sim \mathcal{CN}(\mathbf{0},\boldsymbol{I})$, d_i is the distance between the eNodeB and the SMD i, and $\zeta = 6.25\times10^{-4}$ is the channel power gain at a reference distance of one meter [27]. The

Algorithm 1 The proposed JTORAEH algorithm

Initialize the uplink transmission time $\boldsymbol{t}^u$, the downlink transmission time $\boldsymbol{t}^d$, the task offloading ratio $\boldsymbol{\lambda}$, and the TS factor α.
repeat
  /*JUDTT-OP*/
  Calculate the uplink transmission time $\boldsymbol{t}^u$, and the downlink transmission time $\boldsymbol{t}^d$ by (35) and (36), respectively.
  Update the Lagrange multipliers of (26) by (31), (32), and (34), respectively.
  /*JTORTSF-OP*/
  Calculate the task offloading ratio $\boldsymbol{\lambda}$ by (52).
  Update the Lagrange multipliers of (39) by (46), (47), (48), (50), and (51), respectively.
  The eNodeB determines whether to update the TS factor α every M iterations. If some task offloading delay requirement cannot be satisfied, α can be updated by (54); Otherwise, α is unchanged.
until Algorithm stopping criterion and convergence.

maximal transmission power of each SMD is 0.1 W, and the maximal downlink transmission power of eNodeB is 1 W. The maximal CPU-cycle frequency of the SMDs and MEC server are set to be 50 MHz and 1 GHz, respectively. The conversion coefficients of SMD and MEC server are 10^{-27} and 10^{-28}, respectively [22]. The initial energy of SMD $E_0 = 0.001$ J. The time period of a complete EH and task offloading process $T = 3$ s. The energy conversion efficiency $\eta = 0.8107$ [25]. The data size of the tasks and the number of CPU-cycle requirements that follow Gaussian distributions are $s_i \sim \mathcal{N}(1, 0.1)$ and $c_i \sim \mathcal{N}(100, 10)$, respectively. The data size is measured in MB and the number of CPU-cycle is measured in Megacycles, respectively. The output data size is 10% of the input data size. Each task offloading delay tolerance is 2.2 s.

Convergence analysis of the proposed jTORAEH algorithm

Figure 3 shows the convergence of the proposed JTORAEH algorithm with different M. M can be considered as updating the value of TS factor α every M iterations according to (54). We set $\Delta = \frac{\min_{i\in\mathcal{I}}\{t_i^{\max}\}}{2T}$, $M = 2, 4$, respectively. It is seen that the proposed JTORAEH algorithm can achieve convergence within about 16 iterations. Besides, the proposed JTORAEH algorithm nearly has the same convergence speed no matter what M takes. Therefore, we take $M = 4$ in the following simulation.

In order to verify the effectiveness of the proposed JTORAEH algorithm, the following three algorithms are also simulated in our work for comparison.

- Local computing (LC) algorithm: In this algorithm, all the tasks are only executed by SMDs locally, where $\lambda_i = 0, \forall i \in \mathcal{I}$.
- MEC server computing (MC) algorithm: In this algorithm, all the tasks are offloaded to the MEC server, and only executed by the MEC server, where $\lambda_i = 1, \forall i \in \mathcal{I}$.
- Based on Hungarian and graph coloring (BHGC) algorithm: In this algorithm, the task offloading ratio, uplink power and computation resource allocation can be obtained by utilizing the same algorithm in

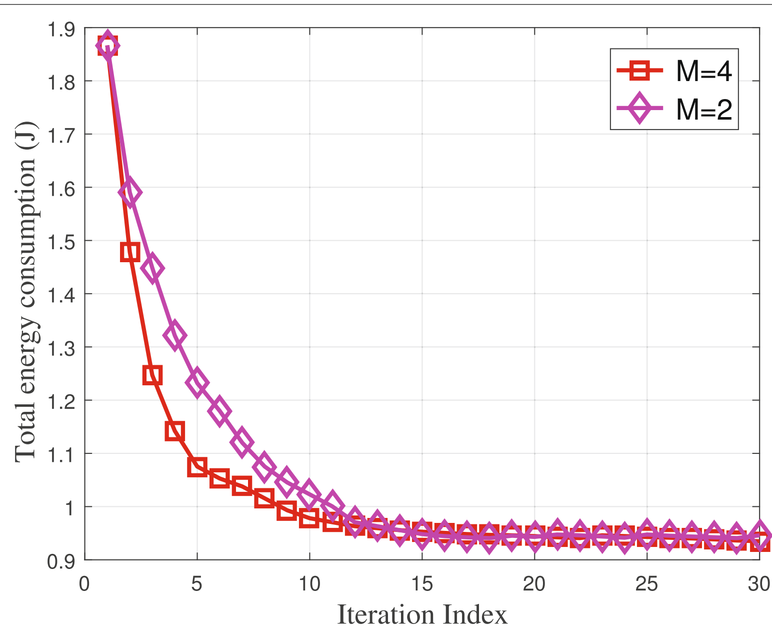

Fig. 3 Total energy consumption versus iterations

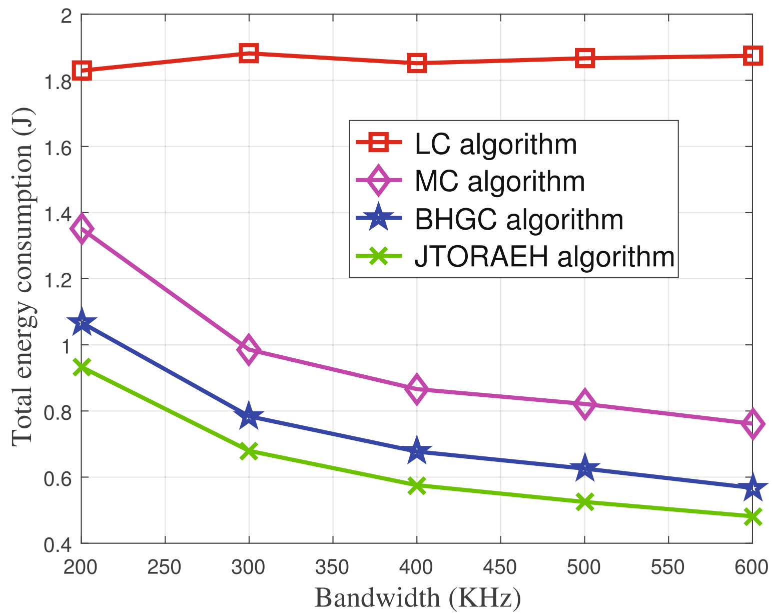

Fig. 4 Total energy consumption versus bandwidth

[28]. To obtain the TS factor α, the same algorithm is used as this work. For the downlink power allocation, the power is allocated averagely for each task.

Effect of the bandwidth on the energy consumption

Figure 4 plots the total energy consumption versus bandwidth. It can be seen that for the LC algorithm, the total energy consumption remains unchanged with the increment of the bandwidth. Because all the tasks are executed by the SMDs locally, the total energy consumption is independent of bandwidth. For the other three algorithms, the total energy consumption decreases as the bandwidth increases. Because higher bandwidth leads to lower transmission power. Compared with the BHGC algorithm, when the bandwidth grows from 200 KHz to 600 KHz, the total energy consumption of JTORAEH

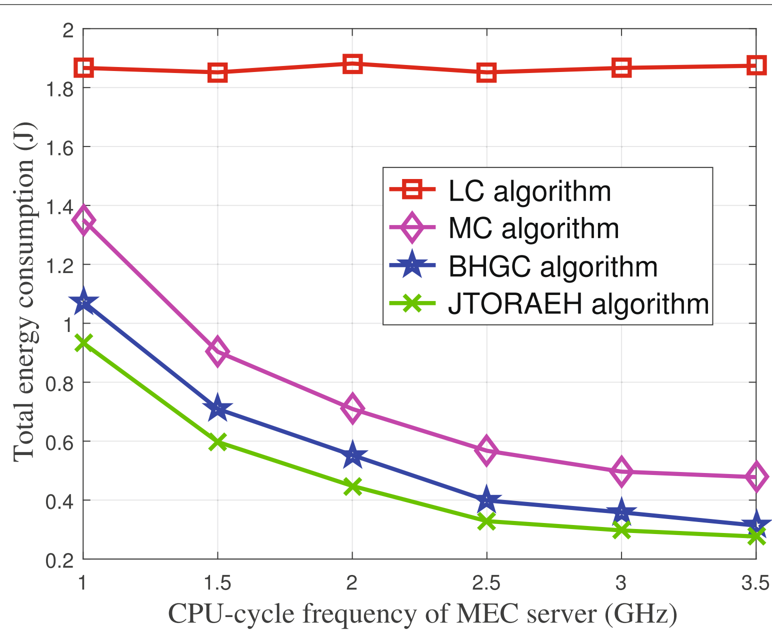

Fig. 5 Total energy consumption versus CPU-cycle frequency of MEC server

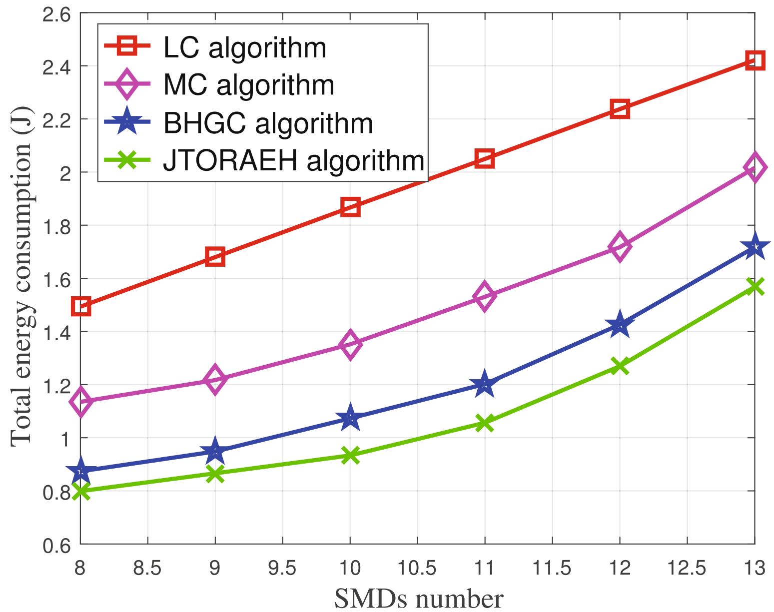

Fig. 6 Total energy consumption versus SMDs number

algorithm can achieve the system performance gain about 14.43%.

Effect of the cPU-cycle frequency of mEC server on the energy consumption

Figure 5 compares the total energy consumption versus CPU-cycle frequency of MEC server. For the LC algorithm, because the total energy consumption is independent of CPU-cycle frequency of MEC server, the total energy consumption remains unchanged with the increase of the CPU-cycle frequency of MEC server. For the other three algorithms, with the increasing of CPU-cycle frequency of MEC server, each task can be allocated more computation resources, and the task computation time can be reduced. Meanwhile, SMDs have more time to transmit tasks to the MEC server, therefore, the uplink transmission energy consumption is reduced. Compared with the BHGC algorithm, when the CPU-cycle frequency

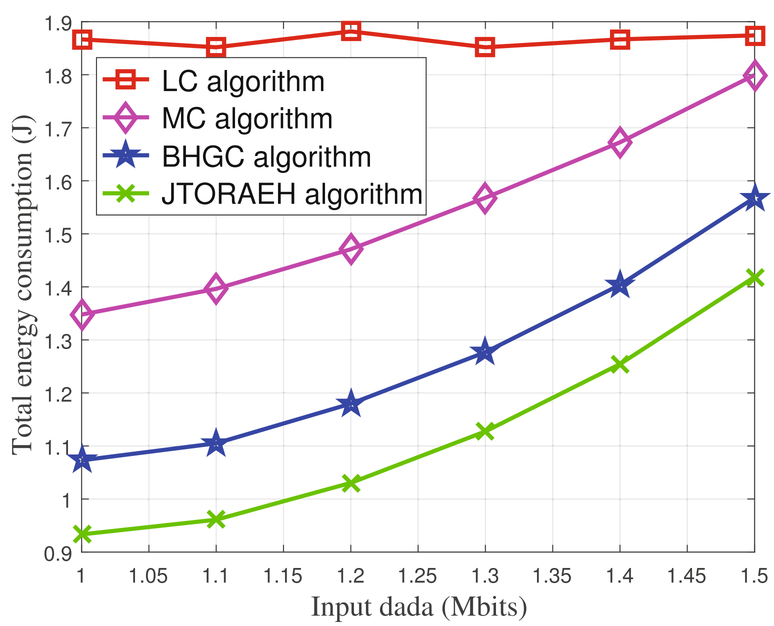

Fig. 7 Total energy consumption versus input computation file size of task

of MEC server grows from 1 GHz to 3.5 GHz, the total energy consumption of JTORAEH algorithm can achieve the system performance gain about 15.73%.

Effect of the sMDs number on the energy consumption

Figure 6 shows the total energy consumption versus SMDs number. It can be observed that the total energy consumption increases with respect to the number of SMDs. Because the resources are limited, when the number of SMDs increases, each SMD will allocate fewer resources, the total energy consumption increases. Meanwhile, the more number of SMDs increases, the faster the total energy consumption of the MC algorithm, BHGC algorithm and JTORAEH algorithm increases. For the LC algorithm, since the energy consumption is only related to the CPU-cycle frequency of SMDs locally, the energy consumption of LC algorithm increases linearly with the number of SMDs increases.

Effect of the input computation file size of task on the energy consumption

Figure 7 plots the total energy consumption versus input computation file size of task. For the LC algorithm, because the tasks are only executed by SMDs locally, and the total energy consumption is independent of the input computation file size of task, the total energy consumption remains unchanged with the increase of the input computation file size of task. For the other three algorithms, with the increment of input computation file size of task, the SMDs need to consume more energy to transmit tasks. Therefore, the total energy consumption increases.

Conclusions

This paper studied the total energy consumption minimization problem for EH-enabled MEC networks by jointly optimizing the task offloading ratio and resource allocation. For such a problem, the task uplink transmission time, MEC computation time and the computation results downloading time were considered at the same time. Since the problem was non-convex, we first reformulated it, and then decomposed it into two subproblems, i.e., JUDTT-OP and JTORTSF-OP. By solving them, JTORAEH algorithm was proposed to solve the considered problem. Simulation results show that compared with other benchmark methods, the proposed JTORAEH algorithm can achieve a better performance in terms of the total energy consumption.

Considering the limited energy of batteries can not provide SMDs with long-term and stable power supply, and rechargeable batteries or power supply through the traditional grid (for example, when SMDs are distributed in remote or dangerous environment) may even be impossible or extremely expensive. We can utilize EH technology to achieve the green communication and durable operation of the SMDs. This article provides us with a solution to reduce energy consumption through EH and resource allocation strategy in the future 6G MEC network.

Abbreviations

MEC: Mobile edge computing; SMDs: Smart mobile devices; 6G: Sixth-generation; EH: Energy harvesting; TS: Time switching; JUDTT-OP: Joint uplink and downlink transmission time optimization subproblem; JTORTSF-OP: Joint task offloading ratio and TS factor optimization subproblem; SWIPT: Simultaneous wireless information and power transfer; JTORAEH: Joint task offloading and resource allocation with energy harvesting

Acknowledgements

The authors would like to thank all anonymous reviewers for their invaluable comments.

Authors' contributions

SL and RJ conceived the original idea and completed the theoretical analysis. ZZ and FZ completed the numerical simulations. NZ and GY improved the system model and algorithm of the article. All authors provided useful discussions and reviewed the manuscript. The authors read and approved the final manuscript.

Funding

This work was partly supported by the National Natural Science Foundation of China (No.61861002, 62101507). Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing (No. GXKL06200121). Guangxi Science and Technology Plan Project (No.AD18281044).

Availability of data and materials

The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Guangxi Key Laboratory of Wireless Broadband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin, China. [2]Department of Electrical and Computer Engineering, University of Windsor, Windsor, Canada. [3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China. [4]School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China.

Received: 25 August 2021 Accepted: 16 June 2022
Published online: 29 June 2022

References

1. Li S, Zhang N, Lin S, Kong L, Katangur A, Khan MK, Ni M, Zhu G (2018) Joint admission control and resource allocation in edge computing for Internet of things. IEEE Netw 32(1):72–79
2. Li S, Lin S, Cai L, Li W, Zhu G (2020) Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing. IEEE Trans Veh Technol 69(3):3384–3398
3. Li S, Zhang N, Chen H, Lin S, Dobre OA, Wang H (2021) Joint road side units selection and resource allocation in vehicular edge computing. IEEE Trans Veh Technol 70(12):13190–13204
4. Li S, Wang Q, Wang Y, Xie J, Tan D, Kou W, Li W (2021) Joint congestion control and resource allocation for delay-aware tasks in mobile edge computing. Wirel Commun Mob Comput 2021(1):1–16
5. Jiang R, Xiong K, Fan P, Zhou L, Zhong Z (2020) Outage probability and throughput of multirelay SWIPT-WPCN networks with nonlinear EH model and imperfect CSI. IEEE Syst J 14(1):1206–1217

6. Chen Y, Zhang N, Zhang Y, Chen X, Wu W, Shen XS (2021) TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing. IEEE Trans Cloud Comput 9(4):1634–1644
7. Wang F, Zhang X (2018) Dynamic computation offloading and resource allocation over mobile edge computing networks with energy harvesting capability. In: 2018 IEEE International Conference on Communications (ICC). pp 1–6. https://doi.org/10.1109/ICC.2018.8422096
8. Jiang R, Xiong K, Fan P, Zhang Y, Zhong Z (2019) Power minimization in SWIPT networks with coexisting power-splitting and time-switching users under nonlinear EH model. IEEE Internet Things J 6(5):8853–8869
9. Zhang G, Chen Y, Shen Z, Wang L (2019) Distributed energy management for multiuser mobile-edge computing systems with energy harvesting devices and QoS constraints. IEEE Internet Things J 6(3):4035–4048
10. Wang F, Xu J, Cui S (2020) Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems. IEEE Trans Wirel Commun 19(4):2443–2459
11. Teng Y, Cheng K, Zhang Y, Wang X (2019) Mixed-timescale joint computational offloading and wireless resource allocation strategy in energy harvesting multi-MEC server systems. IEEE Access 7:74640–74652
12. Hu X, Wong K, Yang K (2018) Wireless powered cooperation-assisted mobile edge computing. IEEE Trans Wirel Commun 17(4):2375–2388
13. Wang F, Xu J, Wang X, Cui S (2018) Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans Wirel Commun 17(3):1784–1797
14. Ji L, Guo S (2019) Energy-efficient cooperative resource allocation in wireless powered mobile edge computing. IEEE Internet Things J 6(3):4744–4754
15. Mahmood A, Ahmed A, Naeem M, Hong Y (2020) Partial offloading in energy harvested mobile edge computing: A direct search approach. IEEE Access 8:36757–36763
16. Liu B, Wang J, Ma S, Zhou F, Ma Y, Lu G (2019) Energy-efficient cooperation in mobile edge computing-enabled cognitive radio networks. IEEE Access 7:45382–45394
17. Zhao Y, Leung VCM, Gao H, Chen Z, Ji H (2018) Uplink resource allocation in mobile edge computing-based heterogeneous networks with multi-band RF energy harvesting. In: 2018 IEEE International Conference on Communications (ICC). IEEE. pp 1–6. https://doi.org/10.1109/ICC.2018.8422201
18. Wang F, Xing H, Xu J (2019) Optimal resource allocation for wireless powered mobile edge computing with dynamic task arrivals. In: 2019 IEEE International Conference on Communications (ICC). IEEE. pp 1–7. https://doi.org/10.1109/ICC.2019.8761143
19. Guo J, Song Z, Cui Y, Liu Z, Ji Y (2017) Energy-efficient resource allocation for multi-user mobile edge computing. In: GLOBECOM 2017 - 2017 IEEE Global Communications Conference. pp 1–7
20. Jiang R, Xiong K, Fan P, Zhong Z, Letaief KB (2020) Information-energy region for SWIPT networks in mobility scenarios. IEEE Trans Veh Technol 69(7):7264–7280
21. Zhao P, Tian H, Qin C, Nie G (2017) Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. IEEE Access 5:11255–11268
22. Miettinen AP, Nurminen JK (2010) Energy efficiency of mobile clients in cloud computing. In: 2010 USENIX Conference on Hot Topics in Cloud Computing (Hotcloud). USENIX Association. pp 4–4. https://doi.org/10.5555/1863103.1863107
23. Choi JH, Bansal A, Meterelliyoz M, Murthy J, Roy K (2007) Self-consistent approach to leakage power and temperature estimation to predict thermal runaway in FinFET circuits. IEEE Trans Comput Aided Des Integr Circ Syst 26(11):2059–2068
24. Wei X, Goth G, Kelly P, Zoodsma R, VanDeventer A (2014) Air-water hybrid cooling for computer servers: A case study for optimum cooling energy allocation. In: Fourteenth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm). IEEE. pp 568–573. https://doi.org/10.1109/ITHERM.2014.6892331
25. Jiang R, Xiong K, Fan P, Zhang Y, Zhong Z (2017) Optimal design of swipt systems with multiple heterogeneous users under non-linear energy harvesting model. IEEE Access 5:11479–11489
26. Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press, Cambridge
27. Wang F, Xu J, Wang X, Cui S (2018) Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans Wirel Commun 17(3):1784–1797
28. Zhang J, Xia W, Yan F, Shen L (2018) Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing. IEEE Access 6:19324–19337

Publisher's Note