TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 11/11 pp239–250
DOI: 10.26599/TST.2019.9010062
Volume 26, Number 2, April 2021

Utility Aware Offloading for Mobile-Edge Computing

Ran Bi*, Qian Liu, Jiankang Ren, and Guozhen Tan

Abstract: Mobile-edge computing casts the computation-intensive and delay-sensitive applications of mobile devices onto network edges. Task offloading incurs extra communication latency and energy cost, and extensive efforts have focused on offloading schemes. Many metrics of the system utility are defined to achieve satisfactory quality of experience. However, most existing works overlook the balance between throughput and fairness. This study investigates the problem of finding an optimal offloading scheme in which the objective of optimization aims to maximize the system utility for leveraging between throughput and fairness. Based on Karush-Kuhn-Tucker condition, the expectation of time complexity is analyzed to derive the optimal scheme. A gradient-based approach for utility-aware task offloading is given. Furthermore, we provide an increment-based greedy approximation algorithm with $1+\frac{1}{e-1}$ ratio. Experimental results show that the proposed algorithms can achieve effective performance in utility and accuracy.

Key words: utility; approximation algorithm; quality of experience; mobile edge computing

1 Introduction

With the advancement of technologies in mobile sensing and wireless communication, Mobile-Edge Computing (MEC) bridges the gap between intensely computational requirements and the restricted capability of individual devices[1, 2]. MEC leverages physical proximity to mobile devices, and brings a promising possibility to enable Internet of Things (IoT)-based computing intensive applications, such as data analytics[3–5] and data mining[6–8], while achieving low delay latency and energy cost[9, 10]. By offloading part of the tasks to a nearby MEC cloud, the system considerably decreases computation latency and energy consumption[11].

Task offloading incurs extra communication latency and energy cost; thus, the offloading decision becomes a critical issue for end-nodes, mobile Base Stations (BSs), and the cloud. According to the optimization goals, the research work can be divided into three categories: delay latency minimization, energy consumption minimization, and joint optimization for offloading and resource allocation. To achieve satisfactory Quality of Experience (QoE), researchers proposed many metrics of system utility. In Ref. [12], the utility measured the improvement in delay and energy consumption by offloading compared with local execution. Tao et al.[13] studied the performance guaranteed scheme, which minimized the energy consumption of mobile devices. Optimal offloading decision was formalized as a latency-constrained energy minimization problem in Ref. [14]. Meng et al.[15] aimed at improving the QoE of end users, in which the trade-off between quality of services and experience of end-users was investigated.

Due to the diversity of applications in MEC, the range of data sizes related to offloading is large, and the distribution of data sizes cannot be predicted. Most existing works on MEC have overlooked the balance between throughput and fairness. Considering the realistic scenarios, we introduce a multi-user and one-BS MEC system. We assume that the computation task set is $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, where N is the number

• Ran Bi, Qian Liu, Jiankang Ren, and Guozhen Tan are with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China. E-mail:{biran, qianliu, rjk, gztan}@dlut.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2019-12-09; accepted: 2019-12-16

of computation tasks. The data size of task $\mathcal{T}_N$ is much larger than that of $\mathcal{T}_i$, for $i \neq N$. The QoE is severely reduced if we make an offloading decision for $\mathcal{T}_N$. Because offloading task $\mathcal{T}_N$ takes a significant computation resource such that other tasks cannot be offloaded.

The computing tasks have different requirements, such as data size, number of CPU cycles, deadline of completion time, and constraint for energy consumption. Apart from considering fairness, the offloading scheme will overlook certain types of computation task, thereby leading to a significant reduction in QoE. The computation capability is always provided in advance. The relation between the requirements of computing tasks and computation capability of the server is neglected.

The aforementioned observation motivates us to investigate the problem of finding an optimal offloading scheme, which maximizes the system utility while balancing the throughput and fairness. We present a formal formulation of the optimization problem above. By using Karush-Kuhn-Tucker (KKT) condition, we prove the expectation of time complexity to derive the optimal scheme. For a given precision parameter, a $\left(1+\frac{1}{\mathrm{e}-1}\right)$-approximation algorithm is provided based on greedy method. Its time complexity and space complexity are $O\left(\frac{N^4\tau}{c_{\min}}\right)$ and $O(N^3)$, respectively, where τ is the computation constraint of server and $c_{\min}$ is the minimum number of CPU circles among the tasks.

The main contributions of this paper are as follows:

- The problem of finding an optimal deployment scheme with different requirements of computing task is defined and formalized as an optimization problem.
- According to KKT condition, the expectation of time complexity for deriving the optimal scheme is proved to be $O(4^N)$.
- We present a gradient-based approach for utility-aware task offloading. Furthermore, an increment-based polynomial-time approximation algorithm is provided, where the ratio bound is $1+\frac{1}{\mathrm{e}-1}$. The complexity of the algorithm is analyzed.
- Simulation experiments are conducted to evaluate the proposed algorithms. Experimental results show that the given algorithms can achieve effective performance in terms of utility and accuracy.

The rest of this paper is organized as follows. Related works on offloading and task scheduling in MEC are surveyed in Section 2. In Section 3, the optimization problem is defined, and the computation complexity is analyzed in Section 4. For utility-aware task offloading, Section 5 provides a gradient-based approach and a greedy-based approximation algorithm. The approximation ratio bound is proved. Performance evaluation is illustrated in Section 6, and Section 7 concludes this paper.

2 Related Work

The task scheduling is critical for distributed network systems, such as wireless networks[16], IoT[17], cyber-physical systems[18], and mobile crowdsensing systems[19,20]. However, the system architecture and quality of service in MEC are different from these networks. Computation offloading for MEC has attracted considerable research effort and many efficient algorithms were proposed in Ref. [21]. Recent work has focused on either offloading decisions or resource allocation with two optimization objectives: latency and energy consumption.

To minimize the average delay, Liu et al.[22] considered a power-constrained delay minimization problem and proposed a 1D search-based approach to find the optimal stochastic computation task scheduling. Ren et al.[23] investigated the problem of minimizing the weighted-sum delay, while considering the constraints on communication and computation resource. To improve the random access possibility, Liu et al.[24] provided an MEC framework for energy internet, in which the local network was used for data collection and compression.

Lei et al.[25] proposed a semi-distributed task scheduling strategy to minimize the long-term average weighted sum of latency and energy cost, where the computation offloading and multiuser scheduling were jointly considered in a narrowband-IoT edge computing system. Closed-form solutions were provided for linear approximation and semi-gradient descent approach. Zhu et al.[26] proposed two approximation algorithms in a more complex scenario, where multiple moving mobile devices shared multiple MEC servers. The designed schedule strategy aimed to minimize the energy consumption while satisfying the latency constraint. To reduce the total energy consumption of all mobile devices, Tao et al.[13] studied the partial offloading decisions with constraints on resource capacity and delay.

Lyu et al.[12] investigated the joint optimization

for offloading decisions and computation resource allocation. The optimization objective was to maximize the system utility, which was defined as a linear combination of improvement in latency and energy cost by offloading compared with the local execution. Wang et al.[27] addressed the minimization problem for time delay and energy cost, where both of the CPU frequency control and uploading cost were considered. Firstly, the offloading decisions were made by a central method, and then the local computation and wireless transmission of devices were optimized. Most existing works overlook the balance between throughput and fairness, and the offloading scheme overlooks certain types of computation task. As a result, the QoE of some users is decreased.

3 System Model and Problem Formulation

In this section, the problem of finding an optimal offloading scheme with different computational requirements is defined and formalized as a general optimization problem. Based on KKT condition, the expectation of time complexity is proved to be $O(4^N)$.

3.1 System model

We consider that the system consists of a base station and N devices (the number of computation tasks equals the number of devices), which can include IoT and mobile devices. Each device needs to offload a computation task to the base station. The devices can access the station resources through a wireless channel, and each device is endowed with different computing and communication capabilities, such as smart meters, tablets, and laptops. The system operates with a slotted structure. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the index set for computation task.

Definition 1 (computing task): A computing task $\mathcal{T}_i$ is represented by a tuple (d_i, c_i, t_i, e_i), where d_i describes the data size of the task from device i, c_i is the total number of CPU cycles required to accomplish the computation task, t_i denotes the deadline of completion time for the task, and e_i is the upper bound of energy consumption, which includes the energy cost of wireless transmission and local computation.

For the uplink transmission, the data rate of device i can be[12, 28]

$$r_i = W\log_2\left(1 + \frac{p_i h_i}{N_0}\right) \tag{1}$$

where W is the bandwidth, p_i represents the transmission power of device i, h_i denotes the channel gain from device i to the base station including path loss and fading, and N_0 is the noise power. We assume that the system bandwidth is B. Thus, no more than $M = B/W$ devices are allowed to transmit at the same time.

3.2 Task offloading model

Based on the communication model, the task duration of offloading consists of time consumption by two parts, i.e., (1) communication time consumed by the uplink transmission and (2) execution time consumed by an edge server. The task delay incurred by offloading can be denoted as

$$t_i^{\text{off}} = \frac{d_i}{r_i} + \frac{c_i}{f_e} = \frac{d_i}{W\log_2\left(1 + p_i h_i N_0^{-1}\right)} + \frac{c_i}{f_e} \tag{2}$$

where f_e is the computation capability of edge server in term of instruction numbers per second.

In the procedure of task offloading, we only consider the energy cost of the upload. Therefore, the energy consumption depends on the data size of the task. If device i offloads a task to edge server, then the energy consumption of device i for uplink transmission can be denoted as follows:

$$e_i^{\text{off}} = \frac{d_i}{r_i} \times p_i = \frac{d_i p_i}{W\log_2\left(1 + p_i h_i N_0^{-1}\right)} \tag{3}$$

3.3 Local computation

For local task computing, let f_i denote the CPU computing capability of device i. The local execution time of the task can be obtained as follows:

$$t_i^{\text{loc}} = \frac{c_i}{f_i} \tag{4}$$

The energy cost of CPU is a superlinear function of computation frequency. We denote p_i^{loc} as the power consumption per CPU cycle. For task $\mathcal{T}_i$, the energy cost of locally processing is expressed as

$$e_i^{\text{loc}} = p_i^{\text{loc}} \times c_i \tag{5}$$

3.4 Problem formulation

For each task $\mathcal{T}_i$, we consider the task completion time and energy consumption simultaneously. Let λ_i denote the fraction of the offloading task for device i. The task delay consists of offloading and local computation as follows:

$$\lambda_i \times t_i^{\text{off}} + (1-\lambda_i) \times t_i^{\text{loc}} = \frac{\lambda_i d_i}{r_i} + \frac{\lambda_i c_i}{f_e} + (1-\lambda_i) \times \frac{c_i}{f_i} \tag{6}$$

Similarly, the energy consumption of each device i includes the uplink communication and local computation as follows:

$$\lambda_i \times e_i^{\text{off}} + (1-\lambda_i) \times e_i^{\text{loc}} = \frac{\lambda_i d_i p_i}{r_i} + (1-\lambda_i) \times p_i^{\text{loc}} c_i \tag{7}$$

In our model, a set of tasks arrive and seek the derived computing service. We adopt an economic point to measure the utility for the edge server performing a computation task. The nonlinear value function presented in this paper is similar to that in the work of Refs. [29,30] based on the law of diminishing marginal utility[31,32]. The utility of the edge server by serving task $\mathcal{T}_i$ can be written as

$$u_i = \log(1 + \lambda_i d_i) \tag{8}$$

Let τ denote the total CPU computation capability, which can be observed as the maximum number of instructions per second. For a given set of computation task $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, the computation capability of the server satisfies the condition that $\sum_{i=1}^{N} c_i > \tau$. That is, the service cannot enable each task to be computed at the BS. In this paper, we aim to maximize the sum of the utility for the given tasks. The optimization problem can be formulated as follows:

$$\max_{\lambda} \sum_{i=1}^{N} \log(1 + \lambda_i d_i) \tag{9}$$

$$\text{s.t.} \quad \frac{\lambda_i d_i}{r_i} + \frac{\lambda_i c_i}{f_e} + (1-\lambda_i) \times \frac{c_i}{f_i} \leqslant t_i, \forall i \in \mathcal{N} \tag{10}$$

$$\frac{\lambda_i d_i p_i}{r_i} + (1-\lambda_i) \times p_i^{\text{loc}} c_i \leqslant e_i, \forall i \in \mathcal{N} \tag{11}$$

$$\sum_{i=1}^{N} \lambda_i c_i \leqslant \tau \tag{12}$$

$$0 \leqslant \lambda_i \leqslant 1, \forall i \in \mathcal{N} \tag{13}$$

4 Computation Complexity Analysis

Lemma 1 The objective function of optimization problem (9) is strictly concave.

Proof For given $i \in \{1, 2, \ldots, N\}$, we can have the following:

$$\frac{\partial \log(1 + \lambda_i d_i)}{\partial \lambda_i} = \frac{d_i}{(1 + \lambda_i d_i) \ln 2}.$$

According to the constraints $\lambda_i > 0$ and $d_i > 0$, the following is easily known:

$$\frac{\partial^2 \log(1 + \lambda_i d_i)}{\partial^2 \lambda_i} = \frac{-{d_i}^2}{(1 + \lambda_i d_i) \ln 2} < 0.$$

Then $\log(1 + \lambda_i d_i)$ is a concave function with respect to λ_i, and the sum of independent concave functions is still a concave function. Based on the preceding discussion, we can determine that the objective function of optimization problem (9) is strictly concave. ■

For any given i, we know that the inequality constraints of the problem are convex function, with regard to λ_i. We can apply the KKT condition to obtain the optimal solutions for the proposed problem (9). For simplification, we introduce notations T_i and E_i, such that $\forall i \in \{1, 2, \ldots, N\}$,

$$U(\lambda) = \sum_{i=1}^{N} \log(1 + \lambda_i d_i), C(\lambda) = \sum_{i=1}^{N} \lambda_i c_i - \tau.$$

$$T_i(\lambda_i) = \frac{\lambda_i d_i}{r_i} + \frac{\lambda_i c_i}{f_e} + (1-\lambda_i) \times \frac{c_i}{f_i} - t_i \tag{14}$$

$$E_i(\lambda_i) = \frac{\lambda_i d_i p_i}{r_i} + (1-\lambda_i) \times p_i^{\text{loc}} c_i - e_i \tag{15}$$

Theorem 1 We assume that $\lambda_1^*, \lambda_2^*, \ldots, \lambda_N^*$ are the optimal solutions to Problem (9), then there must exist $\alpha_1^*, \ldots, \alpha_N^*$, $\beta_1^*, \ldots, \beta_N^*$, and ξ^*, satisfying the following:

$$\nabla U(\lambda^*) - \sum_{j=1}^{N} \alpha_i^* \nabla T_i(\lambda_i^*) - \sum_{k=1}^{N} \beta_k^* \nabla E_k(\lambda_k^*) - \xi^* \nabla C(\lambda^*) = 0 \tag{16}$$

$$\alpha_i^* E_i(\lambda_i^*) = 0, \beta_i^* E_i(\lambda_i^*) = 0, \forall i \in \{1, 2, \ldots, N\} \tag{17}$$

$$\xi^* C(\lambda^*) = 0 \tag{18}$$

$$T_i(\lambda_i^*) \leqslant 0, E_i(\lambda_i^*) \leqslant 0, C(\lambda^*) \leqslant 0, \forall i \in \{1, 2, \ldots, N\} \tag{19}$$

$$\xi^* \geqslant 0, 0 \leqslant \lambda_i^* \leqslant 1, \alpha_i^* \geqslant 0, \beta_i^* \geqslant 0, \forall i \in \{1, 2, \ldots, N\} \tag{20}$$

We define the Lagrangian function as follows:

$$L(\lambda, \alpha, \beta, \xi) = \sum_{i=1}^{N} \log(1 + \lambda_i d_i) + \sum_{i=1}^{N} \alpha_i \left(0 - T_i(\lambda_i)\right) + \sum_{i=1}^{N} \beta_i \left(0 - E_i(\lambda_i)\right) + \xi \left(0 - C(\lambda)\right) \tag{21}$$

Based on Theorem 1, we need to solve $(3N + 1)$ equations, that is, Eqs. (16)–(18). Then, we check if the solutions satisfy Constraints (19) and (20). For given $i \in \{1, \ldots, N\}$, we obtain the optimal solutions by

dealing with the following equations:

$$\begin{cases} \dfrac{d_i}{(1+\lambda_i^* d_i)\ln 2}+\alpha_i^*\left(\dfrac{c_i}{f_i}-\dfrac{d_i}{r_i}-\dfrac{c_i}{f_e}\right)= \\ \qquad -\beta_i^*\left(p_i^{\mathrm{loc}}c_i-\dfrac{d_i p_i}{r_i}\right)+\xi^* c_i & (22a) \\ \beta_i^*\left(\dfrac{\lambda_i^* d_i p_i}{r_i}+(1-\lambda_i^*)\times p_i^{\mathrm{loc}}c_i-e_i\right)=0 & (22b) \\ \xi^*\left(\sum_{i=1}^{N}\lambda_i^* c_i-\tau\right)=0 & (22c) \end{cases}$$

Lemma 2 Without the constraint of Eq. (22c), four cases are used to check the feasibility of the solutions derived by the preceding equations.

Proof For given i, we consider the equations for λ_i, α_i, and β_i.

Case 1 $\alpha_i' \neq 0$ and $\beta_i' = 0$. Then, the following formulas simultaneously hold:

$$\begin{cases} \dfrac{\lambda_i' d_i}{r_i}+\dfrac{\lambda_i' c_i}{f_e}+(1-\lambda_i')\times\dfrac{c_i}{f_i}-t_i=0 & (23a) \\ \dfrac{\lambda_i' d_i p_i}{r_i}+(1-\lambda_i')\times p_i^{\mathrm{loc}}c_i-e_i<0 & (23b) \end{cases}$$

According to Eq. (23a), we can obtain that $\lambda_i' = \dfrac{c_i {f_i}^{-1}-t_i}{c_i {f_i}^{-1}-c_i {f_e}^{-1}-d_i {r_i}^{-1}}$. If λ_i' satisfies Constraint (23b), then λ_i' is the candidate of optimal solutions. If λ_i' does not meet Constraint (23b), then we consider Cases $2-4$.

Case 2 $\alpha_i' = 0$ and $\beta_i' \neq 0$. Similarly, the following formulas simultaneously hold:

$$\begin{cases} \dfrac{\lambda_i' d_i}{r_i}+\dfrac{\lambda_i' c_i}{f_e}+(1-\lambda_i')\times\dfrac{c_i}{f_i}-t_i<0 & (24a) \\ \dfrac{\lambda_i' d_i p_i}{r_i}+(1-\lambda_i')\times p_i^{\mathrm{loc}}c_i-e_i=0 & (24b) \end{cases}$$

According to Eq. (24b), we know that $\lambda_i' = \dfrac{e_i - {p_i}^{\mathrm{loc}}c_i}{d_i p_i {r_i}^{-1}-{p_i}^{\mathrm{loc}}c_i}$. If λ_i' can meet Constraint (24a), then λ_i' is the candidate of optimal solutions. If λ_i' does not satisfy Constraint (24a), then we consider Cases 3 and 4.

Case 3 $\alpha_i' \neq 0$ and $\beta_i' \neq 0$.

$$\begin{cases} \dfrac{\lambda_i' d_i}{r_i}+\dfrac{\lambda_i' c_i}{f_e}+(1-\lambda_i')\times\dfrac{c_i}{f_i}-t_i=0 & (25a) \\ \dfrac{\lambda_i' d_i p_i}{r_i}+(1-\lambda_i')\times p_i^{\mathrm{loc}}c_i-e_i=0 & (25b) \end{cases}$$

Based on Eq. (25a), $\lambda_i' = \dfrac{c_i {f_i}^{-1}-t_i}{c_i {f_i}^{-1}-c_i {f_e}^{-1}-d_i {r_i}^{-1}}$. If λ_i' satisfies Constraint (25b), then λ_i' is the candidate of optimal solutions. Otherwise, we consider Case 4.

Case 4 $\alpha_i' = 0$ and $\beta_i' = 0$. According to $\dfrac{\partial L}{\partial \lambda_i'} = 0$, we know $\lambda_i' = \dfrac{1}{\xi' c_i \ln 2}-\dfrac{1}{d_i}$, which implies that $\xi' > 0$. $\alpha_i' \neq 0$, $\beta_i' \neq 0$ or $\xi_i' \neq 0$ mean that the corresponding optimal point is on the boundary. ■

Theorem 2 For given N tasks, the expectation of time complexity for deriving the optimal points is $O(4^N)$.

Proof For given task $\mathcal{T}_i$, we need to check the feasibility of optimal solution under four cases. If $\lambda_1', \lambda_2', \ldots, \lambda_N'$ are candidates of optimal points, that are derived based on Lemma 2, then ξ' can be obtained by resolving equation $\sum_{i=1}^{N}\lambda_i' c_i - \tau = 0$. We verify whether the candidate points such as $\lambda_1', \ldots, \lambda_N'$, $\alpha_1', \ldots, \alpha_N'$, $\beta_1', \ldots, \beta_N'$, and ξ', meet the KKT conditions in Eqs. (16) – (20).

For a given feasible solution α, β, and ξ, we denote the computation cost to verify KKT conditions as $O(1)$. We assume that the optimal points satisfy the uniform distribution. The expectation of time complexity is analyzed as $1\times\dfrac{1}{4^N}+2\times\dfrac{1}{4^N}+\cdots+4^N\times\dfrac{1}{4^N}=\dfrac{4^N+1}{2}$.

Based on the preceding analysis, the expectation of time complexity to derive the optimal points is $O(4^N)$. ■

5 Algorithm Design for Utility-Aware Task Offloading

As the expectation of time complexity is $O(4^N)$, designing an exact algorithm with polynomial time for optimal task offloading is difficult. We present a gradient-based approach for utility-aware task offloading, in which the performance depends on step size and relative ratio. Moreover, we prove that the objective function of the problem is increasing and submodular. We provide a greedy polynomial-time algorithm with approximation ratio guarantee.

5.1 Gradient-based approach for utility-aware task offloading

We provide a gradient-based approach for utility-aware task offloading. The basic idea of the gradient method is to solve the dual problem. We consider the dual decomposition for Problem (9). Owing to the separability of Constraints (10) and (11), we present the Lagrangian form of the primal problem.

$$L(\lambda,\xi)=\sum_{i=1}^{N}\log(1+\lambda_i d_i)+\xi\left(\tau-\sum_{i=1}^{N}\lambda_i c_i\right) \tag{26}$$

$$\text{s.t.}\quad \frac{\lambda_i d_i}{r_i}+\frac{\lambda_i c_i}{f_e}+(1-\lambda_i)\times\frac{c_i}{f_i}\leqslant t_i,\forall i\in\mathcal{N} \tag{27}$$

$$\frac{\lambda_i d_i p_i}{r_i}+(1-\lambda_i)\times p_i^{\text{loc}}c_i\leqslant e_i,\forall i\in\mathcal{N} \tag{28}$$

$$0\leqslant\lambda_i\leqslant 1,\forall i\in\mathcal{N} \tag{29}$$

We can easily know that

$$L(\lambda,\xi)=\sum_{i=1}^{N}\{\log(1+\lambda_i d_i)-\xi\lambda_i c_i\}+\xi\tau,$$

and $L(\lambda,\xi)$ is a strictly concave function.

Without the constraints, the maximum value of $L(\lambda,\xi)$ can be achieved by a given ξ. For a given ξ, $\lambda_i^\star$ is defined as follows:

$$\lambda_i^\star(\xi)=\arg\max_{\lambda_i\geqslant 0}[\log(1+\lambda_i d_i)-\xi\lambda_i c_i] \tag{30}$$

$\lambda_i^\star(\xi)$ can be obtained by $\frac{\partial L(\lambda,\xi)}{\partial\lambda_i}=0$. For $\forall i\in\mathcal{N}$, we have

$$\frac{\partial L(\lambda,\xi)}{\partial\lambda_i}=\frac{d_i}{(1+\lambda_i d_i)\ln 2}-\xi c_i \tag{31}$$

$$\lambda_i^\star(\xi)=\frac{1}{\xi c_i\ln 2}-\frac{1}{d_i} \tag{32}$$

According to strict concavity, $\lambda_i^\star(\xi)$ is unique. The dual problem is as follows:

$$g(\xi)=\sum_{i=1}^{N}g(\lambda_i^\star,\xi)+\xi\tau \tag{33}$$

$$\text{s.t.}\quad g(\lambda_i^\star,\xi)=\log(1+\lambda_i^\star d_i)-\xi\lambda_i c_i \tag{34}$$

$$\lambda_i^*(\xi)=\frac{1}{\xi c_i\ln 2}-\frac{1}{d_i} \tag{35}$$

$$\xi\geqslant 0 \tag{36}$$

Based on the preceding analysis, the gradient approach can be applied:

$$\xi(l+1)=\left[\xi(l)-\varphi\left(\tau-\sum_{i=1}^{N}\lambda_i\left(\xi(l)\right)\right)\right]^+ \tag{37}$$

where l is the iteration index, $\varphi>0$ is a small positive step size, and $[\cdot]^+$ denotes the projection onto the nonnegative quadrant.

Theorem 3 If we do not consider the Constraints (10) and (11), then the dual variable $\xi(m)$ converges to the dual optimal $\xi^\star$ as $m\to\infty$, and the primal variable $\lambda_i^\star\left(\xi(m)\right)$ also converges to the primal optimal variable $\lambda_i^\star$.

Proof The objective function of Problem (9) is strictly concave. The problem satisfies Slater's condition. Thus, the primal Problem (9) and dual Problem (33) are strong dualities, if we do not consider the constraints on delay latency and energy cost. As the duality gap is zero, the solution of Problem (30) is unique because of the strong duality. Therefore, the dual variable $\xi(m)$ converges to the dual optimal $\xi^\star$ as $m\to\infty$, and the primal variable $\lambda_i^\star\left(\xi(m)\right)$ also converges to the primal optimal variable $\lambda_i^\star$. ■

We can obtain the optimal solution for the primal problem by solving the dual problem. Constraints (10) and (11) are linear functions with respect to λ_i. The feasible solution space is convex. The dual solution $\xi(m+1)$ leads to the primal solution λ, when $m\to\infty$. In the iterative process, we should check the boundary conditions for λ_i. For a given $\mathcal{T}_i=<d_i,c_i,t_i,e_i,p_i,p_i^{\text{loc}}>$, the boundary condition for λ_i is easily obtained due to the separability of Constraints (10) and (11).

According to the pseudocode as shown in Algorithm 1, the computation complexity depends on the δ and initial value of ξ_O.

Algorithm 1 Gradient-based algorithm for utility-aware task offloading

Input: $W,\tau,f_e,\varphi>0,\delta>0,\mathcal{T}_i=<d_i,c_i,t_i,e_i,p_i,p_i^{\text{loc}}>$, for $i=1,\ldots,N$;
Output: $\lambda_1,\lambda_2,\ldots,\lambda_N$;

```
Initialization, let ξ_O and ξ_U equal a nonnegative value;
Initialization, let λ_i = 0;
for i = 1 to N do
    According to Constraints (10) and (11), compute the lower bound L_i and upper bound U_i of λ_i;
end for
Let h = +∞;
while h > 1 + δ do
    ξ_O = ξ_U;
    ξ_U = [ξ_O − φ(τ − Σ_{i=1}^{N} λ_i(ξ_O))]^+;
    for i = 1 to N do
        λ_i = 1/(ξ_U c_i ln 2) − 1/d_i;
        if λ_i > U_i then
            λ_i = U_i;
        end if
        if λ_i < L_i then
            λ_i = L_i;
        end if
    end for
    h = Σ_{i=1}^{N} log(1 + λ_i(ξ_U)d_i) / Σ_{i=1}^{N} log(1 + λ_i(ξ_O)d_i);
end while
return λ_i.
```

5.2 Increment-based approximation algorithm

Designing an exact algorithm with polynomial-time cost for optimal offloading scheme is difficult. In this section, we prove that the objective function of the proposed problem is increasing and submodular. Then, we provide a greedy algorithm called Increment-based Approximation Algorithm for Offloading Scheme (IAAOS). The approximation ratio bound is proved to be $1+\frac{1}{\mathrm{e}-1}$.

Theorem 4 The objective function of Problem (9) is submodular.

Proof We denote $\lambda=\{\lambda_1,\ldots,\lambda_N\}$ as a given feasible offloading scheme. For convenience, we define a unit set $I(k,i)=\{0,\ldots,10^{-k},0,\ldots,0\}$. $\lambda+I(k,i)$ will generate a new set $\hat{\lambda}=\{\lambda_1,\ldots,\lambda_{i-1},\lambda_i+10^{-k},\lambda_{i+1},\ldots,\lambda_N\}$, in which the schemes of all the tasks are the same as λ, and expect λ_i added by 10^{-k}.

We denote $\delta(\lambda,k,i)$ as the increment of the utility where only λ_i is added by 10^{-k}. Thus we can have the following:

$$\delta(\lambda+I(k,i),\lambda)=U(\lambda+I(k,i))-U(\lambda)=\log(1+\lambda_i d_i+10^{-k}d_i)-\log(1+\lambda_i d_i) \quad (38)$$

Since $\log(1+\lambda_i d_i)$ is a monotone increasing function, we know $\delta(\lambda+I(k,i),\lambda)>0$.

For any given h and $j\neq i$, we can have

$$\begin{aligned}&\delta(\lambda+I(h,j)+I(k,i),\lambda+I(h,j))=\\&\log(1+\lambda_j d_j+10^{-h}d_j)+\log(1+\lambda_i d_i+10^{-k}d_i)-\\&\log(1+\lambda_j d_j+10^{-h}d_j)-\log(1+\lambda_i d_i)=\\&\delta(\lambda+I(k,i),\lambda)\end{aligned} \quad (39)$$

For any given h and $j=i$, we can get

$$\begin{aligned}&\delta(\lambda+I(h,i)+I(k,i),\lambda+I(h,i))=\\&\log(1+\lambda_i d_i+(10^{-h}+10^{-k})d_i)-\\&\log(1+\lambda_i d_i+10^{-h}d_i).\end{aligned}$$

Based on Lagrange mean value theorem, it can be known that

$$\begin{cases}\delta(\lambda+I(k,i),\lambda)=\dfrac{d_i 10^{-k}}{(1+x_1 d_i)\ln 2},\\ \delta(\lambda+I(h,i)+I(k,i),\lambda+I(h,i))=\dfrac{d_i 10^{-k}}{(1+x_2 d_i)\ln 2}\end{cases} \quad (40)$$

where $x_1\in(\lambda,\lambda+10^{-k})$ and $x_2\in(\lambda+10^{-h},\lambda+10^{-h}+10^{-k})$. And then we can have

$$\delta(\lambda+I(h,i)+I(k,i),\lambda+I(h,i))<\delta(\lambda+I(k,i),\lambda) \quad (41)$$

According to the definition of submodular set function[33], the objective function of Problem (9) is submodular. ■

In the following, we propose an increment per-cost-based greedy algorithm and analyze the approximation ratio.

Step 1. We enumerate all feasible solutions of cardinality 3.

Step 2. For each feasible solution, we find the offloading task with the largest increment per cost. Let the corresponding λ be increased by 10^{-k}, until Constraints (10)−(13) are not be satisfied. Then, k is the input parameter for precision, which can be defined by the users.

Step 3. Finally, the algorithm returns the offloading scheme with the largest utility among all the schemes derived from Step 2, thereby obtaining the final result.

Based on the preceding analysis, Algorithm 2 presents the pseudocode of calculating the feasible offloading scheme.

Corollary 1 For a given precision parameter k, let $Q_{\mathrm{OPT}}(k)$ denote the utility of the optimal solutions of Problem (9). We denote the utility derived from the solution of Algorithm 2 as $Q_{\mathrm{IAAOS}}(k)$. Then, the approximation ratio $r=\dfrac{Q_{\mathrm{OPT}}(k)}{Q_{\mathrm{IAAOS}}(k)}$ satisfies

$$r\leqslant 1+\frac{1}{\mathrm{e}-1} \quad (42)$$

Proof Based on Theorem 4, the objective function of Problem (9) is increasing and submodular. According to Ref. [34], the provided greedy-based approximation algorithm is with ratio bound $1+\frac{1}{\mathrm{e}-1}$. ■

In Corollary 1, we note that the approximation ratio is also related to precision parameter k. This condition means that for $\forall i\in\{1,2,\ldots,N\}$, $\lambda_i\in\{0,1\times10^{-k},2\times10^{-k},\ldots,10^k\times10^{-k}\}$.

N^3 feasible schemes exist, such that $\sum_{i=1}^{N}\lambda_i\leqslant 3\times10^{-k}$. According to Algorithm 2, the time complexity for enumerating all feasible solutions of cardinality 3 is $O(N^3)$. For each feasible scheme $\lambda(h)$, the complexity for greedily increasing $\lambda_i(h)$ by 10^{-k} is $O\left(\min\left\{\frac{N\tau}{c_{\min}},10^k\right\}\right)$, where $c_{\min}=\min\{c_i|i=1,2,\ldots,N\}$. Then, the cost is $O\left(\frac{N^4\tau}{c_{\min}}\right)$ from Lines 4 − 14. Based on the preceding analysis, the computation complexity is $O\left(\frac{N^4\tau}{c_{\min}}\right)$. The space complexity is

Algorithm 2 Increment-based approximation algorithm for offloading scheme

Input: $k, W, \tau, f_e, \mathcal{T}_i = < d_i, c_i, t_i, e_i, p_i, p_i^{\text{loc}} >$, for $i = 1, \ldots, N$

Output: $\lambda_1, \lambda_2, \ldots, \lambda_N$

Initialization, let $\lambda_i = 0$ for $i \in \{1, 2, \ldots, N\}$.
Enumerate all feasible solutions of cardinality 3, as the initial scheme set $S_3 = \{\lambda(1), \lambda(2), \ldots, \lambda(M)\}$. Such that for each scheme $\lambda(j) = \{\lambda_1(j), \lambda_2(j), \ldots, \lambda_N(j)\}$, $\sum_{i=1}^{N} \lambda_i(j) \leqslant 3 \times 10^{-k}$, and the constraints are satisfied, **i.e.** $\forall i \in \{1, \ldots, N\}$, $T_i(\lambda_i(j)) \leqslant 0, E_i(\lambda_i(j)) \leqslant 0$, and $\sum_{i=1}^{N} \lambda_i(j) c_i \leqslant \tau$.
$U_{\max} = \max\{U(\lambda(j)) | j \in \{1, 2, \ldots, M\}\}$;
for $j = 1$ to M **do**
  while $\sum_{i=1}^{N} \lambda_i(j) c_i + 10^{-k} \times \min\{c_1, \ldots, c_N\} \leqslant \tau$ **do**
    for $h = 1$ to N **do**
      $\varepsilon_h(j) = 0$;
      $\hat{\lambda}_h(j) = \lambda_h(j) + 10^{-k}$;
      if $\left\{ \frac{\hat{\lambda}_h(j) d_h}{r_h} + \frac{\hat{\lambda}_h(j) c_h}{f_e} + (1 - \hat{\lambda}_h(j)) \times \frac{c_h}{f_h} \leqslant t_h \right\} \wedge \left\{ \frac{\hat{\lambda}_h(j) d_h p_h}{r_h} + (1 - \hat{\lambda}_h(j)) \times p_h^{\text{loc}} c_h \leqslant e_h \right\}$ **then**
        if $\sum_{i=1}^{N} \lambda_i(j) c_i + 10^{-k} \times c_h \leqslant \tau$ **then**
          $\varepsilon_h(j) = \{U(\lambda(j) + I(k, h)) - U(\lambda(j))\} / 10^{-k} c_h$
        end if
      end if
    end for
    $g = \arg\max\{\varepsilon_h(j) | h \in \{1, 2, \ldots, N\}\}$;
    $\lambda(j) = \lambda(j) + I(k, g)$;
  end while
  $U_{\max} = \max(U_{\max}, U(\lambda(j)))$
end for
return $U_{\max}$ and the corresponding offloading scheme.

$O(N^3)$. In conclusion, the proposed algorithm can be implemented in time complexity of $O\left(\frac{N^4 \tau}{c_{\min}}\right)$ and space complexity of $O(N^3)$.

6 Performance Evaluation

In this section, a series of experiments have been conducted to evaluate the performance of given algorithms. We compare the performance of the algorithms with the exact solutions, which are calculated by convex optimization Matlab toolbox. In the simulation, the number of offloading tasks is set as 20, and the mobile CPU ability is randomly generated from $\{1, 1.1, \ldots, 2\}$ GHz. The size of the task follows uniform distribution over $(0, 2)$ MB. We set the channel bandwidth $B = 10$ MHz. The transmitting power is uniformly distributed on $(0, 1)$ W. For each i, t_i and e_i are randomly generated.

6.1 Gradient-based algorithm for utility-aware task offloading

The first group of experiments aims to investigate the performance of Gradient-based Algorithm for utility-aware Task Offloading (GATO).

Firstly, we evaluate the performance on utility when the computation capability varies. Figure 1 shows the relationship of the utility and server computation when $\delta = 0.15$. In Algorithm 1, the termination criterion for iteration is the ratio between the objective value of the latest $\lambda(\xi_U)$ and that of $\lambda(\xi_O)$. When the computation capability of the server is low, the constraint on τ makes the utility low and the number of iterations is high. When the computation capability of the server is high, the constraints on delay latency and energy consumption have a strong impact on the termination of iteration.

Figure 2 shows the relationship of the utility and server computation when the size of the task follows a normal distribution over $N(2, 0)$ MB and $\delta = 0.15$. The normal distribution generates a data size that is more uniform than that of uniform distribution. The objective value is higher because of the concavity.

Secondly, we investigate the performance of utility when the number of users varies. Figure 3 plots the relationship between the utility and number of mobile devices. When the number of mobile devices is small, GATO plays a major role in the utility gain. When the number of mobile devices is large, the constraint on bandwidth has a major impact on the utility. Owing to the concavity of the objective function,

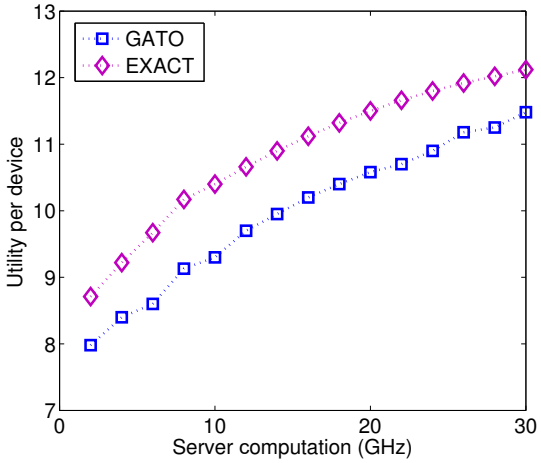


Fig. 1 Relationship between utility per device and server computation when $d_i \sim U(0, 2)$ MB.

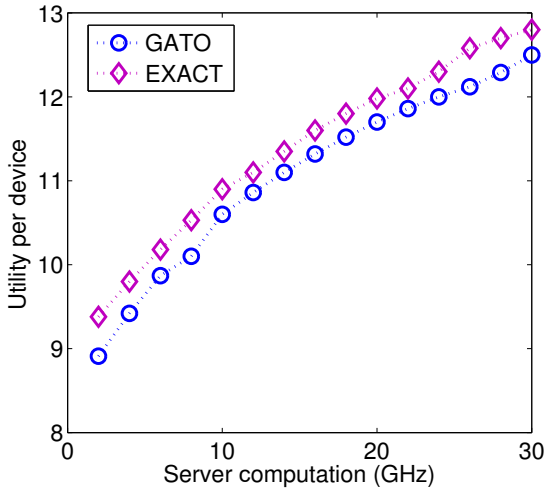


Fig. 2 Relationship between utility per device and server computation when $d_i \sim N(2,0)$ MB.

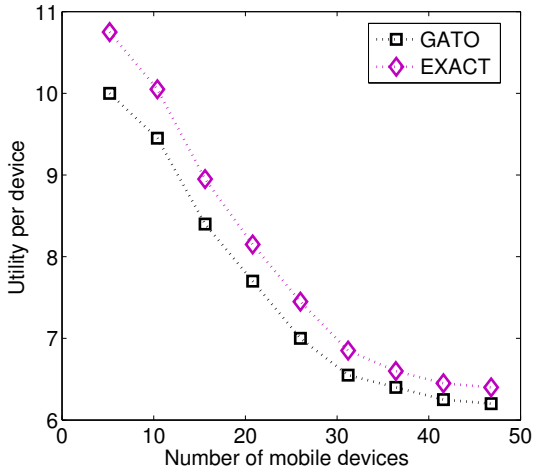


Fig. 3 Relationship of utility per device and number of mobile devices when $d_i \sim N(0,2)$ MB.

multiple small tasks have a larger utility than that of a large task if the amount of offloading data is the same. The constraints on delay latency and energy consumption play a minor role in the utility when the number of available offloading increases. As the termination criterion for iteration is the ratio of the utility, the number of iterations decreases when the utility increases slightly.

Thirdly, we evaluate the computing performance of GATO. In the experiments, Approximation Ratio (AR) is the ratio of $\sum_{i=1}^{N} \log(1+\lambda_i d_i)$ generated by the solutions of exact approach to that of the solution output by GATO. Figure 4 demonstrates the relationship between the AR and server computation where the step sizes are $\varphi = 0.01$ and $\varphi = 0.005$, respectively. Experimental results show that the utility value of

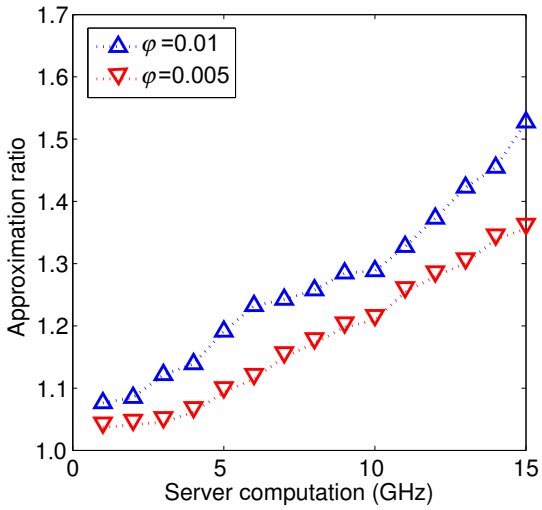


Fig. 4 Relationship of approximation ratio and server computation.

solutions returned by GATO is close to that of the optimal ones. For a given $\delta = 0.1$, the utility value of solutions returned with a small step size is large because the number of iterations is large. The ratio of the termination criterion equals $1+\delta$. The utility value is small when the capability of computation is low. Thus, the AR is extremely close to 1. The AR is larger when the server computation increases.

6.2 Increment-based approximation algorithm for offloading scheme

In the following, we evaluate the performance of IAAOS.

Firstly, we show the utility when the computation capability of server varies. According to Fig. 5, the utility increases when the server computation is enhanced. As the objective function is concave,

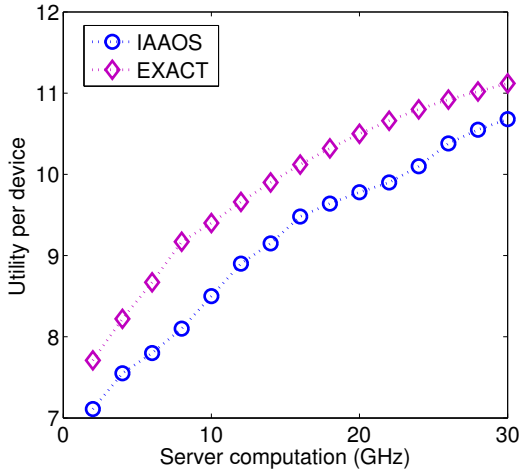


Fig. 5 Relationship between utility per device and server computation.

the increment is gentler with larger computation capability. Owing to the limited bandwidth, the increase is not obvious even if the server computation is more than the total number of CPU circles.

Figure 6 plots the relationship between the number of mobile devices and utility of the proposed approximation algorithm. As the computation capability of the server is given, the utility per device decreases when the number of mobile devices increases. Due to the concavity, the data size of offloading becomes fair when the number of mobile devices increases. Thus, the difference between the utility of approximation algorithm and that of the exact algorithm is minimal.

Then, we investigate the computing performance of IAAOS, and the correctness of the AR is verified. We prove that $\mathrm{AR} \leqslant 1 + \dfrac{1}{\mathrm{e}-1}$. In the experiments, approximation ratio is the ratio of $\sum_{i=1}^{N} \log(1+\lambda_i d_i)$ generated by the solutions of exact approach to that of the approximated solution output by IAAOS. Figure 7 demonstrates the relationship between the AR and server computation, where precision parameter $k = 3$. Experimental results show that the utility value of approximation results returned by IAAOS is very close to that of the optimal ones. The proposed approximation algorithms can achieve high accuracy.

7 Conclusion

By offloading part of the tasks to a nearby MEC cloud, MEC can decrease the computation latency and energy cost by optimizing task offloading and resource allocation. Many factors pose challenges to the offloading decision. To maximize the system utility, we defined the problem of finding the optimal offloading scheme. We studied the expectation of time complexity to derive the optimal scheme based on the KKT condition. We provided a greedy-based polynomial-time algorithm with $\left(1+\dfrac{1}{\mathrm{e}-1}\right)$-approximation ratio. Simulation results show the improved performance of the proposed algorithms for system utility.

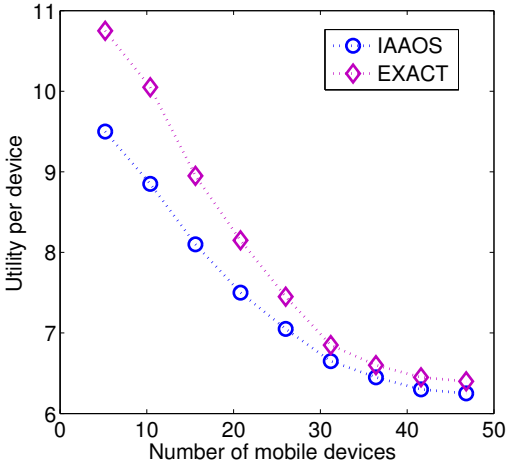


Fig. 6 Relationship of utility per device and number of mobile devices.

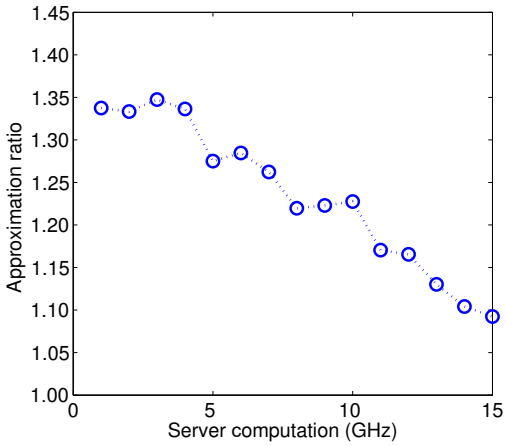


Fig. 7 Relationship of approximation ratio and server computation ($k = 3$).

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 61602084, 61761136019, U1808206, 61772112, and 61972083), the Post-Doctoral Science Foundation of China (No. 2016M600202), the Doctoral Scientific Research Foundation of Liaoning Province (No. 201601041), and the Fundamental Research Fund for the Central Universities (No. DUT19JC53).

References

[1] J. Ren, H. Guo, C. G. Xu, and Y. X. Zhang, Serving at the edge: A scalable IoT architecture based on transparent computing, *IEEE Network*, vol. 31, no. 5, pp. 96–105, 2017.

[2] Z. P. Cai, X. Zheng, and J. G. Yu, A differential-private framework for urban traffic flows estimation via taxi companies, *IEEE Trans. Ind. Inf.*, vol. 15, no. 12, pp. 6492–6499, 2019.

[3] Z. P. Cai and Z. B. He, Trading private range counting over big IoT data, in *Proc. 39th IEEE Int. Conf. Distributed Computing Systems*, Dallas, TX, USA, 2019.

[4] J. W. Shang, C. K. Wang, C. P. Wang, G. Y. Guo, and J. Qian, An attribute-based community search method with graph refining, *The Journal of Supercomputing*, doi: 10.1007/s11227-017-1976-z.

[5] X. Zheng and Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial IoTs, *IEEE Journal on Selected Areas in Communications*, doi: 10.1109/JSAC.2020.2980802.

[6] F. Firouzi, A. M. Firouzi, K. Mankodiya, M. Badaroglu, G. V. Merrett, P. Wong, and B. Farahani, Internet-of-things and big data for smarter healthcare: From device to architecture, applications and analytics, *Future Generation Computer Systems*, vol. 78, pp. 583–586, 2018.

[7] C. K. Wang, C. P. Wang, Z. Wang, X. J. Ye, J. X. Yu, and B. Wang, DeepDirect: Learning directions of social ties with edge-based network embedding, *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2277–2291, 2019.

[8] B. J. Zhou, J. Li, X. Y. Wang, Y. Gu, L. Xu, Y. Q. Hu, and L. H. Zhu, Online internet traffic monitoring system using spark streaming, *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 47–56, 2018.

[9] X. Zheng, Z. P. Cai, J. G. Yu, C. K. Wang, and Y. S. Li, Follow but no track: Privacy preserved profile publishing in cyber-physical social systems, *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1868–1878, 2017.

[10] K. Zhang, S. P. Leng, Y. J. He, S. Maharjan, and Y. Zhang, Mobile edge computing and networking for green and low-latency internet of things, *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 39–45, 2018.

[11] X. L. Fang, Z. P. Cai, W. Y. Tang, G. C. Luo, L. J. Zhou, R. Bi, and H. Gao, Job scheduling to minimize total completion time on multiple edge servers, *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2019.2958281.

[12] X. C. Lyu, H. Tian, C. Sengul, and P. Zhang, Multiuser joint task offloading and resource optimization in proximate clouds, *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.

[13] X. Y. Tao, K. Ota, M. X. Dong, H. Qi, and K. Q. Li, Performance guaranteed computation offloading for mobile-edge cloud computing, *IEEE Wirel. Commun. Lett.*, vol. 6, no. 6, pp. 774–777, 2017.

[14] F. Wang, J. Xu, X. Wang, and S. G. Cui, Joint offloading and computing optimization in wireless powered mobile-edge computing systems, *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 1784–1797, 2018.

[15] Q. Y. Meng, K. Wang, X. M. He, and M. Y. Guo, QoE-driven big data management in pervasive edge computing environment, *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 222–233, 2018.

[16] X. Zheng, Z. P. Cai, J. Z. Li, and H. Gao, A study on application-aware scheduling in wireless networks, *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1787–1801, 2017.

[17] H. Li, K. Ota, and M. X. Dong, Learning IoT in edge: Deep learning for the internet of things with edge computing, *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[18] Z. P. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2018.2830307.

[19] Z. J. Duan, W. Li, and Z. P. Cai, Distributed auctions for task assignment and scheduling in mobile crowdsensing systems, in *Proc. 37^{th} Int. Conf. Distributed Computing Systems*, Atlanta, GA, USA, 2017, pp. 635–644.

[20] Z. J. Duan, W. Li, X. Zheng, and Z. P. Cai, Mutual-preference driven truthful auction mechanism in mobile crowdsensing, in *Proc. 39^{th} IEEE Int. Conf. Distributed Computing Systems*, Dallas, TX, USA, 2019.

[21] Y. Z. Zhou, D. Zhang, and N. X. Xiong, Post-cloud computing paradigms: A survey and comparison, *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 714–732, 2017.

[22] J. Liu, Y. Y. Mao, J. Zhang, and K. B. Letaief, Delay-optimal computation task scheduling for mobile-edge computing systems, in *IEEE International Symposium on Information Theory*, Barcelona, Spain, 2016, pp. 1451–1455.

[23] J. K. Ren, G. D. Yu, Y. L. Cai, and Y. H. He, Latency optimization for resource allocation in mobile-edge computation offloading, *IEEE Trans. Wirel. Commun.*, vol. 17, no. 8, pp. 5506–5519, 2018.

[24] L. N. Liu, X. Chen, Z. M. Lu, L. H. Wang, and X. M. Wen, Mobile-edge computing framework with data compression for wireless network in energy internet, *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 271–280, 2019.

[25] L. Lei, H. J. Xu, X. Xiong, K. Zheng, and W. Xiang, Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system, *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5345–5362, 2019.

[26] T. X. Zhu, T. Shi, J. Z. Li, Z. P. Cai, and X. Zhou, Task scheduling in deadline-aware mobile edge computing systems, *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4854–4866, 2019.

[27] Q. Y. Wang, S. T. Guo, J. D. Liu, and Y. Y. Yang, Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing, *Sustain. Comput.: Inform. Syst.*, vol. 21, pp. 154–164, 2019.

[28] A. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge University Press, 2005.

[29] X. C. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, Optimal schedule of mobile edge computing for internet of things using partial information, *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, 2017.

[30] L. Tang and H. Chen, Joint pricing and capacity planning in the IaaS cloud market, *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 57–70, 2017.

[31] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, Dynamic resource allocation and power management in virtualized data centers, in *IEEE Network Operations and Management Symposium*, Osaka, Japan, 2010, pp. 479–

486.

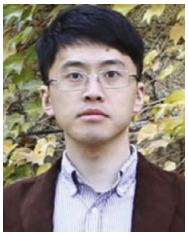
[32] F. M. Liu, Z. Zhou, H. Jin, B. Li, B. C. Li, and H. B. Jiang, On arbitrating the power-performance tradeoff in SaaS clouds, *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2648–2658, 2014.

[33] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, An analysis of approximations for maximizing submodular set functions – I, *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

[34] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, *Oper. Res. Lett.*, vol. 32, no. 1, pp. 41–43, 2004.

Ran Bi received the PhD, MS, and BS degrees from Harbin Institute of Technology, China in 2015, 2010, and 2008, respectively. She is currently an associate professor at the School of Computer Science and Technology, Dalian University of Technology, China. Her research interests include wireless networking, sensory data management, internet of things, and edge computing.

Qian Liu received the BS and MS degrees from Dalian University of Technology, Dalian, China in 2006 and 2009, respectively, and the PhD degree from the State University of New York at Buffalo (SUNY-Buffalo), Buffalo, NY, USA in 2013. She was a postdoctoral fellow at the Ubiquitous Multimedia Laboratory, SUNY-Buffalo from 2013 to 2015. She received the Alexander von Humboldt Fellowship in 2015, and was a postdoctoral fellow at the Chair of Media Technology and the Chair of Communication Networks, Technical University of Munich, from 2016 to 2017. She is now an associate professor at the Department of Computer Science and Technology, Dalian University of Technology, China. Her current research interests include multimedia transmission over MIMO systems, IEEE 802.11 wireless networks and LTE networks, device-to-device communication, energy-aware multimedia delivery, and the tactile internet. She received the best paper runner-up award at the 2012 International Conference on Complex Medical Engineering and was in the finalist for the best student paper award at the 2011 IEEE International Symposium on Circuits and Systems.



Jiankang Ren received the BSc, ME, and PhD degrees in computer science from Dalian University of Technology in 2008, 2011, and 2015, respectively. He was a visiting scholar at the Computer and Information Science Department, University of Pennsylvania, from September 2013 to September 2014. He is currently an associate professor at the School of Computer Science and Technology, Dalian University of Technology. His research interests include Cyber-Physical Systems (CPS), cloud computing, and computational intelligence.

Guozhen Tan received the BS degree from Shenyang University of Technology, Shenyang, China in 1982, the MS degree in computer software and theory from Harbin Institute of Technology, Harbin, China in 1992, and the PhD degree in computer applied technology from Dalian University of Technology, Dalian, China in 2003. He was a visiting scholar at the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, USA from January 2007 to January 2008. He has been a professor at the School of Computer Science and Technology, Dalian University of Technology and the director of Engineering and Technology Research Center for the Internet of Things and Collaborative Sensing, Liaoning Province. His research interests include internet of things, vehicular networking, and intelligent transportation control.