Parallel Offloading in Green and Sustainable Mobile Edge Computing for Delay-Constrained IoT System

Yiqin Deng, *Member, IEEE*, Zhigang Chen, *Member, IEEE*, Xin Yao, *Member, IEEE*, Shahzad Hassan, and Ali. M. A. Ibrahim

***Abstract*—Currently, the Internet of Things (IoT) solutions are playing an important role in numerous areas, especially in smart homes and buildings, health-care, vehicles, and energy. It will continue to expand in various fields in the future. However, some issues limit the further development of IoT technologies. First, the battery-powered feature increases the maintenance cost of replacing batteries for IoT devices. Second, existing Cloud-IoT frameworks are not able to cope with emerging delay-constrained applications in the IoT system due to its centralized mode of operation and the considerable communication delay. Existing studies neither satisfy the demand for the quick response in time-constraint IoT applications nor fundamentally solving the problem of energy sustainability. Therefore, this paper studies the problem of energy sustainability and timeliness in IoT system. Based on Energy Harvesting Technologies (EHT), the Green and Sustainable Mobile Edge Computing (GS-MEC) framework is proposed to make IoT devices self-powered by utilizing the green energy in the IoT environment. In this framework, we formulate the problem of minimizing response time and packet losses of tasks under the limitation of energy queue stability to improve the timeliness and reliability of task processing. Additionally, the dynamic parallel computing offloading and energy management (DPCOEM) algorithm is designed to solve the problem based on the Lyapunov optimization technology. Finally, theoretical analysis demonstrates the effectiveness of the proposed algorithm, and the numerical result of simulation shows that the average performance of the proposed algorithm is an order of magnitude better than state-of-the-art algorithms.**

***Index Terms*—Mobile edge computing, internet of things (IoT), partial computation offloading, energy harvesting, resource allocation, lyapunov optimization.**

I. Introduction

INTERNET of Things (IoT) solutions are playing an important role in numerous areas, especially in smart homes, smart buildings, healthcare, vehicles, and energy. These areas are defined as the sectors that are currently utilizing the IoT to the fullest by the US Government Accountability Office (GAO) [1]. The IoT systems have been prospered by explosive growth in mobile devices whenever ubiquitous availability of wireless connectivity and the fast-falling cost of IoT sensors [2], [3]. It is forecasted that there will be 75 billion IoT devices around the world by 2025 [2]. Also, from 2004 to 2014, the average price of IoT sensors dropped by more than half, and it is projected to shrink another 37% to 0.38 USD by 2020 [3]. According to International Data Corporation (IDC), the IoT is going to stay hot. The investment in the IoT is forecasted to increase by 15.4% in 2019 and consequently 1 trillion USD mark in 2022.

However, some issues limit the further development of IoT technologies. First, the battery-powered feature increases the maintenance cost of IoT devices and reduces the service life of them. The cost of replacing batteries often higher than the cost of the IoT device itself. There would be 274 million battery replacements in IoT devices a day in a 10-year lifespan scenario and the number would be 913 million (globally) per day in case of three-year lifespan [4]. Recent work [5] shows that there is much potential to utilize Energy Harvesting Technologies (EHT) to reduce the need for, or even eliminate, the batteries used in IoT devices. Second, the latest research [6] presents that existing Cloud-IoT frameworks are not able to cope with emerging delay-constrained applications in the IoT system due to its centralized mode of operation and the large communication delay. For instance, the biomedical application allows for a delay of no more than a few milliseconds. Fortunately, the rapid development of Mobile Edge Computing (MEC) in recent years has provided an opportunity for the rise of these delay-constrained applications.

Although the aforementioned problems of battery sustainability and real-time task processing are important, they have been under-investigated. In previous years, many studies have focused on reducing the energy consumption of IoT devices. Unfortunately, with the proliferation of data in IoT systems, these methods are gradually failing. On the other hand, the scheme of using EHT to extend the lifetime of IoT devices has an obvious drawback: dynamics of the harvested energy. That is, the energy input and output may be inconsistent. Moreover, as pointed out by the recent work [7], existing frameworks on IoT communications must be re-engineered in order to meet stringent delay deadlines and be robust to packet losses. MEC can play a vital role in the task processing for IoT, however, this research is new and most of the existing studies just put forward such a concept instead of designing an effective algorithm to guarantee the above two aspects: delay and packet losses.

Manuscript received September 10, 2019; revised September 27, 2019; accepted September 27, 2019. Date of publication October 2, 2019; date of current version December 17, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61672540, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2019JJ50802, in part by the Graduate Research and Innovation Project of Hunan under Grant CX20190124, in part by the Major Program of the National Natural Science Foundation of China under Grant 71633006, and in part by the "Mobile Health" Ministry of Education-China Mobile Joint Laboratory. The review of this article was coordinated by Dr. K. Bian. *(Corresponding author: Zhigang Chen.)*

The authors are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: dengyiqin@csu.edu.cn; czg@csu.edu.cn; xinyao@csu.edu.cn; shahzadadeem@hotmail.com; alimaher_912@hotmail.com).

Digital Object Identifier 10.1109/TVT.2019.2944926

In this paper, we propose the Green and Sustainable Mobile Edge Computing (GS-MEC) framework for delay-constrained IoT systems. The idea of GS-MEC framework is fundamentally different from the traditional communication framework (i.g. Cloud-IoT communication framework) in the IoT system. To ensure the timeliness of task processing, we adopt the parallel offloading strategy in MEC. To ensure the reliability of task processing, we consider packet losses in this framework. In order to make full use of the green energy in the IoT system, we apply EHT to the IoT system and use Lyapunov optimization technology to achieve energy stability and sustainability.

Specifically, we use IoT devices along with the edge server for parallel processing tasks in a collaborative fashion instead of uploading tasks to the remote Cloud centers. Despite this combination in GS-MEC supported delay-constrained IoT system, the problem of minimizing response time and packet losses with the limitation of energy queue stability is formulated. To solve this problem, we design the DPCOEM algorithm based on the Lyapunov optimization technology. In this algorithm, we can get the following important variables of a single time slot: the energy harvested by IoT devices, the transmission power set for IoT devices, the CPU frequency set for IoT devices, as well as offloading decision vectors. Finally, theoretical analysis demonstrates the effectiveness of the proposed DPCOEM algorithm and simulation results show that the average performance of the proposed DPCOEM algorithm is an order of magnitude better than state-of-the-art algorithms.

Our main contributions are summarized as follows:

- We propose the GS-MEC framework for delay-constrained IoT systems. This framework enables the IoT system energy sustainable by incorporating the EHT into IoT devices. Additionally, it processes tasks in parallel both in IoT edge devices and the edge server to reduce task completion time and the ratio of dropping tasks.
- We formulate a minimization problem of response time and dropping tasks with the constraint of energy stability of Energy Harvest (EH) IoT devices. This problem involves coupled variables and continuous variables.
- We use the Lyapunov optimization technique to decompose the formulated problem and use the variable substitution optimization technique decouples variables in the subproblem. Then, the DPCOEM algorithm is developed to get the optimal parameters at every time slot.
- We perform the theoretical analysis and it demonstrates that the proposed DPCOEM algorithm can achieve approximately optimal performance. Results show the proposed DPCOEM algorithm outperforms other methods.

Section II gives related works. System model and problem formulation are presented in Section III. In Section IV, we propose an algorithm to solve the task cost minimization problem. The performance of the proposed algorithm is analyzed in Section V. Finally, extensive simulation's results and conclusions are provided in Sections VI and VII, respectively.

II. Related Work

Existing works on MEC-enabled IoT generally concentrate on energy efficiency and binary offloading. However, energy harvesting and parallel offloading among delay-constrained IoT systems have not been thoroughly investigated. Next, we will elaborate on them from three perspectives.

A. Energy Harvesting

Current researches on the energy of IoT systems is mainly focused on the research of battery-powered equipment, and its primary goal is to save energy [8]–[13]. Ning *et al.* [8] put forward an energy-efficient task scheduling framework for MEC-assisted Internet of Vehicles to minimize the energy consumption of roadside units. Chen *et al.* [9] propose a device-to-device Crowd framework for IoT systems in the fifth-generation era with the help of MEC and develop a graph-matching-based optimal task assignment algorithm to achieve energy efficiency. Lyu *et al.* [10] present an integration architecture of the cloud, MEC, and IoT, and design a selective offloading scheme to minimize the energy consumption of devices and satisfy the latency requirements of different services. Dong *et al.* [11] developed a joint offloading and resource management framework and propose two algorithms to minimize the energy consumption of the embedded system. Xu *et al.* [12] studied the joint task offloading and caching algorithm on edge-cloud computing to reduce the cost of energy consumption. Wang *et al.* [13] incorporated the DVFS technology into task offloading methods in MEC, for the sake of reducing energy consumption and application execution time.

Proliferating data and high maintenance costs of the battery in IoT systems make these energy-efficient or energy-saving methods insufficient to keep the IoT system running sustainably. On the other hand, some companion technologies of IoT have gotten large innovations in recent years, like ultra-low-power wireless sensor networks, IoT-specific network protocols and standards, low-power integrated circuits, and EHT for IoT devices [14]–[18]. The development of these new technologies make green and sustainable IoT systems possible, e.g., incorporating EHT into IoT edge devices [18].

Researches on EHT involve many fields, especially wireless networks [5], [18]–[21]. The research team at Central South University found that the Y6-based solar cell delivers a high-power conversion efficiency of 15.7% [19]. The company of E-peas developed the radio frequency (RF) energy harvesting IC solution–E-peas AEM40904 PMIC, which can harvest RF input currents up to 125 mA with a tiny footprint of 55 mm [5]. Zhang *et al.* [20] studied energy harvesting cognitive radio sensor networks on resource management and allocation, which supplies the sensor nodes with continuous energy by energy harvesting technology to extend network lifetime. Guo *et al.* [21] proposed a joint energy and channel transmission management algorithm for body area networks with energy harvesting devices. Ünlü *et al.* [18] showed that there are abundant energy sources for devices in the IoT environment, such as solar power, thermal energy, wind energy, and kinetic energy, and these EH IoT devices can harvest enough energy to support the operation of these devices. These studies do not incorporate EHT into MEC-enabled IoT systems to achieve energy sustainability. Even the latest report just illustrated the feasibility of applying

EHT to IoT edge devices from a quantitative perspective and it did not propose a framework or solutions [18].

B. Parallel Offloading

Task offloading in wireless networks is an effective approach to save energy for devices and reduce the delay of processing tasks, and it has been extensively studied [22]–[25].

Rodrigues *et al.* [22] proposed a Cloudlets activation scheme that lowers the delay and raises the number of served users based on transmission power control and virtual machine migration in scalable MEC. In [23], the authors presented a method for minimizing the service delay in Edge-Cloud system through virtual machine migration and transmission power control to reduce the processing delay and transmission delay, respectively. Ndikumana *et al.* [24] proposed a collaborative cache allocation and computation offloading method to maximize resource utilization in MEC. Salmani *et al.* [25] solved the problem of minimizing the completion time of several camera sensors that share the transmission and the processing resources for computation offloading. Nevertheless, these methods belong to binary offloading, that is, they either totally process tasks on the mobile device or migrating tasks to the other server (e.g., the edge server and the cloud). These schemes cannot take full advantage of resources in both the mobile device and the server, and still cause a significant delay, which is still far below than the user's expectation.

C. MEC-Enabled IoT

The rapid development of IoT applications in recent years has brought enormous challenges to the transmission and processing of data, especially in delay-constrained applications. Therefore, many kinds of researches have been done in this aspect [26]–[30]. Tang *et al.* [26] proposed a deep learning (DL)-based algorithm to intelligently allocate channels to each link in the software-defined networking (SDN)-based IoT system for improving the transmission quality of it. In [27], the authors presented the DL-based intelligent POC assignment for the highly dynamic large-scale SDN-IoT to avoid network congestion. To overcome the scalability problem of the traditional IoT architecture, Sun *et al.* [28] proposed the edge-IoT architecture for handling the data generated from distributed IoT edge devices. Chen *et al.* [29] developed a resource-efficient edge computing scheme for the emerging intelligent IoT applications to support its computationally intensive task. Guo *et al.* [30] devised a mobile-edge computation offloading strategy for ultradense IoT networks to handle the conflict between the resource-hungry IoT applications and the resource-constrained devices. However, even the MEC-enabled IoT systems are advanced, none of these systems consider reducing computational and transmission delays simultaneously for the delay-constrained IoT system

To fill the research gap, this paper proposes the green and sustainable MEC framework and parallel offloading method to achieve energy sustainability and delay minimization in delay-constrained IoT systems in this framework. These researches are based on EHT and divisible load theory [31]. Moreover, computational and transmission delays are reduced simultaneously in this paper based on the transmission power control and CPU frequency control.

TABLE I
MAIN SYMBOLS AND THEIR MEANINGS

Symbols	Meanings
$\mathcal{A}$	A computing task
$\mathcal{H}$	The size of a task
ϵ	The deadline of a task
θ	The probability of task arrival
$\mathcal{X}$	The offloading decision vector
$x^l(t)$, $x^c(t)$	Percentage of tasks on IoT device or edge server
$x^d(t)$	The decision of dropping tasks
ζ	The CPU size required for processing a unit task
$\mathcal{M}$	The CPU cycles needed to process a task
$f(t)$	The CPU frequency scheduled
$\mathcal{T}^l(t)$, $\mathcal{T}^c(t)$	Time of processing task on IoT device or edge server
$\mathcal{P}^l(t)$, $\mathcal{P}^c(t)$	Energy consumed on IoT device or edge server
r	The transmission rate from the IoT device to edge server
$p(t)$	The transmission power scheduled for the IoT device
$\mathcal{P}^{total}$	The total energy consumption of processing all tasks
$E(t)$	The energy queue of IoT device
$e(t)$	The energy harvested by the IoT device
$\mathcal{W}$	The task cost
$\mathcal{D}(t)$	Time caused by processing task
β	The cost of dropping tasks
ϕ	The auxiliary constant
$\hat{E}(t)$	The virtual energy queue
p_{max}, f_{max}	The maximum transmission power and CPU frequency

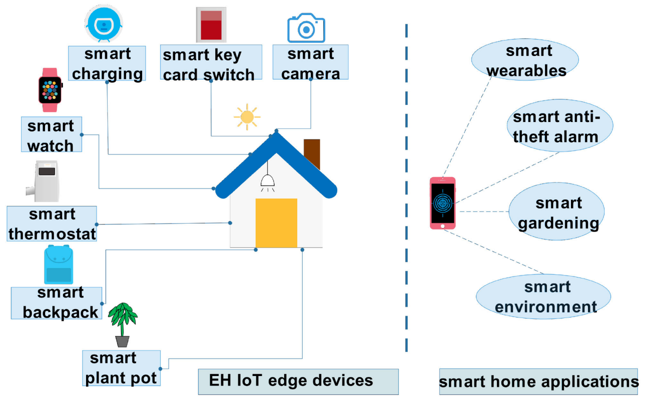


Fig. 1. Energy sustainable smart home scenario.

III. System Model and Problem Formulation

In this section, the computing model, the energy model, the task cost and problem formulation are given. The main symbols used in this paper and their specific meanings are summarized in Table I.

A. Overview

As we can see from Fig. 1, we consider the scenario of a smart home that consists of numerous IoT edge devices (e.g.,

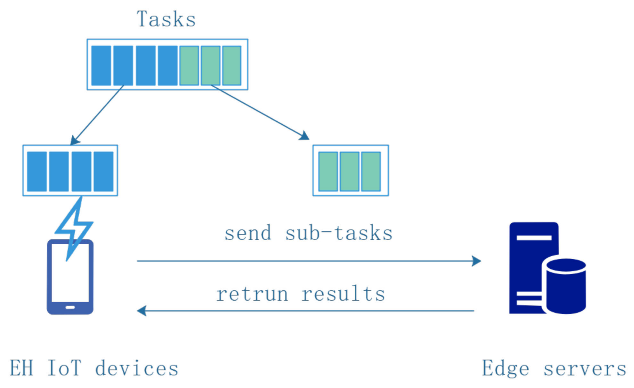


Fig. 2. Parallel task processing for IoT applications under GS-MEC framework.

smartwatch, smart thermostat, smart charging, smart backpack, smart plant pot, smart key card switch, and smart camera.) and applications (e.g., smart gardening, smart anti-theft alarm, smart wearable, and smart environment.) on mobile phone or tablet. IoT edge devices are equipped with energy-harvesting components (e.g., energy harvester) and thus they can convert the captured ambient energy (e.g., solar energy, thermal, vibrational, and RF energy) into electrical energy to power these devices. Furthermore, since energy harvesting and energy consumption may be inconsistent in a time dimension, every EH IoT edge device is furnished with a temporary energy buffer (e.g., super-capacitor) to effectively deliver the energy from the harvester to the IoT edge device. Note that the generated energy of EH IoT devices can cover multiple orders of magnitude, ranging from 0.1 W to 1 W, that is sufficient for the average power consumption of the device (i.e., 100 $\mathrm{W/cm^2}$) [18].

Moreover, the EH IoT edge devices produce the computing task $\mathcal{A}(\mathcal{H},\epsilon)$, where $\mathcal{H}$ and ϵ denote the size and deadline of the task $\mathcal{A}$, respectively. The probability of task arriving is θ, otherwise $1-\theta$. Besides, denote $\mathcal{R}(t)=1$ when a task arrives, otherwise $\mathcal{R}(t)=0$. We have $\mathcal{P}(\mathcal{R}(t)=1)=1-\mathcal{P}(\mathcal{R}(t)=0)=\theta$. Similar to many researches (e.g., [13], [32], [33]), tasks in this paper are separable, as well as they can be partially offloaded to the edge server through the dynamic wireless channel. The wireless channel is independent and identically distributed at every time slot t and $\forall t\in\mathcal{T}$. Note that the edge server is usually powered by the conventional power grid; thus, just the energy consumption of the IoT device is considered in this paper.

B. Computing Model

Define $\mathcal{X}$ as the offloading decision vector and $\mathcal{X}$ consists of three elements: $x^l(t)$, $x^c(t)$ and $x^d(t)$. $x^l(t)$ and $x^c(t)$ denote the percentage of tasks processed locally (on the IoT device) and tasks processed on the edge server, respectively. $x^d(t)$ represents the dropping decision. In particular, dropping tasks means that the entire task will be dropped at one time; thus, the value of $x^d(t)$ is 0 or 1. However, dropping tasks will increase the cost, which will be described in the problem formulation in detail. In short, the three variables are constrained by the following equations.

$$x^l(t)+x^c(t)+x^d(t)=1,\ \forall t\in\mathcal{T}. \tag{1}$$

$$x^l(t),x^c(t)\in[0,1],x^d(t)\in\{0,1\},\ \forall t\in\mathcal{T}. \tag{2}$$

Only Local Processing Model: At this time, $x^l(t)=1,\ x^c(t)=0,\ x^d(t)=0$. The CPU size required for processing a unit task is ζ; thus, $\mathcal{M}=\mathcal{H}\cdot\zeta$ cycles are needed to process the task $\mathcal{A}(\mathcal{H},\epsilon)$. The frequency scheduled for the $\mathcal{M}$ CPU cycles of the IoT device is denoted as $f^m(t)$, $m=1,2,\ldots,\mathcal{M}$. The frequency schedule can be achieved by the dynamic voltage and frequency scaling technique. When the task is processed only on the IoT device, we can get the task delay $\mathcal{T}^l(t)$ as follows.

$$\mathcal{T}^l(t)=\sum_{m=1}^{\mathcal{M}}\frac{1}{f^m(t)},\ \forall t\in\mathcal{T}. \tag{3}$$

Let $\mathcal{P}_n^l(t)$ represent the energy consumed by processing tasks locally. Accordingly, the energy consumed for only local processing can be expressed as

$$\mathcal{P}^l(t)=k\sum_{m=1}^{\mathcal{M}}(f^m(t))^2,\ \forall t\in\mathcal{T}. \tag{4}$$

where k is the effective switched capacitance and depends on the chip architecture.

As documented in [34], the optimal CPU frequency of $\mathcal{M}$ CPU cycles is the same when a task is processed locally. Therefore, we have $\mathcal{T}^l(t)=\frac{\mathcal{M}\cdot x^l(t)}{f(t)}$ and $\mathcal{P}^l(t)=k\cdot\mathcal{M}\cdot x^l(t)\cdot(f(t))^2$. In addition, the frequency f is constrained by the maximum available CPU frequency $f_{\max}$, that is, $f\leq f_{\max},\forall m$.

Only Edge Server Processing Model: At this time, $x^l(t)=0,\ x^c(t)=1,\ x^d(t)=0$. According to Shannon-Hartley formula, the transmission rate from the IoT device to the edge server is $r=\omega log_2(1+\frac{\lambda\cdot p(t)}{\sigma})$, where ω and σ are the system bandwidth and the noise power at the server, respectively. λ denotes the channel power gain from the device to the server. We assume that the edge server has adequate computational resource especially in multi-core CPU based extremely high speed, therefore, the processing time at the edge server is negligible [35]. Additionally, we suppose that the time of downlink transmission can be ignored since the output of tasks is significantly small [36]. When the task is processed only on the edge server, we can get the task delay $\mathcal{T}^c(t)$ as follows

$$\mathcal{T}^c(t)=\frac{\mathcal{H}}{r},\ \forall t\in\mathcal{T}. \tag{5}$$

Let $\mathcal{P}^c(t)$ represent the energy consumed by processing tasks on the edge server. Accordingly, the energy consumed for only the edge server process can be expressed as

$$\mathcal{P}^c(t)=p(t)\cdot\mathcal{T}^c(t)=\frac{p(t)\cdot\mathcal{H}}{r},\ \forall t\in\mathcal{T}. \tag{6}$$

C. Energy Model

Since dropping task does not consume energy, we can get the total energy consumption $\mathcal{P}^{total}(t)$ of processing all task in the time slot t as the following.

$$\mathcal{P}^{total}(t) = x^l(t) \cdot \mathcal{P}^l(t) + x^c(t) \cdot \mathcal{P}^c(t),\ \forall t \in \mathcal{T}. \tag{7}$$

Let $E_n(t)$ denote the energy queue length of the IoT device, which is the energy available in the energy buffer of the IoT device. Let $e(t)$ represent the specific number of energy harvested by the IoT device from the surrounding environment in the time slot t, so the energy queue dynamic of the IoT device can be expressed as follows:

$$E(t+1) = E(t) - \mathcal{P}^{total}(t) + e(t),\ \forall t \in \mathcal{T}. \tag{8}$$

In time slot t, the energy consumption of the IoT device cannot exceed the energy remaining in its energy queue. Therefore,

$$0 \leq P^{total}(t) \leq E(t),\ \forall t \in \mathcal{T}. \tag{9}$$

Moreover, the total number of energy existing in the environment at the time slot t is expressed as $\psi(t)$ and its maximum value is $\psi_{\max}$. Note that $\psi(t)$ is dynamic due to the variable environment and $\psi_{\max}$ is a fixed value that denotes the maximum existing energy in the environment at any time. Assume that $\psi(t)$ is independent and identical distributed in different time slot t [20]. Note that the algorithm designed in this paper does not need to know the specific probability distribution of $\psi(t)$. The energy acquired by the IoT device in time slot t cannot exceed the energy supply in the environment, that is,

$$0 \leq e(t) \leq \psi(t),\ \forall t \in \mathcal{T}. \tag{10}$$

Because of physical size limitations, the IoT device has limited energy buffer capacity. The sum of remaining energy and the harvested energy in time slot t cannot exceed the energy buffer capacity of the IoT device Ω. That is,

$$E(t) + e(t) \leq \Omega,\ \forall t \in \mathcal{T}. \tag{11}$$

D. Task Cost and Problem Formulation

Let $\mathcal{W}(t)$, $\mathcal{D}(t)$ and β (in second) indicate the task cost, the delay caused by the processing task and cost of dropping task, respectively. We have:

$$\mathcal{W}(t) = \alpha{\cdot}\mathcal{D}(t) + \beta{\cdot}\mathcal{I}(t),\ \forall t \in \mathcal{T} \tag{12}$$

where α is the weight of delay and we set it as 1 in this paper, and $\mathcal{I}(t)$ is an indicator function indicating that there are task arrivals and all of them are dropped at time slot t. That is,

$$\mathcal{I}(t) = \begin{cases} 1, & \text{if}\{\mathcal{R}(t) = 1, x^d(t) = 1\} \\ 0, & \text{else.} \end{cases} \tag{13}$$

Delay $\mathcal{D}(t)$ consists of two parts. One part is the computation delay processed locally, and the other part is the transmission delay of the IoT device of transmitting tasks to the edge server. Delay $\mathcal{D}(t)$ is given by

$$\mathcal{D}(t) = 1_{\{\mathcal{R}(t)=1\}} \cdot \max(x^l(t) \cdot \mathcal{T}^l(t), x^c(t) \cdot \mathcal{T}^c(t)),\ \forall t \in \mathcal{T}. \tag{14}$$

Recall that ϵ denote the deadline of the task $\mathcal{A}$; thus, the following inequality should be met:

$$\mathcal{D}(t) \leq \epsilon,\ \forall t \in \mathcal{T}. \tag{15}$$

In summary, the parallel offloading (PO) problem can be formulated as follows

$$\min_{\mathcal{X},\ f(t),\ p(t),\ e(t)} \quad \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \mathbb{E}\left[\sum_{t=0}^{\mathcal{T}-1} \mathcal{W}(t)\right]$$
$$\text{s.t.} \begin{cases} (1), (2), (3), (10), (15). \\ x^l(t) + x^c(t) \leq \mathcal{R}(t),\ \forall\, t \in \mathcal{T}. \\ \mathcal{P}^{total}(t) \leq E_{\max},\ \forall\, t \in \mathcal{T}. \\ 0 \leq p(t) \leq p_{\max} \cdot 1_{\{x^c(t)=1\}},\ \forall\, t \in \mathcal{T}. \\ 0 \leq f(t) \leq f_{\max} \cdot 1_{\{x^l(t)=1\}},\ \forall\, t \in \mathcal{T}. \end{cases} \tag{16}$$

To deal with the coupling problem in formula (9), we introduce a lower bound $E_{\min}$ and an upper bound $E_{\max}$ as follows

$$\mathcal{P}_n^{total}(t) \in 0 \bigcup [E_{\min}, E_{\max}],\ \forall n \in \mathcal{N}. \tag{17}$$

where $0 \leq E_{\min} \leq E_{\max}$.

To this end, we get a new tightened problem based on the problem (16) as

$$\min_{\mathcal{X},\ f(t),\ p(t),\ e(t)} \quad \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \mathbb{E}\left[\sum_{t=0}^{\mathcal{T}-1} \mathcal{W}(t)\right]$$
$$\text{s.t.} \begin{cases} (1), (2), (9), (10), (15). \\ x^l(t) + x^c(t) \leq \mathcal{R}(t),\ \forall\, t \in \mathcal{T}. \\ \mathcal{P}^{total}(t) \leq E_{\max},\ \forall\, t \in \mathcal{T}. \\ 0 \leq p(t) \leq p^{\max} \cdot 1_{\{x^c(t)=1\}},\ \forall\, t \in \mathcal{T}. \\ 0 \leq f(t) \leq f^{\max} \cdot 1_{\{x^l(t)=1\}},\ \forall\, t \in \mathcal{T}. \\ \mathcal{P}^{total}(t) \in 0 \bigcup [E_{\min}, E_{\max}],\ \forall\, t \in \mathcal{T}. \end{cases} \tag{18}$$

Lemma 1: Let $\mathcal{W}^1$ and $\mathcal{W}^*$ represent the optimal average task cost achieved by (16) and (18), respectively. Their relationships are given by $\mathcal{W}^1 \leq \mathcal{W}^* \leq \mathcal{W}^1 + \mu(E_{\min})$, where $\mu(E_{\min}) = \theta[\beta(1-\Lambda) + 1 \cdot_{\{E_{\min} \geq E_{\min}^{\epsilon}\}} \cdot (\beta - \tau_{E_{\min}})]$. Here, $\Lambda = \frac{(2^{\frac{\mathcal{H}}{\epsilon}}-1)\sigma\cdot\epsilon}{E_{\min}}$, $E_{\min}^{\epsilon} = \frac{\kappa\cdot\mathcal{M}^3}{\tau_d^2}$ and $E_{\min}^{\epsilon} = \frac{\sqrt{\kappa}\cdot\sqrt{\mathcal{M}^3}}{\sqrt{E_{\min}}}$. Let $E_{\min}$ infinity close to 0, that is, $\lim_{E_{\min}\to 0} \mu(E_{\min}) = 0$, $\mathcal{W}^*$ is arbitrarily close to $\mathcal{W}^1$.

Proof: Please refer to Appendix A in [36]. ■

IV. Dynamic Parallel Computing Offloading and Energy Management Algorithm

In this section, we propose the DPCOEM algorithm to solve the problem in formula (16).

A. The Problem Analysis

Denote the virtual energy queue as $\hat{E}(t)$ and $\hat{E}(t) = E(t) - \phi$. Here, ϕ is an auxiliary constant and it is bounded by the following inequality

$$\phi \geq \mathcal{P}_{\max} + \frac{\mathcal{V}\cdot\beta}{E_{\min}}. \tag{19}$$

where $P_{\max}$ is the maximum energy consumption and $\mathcal{P}_{\max} = \min\{\max\{\kappa\cdot\mathcal{M}\cdot(f_{\max})^2, p_{\max}\cdot\epsilon\}, E_{\max}\}$. $\mathcal{V}$ is a non-negative weight and it indicates how important the task cost is in the overall goal. By adjusting the value of $\mathcal{V}$, we can achieve the tradeoff between energy queue length and task cost.

Next, define the Lyapunov function as $\mathcal{L}(t)$ and we have

$$\mathcal{L}(t) \triangleq \frac{1}{2}(\hat{E}(t))^2 = \frac{1}{2}(E(t)-\phi)^2. \tag{20}$$

Then, define the Lyapunov drift function $\Delta(t)$ as follows

$$\Delta(t) \triangleq \mathbb{E}[\mathcal{L}(t+1) - \mathcal{L}(t) \mid \hat{E}(t)]. \tag{21}$$

If the Lyapunov drift function $\Delta(t)$ is minimized at every time slot t, the virtual energy queue $\hat{E}(t)$ will stay stable. Thus, the mobile device does not stop working because of energy exhaustion.

The objective of this paper is reducing task cost while extending the life of the IoT device. Therefore, we can construct the drift-plus-cost function $\Delta_{\mathcal{V}}(t)$ as follows

$$\Delta_{\mathcal{V}}(t) \triangleq \Delta(t) + \mathcal{V}\cdot[\mathcal{D}(t) + \beta\cdot 1_{\{\mathcal{R}(t)=1, x^d(t)=1\}} \mid \hat{E}(t)]. \tag{22}$$

Lemma 2: For any time slot t, the upper bound of $\Delta_{\mathcal{V}}(t)$ is

$$\begin{aligned}\Delta_{\mathcal{V}}(t) \le{} & [(\hat{E}(t))[e(t) - \mathcal{P}^{total}] + \mathcal{V} \\ & \cdot [\mathcal{D}(t) + \beta\cdot 1_{\{\mathcal{R}(t)=1, x^d(t)=1\}} \mid \hat{E}(t)]] + \mathcal{B}. \end{aligned} \tag{23}$$

where $\mathcal{B}$ is determined by system parameters, regardless of variables and $\mathcal{V}$, and it can be expressed as $\mathcal{B} = \frac{(\psi_{\max})^2 + (\mathcal{P}^{total}_{\max})^2}{2}$.

Proof: Please refer to Appendix A. ■

In addition to the constant $\mathcal{B}$, formula (23) mainly consists of two parts: one part concerning the harvesting energy and the other regarding the offloading decision. Therefore, the problem of minimizing $\Delta_{\mathcal{V}}(t)$ can be decomposed into two sub-problems: energy management and parallel computing offloading.

B. The Proposed DPCOEM Algorithm

Energy management: Considering terms with respect to $e(t)$ in (23), we can construct an energy management (EM) problem as follows

$$\begin{aligned}\min_{e(t)} \quad & \hat{E}(t)e(t) \\ \text{s.t.} \quad & \begin{cases} 0 \le e(t) \le \psi(t). \\ E(t) + e(t) \le \Omega. \end{cases}\end{aligned} \tag{24}$$

The energy management problem in formula (24) is a linear programming problem and it is easy to be solved. If the energy buffer of the IoT device can accommodate more power at the beginning of the time slot t, i.e., the IoT device continues to acquire energy; otherwise, the IoT device no longer acquires energy. In general, with energy management, the IoT device will harvest as much energy as possible to fill the energy buffer. The optimal energy acquisition $e^*(t)$ is as follows

$$e^*(t) = \begin{cases} \min(\Omega - E(t), \psi(t)), & \text{if } \hat{E}(t) \le 0 \\ 0, & \text{else.} \end{cases} \tag{25}$$

Parallel computing offloading: After separating $e(t)$ from formula (23), we can construct a dynamic parallel computing offloading (DPCO) problem related to $\mathcal{X}, \ f(t), \ p(t)$.

$$\begin{aligned}\min_{\mathcal{X},\, f(t),\, p(t)} \quad & -\hat{E}(t)\cdot\mathcal{P}^{total}(t) + \mathcal{V}[\mathcal{D}(t) + \beta\cdot 1_{\{\mathcal{R}(t)=1, x^d(t)=1\}}] \\ \text{s.t.} \quad & (1), (2), (5) \\ & \mathcal{P}^{total}(t) \in [E_{\min}, E_{\max}], \ \forall\, t \in \mathcal{T}. \end{aligned} \tag{26}$$

Parallel computing offloading problem contains three sub-problems: task allocation $\mathcal{X}$, transmission power $p(t)$ and CPU frequency $f(t)$. It is difficult to solve these three sub-problems directly because these problems affect each other and have higher coupling [37].

When the task is dropped, that is, $x^d(t) = 1, \ x^l(t) = x^c(t) = 0$, the mobile device does not need to process tasks or transmit tasks to the edge server. To this end, we have $f(t) = p(t) = 0$.

Next, we consider the case of no task dropping. In order to deal with the high coupling of these three subproblems, we utilize the alternative optimization techniques and convert them to the following three equivalent subproblems: (i) Task offloading problem: when the transmission power and CPU frequency are given, e.g., $p(t) = p^0, \ f(t) = f^0$, we can obtain the optimal solution $\mathcal{X}^*$. (ii) Transmission power problem: when the task allocation and CPU frequency are fixed, e.g., $\mathcal{X} = \mathcal{X}^0, \ f(t) = f^0$, we can get the optimal solution $p^*(t)$. (iii) CPU frequency problem: when the Task allocation and transmission power are given, e.g., $\mathcal{X} = \mathcal{X}^0, \ p(t) = p^0$, the optimal solution $f^*(t)$ can be obtained. First, we discuss the task offloading (TO) problem as the following

$$\begin{aligned}\min_{\mathcal{X}} \quad & -\hat{E}_n(t)[x^l(t)\cdot\mathcal{P}^l_n(t) + x^c(t)\cdot\mathcal{P}^c(t)] + \mathcal{V}\cdot\{1_{\{\mathcal{R}(t)=1\}} \\ & \cdot\max(x^l(t)\cdot\mathcal{T}^l(t), x^c(t)\cdot\mathcal{T}^c(t)) + \beta\cdot 1_{\{\mathcal{R}(t)=1\}}\} \\ \text{s.t.} \quad & (1), (2), (15) \\ & P^{total}(t) \in [E_{\min}, E_{\max}], \ \forall\, t \in \mathcal{T}. \end{aligned} \tag{27}$$

The formula (27) is a convex optimization problem with respect to the variable $\mathcal{X}$ since it is composed of several convex functions added together. To this end, the optimal offloading decision $\mathcal{X}^*$ can be obtained with already mature convex optimization techniques, such as the interior point method.

Second, the transmission power (TP) problem is discussed

$$\begin{aligned}\min_{p(t)} \quad & -\hat{E}(t)\left[x^l(t)\cdot\mathcal{P}^l(t) + x^c(t)\cdot\frac{p(t)\cdot\mathcal{H}}{\omega log_2(1+\frac{\lambda\cdot p(t)}{\sigma})}\right] + \mathcal{V} \\ & \cdot\left\{1_{\{\mathcal{R}(t)=1\}}\cdot\max\left(x^l(t)\cdot\mathcal{T}^l(t), x^c(t)\cdot\frac{\mathcal{H}\cdot x^c(t)}{\omega log_2(1+\frac{\lambda\cdot p(t)}{\sigma})}\right)\right. \\ & \left. + \beta\cdot 1_{\{\mathcal{R}(t)=1\}}\right\} \\ \text{s.t.} \quad & \mathcal{D}(t) \le \epsilon, \ \forall t \in \mathcal{T} \\ & 0 \le p(t) \le p_{\max}, \ \forall t \in \mathcal{T} \\ & \mathcal{P}^{total}(t) \in [E_{\min}, E_{\max}], \ \forall\, t \in \mathcal{T}. \end{aligned} \tag{28}$$

Solve the problem in formula (28), we can get the optimal transmission power $p^*(t)$ as follows

$$p^*(t) = \begin{cases} p_{\mathcal{U}}, & \text{if } \hat{E}(t) \geq 0 \text{ or } \hat{E}(t) < 0,\ p_{\mathcal{I}} > p_{\mathcal{U}} \\ p_{\mathcal{I}}, & \text{if } \hat{E}(t) < 0,\ p_{\mathcal{L}} \leq p_{\mathcal{I}} \leq p_{\mathcal{U}} \\ p_{\mathcal{L}}, & \text{if } \hat{E}(t) < 0,\ p_{\mathcal{I}} < p_{\mathcal{L}}. \end{cases} \tag{29}$$

where $p_{\mathcal{I}}$ is the solution for the equation $-\hat{E}(t)log_2(1 + \frac{\lambda \cdot p(t)}{\sigma}) - \frac{\lambda}{(\sigma + \lambda \cdot p(t))ln2}(\mathcal{V} - p(t) \cdot \hat{E}(t)) = 0$; $p_{\mathcal{L}}$ and $p_{\mathcal{U}}$ are defined as the following

$$p_{\mathcal{L}} = \begin{cases} p_{\mathcal{L},\epsilon}, & \text{if } \frac{\sigma \cdot \mathcal{H} \cdot ln2}{\omega h} \geq E_{\min}, \\ \max\{p_{\mathcal{L},\epsilon},\ p_{E\min}\}, & \text{else.} \end{cases} \tag{30}$$

where $p_{\mathcal{L},\epsilon} \triangleq \frac{\sigma(2^{\frac{\mathcal{H}}{\omega \cdot \epsilon}} - 1)}{\lambda}$. $p_{E\min}$ satisfies the following two formulas: $p_{E\min} \cdot \mathcal{H} = r(\lambda, p_{E\min}) \cdot E_{\min}$ and $\frac{\sigma \cdot \mathcal{H} \cdot ln2}{\omega h} \geq E_{\min}$.

$$p_{\mathcal{U}} = \begin{cases} \min\{p^{\max},\ p_{E\max}\}, & \text{if } \frac{\sigma \cdot \mathcal{H} \cdot ln2}{\omega h} < E_{\max}, \\ 0, & \text{else.} \end{cases} \tag{31}$$

where $p_{E\max}$ satisfies the following two formulas: $p_{E\max} \cdot \mathcal{H} = r(\lambda, p_{E\max}) \cdot E_{\max}$ and $\frac{\sigma \cdot \mathcal{H} \cdot ln2}{\omega h} < E_{\max}$. The detailed solution process of $p^*(t)$ is omitted due to the limited space.

Then, we discuss the CPU frequency (CF) problem as follows

$$\begin{aligned} \min_{f(t)} \quad & -\hat{E}(t)[x^l(t) \cdot k \cdot \mathcal{M} \cdot (f(t))^2 + x^c(t) \cdot \mathcal{P}^c(t)] + \mathcal{V} \\ & \cdot \left\{ 1_{\{\mathcal{R}(t)=1\}} \cdot \max\left(\frac{x^l(t) \cdot \mathcal{M}}{f(t)}, \frac{x^c(t) \cdot \mathcal{H}}{r} \right) \right. \\ & \left. + \beta \cdot 1_{\{\mathcal{R}(t)=1\}} \right\} \\ \text{s.t.} \quad & \mathcal{D}(t) \leq \epsilon,\ \forall t \in \mathcal{T} \\ & 0 \leq f(t) \leq f_{\max},\ \forall t \in \mathcal{T} \\ & \mathcal{P}^{total}(t) \in [E_{\min}, E_{\max}],\ \forall t \in \mathcal{T}. \end{aligned} \tag{32}$$

By solving the problem in formula (32), we can get the optimal CPU frequency $f^*(t)$ as follows

$$f^*(t) = \begin{cases} f_{\mathcal{U}}, & \text{if } \hat{E}(t) \geq 0 \text{ or } \hat{E}(t) < 0,\ f_{\mathcal{I}} > f_{\mathcal{U}} \\ f_{\mathcal{I}}, & \text{if } \hat{E}(t) < 0,\ f_{\mathcal{L}} \leq f_{\mathcal{I}} \leq f_U \\ f_{\mathcal{L}}, & \text{if } \hat{E}(t) < 0,\ f_{\mathcal{I}} < f_{\mathcal{L}}. \end{cases} \tag{33}$$

where $f_{\mathcal{I}} = \sqrt[3]{\frac{\mathcal{V}}{-2k \cdot \hat{E}(t)}}$, $f_{\mathcal{L}} = \max\{\sqrt{\frac{E_{\min}}{k \cdot \mathcal{M}}}, \frac{\mathcal{M}}{\epsilon}\}$ and $f_{\mathcal{U}} = \min\{\sqrt{\frac{E_{\max}}{k \cdot \mathcal{M}}}, f_{\max}\}$. The detailed solution process of $f^*(t)$ is omitted due to the limited space.

The proposed DPCOEM algorithm is summarized as Algorithm IV-B.

Algorithm 1: The Proposed DPCOEM Algorithm.

```
   Input: $\mathcal{R}(t), \psi(t), h, \hat{E}(t)$
   Output: $e^*(t)$, $\mathcal{X}^*$, $f^*(t)$, $p^*(t)$, $\hat{E}(t+1)$
   /* Energy Management                          */
1  Solve EM problem as Eq. (24);
2  if $\hat{E}(t) \leq 0$ then
3      $e^*(t) = \min(\Omega - E(t), \psi(t))$;
4  else
5      $e^*(t) = 0$
   /* Parallel Computing Offloading              */
6  Solve TO problem as Eq. (27) to get $\mathcal{X}^*$;
7  Solve TP problem as Eq. (28) to get $p^*(t)$ as Eq. (29);
8  Solve CP problem as Eq. (32) to get $f^*(t)$ as Eq. (33);
   /* Queues Updating                            */
9  Compute $E(t+1)$ based on Eq. (8);
10 Compute $\hat{E}(t+1)$ based on $\hat{E}(t+1) = E(t+1) - \phi$;
```

V. Performance Analysis

A. The Optimality Analysis of the Proposed DPCOEM Algorithm

Lemma 3: (Optimality). Recall that $\mathcal{W}^1$ is the optimal time average task cost obtained by the origin problem in formula (16) and $\mathcal{W}^*$ is the time average task cost obtained by the proposed DPCOEM algorithm (solving formula (18)). To this end, the relationship between $\mathcal{W}^*$ and $\mathcal{W}^1$ is expressed as follows:

$$\mathcal{W}^* \leq \mathcal{W}^1 + \mu(E_{\min}) + \frac{\mathcal{B}}{\mathcal{V}}. \tag{34}$$

Proof: Please refer to Appendix B. ■

Lemma 3 demonstrates that there is a positive correlation between the performance of the proposed DPCOEM algorithm and $E_{\min}$, as well as a negative correlation between the performance of the proposed DPCOEM algorithm and $\mathcal{V}$. Thus, we can achieve optimal performance by increasing $\mathcal{V}$ and decreasing $E_{\min}$. Nevertheless, the bigger $\mathcal{V}$ and the smaller $E_{\min}$ will bring the longer energy queue; that is, the IoT device is required a bigger energy buffer capacity.

VI. Simulation Results

In this section, the performance of the proposed DPCOEM algorithm is evaluated through Matlab simulations [38]. The parameter settings are as follows. By following the assumption in [20], the energy supply in the environment $\psi_n(t)$ is uniformly distributed. Considering the parameter setting in [36], the channel power is exponentially distributed with the mean of $q \cdot d^{-4}$, where $d = 50$ m and $q = -40$ dB represent the distance between the edge server and IoT device, and the path-loss constant, respectively. As the setting in [36], the following values are taken: $k = 10^{-16}$, $\beta = 0.002$ s, $\omega = 1$ MHz, $\sigma = 10^{-13}$ W, $p_{\max} = 0.1$ W, $f_{\max} = 1.5$ GHz, $E_{\max} = 0.06$ J, $\mathcal{H} = 100$ bits. Furthermore, $\zeta = 50$ cycles per byte, which corresponds to the workload of compressing a general medical image [39]. For simulation, constant ϕ is chosen as the lower bound of Eq. (19). In addition, we use the parameter setting provided by [36], i.e., $\psi_{\mathcal{H}} = 12$ mW, $\theta = 0.5$ and $\epsilon = 1.5$ ms except that they are varied as variable parameters in the simulation.

Further, we introduce three related methods for comparison: complete local execution with greedy energy allocation (Complete local execution), complete edge server execution with greedy energy allocation (Complete edge server execution) and 0–1 offloading with dynamic energy allocation (0–1 offloading) [36]. In the following, we just present the discussion of the case when a task arrives, i.g., $\mathcal{R}_n(t) = 1$, since both the CPU frequency and the transmission power are equal to 0 when there is no arriving task. Details about these methods are as follows:

Complete local execution: In the complete local execution, there is no offloading and the computation tasks are executed at the IoT device all the time. If $\mathcal{M}/f_u \leq \epsilon$, the computation tasks will be executed with the CPU frequency of f_u; otherwise, the task is failure and will be dropped. Here, $f_u = \min\{f_{\max}, \sqrt{\frac{\min\{E_n(t), E_{\max}\}}{k \cdot \mathcal{M}}}\}$.

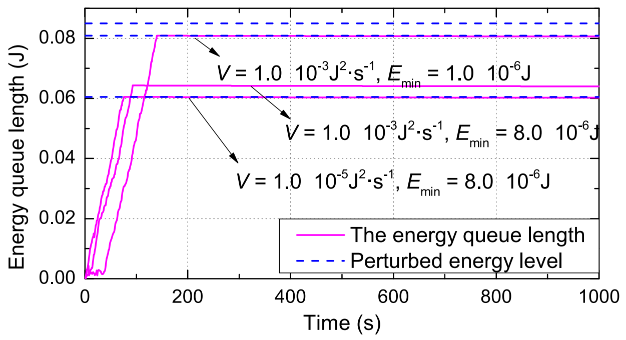
Complete edge server execution: In the complete edge server execution, there is also no choice and the computation tasks are offloaded to edge server all the time. If $\mathcal{M}/r \leq \epsilon$, the computation tasks will be transferred to the edge server with transmission power of p_u; otherwise, the task is failure and will be dropped. Here, $p_u = \min\{p_{\max}, p_{\min\{E_n(t), E_{\max}\}}\}$ if $\frac{\sigma \cdot \mathcal{H} \cdot ln2}{\omega h} < \min\{E_n(t), E_{\max}\}$, where $p_{\min\{E_n(t), E_{\max}\}}$ is the unique solution of $p \cdot \mathcal{H} = r \cdot \min\{E_n(t), E_{\max}\}$.

0–1 offloading: In the 0–1 offloading, tasks are either all executed locally or all offloaded to the edge server. If the deadline of the task could be meet, a decision of the lower task execution cost will be made; otherwise, the task is a failure and will be dropped. The CPU frequency and transmission power are computed the same as in the complete local execution and complete edge server execution methods, respectively.

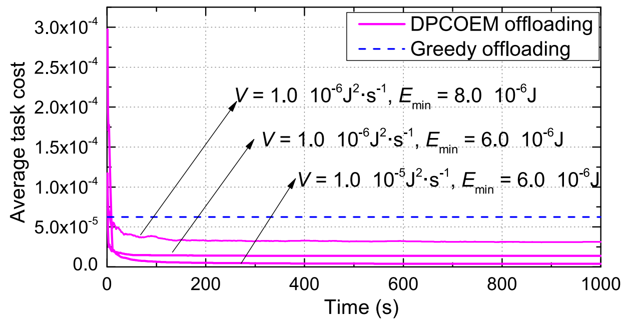
A. *Performance Analysis*

As Fig. 3(a) demonstrates, the energy queue length of the proposed DPCOEM algorithm increases at an early period and then it converges to the optimal value. The reason for this tendency is that the proposed DPCOEM algorithm includes energy management as in the formula (25). In addition, the energy queue length converges to a larger value as $\mathcal{V}$ increases or as $E_{\min}$ decreases. In Fig. 3(b), the average task cost of the proposed DPCOEM algorithm decreases as $\mathcal{V}$ increases or as $E_{\min}$ decreases while it needs more time to converge to a stable value. Moreover, the average task cost of the proposed DPCOEM algorithm is far below that of Greedy offloading. The reason is that the Greedy offloading algorithm processes all tasks locally; thus, it causes an amount of dropped tasks.

Fig. 4(a) and Fig. 4(b) reveal the relationship of the average task cost and the required battery capacity with different V, respectively. In Fig. 4(a), it seems that V has no impact on the average task cost for different methods. Actually, the proposed DPCOEM algorithm decreases as V increases and it converges to the optimal value. It corresponds to the Lyapunov drift-plus-penalty function as in formula (22). Specifically, when $\Delta_{\mathcal{V}}(t)$ is minimized, the task cost will be inversely proportional to V. Fig. 4(b) demonstrates that the required battery capacity increases proportionally with V. It also matches precisely with Eq. (22). When $\Delta_{\mathcal{V}}(t)$ is minimized, $\Delta(t)$ decreases as V increases. In other words, the energy queue should be more stable when V increases, which means that larger battery capacity

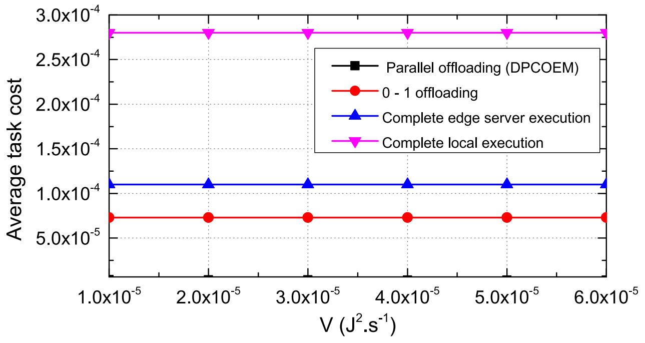


(a)

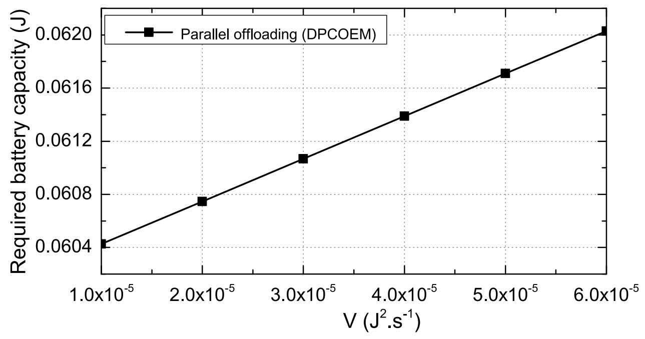


(b)

Fig. 3. The energy queue length and average task cost on different time slot when the task arrival probability is 0.5. (a) The energy queue length vs. different time slot. (b) Average task cost vs. different time slot.

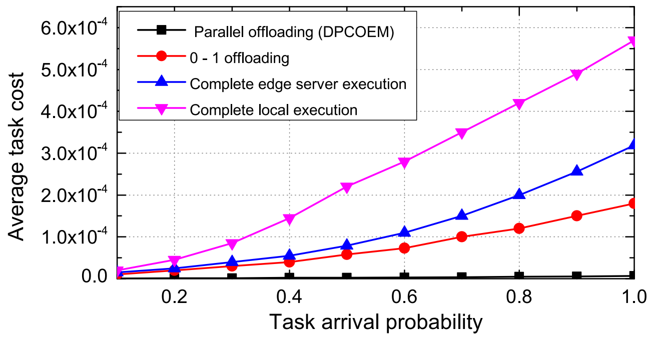


(a)

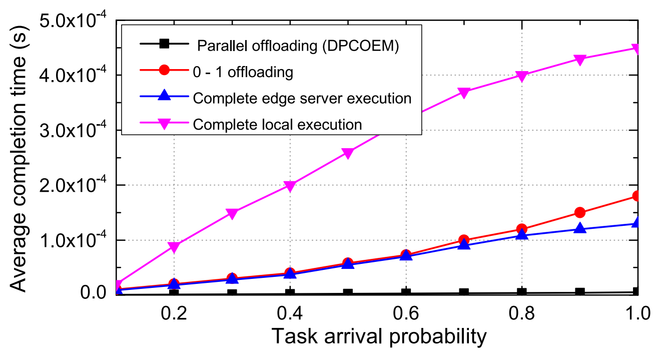


(b)

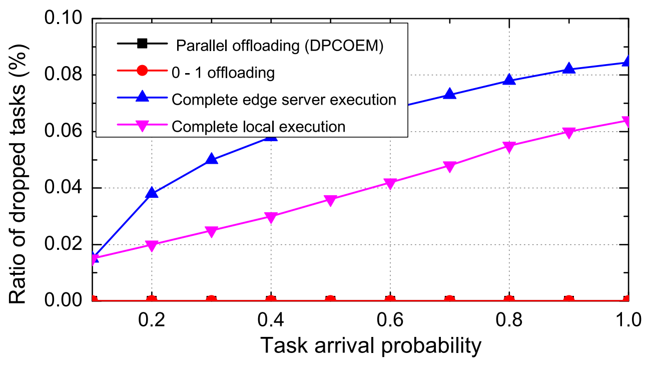
Fig. 4. Average task cost and required battery capacity on different V when the task arrival probability is 0.5 and $E_{\min} = 10^{-6}$ J. (a) Average task cost vs. different V. (b) Required battery capacity vs. different V.



(a)

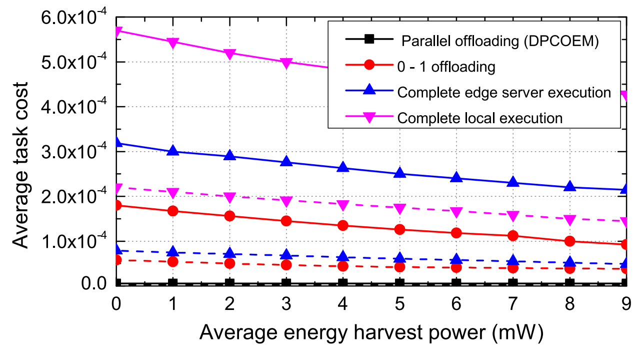


(b)

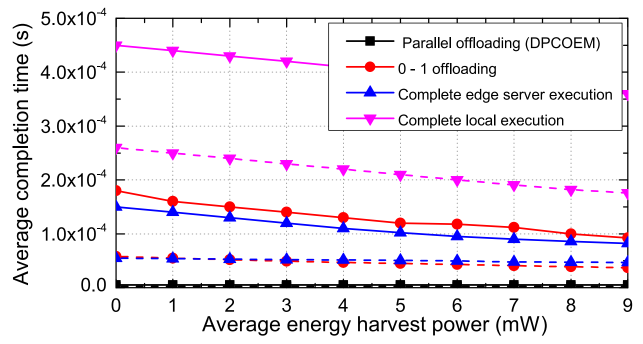


(c)

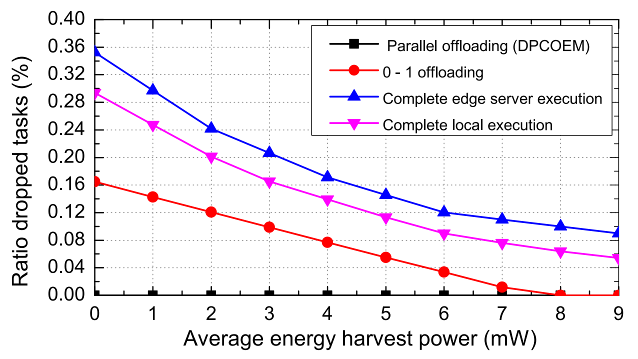
Fig. 5. Performance on different task arrival probability. (a) Average task cost vs. task arrival probability. (b) Average completion time vs. task arrival probability. (c) Ratio of dropped tasks vs. task arrival probability.



(a)



(b)



(c)

Fig. 6. Performance on different average energy harvest power. The solid line refers to task arrival probability equals to 1 and the dash line refers to task arrival probability equals to 0.5. (a) Average task cost vs. average energy harvest power. (b) Average completion time vs. average energy harvest power. (c) Ratio of dropped tasks vs. average energy harvest power.

is required to harvest more energy from the surroundings for balancing energy consumption. Thus, considering the above two aspects, an optimal V needs to be carefully chosen to achieve optimal performance.

B. Parameter Analysis

First, we show the system performance with respect to the task arrival probability from 0.1 to 1 in Fig. 5. From Fig. 5(a), the average task cost increases with the increase of task arrival probability for all the four methods. In addition, the proposed DPCOEM algorithm achieves the lowest average task cost. Fig. 5(b) presents the average completion time with different task arrival probability. It can be seen that the proposed DPCOEM algorithm has the lowest average completion time and it just increases a little with the increase of task arrival probability. The reason is that the proposed DPCOEM algorithm adopts the way of parallel offloading and makes full use of computation resources of both the IoT device and the edge server. In Fig. 5(c), it seems that both the proposed DPCOEM algorithm and the 0–1 offloading algorithm almost have the same ratio of dropped tasks. Actually, the proposed DPCOEM algorithm has a lower ratio of dropped tasks than that of 0–1 offloading algorithm since the cost of dropping tasks has been taken into consideration in the objective function.

Second, let the average energy harvest power as a variable from 0 to 9 mW and then the corresponding change of system performance can be seen from Fig. 6. Fig. 6(a) shows that the average task cost of these four methods decreases with the increase of average energy harvest power, and the proposed DPCOEM

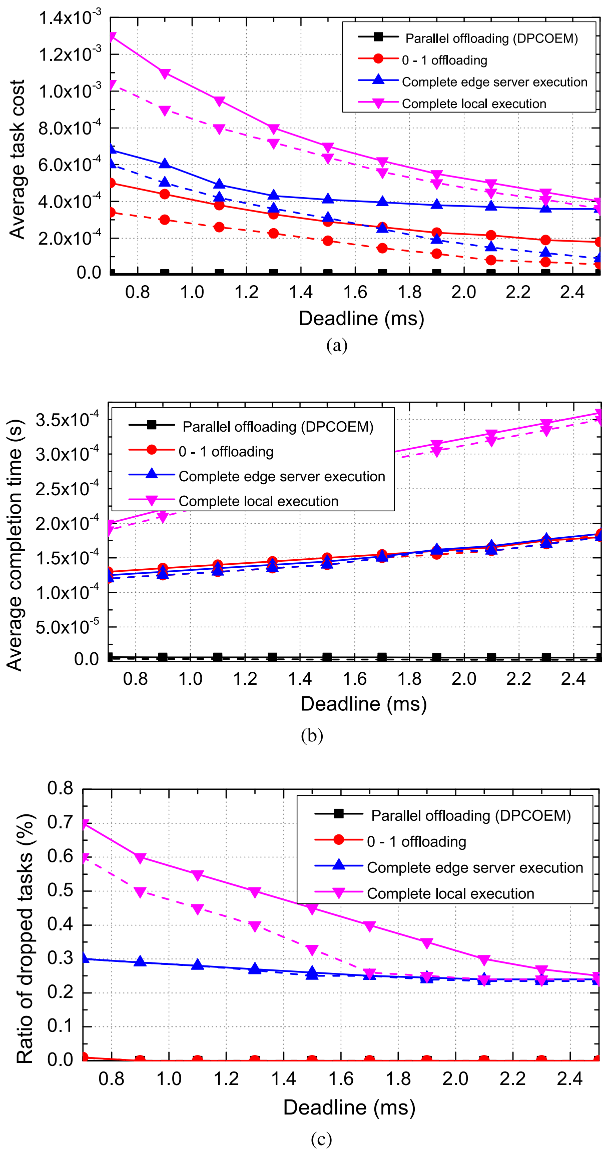


Fig. 7. Performance on different deadline. The solid line refers to task arrival probability equals to 1 and the dash line refers to task arrival probability equals to 0.5. (a) Average task cost vs. deadline. (b) Average completion time vs. deadline. (c) Ratio of dropped tasks vs. deadline.

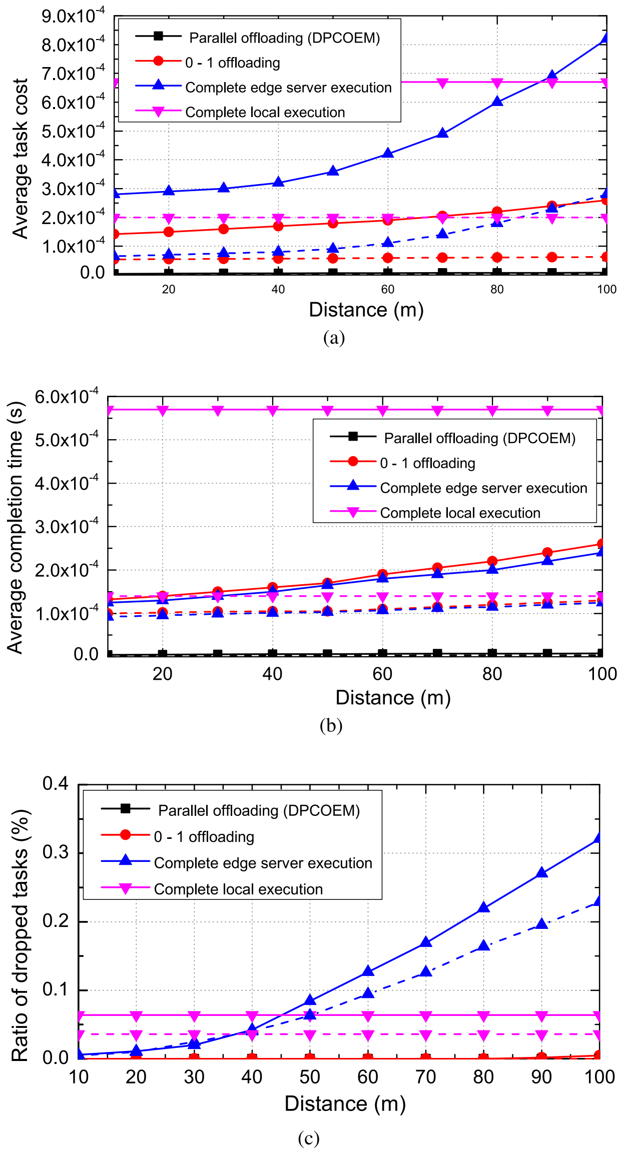


Fig. 8. Performance on different distance. The solid line refers to task arrival probability equals to 1 and the dash line refers to task arrival probability equals to 0.5. (a) Average task cost vs. distance. (b) Average completion time vs. distance. (c) Ratio of dropped tasks vs. distance.

algorithm achieves the lowest average task cost. Besides, the proposed DPCOEM algorithm is not very sensitive to the change of average energy harvest power since the edge server plays a vital role when the harvested energy is low. Fig. 6(b) and Fig. 6(c) show that the proposed DPCOEM algorithm achieves the lowest task completion time and the lowest ratio of dropped tasks, respectively.

Third, the relationship between system performance and the task deadline is shown in Fig. 7. Similar to the conclusion mentioned earlier, we find that the proposed DPCOEM algorithm achieves the lowest average cost, the lowest task completion time, and the lowest ratio of dropped tasks, as shown in Fig. 7(a), Fig. 7(b), and Fig. 7(c). As the time requirement of task processing gradually decreases then the deadline is increasing eventually, all methods will cause less cost because more tasks can be completed in time to cause less task dropping, corresponding to Eq. (12). The DPCOEM algorithm can process tasks parallelly at two places, so the delay is the smallest and the ratio of dropped tasks is the lowest, resulting in the smallest task cost. We find that the completion time increases slightly with the increase of the deadline since the offloading policy of energy saving will be selected when the delay allows, resulting in larger completion time.

Finally, Fig. 8 shows the system performance on different distances between IoT devices and edge servers. As we can see from Fig. 8(a), Fig. 8(b) and Fig. 8(c), task cost, completion time, and the ratio of dropped tasks grow as the distance increases in DPCOEM, 0–1 offloading, and complete edge server execution. The reason is that the farther distance leads to greater transmission time. Furthermore, the system performance of complete local execution does not change with the change of distance in complete local execution since it does not rely on the edge server to process tasks. The proposed DPCOEM algorithm outperforms the other three benchmark algorithms in terms of three metrics because of its flexibility to offload on local devices and edge server.

To sum up, the proposed DPCOEM algorithm outperforms the other three algorithms in all aspects considered in simulations, based on the different parameter of task arrival probability, average energy harvest power, deadline, and distance. The proposed DPCOEM algorithm utilizes a parallel processing mode, which will consume slightly more total energy than processing in a single device (i.e., 0–1 offloading, Complete edge server execution and Complete local execution). However, edge servers usually have a stable energy source (e.g., power grid). Thus, we can ignore the energy consumed by edge servers. On the other hand, the proposed DPCOEM algorithm can provide enough energy for IoT devices because the algorithm controls both the CPU frequencies and transmission powers to manage energy as in Section IV-B.

VII. Conclusion

This paper investigates emerging delay-constrained applications in the IoT system. To reduce the maintenance cost of replacing batteries for IoT devices, and to utilize the harvested energy in IoT environment, the GS-MEC framework is proposed. In this framework, we formulate the problem of minimizing response time and packet losses under the limitation of energy queue stability to improve the timeliness and reliability of task processing. To solve the formulated problem, the DPCOEM algorithm is designed. According to the algorithm, the following optimal variables of a single time slot can be determined: the energy harvested by IoT devices, the transmission power scheduled for IoT devices, the CPU frequency scheduled for IoT devices, and offloading decision vectors. With the help of theoretical analysis and simulation, we found that the system performance of the proposed DPCOEM algorithm outperforms the other three algorithms in all aspects considering the different parameter of task arrival probability, average energy harvest power, deadline, and distance. In future researches, we may extend the proposed algorithm to other scenarios (e.g., Internet of Vehicles).

Appendix

A. Proof of Lemma 2

Square the two sides of equation (11) and we can get the formula (35).

$$\begin{aligned}
&\frac{1}{2}[(\hat{E}(t+1))^2 - (\hat{E}(t))^2] \\
&= \frac{1}{2}[(E(t+1)-\phi)^2 - (E(t)-\phi)^2] \\
&\le \frac{(P^{total}(t))^2 + (e(t))^2 + 2\hat{E}(t)(e(t) - P^{total}(t))}{2} \\
&\le \frac{(\psi_{\max})^2 + (P_{\max})^2}{2} + \hat{E}(t)(e(t) - P^{total}(t)) \qquad (35)
\end{aligned}$$

Then, combine the result of the formula (35) with (12), we can proof Lemma 2 as (23).

B. Proof of Lemma 3

First, to help the proof of Lemma 3, an auxiliary problem is constructed as follows

$$\min_{\mathcal{X},\, f(t),\, p(t),\, e(t)} \quad \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} W(t)\right]$$

$$\text{s.t.} \quad \begin{cases} (1), (2), (9), (10), (15), (18). \\ x^l(t) + x^c(t) \le R(t),\ \forall\, t \in \mathcal{T}. \\ P^{total}(t) \le E_{\max},\ \forall\, t \in \mathcal{T}. \\ 0 \le p(t) \le p_{\max}\cdot 1_{\{x^c(t)=1\}},\ \forall\, t \in \mathcal{T}. \\ 0 \le f(t) \le f_{\max}\cdot 1_{\{x^l(t)=1\}},\ \forall\, t \in \mathcal{T}. \\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[P^{total}(t) - e(t)] = 0,\ \forall\, t \in \mathcal{T}. \end{cases} \qquad (36)$$

Let the optimal solution of the problem (37) be W^{Π} and we know that $W^{\Pi} \le W^*$, which can be found in Lemma 5 in [36].

According to formula (23) and the proposed DPCOEM algorithm, we have

$$\begin{aligned}
\Delta_V(t) &\le \mathbb{E}[(\hat{E}(t))[e^*(t) - P^{total}(t)] + V\cdot[W^*] + B \\
&\le \mathbb{E}[(\hat{E}(t))[e^{\Pi}(t) - P^{total}] + V\cdot[W^{\Pi}] + B \\
&= \hat{E}(t)\mathbb{E}[e^{\Pi}(t) - P^{total}] + V\cdot\mathbb{E}[W^{\Pi}] + B \\
&\le \max\{\phi, \psi_{\max}\}\cdot\varpi\cdot\eta + V(W^{\Pi} + \eta) + B. \qquad (37)
\end{aligned}$$

Next, let η tend to 0, we can get

$$\Delta_V(t) \le V\cdot W^{\Pi} + B. \qquad (38)$$

After making several mathematical transformations of (38), we can get the following inequality relationship

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{\Delta_V(t)\} \le \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{V\cdot W^{\Pi} + B\}. \qquad (39)$$

Then, divide by V on the both sides of (39), we can get the following inequality

$$W^* \le W^{\Pi} + \frac{B}{V}. \qquad (40)$$

Furthermore, considering Lemma 1 and $W^{\Pi} \le W^*$, we can get the conclusion $W^* \le W^1 + \mu(E_{\min}) + \frac{B}{V}$.

References

[1] The Network, “8 industries where IoT is working the best,” 2018. [Online]. Available: https://www.networkworld.com/article/3200783/8-industries-where-iot-is-working-the-best.html

[2] A. Bera, “80 IoT statistics (Infographic),” 2019. [Online]. Available: https://safeatlast.co/blog/iot-statistics/

[3] E. Dukes, “The cost of IoT sensors is dropping fast,” 2018. [Online]. Available: https://www.iofficecorp.com/blog/cost-of-iot-sensors

[4] P. Brown, “Is battery life hindering the growth of Internet of Things devices?,” 2018. [Online]. Available: https://www.psikick.com/battery-life-hindering-growth-internet-things-d evices/

[5] e-peas, "AEM40940-Energy harvesting power management IC," 2019. [Online]. Available: https://e-peas.com/products/energy-harvesting/rf/aem40940/
[6] A. Alnoman, S. K. Sharma, W. Ejaz, and A. Anpalagan, "Emerging edge computing technologies for distributed IoT systems," *IEEE Netw.*, to be published, doi: 10.1109/MNET.2019.1800543.
[7] M. R. Palattella *et al.*, "Internet of Things in the 5G era: Enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.
[8] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of Vehicles: Toward energy-efficient scheduling," *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep.–Oct. 2019.
[9] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 3, no. 4, pp. 1140–1151, Apr. 2019.
[10] X. Lyu *et al.*, "Selective offloading in mobile edge computing for the green Internet of Things," *IEEE Netw.*, vol. 32, no. 1, pp. 54–60, Jan. 2018.
[11] Z. Dong *et al.*, "An energy-efficient offloading framework with predictable temporal correctness," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, Oct. 2017, Art. no. 19.
[12] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 16–19, 2018, doi: 10.1109/INFOCOM.2018.8485977.
[13] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[14] "Energy harvesting: How we will build the Internet of Perpetual Things," Jabil, St. Petersburg, FL, USA, White Paper, 2019. [Online]. Available: https://www.jabil.com/content/dam/insights/white-papers/en/energy-harve sting-how-we-will-build-the-internet-of-perpetual-things
[15] B. Feng, C. Zhang, H. Ding, and Y. Fang, "Exploiting wireless broadcast advantage for energy efficient packet overhearing in WiFi," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3586–3599, Apr. 2019.
[16] H. Ding and Y. Fang, "Virtual infrastructure at traffic lights: Vehicular temporary storage assisted data transportation at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12452–12456, Dec. 2018.
[17] H. Ding, C. Zhang, B. Lorenzo, and Y. Fang, "Access point recruitment in a vehicular cognitive capability harvesting network: How much data can be uploaded?," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6438–6445, Jul. 2018.
[18] F. Ünlü, L. Wawrla, and A. Dìaz, "Energy harvesting technologies for IoT edge devices," 4E, Int. Energy Agency, Paris, France, Jul. 2018. [Online]. Available: https://www.iea-4e.org/document/417/energy-harvesting-technologies-for-iot-edge-devices
[19] J. Yuan *et al.*, "Single-junction organic solar cell with over 15 percent efficiency using fused-ring acceptor with electron-deficient core," *Joule*, vol. 3, no. 4, pp. 1140–1151, Apr. 17, 2019.
[20] D. Zhang, Z. Chen, M. K. Awad, N. Zhang, H. Zhou, and X. S. Shen, "Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3552–3565, Dec. 2016.
[21] L. Guo, Z. Chen, D. Zhang, J. Liu, and J. Pan, "Sustainability in body sensor networks with transmission scheduling and energy harvesting," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2019.2930076.
[22] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.
[23] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2016.
[24] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran, and C. S. Hong, "Collaborative cache allocation and computation offloading in mobile edge computing," in *Proc. 19th Asia-Pacific Netw. Oper. Manage. Symp.*, Sep. 2017, pp. 366–369.
[25] M. Salmani and T. N. Davidson, "Multiple access partial computational offloading: Two-user case," in *Proc. 23rd Asia-Pacific Conf. Commun.*, Dec. 2017, pp. 1–6.
[26] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[27] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 80–86, Sep. 2018.
[28] X. Sun and N. Ansari, "Edge IoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.
[29] X. Chen, Q. Shi, L. Yang, and J. Xu, "Thrifty Edge: Resource-efficient edge computing for intelligent IoT applications," *IEEE Netw.*, vol. 32, no. 1, pp. 61–65, Jan./Feb. 2018.
[30] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.
[31] T. Robertazzi, "Ten reasons to use divisible load theory," *Computer*, vol. 36, no. 5, pp. 63–68, Sep. 2003.
[32] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.
[33] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.
[34] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *Proc. Int. Conf. Commun.*, Jun. 2015, pp. 5529–5534.
[35] C. You, K. Huang and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.
[36] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[37] Y. Lin, Y. Liu, W.-N. Chen, and J. Zhang, "A hybrid differential evolution algorithm for mixed-variable optimization problems," *Inf. Sci.*, vol. 466, no. 1, pp. 170–188, Oct. 2018.
[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[39] R. Starosolski, "Simple fast and adaptive lossless image compression algorithm," *Softw. Pract. Experience*, vol. 37, no. 1, pp. 65–91, Jan. 2007.

Yiqin Deng received the B.S. degree in project management from the Hunan Institute of Engineering, Xiangtan, China, in 2014, and the M.S. degree in software engineering from Central South University, Changsha, China, in 2017, where she is currently working toward the Ph.D. degree in computer science and technology. Her research interests are primarily focused on edge/fog computing, Internet of Vehicle, smart city, and resource management. She is a member of the China Computer Federation.

Zhigang Chen received the B.S., M.S., and Ph.D. degrees from Central South University, Changsha, China in 1984, 1987, and 1998, respectively. He is currently a Professor, Ph.D. Supervisor, and Dean with the School of Software, Central South University. His research interests cover the general area of cluster computing, parallel and distributed system, computer security and wireless networks. He is a Director and Advanced Member of the China Computer Federation (CCF), and a member of the Pervasive Computing Committee of CCF.

Xin Yao received the B.S. degree in computer science from Xidian University, Xi'an, China, in 2011, and the M.S. degree in software engineering and the Ph.D. degree in computer science and technology from Hunan University, Changsha, China, in 2013 and 2018, respectively. From 2015 to 2017, he was a Visiting Scholar with Arizona State University. He is currently an Assistant Professor with Central South University, Changsha, China. His research interests include security and privacy issues in social network, Internet of Things, cloud computing, and big data. He is a member of the China Computer Federation.

Shahzad Hassan received the B.S. degree in software engineering from Riphah International University, Rawalpindi, Pakistan, in 2013, and the M.Eng. degree in software engineering from Central South University, Changsha, China, in 2016, where he is currently working toward the Ph.D. degree in computer application and technology. His research interests are primarily focused on sensors, IoT, fog computing, and edge computing.

Ali. M. A. Ibrahim received the B.Sc. and M.Sc. degrees in communication engineering from the University of Gezira, Wad Madani, Sudan, in 2012 and 2016, respectively. He is currently working toward the Ph.D. degree in computer science and engineering with Central South University, Changsha, China. He is a Lecturer with the Managil University of Science and Technology, El Manaqil, Sudan. His research interests are wireless networks, massive MIMO, D2D communication, mobile social networks, mobile edge computing, and fog RAN.