Energy-latency Trade-off for Energy-aware Offloading in Mobile Edge Computing Networks

Jiao Zhang, Xiping Hu, Zhaolong Ning, Edith C.-H. Ngai, Li Zhou, Jibo Wei, Jun Cheng, Bin Hu

***Abstract*—Mobile edge computing (MEC) brings computation capacity to the edge of mobile networks in close proximity to smart mobile devices and contributes to energy saving compared with local computing, but resulting in increased network load and transmission latency. To investigate the trade-off between energy consumption and latency, we present an energy-aware offloading scheme, which jointly optimizes communication and computation resource allocation under the limited energy and sensitive latency. In this paper, single and multi-cell MEC network scenarios are considered at the same time. The residual energy of smart devices' battery is introduced into the definition of the weighting factor of energy consumption and latency. In terms of the mixed integer nonlinear problem (MINLP) for computation offloading and resource allocation, we propose an iterative search algorithm combining interior penalty function with D.C. (the difference of two convex functions/sets) programming (IPDC) to find the optimal solution. Numerical results show that the proposed algorithm can obtain lower total cost (i.e., the weighted sum of energy consumption and execution latency) comparing with the baseline algorithms, and the energy-aware weighting factor is of great significance to maintain the lifetime of smart mobile devices.**

***Index Terms*—Mobile edge computing; Energy-aware offloading; Resource allocation**

I. Introduction

Smart mobile devices (SMDs) are attracting enormous popularity with the emerging of mobile technologies like Internet of Things (IoTs) and wearable devices, which can provide a powerful platform to support some novel mobile applications (e.g. interaction gaming, face recognition and natural language processing [1]–[3]). Such computing-intensive applications require higher computing capacity and more energy than traditional applications on SMDs [4]. In general, SMDs have limited computation resources (e.g., central process unit (CPU) frequency and memory) and battery lifetime, bringing in unprecedented challenge to effectively execute these mobile applications [5]–[7]. Since the cloud sever has higher computation capacity and storage than the SMD, mobile cloud computing (MCC) is envisioned as a potential approach to react the challenge via migrating computations from the SMD to the cloud sever [8], which is referred to as computation offloading. However, the cloud severs are spatially far from SMDs, which causes high transmission latency and detains the latency-sensitive applications.

Mobile edge computing [9], [10], as a new architecture and key technology for 5G networks, relocates the cloud computation resource close to SMDs. Compared with MCC, MEC can provide lower latency and computing agility in computation offloading. However, considering the economic and scalable deployment, the computation capacity of the MEC server is limited. In addition, the computation offloading especially in the ultra dense networks (UDN) causes more interference and results in unexpected transmission delay [11]. Thus, it is impossible to offload all computation tasks to the MEC sever, and some of them should be executed on SMDs (i.e., local computing). Although local computing consumes more energy, it can significantly minimize the execution latency without additional communication or waiting delay. Thus, it is critical to make efficient offloading decision and investigate the trade-off between energy consumption of SMDs and execution latency of the corresponding tasks.

In this paper, we propose an energy-aware offloading scheme to investigate the trade-off between energy consumption of SMDs and latency of their tasks. The motivations behind our work are attributed to the following observations: (1) With the finite computation resources in MEC server and severe interference among networks, all tasks can not be offloaded to the MEC server.. The computation offloading decision should be rationally determined. (2) Both the energy consumption and latency are of great significance for SMDs. The energy consumption and latency depend mainly on the transmission power and communication channel when offloading the task to the MEC server. However, they depend mainly on the CPU-cycle frequency when the task is locally computed. (3) According to the service condition of the battery and user-specific demands, the user preference (i.e., the weighting factor) should be defined to allow SMDs to choose different optimal objectives. Therefore, based on the above considerations, our scheme jointly optimizes computation offloading and resource allocation to make a trade-off between energy consumption and latency considering the limited battery lifetime and latency sensitive tasks. The main contributions of this paper are as follows.

- We present an integrated framework for computation offloading and resource allocation in MEC networks, where both single and multi-cell network scenarios are

J. Zhang, L. Zhou and J. Wei are with the College of Electronic Science and Engineering, National University of Defense Technology, Changsha, China. E-mail: {zhangjiao16,zhouli2035,wjbhw}@nudt.edu.cn.

J. Zhang, X. Hu (Co-corresponding author) and J. Cheng are with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. E-mail: {xp.hu,jun.cheng}@siat.ac.cn.

Z. Ning (Co-corresponding author) is with the School of Software, Dalian University of Technology, Dalian, China. E-mail: zhaolongning@dlut.edu.cn.

E. Ngai is with the Department of Information Technology, Uppsala University, Uppsala, Sweden. Email: edith.ngai@it.uu.se.

B. Hu (Co-corresponding author) is with School of Information Science and Engineering, Lanzhou University, China. Email: bh@lzu.edu.cn.

considered.

- From the perspective of SMDs, we propose an energy-aware offloading scheme that associates with computation and communication resource allocation to minimize the weighted sum of energy consumption and latency.
- Compared with the weighting factor of energy consumption and execution latency subjectively defined by users, we introduce the residual energy of SMDs into the definition of the weighting factor, which takes the service condition of the battery into account and contributes to maintain the lifetime of SMDs.
- We propose an iterative search algorithm combining the interior penalty function method with D.C. programming to obtain optimal offloading decision and resource allocation, which optimizes local computing frequency scheduling, channel allocation, power allocation and computation offloading in a distribution method.

The remainder of the paper is organized as follows. In section II, we review the related works. The system model and problem formulation are introduced in Section III. Section IV and Section V present the optimal computation offloading and resource allocation schemes for single and multi-cell MEC networks, respectively. Simulation results are discussed in Section VI and Section VII concludes the paper.

II. Related Works

The computation offloading and resource allocation are of great importance in MEC networks, which have been attracting more attention in recent years [12], [13]. The energy consumption [6], [14]–[18] and latency [19]–[21] are usually considered as the criterions for performance evaluation. The authors in [15] investigated the power minimization problem for SMDs, where backhaul capacity limitation, interference level and tolerable delay are taken into account. Sardellitti et al. in [16] jointly optimized the users' transmission precoding matrices and the cloud's computational resource to minimize the SMDs' energy consumption. Munoz et al. in [17] minimized the SMD's energy consumption by jointly optimizing the transmission time and the amount of data offloaded to a femto access point (AP). The authors in [18] formulated the computation offloading as a constrained Markov decision process, which aimed to minimize the energy consumption at the user equipment (UE) while satisfying the execution delay of the applications. In [19], the authors proposed a low-complexity Lyapunov optimization-based dynamic computation offloading algorithm to reduce execution time. In [20], Yang et al designed a heuristic method to partition the users' computation tasks so as to minimize the average completion time of all the users. In [21], Ni et al. proposed a resource allocation strategy based on priced timed Petri nets, which considered the price cost, time cost and the credibility evaluation of both users and fog resources. Furthermore, Wang et al. in [22] investigated the energy consumption minimization and execution latency minimization respectively by jointly optimizing the computation speed and transmission power of SMDs.

There are some papers focusing on the trade-off between energy consumption and execution latency. The authors in [23] aimed to minimize both total tasks' execution latency and the SMD's energy consumption by jointly optimizing the task offloading decision and the SMD's CPU-cycle frequency, which considered the fixed CPU frequency and elastic CPU frequency. In [24], the authors formulated the computation offloading decision, physical resource block allocation, and MEC computation resource allocation as the optimization problem to minimize the overall consumption of the entire system, in terms of time and energy. In [25], Lin et al. developed a Ternary Decision Maker offloading framework to shorten response time and reduce energy consumption at the same time. In [11], Deng et al. proposed an adaptive sequential offloading game approach to offload decision, and the optimal objective function is the weighted sum of energy and computational time. In order to investigate the trade-off between energy consumption and execution latency, Hong et al. in [26] formulated the problem of data offloading scheduling as a dynamic programming problem. They adopted a weighting factor to define the weighted sum of energy consumption and latency as the research objective in these works. However, most of the existing studies do not specify the definition of the weighting factor of the energy consumption and execution latency.

Different from these studies, this paper proposes an energy-aware offloading scheme to exploit the trade-off between energy consumption of SMDs and execution latency of their tasks by jointly optimizing CPU-cycle frequency, transmission power and channel resource allocation, where the weighting factor is specifically defined based on the residual energy of SMD battery.

III. system model and problem formulations

A. System Model

A 5G heterogeneous MEC network with one macro cell and M small cells is considered, as shown in Fig. 1. The macro eNodeB (MeNB) is equipped with an MEC server which is capable of executing multiple computing-intensive tasks, and small cell eNodeBs (SeNBs) are overlaid by the MeNB. The SeNBs are connected to the MeNB through wired link [27]. Each SeNB serves up to $U_j (j \in \{1, 2, ..., M\})$ SMDs. In order to reuse spectrum, we assume that multiple SeNBs operate in the same frequency band, where the interference between small cells exists. The bandwidth B is divided into N channels. SMDs associate with the SeNBs in Orthogonal Frequency-Division Multiple Access (OFDMA), where the channel of each SMD in the same SeNB is orthogonal to others. In this network, SMD i in SeNB j has a computation task $\tau_{i,j} = \{d_{i,j}, c_{i,j}, T_{i,j}^{\max}\}$ to be completed, where $d_{i,j}$ is the size of input data, $c_{i,j}$ denotes the total number of CPU cycles required to accomplish the computation task, and $T_{i,j}^{\max}$ denotes the maximum tolerance latency. For each SMD, its task can be executed either locally on itself or remotely in the MEC server via computation offloading. Let $s_{i,j}$ be the offloading decision of SMD i in cell j. If the SMD offloads its task to the MEC server, $s_{i,j} = 1$, otherwise, $s_{i,j} = 0$.

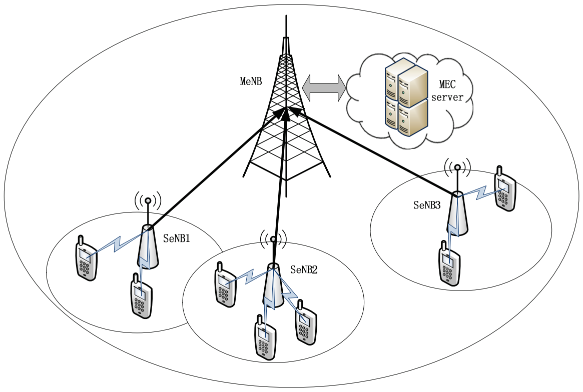


Fig. 1: Network model.

(1) Local computing

We define $f_{i,j}^{l}$ as the computation ability (i.e., CPU cycles per second) of the SMD. When the task $\tau_{i,j}$ is executed locally on the SMD, the computation execution time $t_{i,j}^{L}$ is

$$t_{i,j}^{L} = \frac{c_{i,j}}{f_{i,j}^{l}}. \tag{1}$$

The energy consumption of the SMD can be calculated as

$$e_{i,j}^{L} = \kappa (f_{i,j}^{l})^2 c_{i,j}, \tag{2}$$

where $\kappa = 10^{-26}$, and it is a coefficient depending on the chip architecture [28], [29].

Considering that $f_{i,j}^{l}$ affects the computation execution time and energy consumptions simultaneously. It is allowed to schedule the CPU-cycle frequency via dynamic voltage and frequency scaling (DVS) technology [22].

(2) Edge computing

When the input data is transmitted to the MEC server through SeNB, the transmission expenditure between the MEC sever and the SeNB is ignored [24], [30]. If the SMD accesses the SeNB on channel n, the achievable uplink transmission rate can be expressed as

$$r_{i,j,n} = w\log_2(1 + \frac{p_{i,j,n} h_{i,j,n}}{\sigma^2 + I_{i,j,n}}), \tag{3}$$

where w is the bandwidth, $w = B/N$. $p_{i,j,n}$ and $h_{i,j,n}$ are the transmission power and the channel gain between the SMD i and the SeNB j on channel n, respectively. σ^2 is the noise power. $I_{i,j,n}$ denotes the interference of SMD i in cell j suffering from other SMDs in neighboring cells on the same channel n, and it can be represented as

$$I_{i,j,n} = \sum_{k=1}^{U_l} \sum_{l=1, l \neq j}^{M} a_{k,l,n} p_{k,l,n} h_{k,l,n}^{j}, \tag{4}$$

where l is the lth except the jth small cell, $h_{k,l,n}^{j}$ is the channel gain from SMD k in cell l to cell j on channel n, and U_l is the number of SMDs in small cell l. Hence, the total uplink transmission rate for SMD i in cell j can be obtained as

$$r_{i,j} = \sum_{n=1}^{N} a_{i,j,n} r_{i,j,n}, \tag{5}$$

where $a_{i,j,n} \in \{0,1\}$. $a_{i,j,n} = 1$ denotes the channel n is assigned to SMD i in cell j to offload its task, otherwise, $a_{i,j,n} = 0$. Let f^{C} denotes the CPU-cycle frequency of the MEC server, which is fixed for the duration of computation task execution [17], [22]. Then the total edge computing execution time of the task includes the transmission time and computation time on MEC server, which can be given as

$$t_{i,j}^{C} = \frac{d_{i,j}}{r_{i,j}} + \frac{c_{i,j}}{f^{C}}. \tag{6}$$

The energy consumption of the SMD is expressed as

$$e_{i,j}^{C} = \sum_{n=1}^{N} a_{i,j,n} p_{i,j,n} \frac{d_{i,j}}{r_{i,j}}. \tag{7}$$

Here, the time and energy consumption of outcome from the MEC server to the SMD are ignored in this case, due to the fact that the size of outcome data is much smaller than the size of input data, which is similar to the studies [30], [31].

In the process of task execution, both execution latency and energy consumption are vital for the SMDs, which attributes to user experience and the battery energy limitation of SMDs. In general, a weighting factor $w_{i,j}$ ($w_{i,j} \in [0,1]$) will be used to investigate the trade-off between energy consumption and latency, which can be defined by different SMDs for the purpose of meeting the user-specific demands [8], [24]. Saving more energy or reducing latency can be implemented by adjusting the weighting factor. However, we bring the residual energy rate $r_{i,j}^{E}$ of the battery into the weighting factor in our model. It is defined as

$$w'_{i,j} = w_{i,j} r_{i,j}^{E}, \tag{8}$$

where $r_{i,j}^{E} = E_{i,j}^{\max}/E^{total}$. $E_{i,j}^{\max}$ is the maximum residual energy aware of the battery of SMD i in cell j and E^{total} is the battery capacity in Joules [26]. Unlike $w_{i,j}$, $r_{i,j}^{E}$ is a definite value, which reflects the real-time service condition of the battery.

According to (1) and (2), the overhead of the task locally computed on SMD i in cell j, namely the weighted sum of energy consumption and latency $G_{i,j}^{L}$, can be defined as

$$G_{i,j}^{L} = w'_{i,j} t_{i,j}^{L} + (1 - w'_{i,j}) \alpha e_{i,j}^{L}, \tag{9}$$

where α is the normalizing factor, which is introduced to implement the unitless combination of energy consumption and latency. The normalizing factor can be defined as the ratio of average latency of all tasks to average energy consumption of all SMDs. Let $w_{i,j}^{t} = w'_{i,j}$ and $w_{i,j}^{e} = (1 - w'_{i,j})\alpha$, then they denote the weightings of execution latency and energy consumption, respectively. Hence, the equation (9) can be represented as $G_{i,j}^{L} = w_{i,j}^{t} t_{i,j}^{L} + w_{i,j}^{e} e_{i,j}^{L}$.

Similarly, the overhead of the task computed in the MEC server can be calculated as

$$G_{i,j}^{C} = w_{i,j}^{t} t_{i,j}^{C} + w_{i,j}^{e} e_{i,j}^{C}. \tag{10}$$

Therefore, the overhead of the SMD i in cell j can be obtained by

$$G_{i,j} = s_{i,j} G_{i,j}^{C} + (1 - s_{i,j}) G_{i,j}^{L}. \tag{11}$$

B. Problem Formulation

As mentioned above, we formulate the problem in two scenarios: single and multi-cell in this section.

(1) Trade-off between energy and latency for single cell

We firstly investigate the trade-off between energy and latency for single cell. In single cell (i.e., $M=1$), SMDs do not experience the interference from other cells. Assume that the number of channels is enough and channel allocation is also ignored. Therefore, the optimization problem is formulated as follows.

$$\begin{array}{lll} P1: & \min\limits_{\mathbf{s},\mathbf{p},\mathbf{f}} & \sum\limits_{i=1}^{U_1}\{s_i[w_i^t(\frac{d_i}{r_i}+\frac{c_i}{f^C})+w_i^e p_i\frac{d_i}{r_i}]\} \\ & + & \sum\limits_{i=1}^{U_1}\{(1-s_i)[w_i^t\frac{c_i}{f_i^l}+w_i^e\kappa(f_i^l)^2 c_i]\} \\ s.t. & C1: & s_i(\frac{d_i}{r_i}+\frac{c_i}{f^C})+(1-s_i)\frac{c_i}{f_i^l}\leqslant T_i^{\max},\forall i \\ & C2: & s_i p_i\frac{d_i}{r_i}+(1-s_i)\kappa(f_i^l)^2 c_i\leqslant E_i^{\max} \\ & C3: & f_{\min}^L\leqslant f_i^l\leqslant f_{\max}^L \\ & C4: & 0\leqslant p_i\leqslant p_{\max} \\ & C5: & s_i\in\{0,1\} \end{array} \tag{12}$$

The constraint C1 is the maximum tolerance latency to execute the SMD's task. C2 ensures that the energy consumption should not exceed the residual energy of the SMD. C3 restricts the local CPU-cycle frequency into a finite set of values. C4 guarantees the maximum transmission power. C5 denotes offloading decision as the binary variables.

(2) Trade-off between energy and latency for multi-cell

In multi-cell network, interference management and channel allocation are considered as well. The minimization overhead of SMDs problem is formulated as follows.

$$\begin{array}{lll} P2: & \min\limits_{\mathbf{s},\mathbf{p},\mathbf{a},\mathbf{f}} & \sum\limits_{i=1}^{U_j}\sum\limits_{j=1}^{M}\{s_{i,j}[w_{i,j}^t(\frac{d_{i,j}}{r_{i,j}}+\frac{c_{i,j}}{f^C})+w_{i,j}^e\sum\limits_{n=1}^{N}a_{i,j,n}\cdot \\ \multicolumn{3}{l}{p_{i,j,n}\frac{d_{i,j}}{r_{i,j}}]\}+\sum\limits_{i=1}^{U_j}\sum\limits_{j=1}^{M}\{(1-s_{i,j})[w_{i,j}^t\frac{c_{i,j}}{f_{i,j}^l}+w_{i,j}^e\kappa(f_{i,j}^l)^2 c_{i,j}]\}} \\ s.t. & C1: & s_{i,j}(\frac{d_{i,j}}{r_{i,j}}+\frac{c_{i,j}}{f^C})+(1-s_{i,j})\frac{c_{i,j}}{f_{i,j}^l}\leqslant T_{i,j}^{\max},\forall i,j \\ & C2: & s_{i,j}\sum\limits_{n=1}^{N}a_{i,j,n}p_{i,j,n}\frac{d_{i,j}}{r_{i,j}}+(1-s_{i,j})\kappa(f_{i,j}^l)^2 c_{i,j} \\ & & \leqslant E_{i,j}^{\max},\forall i,j \\ & C3: & f_{\min}^L\leqslant f_{i,j}^l\leqslant f_{\max}^L,\forall i,j \\ & C4: & 0\leqslant\sum\limits_{n=1}^{N}a_{i,j,n}p_{i,j,n}\leqslant p_{\max},\forall i,j \\ & C5: & \sum\limits_{k=1}^{U_l}\sum\limits_{l=1,l\neq j}^{M}a_{k,l,n}p_{k,l,n}h_{k,l,n}\leqslant I_{th},\forall j,n \\ & C6: & \sum\limits_{n=1}^{N}a_{i,j,n}\leqslant 1,\forall i,j \\ & C7: & a_{i,j,n}\in\{0,1\},\forall i,j,n \\ & C8: & s_{i,j}\in\{0,1\},\forall i,j \end{array} \tag{13}$$

where C1 and C2 are the execution latency and energy consumption constraints, respectively. C3 restricts the local CPU-cycle frequency and C4 guarantees the maximum transmission power on SMD i in cell j. C5 addresses that the interference on SeNB j caused by SMDs with offloading task in other cells on each channel does not exceed the predefined threshold. C6 indicates that each SMD can only be allocated to at most one channel. C7 and C8 state that the binary variables are used to represent channel allocation and offloading decision. For $\sum\limits_{n=1}^{N}a_{i,j,n}>0$, $s_{i,j}=1$ holds.

With the nonconvexity of objective function and constraints, $P2$ is a nonconvex problem and belongs to a mixed integer nonlinear problem, in which the optimal solution is intractable.

IV. OPTIMAL COMPUTATION OFFLOADING AND RESOURCE ALLOCATION SCHEME FOR SINGLE MEC NETWORK

In this section, we focus on solving the problem $P1$. We propose the optimal computation offloading and resource allocation scheme for single MEC network. In problem $P1$, it can be seen that the association objective function can be separated into two independent parts: local overhead and edge overhead. Hence, $P1$ can be transformed to $P1.1$ and $P1.2$ shown as follows.

A. Optimal local computing via CPU-cycle frequency scheduling

In order to minimize the local overhead, we can schedule the CPU-cycle frequency of the SMD. Considering the constraints of C1, C2 and C3 in problem $P1$, $P1.1$ can be formed as follows.

$$\begin{array}{lll} P1.1: & \min\limits_{\mathbf{f}} & w_i^t\frac{c_i}{f_i^l}+w_i^e\kappa(f_i^l)^2 c_i \\ s.t. & C1: & \frac{c_i}{f_i^l}\leqslant T_i^{\max} \\ & C2: & \kappa(f_i^l)^2 c_i\leqslant E_i^{\max} \\ & C3: & f_{\min}^L\leqslant f_i^l\leqslant f_{\max}^L \end{array} \tag{14}$$

Let $G_i^L(f_i^l)=w_i^t\frac{c_i}{f_i^l}+w_i^e\kappa(f_i^l)^2 c_i$, we can see that the value of $G_i^L(f_i^l)$ depends only on f_i^l. We take the derivative of $G_i^L(f_i^l)$ with respect to f_i^l, and set it to be zero. Then we can obtain

$$f^*=\sqrt[3]{\frac{w_i^t}{2w_i^e\kappa}}. \tag{15}$$

For $f_i^l>f^*$, $G_i^L(f_i^l)$ monotonously increases with the increase of f_i^l. Otherwise, it monotonously decreases with the increase of f_i^l.

Besides, according to C1 and C2, we have

$$\begin{array}{l} f_i^l\geqslant\frac{c_i}{T_i^{\max}}=f_l \\ f_i^l\leqslant\sqrt{\frac{E_i^{\max}}{\kappa c_i}}=f_h \end{array}. \tag{16}$$

Combining (16) with the constraint C3, we define

$$\begin{array}{l} f_l'=\max\{f_{\min}^L,\ f_l\} \\ f_h'=\min\{f_{\max}^L,f_h\} \end{array}. \tag{17}$$

To ensure the feasible region of f_i^l to be non-empty, $f_l' \leqslant f_h'$ should hold. Hence, the optimal $G_i^L(f_i^l)$ can be calculated as

$$G_i^{L*}(f_i^l) = \begin{cases} G_i^L(f_l') & f^* \leqslant f_l' \\ G_i^L(f^*) & f_l' < f^* \leqslant f_h' \\ G_i^L(f_h') & f^* > f_h' \end{cases} \tag{18}$$

B. Optimal edge computing via power allocation

The overhead of SMDs can be deduced when their tasks are offloaded to the MEC sever. Considering the constraints of C1, C2 and C4 of problem $P1$, $P1.2$ can be formed as follows.

$$\begin{array}{ll} P1.2: & \min\limits_{\mathbf{p}} w_i^t(\frac{d_i}{r_i} + \frac{c_i}{f^C}) + w_i^e p_i \frac{d_i}{r_i} \\ s.t. \quad C1: & \frac{d_i}{r_i} + \frac{c_i}{f^C} \leqslant T_i^{\max} \\ C2: & p_i \frac{d_i}{r_i} \leqslant E_i^{\max} \\ C4: & 0 \leqslant p_i \leqslant p_{\max} \end{array} . \tag{19}$$

Let $G_i^C(p_i) = w_i^t(\frac{d_i}{r_i} + \frac{c_i}{f^C}) + w_i^e p_i \frac{d_i}{r_i}$, it can be seen that the value of $G_i^C(p_i)$ is only decided by p_i, which can be expanded as

$$\begin{aligned} G_i^C(p_i) &= \frac{d_i(w_i^e p_i + w_i^t)}{w\log_2(1 + \frac{p_i h_i}{\sigma^2})} + \frac{w_i^t c_i}{f^C} \\ &= g_i^C(p_i) + \frac{w_i^t c_i}{f^C} \end{aligned} . \tag{20}$$

We observe that $G_i^C(p_i)$ becomes optimal when $g_i^C(p_i)$ gets to its minimum.

Lemma 1: The function $g_i^C(p_i)$ is unimodal.

Proof: It can be proved referring to [22].

According to Lemma 1, if $g_i^C(p_i)$ has a minimum, the minimum is unique.

Taking C1 into consideration, we can obtain

$$p_i \geqslant \frac{\sigma^2}{h_i}(2^{\frac{d_i}{w(T_i^{\max} - c_i/f^C)}} - 1) = p_0. \tag{21}$$

Expanding constraint C2, we can see that the left term in (22) satisfies the Lemma 1 as well.

$$\frac{p_i}{w\log_2(1 + \frac{p_i h_i}{\sigma^2})} \leqslant \frac{E_i^{\max}}{d_i}. \tag{22}$$

Hence, the solution of (22) is a feasible region $[p_l, p_h]$ of p_i. When the equality holds up, the solutions of (22) involve with the Lambert-W function.

Similar to local computing, we define p_l' and p_h' as $\max\{p_0, p_l\}$ and $\min\{p_{\max}, p_h\}$ respectively, then we can acquire the global optimal solution p_i^* and the minimum edge overhead $G_i^{C*}(p_i)$.

$$G_i^{C*}(p_i) = \begin{cases} G_i^C(p_l') & p^* \leqslant p_l' \\ G_i^C(p^*) & p_l' < p^* \leqslant p_h' \\ G_i^C(p_h') & p^* > p_h' \end{cases} \tag{23}$$

where p^* denotes the transmission power when $\nabla G_i^C(p_i)|_{p_i = p^*} = 0$.

C. Offloading decision

In contrast, the SMD prefers to use the computing method with minimum overload to execute its task. The offloading decision is made on comparison of the local and edge computation overhead as follows.

$$s_i = \begin{cases} 1, & G_i^L > G_i^C \\ 0, & G_i^L \leqslant G_i^C \end{cases} \tag{24}$$

When the overhead of each SMD is minimum, the optimal total overhead G^* of the network achieves optimum.

$$G^* = \sum_{i=1}^{U_1} \left\{ s_i G_i^{C*} + (1 - s_i) G_i^{L*} \right\}. \tag{25}$$

Thereafter, the optimal computation offloading and resource allocation for single MEC network is given in Algorithm 1.

Algorithm 1: Optimal computation offloading and resource allocation for single MEC network

```
Input: the set of task $\tau$.
Output: offloading decision $\mathbf{s}$ and the total overhead of SMDs.
Calculate the local CPU-cycle frequency $f^*$, $f_l$ and $f_h$.
Determine the optimal overhead of local computing according to (18).
Calculate the transmission power $p^*$, $p_l$ and $p_h$.
Determine the optimal overhead of edge computing according to (23).
if $G_i^{C*}(p_i) < G_i^{L*}(f_i^l)$ then
    $s_i = 1$.
    else
    $s_i = 0$.
end
Calculate the total overhead of SMDs by (25).
```

V. Optimal computation offloading and resource allocation scheme for multi-cell MEC networks

In multi-cell scenario, we aim to jointly consider local CPU-cycle frequency scheduling, power and channel allocation, interference management and computation offloading to minimize the weighted sum of energy consumption and execution latency of the SMD. The problem is an intractable MINLP due to two binary variables, large amount of variables, the existence of interference term and the product among different variables. Therefore, the problem is non-convex and NP-hard, and we propose an iterative search algorithm combining interior penalty function method with D.C. programming (IPDC) to find the optimal computation offloading and resource allocation scheme. The proposed algorithm comprises three parts: One is to find the optimal local computing overhead (i.e., $P2.1$), another is to find the optimal channel allocation, and the last is to implement optimal power allocation and computation offloading.

A. Optimal local computing

Considering the constraints of C1, C2 and C3 in problem $P2$, $P2.1$ can be formed and the local overhead of the SMD is only related to $f_{i,j}^{l}$.

$$\begin{aligned} P2.1: \quad & \min_{\mathbf{f}} \; w_{i,j}^{t}\frac{c_{i,j}}{f_{i,j}^{l}} + w_{i,j}^{e}\kappa(f_{i,j}^{l})^{2}c_{i,j} \\ s.t. \quad & C1: \quad \tfrac{c_{i,j}}{f_{i,j}^{l}} \leqslant T_{i,j}^{\max} \\ & C2: \quad \kappa(f_{i,j}^{l})^{2}c_{i,j} \leqslant E_{i,j}^{\max} \\ & C3: \quad f_{\min}^{L} \leqslant f_{i,j}^{l} \leqslant f_{\max}^{L} \end{aligned} \tag{26}$$

We define $G_{i,j}^{L}(f_{i,j}^{l}) = w_{i,j}^{t}\frac{c_{i,j}}{f_{i,j}^{l}} + w_{i,j}^{e}\kappa(f_{i,j}^{l})^{2}c_{i,j}$. The value of f^{*}, f_{l}' and f_{h}' can be represented as $f^{*} = \sqrt[3]{\frac{w_{i,j}^{t}}{2w_{i,j}^{e}\kappa}}$, $f_{l}' = \max\{ f_{\min}^{L}, \frac{c_{i,j}}{T_{i,j}^{\max}}\}$ and $f_{h}' = \min\{f_{\max}^{L}, \sqrt{\frac{E_{i,j}^{\max}}{\kappa c_{i,j}}}\}$ respectively. Thus, the optimal local computing overhead is defined as

$$G_{i,j}^{L*}(f_{i,j}^{l}) = \begin{cases} G_{i,j}^{L}(f_{l}') & f^{*} \leqslant f_{l}' \\ G_{i,j}^{L}(f^{*}) & f_{l}' < f^{*} \leqslant f_{h}' \\ G_{i,j}^{L}(f_{h}') & f^{*} > f_{h}' \end{cases} \tag{27}$$

After obtaining $G_{i,j}^{L*}(f_{i,j}^{l})$, the $P2$ can be transformed into the form of (28) as follows.

$$\begin{aligned} P2: \quad & \min_{\mathbf{s},\mathbf{p},\mathbf{a}} \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{s_{i,j}[w_{i,j}^{t}(\frac{d_{i,j}}{r_{i,j}} + \frac{c_{i,j}}{f^{C}}) + w_{i,j}^{e}\cdot \\ & \sum_{n=1}^{N} a_{i,j,n}p_{i,j,n}\frac{d_{i,j}}{r_{i,j}}]\} + \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{(1-s_{i,j})G_{i,j}^{L*}(f_{i,j}^{l})\} \\ & s.t. \quad C1, C2, C4, C5, C6, C7, C8. \end{aligned} \tag{28}$$

However, (28) is still a MINLP and it is NP hard to find its optimal solution. Next, we propose a suboptimal iterative method to solve it, which separates the channel allocation from power allocation and offloading decision.

B. Channel allocation

To minimize the edge computing overhead, the channel quality of offloading tasks should be ensured. Let $G_{i,j}^{C}(\mathbf{p}) = w_{i,j}^{t}(\frac{d_{i,j}}{r_{i,j}} + \frac{c_{i,j}}{f^{C}}) + w_{i,j}^{e}\sum_{n=1}^{N} a_{i,j,n}p_{i,j,n}\frac{d_{i,j}}{r_{i,j}}$, it can be expanded as

$$\begin{aligned} G_{i,j}^{C}(\mathbf{p}) &= \sum_{n=1}^{N}\frac{a_{i,j,n}d_{i,j}(w_{i,j}^{e}p_{i,j,n} + w_{i,j}^{t})}{w\log_2(1 + \frac{p_{i,j,n}h_{i,j,n}}{\sigma^{2}+I_{i,j,n}})} + \frac{w_{i,j}^{t}c_{i,j}}{f^{C}} \\ &= \sum_{n=1}^{N} a_{i,j,n}g_{i,j,n}^{C}(p_{i,j,n}, EI_{i,j,n}) + \frac{w_{i,j}^{t}c_{i,j}}{f^{C}} \end{aligned} \tag{29}$$

where $EI_{i,j,n} = \frac{h_{i,j,n}}{\sigma^{2}+I_{i,j,n}}$, which can be defined as the effective interference for the SMD on channel n [15], [32]. In (29), it can be seen that the optimal $G_{i,j}^{C}(\mathbf{p})$ is positively related to $g_{i,j,n}^{C}(p_{i,j,n}, EI_{i,j,n})$. Given the transmission power of the SMD $p_{i,j,n}$, the value of $G_{i,j}^{C}(\mathbf{p})$ is solely decided by the effective interference $EI_{i,j,n}$. Therefore, in order to minimize the edge computing overhead, each channel should be allocated to the SMD with the highest effective interference.

The channel allocated to SMD i can be obtained according to the following strategy.

$$a_{i,j,\hat{n}} = 1|_{\hat{n}=\max EI_{i,j,n}}, \forall i, j. \tag{30}$$

C. Power allocation and offloading decision

After channel allocation, power allocation and offloading decision are two mainly considerations remaining to be solved for $P2$, which can be simplified as follows.

$$\begin{aligned} P2: \quad & \min_{\mathbf{s},\mathbf{p}} \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{s_{i,j}[\frac{d_{i,j}(w_{i,j}^{e}p_{i,j,\hat{n}} + w_{i,j}^{t})}{w\log_2(1 + \frac{p_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^{2}+I_{i,j,\hat{n}}})} \\ & + \tfrac{w_{i,j}^{t}c_{i,j}}{f^{C}}]\} + \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{(1-s_{i,j})G_{i,j}^{L*}(f_{i,j}^{l})\} \\ s.t. \quad & C1: \quad s_{i,j}(\tfrac{d_{i,j}}{r_{i,j}} + \tfrac{c_{i,j}}{f^{C}}) \leqslant T_{i,j}^{\max} \quad \forall i, j \\ & C2: \quad s_{i,j}p_{i,j,\hat{n}}\tfrac{d_{i,j}}{r_{i,j}} \leqslant E_{i,j}^{\max} \quad \forall i, j \\ & C4: \quad 0 \leqslant s_{i,j}p_{i,j,\hat{n}} \leqslant p_{\max} \quad \forall i, j \\ & C5: \quad \sum_{k=1}^{U_l}\sum_{l=1,l\neq j}^{M} s_{i,j}p_{k,l,\hat{n}}h_{k,l,\hat{n}} \leqslant I_{th} \quad \forall j, \hat{n} \\ & C8: \quad s_{i,j} \in \{0, 1\} \quad \forall i, j \end{aligned} \tag{31}$$

Since the value of $s_{i,j}$ is a binary variable, i.e., 0 or 1, we can redefine the transmission power of the SMD as follows.

$$\hat{p}_{i,j,\hat{n}} = s_{i,j}p_{i,j,\hat{n}}. \tag{32}$$

Next, we relax the integer variable $s_{i,j}$ to real number $s_{i,j} \in [0, 1]$ when $s_{i,j}(1 - s_{i,j}) = 0$ holds. The constraint of C8 can be converted as

$$C8: \quad s_{i,j} \in [0, 1] \quad \forall i, j \ . \tag{33}$$

Therefore, the $P2$ can be transformed into the following problem.

$$\begin{aligned} P2: \quad & \min_{\mathbf{s},\hat{\mathbf{p}}} \sum_{i=1}^{U_j}\sum_{j=1}^{M}[\tfrac{d_{i,j}(w_{i,j}^{e}\hat{p}_{i,j,\hat{n}} + w_{i,j}^{t}s_{i,j})}{w\log_2(1+\frac{\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^{2}+\hat{I}_{i,j,\hat{n}}})} + \tfrac{w_{i,j}^{t}c_{i,j}s_{i,j}}{f^{C}}] \\ & + \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{(1-s_{i,j})G_{i,j}^{L*}(f_{i,j}^{l})\} \\ s.t. \quad & C1: \quad \log_2(1 + \tfrac{\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^{2}+\hat{I}_{i,j,\hat{n}}}) \geqslant s_{i,j}R_{i,j} \quad \forall i, j \\ & C2: \quad \tfrac{\hat{p}_{i,j,\hat{n}}}{\log_2(1+\frac{\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^{2}+\hat{I}_{i,j,\hat{n}}})} \leqslant s_{i,j}\tfrac{E_{i,j}^{\max}w}{d_{i,j}} \quad \forall i, j \\ & C4: \quad 0 \leqslant \hat{p}_{i,j,\hat{n}} \leqslant s_{i,j}p_{\max} \quad \forall i, j \\ & C5: \quad \sum_{k=1}^{U_l}\sum_{l=1,l\neq j}^{M} \hat{p}_{k,l,\hat{n}}h_{k,l,\hat{n}} \leqslant I_{th} \quad \forall j, n \\ & C8: \quad s_{i,j} \in [0, 1] \quad \forall i, j \end{aligned} \tag{34}$$

where $\hat{I}_{i,j,n} = \sum_{k=1}^{U_l}\sum_{l=1,l\neq j}^{M} \hat{p}_{k,l,\hat{n}}h_{k,l,\hat{n}}^{j}$. $R_{i,j} = \frac{d_{i,j}}{w(T_{i,j}^{\max} - \frac{c_{i,j}}{f^{C}})}$ is the required lowest transmission rate for the SMD. In order to find the integer solution of $s_{i,j}$, we use the Lagrangian of (34) to handle the problem, which is defined as

$$\begin{aligned} L(\hat{\mathbf{p}},\mathbf{s},\lambda) = & \sum_{i=1}^{U_j}\sum_{j=1}^{M}[\frac{d_{i,j}(w_{i,j}^{e}\hat{p}_{i,j,\hat{n}}+w_{i,j}^{t}s_{i,j})}{w\log_2(1+\frac{\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^2+\hat{I}_{i,j,\hat{n}}})} \\ & +\frac{w_{i,j}^{t}c_{i,j}s_{i,j}}{f^C}]+\sum_{i=1}^{U_j}\sum_{j=1}^{M}(1-s_{i,j})G_{i,j}^{L*}(f_{i,j}^{l}) \quad , \\ & +\lambda\sum_{i=1}^{U_j}\sum_{j=1}^{Nc}s_{i,j}(1-s_{i,j}) \end{aligned} \tag{35}$$

where $\lambda \gg 1$ is the penalty factor. When the value of λ is large enough, the objective function of $P2$ and (35) have the same optimal value. Hence, the $P2$ can be modified as

$$\begin{aligned} & \min L(\hat{\mathbf{p}},\mathbf{s},\lambda) \\ & s.t. \quad C1, C2, C4, C5, C8. \end{aligned} \tag{36}$$

Considering (35) as a MINLP, we adopt the interior penalty function method to iteratively solve the problem. We define the $f_1(\hat{\mathbf{p}},\mathbf{s})$ and $f_2(\mathbf{s})$ as

$$\begin{aligned} f_1(\hat{\mathbf{p}},\mathbf{s}) = & \sum_{i=1}^{U_j}\sum_{j=1}^{M}[\frac{d_{i,j}(w_{i,j}^{e}\hat{p}_{i,j,\hat{n}}+w_{i,j}^{t}s_{i,j})}{w\log_2(1+\frac{\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}}{\sigma^2+\hat{I}_{i,j,\hat{n}}})} \\ & +\frac{w_{i,j}^{t}c_{i,j}s_{i,j}}{f^C}]+\sum_{i=1}^{U_j}\sum_{j=1}^{M}((1-s_{i,j})G_{i,j}^{L*}(f_{i,j}^{l})+\lambda s_{i,j}) \\ f_2(\mathbf{s}) = & \lambda\sum_{i=1}^{U_j}\sum_{j=1}^{M}s_{i,j}^2 \end{aligned} \tag{37}$$

Then we can rewrite the objective function (35) as $f_1(\hat{\mathbf{p}},\mathbf{s})-f_2(\mathbf{s})$, where $f_1(\hat{\mathbf{p}},\mathbf{s})$ is non-convex and $f_2(\mathbf{s})$ is convex [33]. Hence, the objective function is non-convex and we can adopt the FW (Frankand-Wolf) procedure [34] to iteratively search the optimal solution. The objective function can be transformed as

$$f_1(\hat{\mathbf{p}},\mathbf{s})-f_2(\mathbf{s}^k)-\left\langle \nabla f_2(\mathbf{s}^k),\mathbf{s}-\mathbf{s}^k\right\rangle, \tag{38}$$

where $\mathbf{s}^k$ is the solution of $\mathbf{s}$ at the kth iteration. Furthermore, since the constraints C1 and C2 are non-concave, it is hard to find the solution. We let $g_1(\hat{\mathbf{p}})$ and $g_2(\hat{\mathbf{p}})$ be

$$\begin{aligned} g_1(\hat{\mathbf{p}}) &= log_2(\hat{p}_{i,j,\hat{n}}h_{i,j,\hat{n}}+\hat{I}_{i,j,n}+\sigma^2) \\ g_2(\hat{\mathbf{p}}) &= log_2(\hat{I}_{i,j,n}+\sigma^2) \end{aligned} \tag{39}$$

We can find that both of them contain the D.C. structure [35]. Hence, we can rewrite C1 and C2 as follows.

$$\begin{aligned} C1: & \quad g_1(\hat{\mathbf{p}})-\tilde{g}_2(\hat{\mathbf{p}}) \geqslant s_{i,j}R_{i,j} \\ C2: & \quad \hat{p}_{i,j,\hat{n}}-s_{i,j}\frac{E_{i,j}^{\max}w}{d_{i,j}}(g_1(\hat{\mathbf{p}})-\tilde{g}_2(\hat{\mathbf{p}})) \leqslant 0 \end{aligned} \tag{40}$$

where $\tilde{g}_2(\hat{\mathbf{p}})$ is the first order Tayler approximation of $g_2(\hat{\mathbf{p}})$, it can be obtained by

$$\tilde{g}_2(\hat{\mathbf{p}}) = g_2(\hat{\mathbf{p}}^k)+\left\langle \nabla g_2(\hat{\mathbf{p}}^k),\hat{\mathbf{p}}-\hat{\mathbf{p}}^k\right\rangle, \tag{41}$$

where $\hat{\mathbf{p}}^k$ is the solution of $\hat{\mathbf{p}}$ at the kth iteration. Based on the above, the problem in (35) can be transformed into

$$\begin{aligned} \min & \quad f_1(\hat{\mathbf{p}},\mathbf{s})-f_2(\mathbf{s}^k)-\left\langle \nabla f_2(\mathbf{s}^k),\mathbf{s}-\mathbf{s}^k\right\rangle \\ s.t. & \quad (40), C4, C5, C8 \end{aligned} \tag{42}$$

Therefore, initializing from a feasible $(\hat{\mathbf{p}}^{(0)},\mathbf{s}^{(0)})$, the iterative search starts. The process of IPDC is illustrated in Algorithm 2. The optimal overhead of local computing is obtained at first. Then, channel allocation is completed and the result $\mathbf{a}$ is imported into the stage of power allocation and offloading decision. After performing the third part, $\mathbf{p}$ and $\mathbf{s}$ are updated. The whole process are iteratively operated, which stops and the solutions can be solved when the number of iterations achieves maximum and the difference of $L(\hat{\mathbf{p}},\mathbf{s},\lambda)$ is less than a smaller value ε. Due to the decomposed process of computation offloading and resource allocation, the number of iterations are decreased and the computation complexity is lowered. However, it brings limitations to the optimality of the final solution in return.

Algorithm 2: Optimal computation offloading and resource allocation for multi-cell MEC network (IPDC)

```
Input: the set of task $\tau$.
Output: offloading decision s and the total overhead of
        SMDs.
Initialization: $p$, $a$, $s$, $cntmax$ and $\lambda$.
Local overhead
Calculate the local CPU-cycle frequency $f^*$, $f_l$ and $f_h$.
Determine the optimal overhead of local computing
according to (27).
while $cnt < cntmax$ do
    Channel allocation
    Calculate $EI_{i,j,n}(\forall i,j,n)$.
    find $\hat{n} = \arg\max(EI_{i,j,\hat{n}})$ based on (30).
    $a_{i,j,\hat{n}} = 1$.
    ChannelSet=ChannelSet-$\hat{n}$.
    Power allocation and offloading decision
    while $\left|L(\hat{\mathbf{p}}^{k+1},\mathbf{s}^{k+1},\lambda)-L(\hat{\mathbf{p}}^{k},\mathbf{s}^{k},\lambda)\right| > \varepsilon$ do
        Find $p$ and $s$ using interior point method.
        Update $p$ and $s$ according to the solution of (42).
        Update $\lambda$.
    end
    cnt=cnt+1.
end
Calculate the total overhead of SMDs according to (34).
```

VI. Simulation results

In order to evaluate the performance of our proposed algorithm, we consider a centralized MEC network, where M small cells with $100m$ in radius are randomly scattered over the network. The MEC server is located in the MeNB, and its computation frequency is $4GHz/cycle$. Every SMD has a task needed to be executed. The data size and the required number of CPU cycles of the task are randomly generated between [300, 1200] KB and [0.1,1] GHz. The tolerance time of the task is randomly distributed between 0.5s and 5s. The elastic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2786343, IEEE Internet of Things Journal

CPU frequency of SMDs is ranging from $0.2GHz/cycle$ to $1GHz/cycle$. Moreover, the maximum transmission power is $23dBm$. The path loss model between the SMD and the SeNB is considered as the lognormal distribution [36]. There are 10 channels for each cell and the bandwidth is $0.2MHz$.

The numerical experiments are developed by the Monte Carlo method. We compare the proposed algorithm with five baseline algorithms. 'All local' and 'All MEC' represent the all tasks computed by SMDs and the MEC server, respectively. 'EP' shows that the transmission power of each SMD is equal. 'FF' indicates that SMDs have a fixed CPU-cycle frequency. 'RC' stands for the channel being randomly allocated to the offloading tasks. Considering two models of computation offloading and resource allocation for single cell and multi-cell, this section concludes two parts: single cell and multi-cell.

A. Single cell

We observe the trade-off between the energy consumption of SMDs and the execution latency of their tasks for different number of SMDs in Fig. 2. It is assumed that the number of channels is not considered and every SMD accesses SeNB through OFDMA in a single cell, hence there is no interference among SMDs. Fig. 2 shows that more energy is consumed when all tasks are executed locally. The task offloaded to the MEC sever has higher latency, resulting in the bad user experience. From the perspective of the SMD, the overhead of the SMD mainly depends on its transmission power and computation CPU-cycle frequency. Our proposed optimal offloading strategy jointly considers the allocation and scheduling of computation and communication resources to decrease the expenditure of energy and time. The definition of the weighting factor considers the residual energy aware of the SMD. Compared with the subjective weighting, our weighting factor can achieve a better compromise, which can save more energy than '$w_i = 0.8$' and decrease latency comparing with '$w_i = 0.2$'. From Fig. 2, it can also be seen that our proposed algorithm can obtain lower cost.

Fig. 3 lists 10 tasks to examine the impact of different weightings on energy consumption, latency and the offloading decision. We can observe that the tasks are executed locally through our proposed algorithm when the residual energy of SMDs is enough, like task 4, 6, 7, 8 and 9, which is similar to '$w_i = 0.8$'. However, '$w_i = 0.2$' still requires the tasks to be offloaded to the MEC server and causes more latency. When the residual energy of the SMD is insufficient, the tasks are executed in the MEC server through our proposed algorithm. However, task 5 and task 10 are executed locally for '$w_i = 0.8$'. '$w_i = 0.8$' represents that the system mainly works on decreasing the latency, which means that more energy will be consumed. It is a challenge for the SMD because it may run out of energy and cannot work continually. For '$w_i = 0.5$', more tasks are offloaded to the MEC server, due to the fact that local energy consumption is more than transmission time.

The residual energy is introduced into the definition of the weighting factor in our model. A further observation about the impact of the weightings on the residual energy of the SMD is carried out in Fig. 4. We select two SMDs with capacity of 3000mAh as the research objects. We assume that one has 80% residual energy and the other has 20% residual energy. From Fig. 4, we can see that the energy consumption based on our proposed weighting $w_i' = w_i * r_i^E$ is less than the one based on the fixed weighting with the increase of the number of utilization for both SMDs. The reason is that the weighting factor w_i' decreases with the increase of the number of utilization, then $1 - w_i'$ will increase and the system has a tilt to the energy consumption. On the other hand, if a random weighting is adopted, more energy can be saved for the SMD with 80% residual energy and more energy is consumed for the SMD with 20% residual energy. In fact, energy saving is more important for the SMD with 20% residual energy.

B. Multi-cell

In multi-cell scenario, every cell shares the same spectrum so as to improve the spectrum efficiency. However, SMDs access each SeNB through OFDMA, they suffer from the interference from neighboring cells. It is assumed that each cell has $N = 10$ channels and $U_j (j \in \{1, 2, ..., M\})$ SMDs. The trade-off between energy consumption of SMDs and the execution latency of their tasks for different number of cells is investigated in Fig. 5. We can observe that our proposed algorithm can save more energy than 'All local' and '$w_{i,j} = 0.8$' and use lower latency than 'All MEC' and '$w_{i,j} = 0.2$'. The total cost we obtain is lower than that of most of baselines as well. Note that the cost of different weightings and that of 'All local' as well as 'All MEC' are not compared due to the fact that the weightings of 'All local' and 'All MEC' are the weightings of our proposed algorithm. Hence, the trade-off between energy consumption and execution latency for multi-cell is consistent with that for single cell.

Fig. 6 shows the impact of the transmission power of SMDs on the energy consumption and execution latency under different number of small cells. We select three equal transmission powers (i.e., 15, 18 and 21dBm) to conduct this test. From Fig. 6, we can see that more energy is consumed with the increase of the transmission power but the difference in latency under different powers is very small, which indicates that the transmission power is more significant to energy consumption than latency. Moreover, it can be seen that our proposed algorithm has higher latency than equal power (EP) in most of the time. This can mainly attribute to two reasons: the first one is that the transmission rate increases and latency decreases when the transmission power increases; and the other one is that increasing the transmission power results in the aggravation of the interference among neighboring cells, so that more SMDs decide to execute their tasks locally. On the other hand, the increase of the number of small cells can also cause the aggravation of the interference, which explains the energy consumption under different equal transmission power is more than that of our proposed algorithm for $M = 6$.

Fig. 7 depicts the impact of the computation CPU-cycle frequency of SMDs on the energy consumption and execution latency under different number of small cells. $f_{\max} =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2786343, IEEE Internet of Things Journal

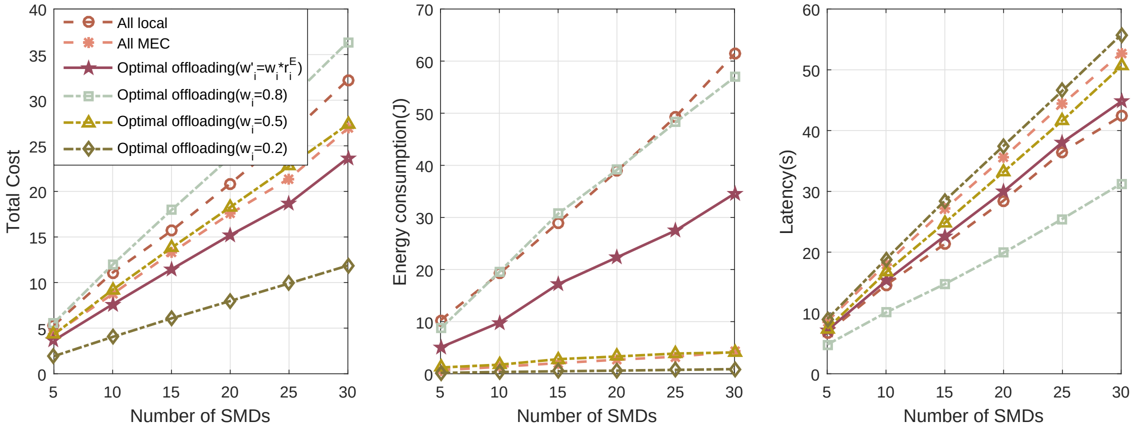


Fig. 2: The trade-off between energy consumption of SMDs and execution latency of their tasks under different number of SMDs.

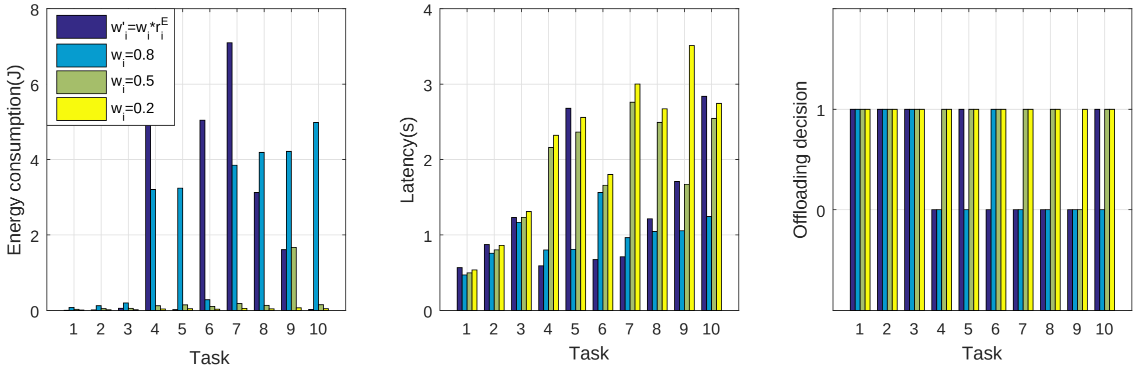


Fig. 3: The compact of weightings on energy consumption, latency and the offloading decision.

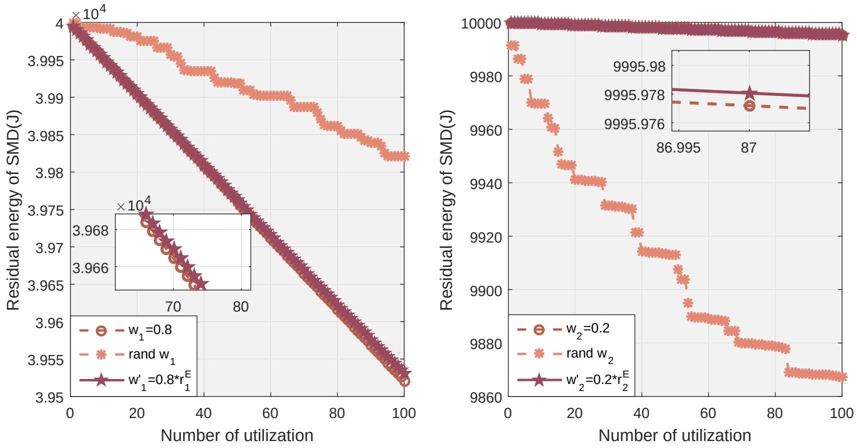


Fig. 4: The impact of the weightings on the residual energy of SMDs.

$1GHz/cycle$ and $f_{\min} = 0.2GHz/cycle$ are the maximum and minimum computation CPU-cycle frequency, and $f_{\text{med}} = (f_{\min} + f_{\max})/2$. We can observe that the energy consumption under $f_{\max}$ is more than that under $f_{\min}$ and the latency under $f_{\max}$ is lower than that under $f_{\min}$ with increased number of small cells, which is consistent with the definition of the local energy consumption and latency. Our proposed algorithm can obtain less energy consumption than $f_{\max}$ and lower latency than $f_{\min}$ to achieve a trade-off through computation frequency scheduling. Last but not least, it is noted that both the energy consumption under $f_{\max}$ and $f_{\min}$ are lower than that under f_{med} for $M = 2$ due to more tasks offloaded to the

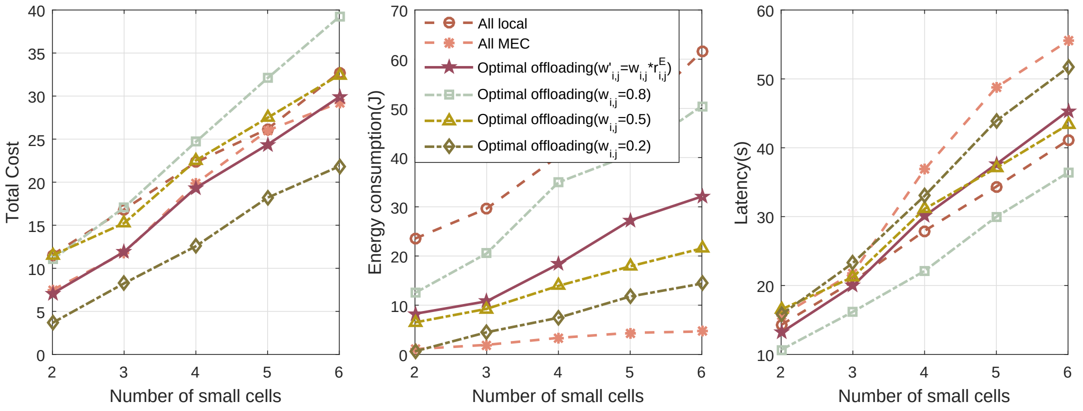
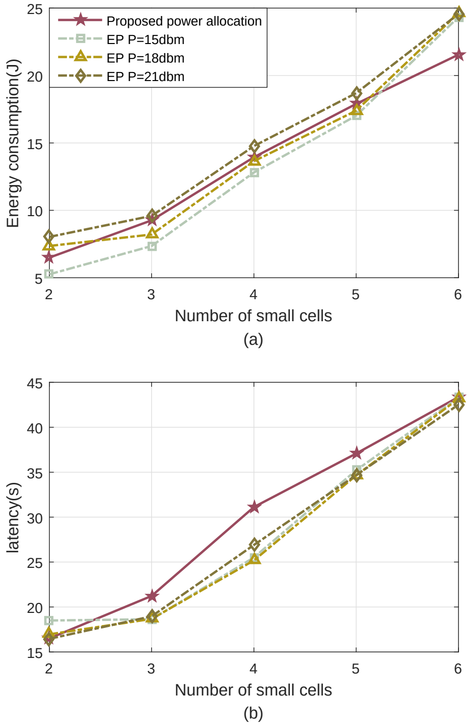


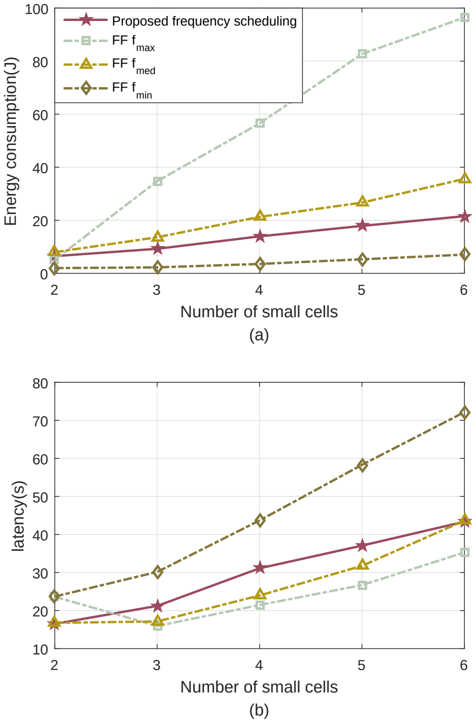
Fig. 5: The trade-off between energy consumption of SMDs and execution latency of their tasks under different number of small cells.



(a)

(b)

Fig. 6: The impact of the transmission power of SMDs on the energy consumption and execution latency under different number of small cells.



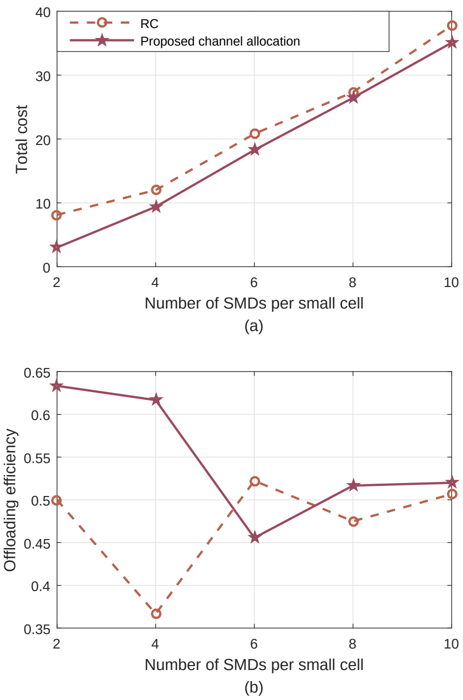
(a)

(b)

Fig. 7: The impact of the computation CPU-cycle frequency of SMDs on the energy consumption and execution latency under different number of small cells.

MEC server.

In Fig. 8, we consider the impact of channel allocation on total cost and the offloading efficiency. $M = 3$ cells are selected as the objective scenario, each cell has $N = 10$ channels and a certain number of SMDs ranging from 2 to 10. We can see that our proposed algorithm can obtain lower total cost compared with random channel allocation from Fig. 8(a), which is justified. Since random channel allocation neglected the interference between different cells, resulting in the fact that the channel quality cannot be guaranteed. Our proposed channel allocation based on the efficient interference ensures the communication quality of SMDs with the same channel in different cells, which is beneficial for tasks to be offloaded to the MEC server. Besides, the offloading efficiency refers to the ratio of the number of tasks offloaded to the MEC server to the total tasks in Fig. 8(b). the offloading efficiency we obtain declines gradually with the increase of the number of SMDs and tends to be stable, while the offloading efficiency random channel allocation has a fluctuating trend. Therefore, our proposed algorithm can obtain more stable offloading efficiency than random channel allocation.



(a)

(b)

Fig. 8: The impact of channel allocation on total cost and the offloading efficiency.

VII. Conclusion

In this paper, we investigate the trade-off between energy consumption of SMDs and execution latency of their tasks for energy-aware MEC network. To minimize the total cost of the SMDs, we jointly consider the computation offloading, computation and communication resource allocation, and define the weighting factor based on the residual energy aware of the SMDs. The single and multi-cell MEC network scenarios are addressed in our paper at the same time. We can find global solutions for single MEC network. In multi-cell MEC networks, due to the intractable MINLP, we propose an iterative search algorithm (IPDC) to search the optimal solutions. The algorithm simplifies the problems and lowers the computation complexity, while the final solutions are suboptimal. Therefore, in order to obtain approximate optimal solution, we will consider adopting the intelligent algorithm or designing heuristic method to solve the problem in the future.

VIII. Acknowledgement

This work was partially supported by the Shenzhen-Hongkong Innovative Project (SGLH20161212140718841), Shenzhen Engineering Laboratory for 3D Content Generating Technologies (NO.[2017]476), Guangdong Technology Project (2016B010108010, 2016B010125003, 2017B010110007), National Basic Research Program of China (973 Program) (No.2014CB744600), National Nature Science Foundation of China (Nos.61403365, 61402458, 61502075, 61632014, 61772508), Program of International S&T Cooperation of MOST (No.2013DFA11140).

References

[1] J. Huang, Q. Duan, C. Xing, and H. Wang, "Topology control for building large-scale and energy-efficient internet of things," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 67–73, 2017.

[2] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social internet of vehicles," *IEEE Internet of Things Journal, Doi: 10.1109/JIOT.2017.2764259*, 2017.

[3] J. Huang, Q. Duan, Y. Zhao, Z. Zheng, and W. Wang, "Simultaneous wireless information and power transfer: Technologies, applications, and research challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 215–224, 2017.

[4] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.

[5] Z. Ning, X. Wang, X. Kong, and W. Hou, "A social-aware group formation framework for information diffusion in narrowband internet of things," *IEEE Internet of Things Journal, Doi: 10.1109/JIOT.2017.2777480*, 2017.

[6] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, no. 99, pp. 5896–5907, 2017.

[7] J. Huang, C. Xing, and C. Wang, "Multicast routing for multimedia communications in the internet of things," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 26–32, 2017.

[8] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *IEEE INFOCOM 2016-the IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[9] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, O. Briante, A. Iera, and G. Ruggeri, "Edge computing and social internet of things for large-scale smart environments development," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[10] J. Pan and J. Mcelhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[11] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *International Conference on Telecommunications*, pp. 1–5, 2016.

[12] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[14] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[15] M. Masoudi, B. Khamidehi, and C. Cavdar, "Green cloud computing for multi cell networks," in *Wireless Communications and NETWORKING Conference*, 2017.

[16] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal & Information Processing Over Networks*, vol. 1, no. 2, pp. 89–103, 2014.

[17] O. Muoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.

[18] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *IEEE International Conference on Communications*, pp. 5529–5534, 2015.

[19] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[20] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.

[21] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed petri nets," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

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

[23] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, “Offloading in mobile edge computing: Task allocation and computational frequency scaling,” *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[24] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[25] Y. D. Lin, T. H. Chu, Y. C. Lai, and T. J. Huang, “Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds,” *IEEE Systems Journal*, vol. 9, no. 2, pp. 393–405, 2015.

[26] S. T. Hong and H. Kim, “Qoe-aware computation offloading scheduling to capture energy-latency tradeoff in mobile clouds,” in *IEEE International Conference on Sensing, Communication, and NETWORKING*, pp. 1–9, 2016.

[27] A. H. Jafari, D. Lpez-Prez, H. Song, H. Claussen, L. Ho, and J. Zhang, “Small cell backhaul: challenges and prospective solutions,” *Eurasip Journal on Wireless Communications & Networking*, vol. 2015, no. 1, p. 206, 2015.

[28] W. Zhang, Y. Wen, K. Guan, K. Dan, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[29] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Usenix Conference on Hot Topics in Cloud Computing*, pp. 4–4, 2010.

[30] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, “Computation offloading and resource allocation in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[31] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[32] M. Masoudi, H. Zaefarani, A. Mohammadi, and C. Cavdar, “Energy efficient resource allocation in two-tier ofdma networks with qos guarantees,” *Wireless Networks*, pp. 1–15, 2017.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambrige University Press, 2004.

[34] P. Apkarian and H. D. Tuan, “Robust control via concave minimization local and global algorithms,” *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 299–305, 2000.

[35] H. H. Kha, H. D. Tuan, and H. H. Nguyen, “Fast global optimal power allocation in wireless networks by local d.c. programming,” *Wireless Communications IEEE Transactions on*, vol. 11, no. 2, pp. 510–515, 2012.

[36] 3GPP, “Evolved universal terrestrial radio access (E-UTRA): Radio frequency(RF) system scenarios,” *TR 36.942 V11. 0. 0*, 2012.

Jiao Zhang received the B.S. and M.S. degrees from Xiangtan University and Xidian University, China, in 2013 and 2016, respectively. She is currently working toward the Ph.D. degree with the College of Electronic science from National University of Defense Technology (NUDT), Changsha, China. Her research interests include mobile edge computing and resource allocation in heterogeneous networks.

Xiping Hu received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada.

He is currenlty a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He was the Co-Founder and CTO of Bravolol Ltd., Hong Kong, a leading language learning mobile application company with over 100 million users, and listed as the top 2 language education platform globally. He has authored or co-authored around 50 papers published and presented in prestigious conferences and journals such as the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, ACM TOMM, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE Communications Magazine, IEEE Network, HICSS, ACM MobiCom, and WWW. His current research interests include mobile cyber-physical systems, crowdsensing, social networks, and cloud computing.

Dr. Hu has been serving as the Lead Guest Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and WCMC.

Zhaolong Ning (M’14) received the M.S. and PhD degrees from Northeastern University, Shenyang, China. He was a Research Fellow at Kyushu University, Japan. He is an assistant professor in the School of Software, Dalian University of Technology, China. His research interests include social computing, edge computing, and vehicular networks.

Edith C.-H. Ngai is currently an Associate Professor in Department of Information Technology, Uppsala University, Sweden. She has been a visiting researcher in Ericsson Research in 2015-2017. She received her PhD from The Chinese University of Hong Kong in 2007. She did her post-doc in Imperial College London, United Kingdom. She has also been a visiting researcher in Simon Fraser University, Tsinghua University, and UCLA. Her research interests include Internet-of-Things, mobile cloud computing, network security and privacy, smart cities and urban computing. She is a project leader of the national project, GreenIoT (2014-2017), for open data and sustainable development in Sweden. Edith is a VINNMER Fellow (2009) awarded by the Swedish governmental agency VINNOVA. Her co-authored papers have received best paper runner-up awards in IEEE IWQoS 2010 and ACM/IEEE IPSN 2013. She served as a TPC co-chair of IEEE SmartCity 2015, IEEE ISSNIP 2015, and ICNC 2018 Network Algorithm and Performance Evaluation Symposium. She is an Associate Editor for IEEE Access and IEEE Transactions of Industrial Informatics. She is a senior member of ACM and IEEE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2786343, IEEE Internet of Things Journal

Li Zhou received his B.S., M.S. and Ph.D. degrees from National University of Defense Technology (NUDT), China in 2009, 2011 and 2015 respectively. From Sept. 2013 to Sept. 2014 he worked as a visiting scholar at The University of British Columbia, Canada. He is currently an assistant professor at College of Electronic science, NUDT, China. His research interests are in the area of software defined radios (SDRs), software defined networks (SDNs) and heterogeneous networks (HetNets). Dr. Zhou served as a TPC member in IEEE CIT 2017, keynote speaker in ICWCNT 2016 and co-chair in ITA 2016. His research contributions have been published and presented in more than 20 prestigious journals and conferences, such as IEEE Transactions on Vehicular Technology, Ad Hoc Networks, WInnComm 2017, IEEE INFOCOM 2015 and IEEE GLOBECOM 2014.

Jibo Wei received his B.S. degree and M.S. degree from National University of Defense Technology (NUDT), Changsha, China, in 1989 and 1992, respectively, and the Ph.D. degree from Southeast University, Nanjing, China, in 1998, all in electronic engineering. He is currently the director and a professor of the Department of Communication Engineering of NUDT. His research interests include wireless network protocol and signal processing in communications, more specially, the areas of MIMO, Multicarrier transmission, cooperative communication, and cognitive network. He is the member of the IEEE Communication Society and also the member of the IEEE VTS. He also works as one of the editors of Journal on Communications and the senior member of China Institute of Communications and Electronics respectively.

Jun Cheng received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2006. He is currently a Professor and the Founding Director of the Laboratory for Human Machine Control, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He has authored or co-authored about 110 articles. His current research interests include computer visions, robotics, and machine intelligence and control.

Bin Hu (M'10, SM'15) is currently a Professor and the Dean of the School of Information Science and Engineering with Lanzhou University, Lanzhou, China, an Adjunct Professor with Tsinghua University, Beijing, China, and a Guest Professor with ETH Zurich, Zrich, Switzerland. He has authored or co-authored over 200 papers in peer-reviewed journals, conferences, and book chapters including Science (Suppl.), the Journal of Alzheimers Disease, IEEE TRANSACTIONS, IEEE Intelligent Systems, AAAI, BIBM, EMBS, CIKM, and ACM SIGIR.

Prof. Hu is the Co-Chair of IEEE SMC TC on Cognitive Computing, a member at large of ACM China, and the Vice President of the International Society for Social Neuroscience (China Committee). His work has been funded as a PI by the Ministry of Science and Technology, National Science Foundation China, European Framework Programme 7, EPSRC, and HEFCE U.K. He has served guest editor of Science in special issue - Advances in Computational Psychophysiology, Associate Editor of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, Brain Informatics, IET Communications, Cluster Computing, Wireless Communications and Mobile Computing, and Wileys Security and Communication Networks. He is an IET Fellow.