Optimizing Energy-Latency Tradeoff for Computation Offloading in SDIN-Enabled MEC-based IIoT

Xinchang Zhang[1], Changsen Xia[2], Tinghuai Ma[2], Lejun Zhang[3] , and Zilong Jin[2, 4*]
[1] Nanjing University of Information Science and Technology
Nanjing 210044, China
[e-mail: xinzc@nuist.edu.cn]
[2] School of Software, Nanjing University of Information Science and Technology, Nanjing
[e-mail: changsenxia@163.com, thma@nuist.edu.cn, zljin@nuist.edu.cn]
[3] College of Information Engineering, Yangzhou University
Yangzhou 225127, China
[e-mail: zhanglejun@yzu.edu.cn]
[4] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET)
Nanjing University of Information Science and Technology, Nanjing 210044, China
*Corresponding author: Zilong Jin

Received January 20, 2022; revised September 6, 2022; accepted September 23, 2022; published December 31, 2022

Abstract

With the aim of tackling the contradiction between computation intensive industrial applications and resource-weak Edge Devices (EDs) in Industrial Internet of Things (IIoT), a novel computation task offloading scheme in SDIN-enabled MEC based IIoT is proposed in this paper. With the aim of reducing the task accomplished latency and energy consumption of EDs, a joint optimization method is proposed for optimizing the local CPU-cycle frequency, offloading decision, and wireless and computation resources allocation jointly. Based on the optimization, the task offloading problem is formulated into a Mixed Integer Nonlinear Programming (MINLP) problem which is a large-scale NP-hard problem. In order to solve this problem in an accessible time complexity, a sub-optimal algorithm GPCOA, which is based on hybrid evolutionary computation, is proposed. Outcomes of emulation revel that the proposed method outperforms other baseline methods, and the optimization result shows that the latency-related weight is efficient for reducing the task execution delay and improving the energy efficiency.

Keywords: SDIN, Edge computing, Computation offloading, Evolution Computation.

This work is sponsored by the National Natural Science Foundation of China under grant number No. 42175194 and and the Project through the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institution.

http://doi.org/10.3837/tiis.2022.12.017 **ISSN : 1976-7277**

1. Introduction

With the development of industry 4.0, Industrial Internet of things (IIoT) has become a promising technology to provide a strong and efficient backbone in intelligent manufacturing, in which there are many computation-intensive and latency-sensitive applications [1]. For these applications, the Mobile Edge Computing technology is introduced to enhance computation performances of Edge Devices (EDs). The MEC servers are closer than cloud center to EDs, thus it can ease the Internet traffic load, and be efficient for reducing communication latency and energy consumption through computation offloading. The MEC servers are also efficient for providing IaaS and PaaS services to guarantee real-time and energy-efficient task processing [2]. However, even the MEC technology can provide the above benefits, the protocol stacks in EDs are not efficient. Typically, in EDs, the control plane and data plane are independent (the control plane is responsible for controlling data forwarding, and the data plane completes data forwarding), thus the protocol overhead will increase the delay of task offloading. e.g., When a new device is connected to a IIoT environment, the control plane should be configured separately to complete the network protocol processing and computation functions [3], which will influence the real-time performance of IIoT applications.

Therefore, in IIoT, Software Defined Industrial Network (SDIN) architecture is utilized to separates the control plane from the EDs. Based on the SDIN architecture the EDs contain data plane, and the control plane is independent of the EDs. The SDIN architecture reduces the limitation of the network hardware, and realizes the automatic deployment and adjustment of the networks, reducing the delay of reconfiguring the control plane when new EDs join the IIoT network.

This paper considers SDIN-enabled MEC based IIoT network which combines the MEC and SDIN architecture. The network is similar to clustered network verified to be the network topology that is suitable for the IIoT environment. Due to the strict requirements of real-time and endurance in the IIoT, computation tasks are generated by industrial applications need to be accomplished in real-time. Thus, it is necessary to investigate the tradeoff between latency of accomplishing computation tasks (i.e., computation delay in local computing or latency of communication and computation in edge computing) and energy consumption for the optimal computation offloading in the SDIN-enabled MEC network.

The existing issues can be summarized as: 1) Energy consumption and latency are related to every process of completing the task (i.e., local computing, wireless communication and edge computing), thus it is necessary to jointly optimize each phase of computation offloading; 2) Resource allocation in industrial networks generally cannot meet the QoS requirements of heterogeneous networks, while the superiority of SDIN is that the controller can have a global view of network resources, developing offloading strategies easier to guarantee diverse QoS demand.

With the aim of addressing the above problems, a tradeoff optimizing method is proposed in this paper which can jointly optimize offloading decision, local computing frequency, channel assignment, and transmission power while considering the limitation of computation resource. Based on the optimization, the proposed task offloading not only solves the issue of whether offload, but also tackles the tough issue of which MEC server is the optimal choice for task offloading. The contributions of this paper are listed below.

- In order to realize the real-time processing of computation tasks in IIoT, the computation offloading model is mathematically developed in SDIN-enabled MEC based IIoT, including wireless communication model and task computation model.

- A latency-aware offloading scheme is proposed in this paper, which optimize the offloading decision, local frequency, wireless and computation resources allocation jointly.
- The issue of optimal offloading scheme is mathematically expressed as a MINLP problem which is a large-scale NP-hard problem. A sub-optimization scheme GPCOA is proposed to tackle this issue in an accessible complexity.

In proposed optimization algorithm, the variable decomposition approach is utilized to ensure that the original problem is split and transformed into two parts, both of which are independent: 1) local overhead and 2) edge overhead. A local frequency scheduling is proposed first to obtain the optimal local overhead, based on which, the remaining variables will be optimized by combining GA and PSO.

The organization of the rest of this paper is as the following. Section II review the related work. The system model of SDIN-enabled MEC network is represented in section III. In section Ⅳ, we formulate the optimization problem into a MINLP problem mathematically and analyze the problem from the point of problem complexity. The proposed GPCOA is shown in section Ⅴ. The simulation outcomes are demonstrated in section Ⅵ and this paper is concluded in Ⅶ.

2. Related Work

MEC can reduce the delay caused by core network congestion and data retransmission, which also cause additional energy consumptions, compared to the tasks processed in central cloud computing servers in IIoT. Computation offloading is a key technology in MEC, attracting more attentions in academia and industry communities. The energy consumption of EDs and delay of completing the computation task.

Starting from the optimization of energy consumption in MEC, there are many research efforts utilizing heuristic algorithms [4] and machine learning algorithms [5] to optimize the related optimization problem in computation offloading and resource allocation to optimize the energy consumption in MEC based IIoT. In addition, with the aim of reducing the task accomplishing latency, some research efforts try to reduce the computation delay [6] and communication delay [7] through different optimization techniques.

After the separate optimization of energy consumption and task accomplishing delay, there are many researches focusing on the joint optimization of both of important indicator based on Lyapunov optimization and convex optimization [8], which jointly optimize the offloading decision, local CPU frequency and resource allocation.

Nevertheless, with the development of intelligent factory, pure MEC may no longer meet the delay and energy consumption requirements of heterogeneous IIoT networks. SDIN has been supposed to be a promising architecture, which can improve the QoS of the industrial applications by flexibly decoupling data plane and the control plane to control the industrial network centrally. At present, combining SDIN with IIoT is recognized as an efficient way to optimize data transmission delay [9].

Different from the existing studies, a comprehensive computation offloading scheme in SDIN-enabled MEC based IIoT is proposed in this paper to exploit the tradeoff between energy consumption of EDs and delay of accomplishing corresponding tasks of EDs via optimizing local frequency, offloading decision, wireless and computation resources allocation, where the utilized weighting factor is delimited on the basis of the predicted latency

of the corresponding computation task. The optimization problem is proved to be a large-scale and also a NP-hard problem. In order to tackle such a complex problem, a sub-optimal algorithm GPCOA is proposed, and the details are in Sec.V.

3. System Model

3.1 SDIN-enabled MEC network

Consider the SDIN-enabled MEC based IIoT comprising a macro base station (MBS) and M small base station(SBSs) and U EDs as demonstrated in **Fig. 1**. For presentation, the BS (including SBSs and MBS) set and the EDs set are denoted as $N_B = \{1,2 \dots, j, \dots, M, M+1\}$ and $U = \{1,2, \dots, i, \dots, U\}$,respectively. With the aim of reusing spectrum, the multiple SBSs are assumed that operate in the same frequency band, which means that a small cell will interfere with the neighbor cells. The bandwidth is partitioned into N channels and the set of channels is denoted as $C = \{1,2 \dots, n, \dots N\}$. There are MEC servers equipped in MBS and SBSs, and the computation capacity of each SBS is variant due to the heterogeneity considered in the proposed SDIN-enabled MEC based IIoT. In this model, ED i in cell j is assumed to have a computation task $T_{i,j} = \{d_{i,j}, c_{i,j}, \tau_{i,j}^{max}, t_{i,j}^{pre}\}$ to accomplish. For task $T_{i,j}$, $d_{i,j}$ represents for the size of input data, $c_{i,j}$ denotes the total number of CPU cycles required to complete the computation task, $\tau_{i,j}^{max}$ denotes the maximum tolerance latency which is required by corresponding computation task, $t_{i,j}^{pre}$ represents the predicted latency of task which can be acquired by machine learning or other technology [10, 11]. We assume that the predicted latency of corresponding task is known in this paper. With aim of accomplishing the generated task, the EDs can either execute tasks locally, or offload the computation task to MEC servers. It is up to SDIN controller integrated in MBS to determine which MEC server to execute the offloading task.

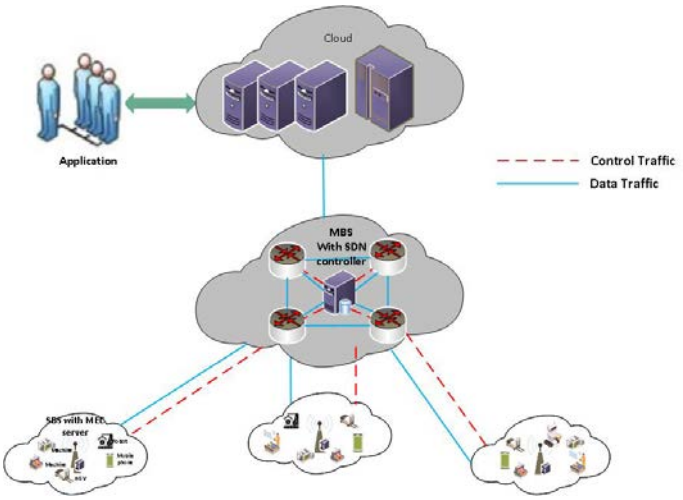

Fig. 1. SDIN-enabled MEC network

Computation offloading in SDN-enabled edge computing primarily includes three phases: 1) transmitting raw data via the wireless channel, 2) MBS with SDN controller decides which MEC server would be assigned to execute the task, 3) executing the task on ME server, 4) downloading outcomes from the servers back to the EDs. The total cost of the fourth phase are

not considered in this paper because of the truth that the size of resulting data is much smaller than input data and the bandwidth of the download is much larger than that of the upload [8].

3.2 Communication Model

In this subsection, a communication model is presented to describe the first phase of computation offloading.

Here, the offloading decision of ED i is defined as a binary variable $a_{i,j}$, which indicates not only the task $T_{i,j}$ whether offload but also where to offload decided by the SDIN controller in MBS server. In the case of the task of ED i is offloaded to MEC server, we have $a_{i,j} = 1$, otherwise, $a_{i,j} = 0$. Particularly, it satisfies $\sum_{j \in N_B} a_{i,j} \leq 1$, indicating that one edge device can offload task to no more than one MEC server equipped in SBSs or MBS at the same time.

In case of that ED i accesses the SBS on channel n, the achievable uplink transmission rate can be represented as:

$$r_{i,j,n} = wlog_2(1 + \frac{p_{i,j,n}h_{i,j,n}}{\sigma^2 + I_{i,j,n}}) \tag{1}$$

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 EDi and SBS j on channel n, respectively. σ^2 represents for the background noise, which is assumed to follow Gaussian distribution. $I_{i,j,n}$ represents for the cahnnel interference of ED i in cell suffering from other EDs in the adjacent cells on the same channel, which can be expressed as

$$\mathrm{I}_{\mathrm{i,j,n}} = \sum_{k=l}^{U_l} \sum_{l=1, l \neq i}^{M} b_{k,l,n} p_{k,l,n} h_{k,l,n}^{j} \tag{2}$$

where l denotes the $l-th$ except the $j-th$ SBS, the channel gain from ED k in cell l to cell j on channel n id denoted by $h_{k,l,n}^{j}$, and U_l represents for the number of the EDs in the area covered by SBS l. Therefore, the total uplink transmission rate for ED i can be expressed as

$$r_{i,j} = \sum_{n=1}^{N} b_{i,j,n} r_{i,j,n}, \tag{3}$$

where $b_{i,j,n} \in \{0,1\}$. When channel k is allocated to ED i in cell j, $b_{i,j,n}$=1, otherwise, $b_{i,j,n} = 0$.

3.3 Computation Model

The computation model is presented in this subsection, describing how the computation task either executed on the EDs locally or offloaded.

1) Local computing: we denote the local computation capacity (i.e., CPU $cycle/second$ (HZ)) of the ED i as $f_{i,j}^{L}$, in the case of the computation task is executed locally, the delay can be expressed as

$$t_{i,j}^{L} = \frac{c_{i,j}}{f_{i,j}^{L}}, \tag{4}$$

The energy consumption of the ED can be represented as

$$e_{i,j}^{L} = k'(f_{i,j}^{L})^2 c_{i,j}, \tag{5}$$

Where k' is a variable relying on the architecture of chip, the value of which equals 10^{-26} [12].

Through the local computing model above, we can find out that both delay and energy consumption of local computing are affected by the local computation ability $f_{i,j}^{L}$. Thus, in this paper, it is allowed that to the $f_{i,j}^{L}$ can be schemed through dynamic voltage and frequency

scaling technology (DVFS).

2) Edge Computing: when the offloading decisions have been made by SDIN controller at the MBS, the computation task will be transferred to the aimed server equipped in the SBSs or MBS (i.e., the first two period). The computation task then will be executed on the target MEC server (the third stage). On the basis of the model demonstrated above, the latency and the energy consumption due to transmission can be given below, respectively.

$$t_{i,j}^{E} = \frac{d_{i,j}}{r_{i,j}} + \frac{c_{i,j}}{F_{i,j}} \tag{6}$$

and

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

where $F_{i,j}$ represents for the computation resource assigned to ED i from MEC server j.

The finite computation capacity of various MEC server, which is defined as F_j, when j= M+1, F_{M+1} represents for the limited computation resources of the MEC server equipped in MBS. Both delay and energy consumption are fatal for EDs in the process of completing the computation task. Thus, a weighting factor $w_{i,j}$ ($w_{i,j} \in [0,1]$) have been introduced to exploit the tradeoff between both of them in SDIN-enabled edge computing, which can be defined by various EDs for different purpose. By means of revising the value of weighting factor, we can save more energy or reduce more latency. Besides, we bring the prediction latency $t_{i,j}^{pre}$ into the definition of indicator in our model, which can be expressed as

$$w'_{i,j} = w_{i,j} r_{i,j}^{t} \tag{8}$$

where $r_{i,j}^{t} = \frac{\tau_{i,j}^{max} - t_{i,j}^{pre}}{\tau_{i,j}^{max}}$. Different from $w_{i,j}$, $r_{i,j}^{t}$, which is a definite value related to delay, reflects the significance of latency optimization for each computation task T_{ij}, the smaller the value of $r_{i,j}^{t}$,the more important the computation task $T_{i,j}$ is to focus on optimization of latency. Given the offloading decision of corresponding ED i, ED i is allocated with part of computation resource $F_{i,j}$ from servers, e the computation resource allocation set $F = \{F_{i,j}, j \in N_B, i \in U\}$ can be obtained. Taking the limited and various computation resource of variant MEC server into consideration, F must meet limit: $\sum_{i \in U} F_{i,j} \leq F_i, i \in N_B$, meaning that the computation resource required to complete the task of EDs must under the computation capacity of corresponding server.

Through the analysis above, the total cost of ED i $S_{i,j}^{L}$ can be defined as

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

where α, which is used to realize the combination of two variables with variant units, can be expressed as the ratio of average energy consumption and average latency of all tasks. We let $w_{i,j}^{t} = 1 - w'_{i,j}$ and $w_{i,j}^{e} = w'_{i,j}$, and they indicate the weightings of execution energy consumption and latency, respectively. Thus, the total cost of single ED i can be replaced as $S_{i,j}^{L} = w_{i,j}^{t} t_{i,j}^{L} + w_{i,j}^{e} e_{i,j}^{L}$.

Analogously, the cost of the computation task executed remotely can be represented as

$$S_{i,j}^{E} = w_{i,j}^{t} t_{i,j}^{E} + w_{i,j}^{e} e_{i,j}^{E} \tag{10}$$

Therefore, the total cost of ED i can be denoted by

$$S_{i,j} = a_{i,j} S_{i,j}^{E} + (1 - a_{i,j}) S_{i,j}^{L}. \tag{11}$$

4. Problem Formulation and Analysis

4.1 Problem Formulation

For given the set of offloading decisions A, the set of channel assignment B and the set of computation resources allocation F, we obtain the total cost for accomplishing the computation task of ED i. The main problem needed to be addressed in this paper is shown in (12).

$$
\begin{aligned}
P_0: \quad & \min_{a,b,p,F,f} = \sum_{i=1}^{U_j}\sum_{j=1}^{M}\{a_{i,j}[w_{i,j}^{t}(\frac{d_{i,j}}{r_{i,j}}+\frac{c_{i,j}}{F_{i,j}})+w_{i,j}^{e}\mathrm{e}_{i,j}^{E}]\} \\
& \quad +\sum_{i=1}^{U_j}\sum_{j=1}^{M}\left\{\left(1-a_{i,j}\right)\left[w_{i,j}^{t}\frac{c_{i,j}}{f_{i,j}^{l}}+w_{i,j}^{e}\kappa\left(f_{i,j}^{l}\right)^{2}c_{i,j}\right]\right\} \\
s.t. \quad & \mathrm{C1}: a_{i,j}\left(\frac{d_{i,j}}{r_{i,j}}+\frac{c_{i,j}}{F_{i,j}}\right)+\left(1-a_{i,j}\right)\frac{c_{i,j}}{f_{i,i}^{l}}\le\tau_{i,j}^{\max} \\
& \mathrm{C2}: a_{i,j}\sum_{n=1}^{N}b_{i,j,n}p_{i,j,n}\frac{d_{i,j}}{r_{i,j}}+\left(1-a_{i,j}\right)k\left(f_{i,j}^{l}\right)^{2}c_{i,j}\le E_{i,j}^{\max} \\
& \mathrm{C3}: f_{\min}^{L}\le f_{i,j}^{L}\le f_{\max}^{L} \\
& \mathrm{C4}: \sum\nolimits_{j\in N_B}a_{i,j}\le 1 \\
& \mathrm{C5}: 0\le\sum_{n=1}^{N}b_{i,j,n}p_{i,j,n}\le p_{\max} \\
& \mathrm{C6}: \sum_{k=1}^{U_l}\sum_{l=1,l\neq j}^{M}a_{k,l,n}p_{k,l,n}h_{k,l,n}\le I_{th} \quad \forall j,n \\
& \mathrm{C7}: \sum_{j\in N_B}\sum_{n=1}^{N}a_{i,j}b_{i,j,n}\le 1 \\
& \mathrm{C8}: \mathrm{a}_{i,j}\in\{0,1\};\ \mathrm{b}_{i,j,n}\in\{0,1\} \\
& \mathrm{C9}: 0\le\mathrm{F}_{i,j}\le F_j
\end{aligned}
\tag{12}
$$

C1 and C2 represents for the constraints of latency and energy consumption of corresponding task, respectively. C3 limits the range of local frequency and C4 ensures that each ED can only offload the computation task to no more than one server. C5 ensures the maximum transmission power of ED i in cell j . C6 indicates that the impact of offloading task on SENB caused by other EDs must under the predefined threshold. C7 means that each ED can be assigned no more than one channel. C8 indicate that binary variables are utilized to demonstrate offloading decision and channel assignment. C9 represents that the computation resource assigned to ED i by MEC server j cannot exceed its capacity.

4.2 Problem Analysis

The formulation P_0 in (12) aims to reduce the total cost of all EDs, through optimizing offloading decision, channel allocation, transmission power, and computation resource

assignment. Since computation offloading decision $a_{i,j}$ and channel allocation indicator $b_{i,j,n}$ are binary variables, the formulated problem P_0 is a mixed inter non-linear programming problem (MINLP), which is a NP-hard problem, making P_0 non-smooth and non-continuous, and the optimal solution of P_0 is intractable.

In addition, in SDIN-enabled edge computing networks, this may be a large scale MINLP problem. As IIoT evolving, the numbers of both EDs and SBSs are in exponential growth, resulting in higher complexity of the formulated problem. Thus, it is impractical to do an exhausting search.

5. Proposed Algorithm

With above analyses, with the aim of getting the optimal solution to problem P_0, local CPU-cycle frequency, the computation offloading decision, optimization of transmission power and resource (i.e., wireless and computation resources) allocation need to be considered concurrently. In SDIN-enabled edge computing network, the numbers of EDs and SBSs may up to hundreds and tens, respectively, resulting in the extremely high complexity of computation. Thus, low-complexity suboptimal algorithms are needed more than ever.

Despite that P_0 is non-convex, the problem P_0 is split and transform into two parts: 1) local overhead and 2) edge overhead.

A suboptimal scheme based on hybrid evolutionary computation, which combine GA and PSO, named as GPCOA is proposed. The proposed GPCOA consists of two parts: The first part aims to obtain the optimal local CPU-cycle frequency to reduce local computing overhead, then the GPCOA is proposed to obtain a sub-optimal solution to the optimization problem in an accessible time complexity.

The proposed GPCOA is a meta-heuristic algorithm eventually, and the reason why we call it sub-optimal algorithm is that some operations in heuristic algorithm, such as selection, crossover and mutation, have random nature, and is easy to trap into local optimal solution.

Algorithm 1 GPCOA

1: **Input**: Population size K, the maximum number of iterations: T, convergence criteria ε, the parameters of evolution computing: p_m, p_c, T_1 and ω, c_1, c_2,T2, respectively.
2: **Outcome**: The historical optimal individual.
3: **Initialization**: Each individual of population is initialized via (17) and the value of t equal 0.
4: **Step 1**: Obtain the optimal local frequency f^*,f_l and f_h via (14).
5: **Step 2**:if t>T or *Fitness(t)-Fitness(t-1)* <= ε. break; else conduct next step.
6: **Step 3**: Perform GA as shown in Algorithm 2.
7: **Step 4**: Conduct PSO.
8: **Step 5**: t=t+1, back to step 2.

5.1 The Process of GPCOA

All tasks generated by devices can either be executed locally or remotely via computation offloading, and one of the variable we need optimize is the offloading decision. The reason why we decouple the origin problem into two problem is that the formulated problem involve not only the cost of local computing but also the remote computing via computation offloading. Therefore, We can optimize the local computation to get an overall optimal. In order to tackle

such a large-scale and NP-hard problem, we first optimize the overhead of local computing by scheduling the local frequency of EDs, taking the constraints C1-C3 in formulated problem p_0 into consideration, the minimization of local computing overhead can be formulated below:

$$\begin{aligned} \text{P1: } & \min_{f} w_{i,j}^{t} \frac{c_{i,j}}{f_{i,j}^{l}} + w_{i,j}^{e} \kappa \left(f_{i,j}^{l} \right)^{2} c_{i,j} \\ \text{s.t. } & C1: \frac{c_{i,j}}{f_{i,j}^{l}} \le T_{i,j}^{\max} \\ & C2: \kappa \left(f_{i,j}^{l} \right)^{2} c_{i,j} \le E_{i,j}^{\max} \\ & C3: f_{\min^{L}} \le f_{i,j}^{l} \le f_{\max^{L}} \end{aligned} \tag{13}$$

We can see that the overhead of ED is only related to $f_{i,j}^{l}$. We define $S_{i,j}^{L}(f_{i,j}^{l}) = w_{i,j}^{t}(c_{i,j}/f_{i,j}^{l}) + w_{i,j}^{e}k'(f_{i,j}^{l})^{2}c_{i,j}$, $S_{i,j}^{L}(f_{i,j}^{l})$ is taken the derivative with respect to $f_{i,j}^{l}$, and let it to be zero, thus, we can obtain $f^{*} = \sqrt[3]{(w_{i,j}^{t}/2w_{i,j}^{e}k)}$. For $f_{i,j}^{l} < f^{*}$, the monotony of $S_{i,j}^{L}$ is the opposite of the monotony of $f_{i,j}^{l}$, otherwise, the monotony of them is the same. In addition, from C1-C2, we have $f_{i,j}^{l} \ge c_{i,j}/T_{i,j}^{max} = f_{l}$ and $f_{i,j}^{l} \le \sqrt{E_{i,j}^{max}/k'c_{i,j}} = f_{h}$, respectively. Combining that result with the constraint C3, we can obtain$f_{l}' = max\{f_{min}^{L}, (c_{i,j}/T_{i,j}^{max})\}$ and $f_{h}' = min\{f_{max}^{L}, \sqrt{E_{i,j}^{max}/k'c_{i,j}}\}$, respectively. Thus, the optimal $S_{i,j}^{L}(f_{i,j}^{l})$ can be calculated as the following:

$$S_{i,j}^{L^{*}}\left(f_{i,j}^{l} \right) = \begin{cases} S_{i,j}^{L}\left(f_{l}^{'} \right), & f^{*} \le f_{l}^{'} \\ S_{i,j}^{L}\left(f^{*} \right), & f_{l}^{'} < f^{*} \le f_{h}^{'} \\ S_{i,j}^{L}\left(f_{h}^{'} \right), & f^{*} > f_{h}^{'} \end{cases} \tag{14}$$

After getting $S_{i,j}^{L*}(f_{i,j}^{l})$, the P_0 can be transformed into the form of the problem P_2 as follows:

$$\begin{aligned} P2: \min_{a,b,p,F} & \sum_{i=1}^{U_j} \sum_{j=1}^{M} \left\{ a_{i,j} \left[w_{i,j}^{t} \left(\frac{d_{i,j}}{r_{i,j}} + \frac{c_{i,j}}{f^{C}} \right) + w_{i,j}^{e} \times \sum_{n=1}^{N} b_{i,j,n} p_{i,j,n} \frac{d_{i,j}}{r_{i,j}} \right] \right\} \\ & + \sum_{i=1}^{U_j} \sum_{j=1}^{M} \left\{ \left(1 - a_{i,j} \right) S_{i,j}^{L^{*}}\left(f_{i,j}^{l} \right) \right\} \\ s.t. & C1, C2, C4, C5, C6, C7, C8, C9 \end{aligned} \tag{15}$$

Nevertheless, the new transformed problem is still a large-scale NP-hard issue.

There are many meta-heuristic algorithms can tackle MINLP problem. GA adopt the idea of evolution, its effect on MIP problem has been proved. However, it converges lower than other algorithms. The characteristics of PSO, which is easy to conducted and the speed of convergence is fast, are the opposite. Therefore, PSO is skilled at hunting for the local region but is weak in ferreting about the global zone. The performance of GPCOA is promoted via utilizing PSO as a complement to GA. We utilize GA for a rough global search while PSO is utilized for more detailed local search as demonstrated in Algorithm 1.

In GPCOA, we first obtain the optimal local computing overhead and initialize the population, then the original solutions will be optimized by two variant evolution computing, each of which does good in different aspect, iteratively until algorithm convergence or the value of t equal *T*. In each iteration of GPCOA, the individuals are operated by GA for T_1 iteration, then the optimized results conducted by GA are input into the PSO and the parameters of PSO will be initialized. After particles optimized by PSO for T_2 iterations, one iteration of GPCOA is done. The outcome of PSO will be optimized by another evolution computing again. Namely, the outcome of GA and algorithm 3 are input and output to each other except the first iteration of GPCOA. The initialization of algorithm 2 is expressed as (17). Two evolution computing algorithms utilized in GPCOA are described in details in the following subsections.

5.2 GA

GA is a metaheuristic algorithm, which tackle complex problem by mimicking biological evolution. The only requirement of GA is that the problem to be tackled is computable. GA optimizes the initial solutions via some genetic operations (i.e., selection, crossover and mutation) until that an acceptable solution is obtained or reach the maximum number of iterations. In particular, the crossover and mutation operation of GA can amplify the search region, thus, GA does well in searching the global domain.

The GA utilized in GPCOA is described below.

1)Architecture of Chromosome: The coding mode of GA is selected as real coded for the purpose of simplicity. The chromosome in GA is the solution to problem P2.

2) Fitness Function: Proposed function is utilized to estimate how suitable the individual is, which is shown as the following

$$Fitness = \sum_{i\in U, j\in N_B} s_{i,j}\,(A,B,P,F) + \alpha_i \sum_{j\in N_B, i\in U}(T_{i,j}(A,B,P,F,) - \tau_{i,j}^{max}) \tag{16}$$

The fitness function is composed of two parts, the former part is the objective function to be optimized, and the later part is the penalty function (α_i is the penalty factor). Fitness function is composed of two parts, the former part is the objective function to be optimized.

3) genetic operations of GA: In this paper, the first iteration of GPCOA is a random generation, but must fit the constraints in problem p_0. The genes of initial individual will be created as the following

$$\begin{aligned} a_{i,j}(0) &= random\left(\{0\}\cup N_B\right) \\ b_{i,j}(0) &= random\left(\{0\}\cup \mathrm{C}\right) \\ p_{i,j}(0) &= rand\left(p_{\max}\right) \\ f_{i,j}(0) &= \mathrm{rand}\left(F_{a_{i,j}(0)}\right) \end{aligned} \tag{17}$$

where $random(\sim)$ is a function, the function of which is to output a value from the set of $\sim$ randomly? $rand(x)$ can output a number between 0 and *x* randomly.

For the selection operation, the tournament is employed in this paper. Genetic operations (i.e., selection, crossover and mutation) are utilized iteratively to initial population to generate a better solution to the problem. For crossover, the pre-defined crossover probability p_c is utilized to select two individuals to exchange corresponding segments randomly to generate offspring. For mutation operation, an individual will be selected firstly, after that, each substance of individual will mutate with p_m within the range of variables defined in problem p_0.

Algorithm 2 GA

1: **Input**: The size of the population K, the iteration of GA T_1, pm, pc.
2: **Outcome**: The historical best individual and K optimized individual.
3: **Initialization**: K individuals are initialized using (17) and set t_1=0.
4: **Step 1**: if t_1>T_1 break; else conduct step 2.
5: **Step 2**: Compute the fitness function of individuals to estimate how suitable it is.
6: **Step 3**: Seek out the best individual P_{t1}^{best}, and record it in the historically optimal set O_{best}. If P_{t1}^{best} is better than P_{best}, then replace the value of P_{best} with P_{t1}^{best}, else remain the same.
7: **Step** 4: Choose K individuals on the basis of tournament rule.
8: **Step 5:** Select two individuals at random performing the crossover operation with P_c.
9: **Step 6:** Perform mutation operation with probability P_m on each individual.
10: **Step 7**: t_1=t_1+1, and go to step 1.

5.3 PSO

The traditional PSO is utilized in GPCOA to improve the convergence. The position and velocity of particles are updated according to the below principles, respectively.

$$V_{r,d}^{t+1} = wV_{r,d}^{t} + r_1c_1\left(\text{ pbest }_{r,d} - X_{r,d}^{t}\right) + r_2c_2\left(gbest_{r,d} - X_{r,d}^{t}\right) \tag{18}$$

$$X_{r,d}^{t+1} = \begin{cases} \text{round}\left(X_{r,d}^{t} + V_{r,d}^{t+1}\right), & r = 1,2 \\ X_{r,d}^{t} + V_{r,d}^{t+1} & r = 3,4 \end{cases} \tag{19}$$

Where ω is an inertia weight, c_1 *and* c_2 are acceleration instants r_1 *and* r_2 are both generated between [0,1], and the function of which is to enhance the randomness of the search. The utilized PSO is a traditional version, so the description of which is skipped.

6. Simulation Results and Discussion

6.1 Simulation Settings

With aim of estimating the behavior of the proposed GPCOA, a SDIN-enabled MEC network is considered, where M cells with 100m in radius are randomly scattered over the network. The MBS with SDIN controller locates at the origin while SBSs and EDs are distributed at random in this area, and other significant parameters are listed in **Table 1**.

Table 1. The Emulation Parameters

Parameters	Value
The channel bandwidth B	180KHz
Background noise N_0	10^{-13}
Local energy consumption	$1\ J/Gcycles$
The maximum transmission power of EDs p_{max}	1 W
Data size to offload $d_{i,j}$	200~400KB
CPU cycles per bit required by offloading task	500~1000
Computation capacity of EDs $F_{i,j}^{L}$	0.2~1 GHz
Computation capacity of SBSs F_j	2.5GHz
Computation capacity of MBS F_0	25GHz
The size of population K	18
Mutation probability p_m	0.02
Probability of crossover p_c	0.54
The number of selected individuals in tournament N_t	5
The parameters of PSO w/c_1/c_2	0.45/1.495/1.495

To evaluate the algorithm proposed in this paper, we first compare the proposed optimal frequency scheduling with weight related to predicted latency with five baseline algorithms to prove the effectiveness of the scheme. “AL” and “AM” indicate all tasks executed by EDs and the MEC server, respectively. Then we compare the convergence and the performance of GPCOA with four baselines: GACA, which is based on GA, local computing (LC), random offloading algorithm (ROCA) and PSO based offloading algorithm (PSOA). In ROCA, the principle of task offloading is random or executed locally and select transmission power, and the SBSs randomly allocate channels and computation resource to EDs as well. In LCA, all computation of EDs will be executed locally. Finally, the tradeoff between energy consumption of EDs and latency of accomplishing corresponding tasks is investigated.

6.2 Evaluation on frequency scheduling

In this subsection, we estimate proposed the frequency scheduling with weighting factor related to predicted latency algorithm. Taking a multicell scenario into consideration, where each cell shares the same spectrum. EDs access each SeNB through OFDMA. We assume that each SBS has ten channels while provide service for U_j EDs. **Fig. 2** describes the influence of the local computing frequency of EDs on the delay and energy consumption under different numbers of small cells.$f_{min} = 0.2\ GHz\ and\ f_{max} = 1GHz$ are the minimum and maximum local computing CPU-cycle frequency, respectively.

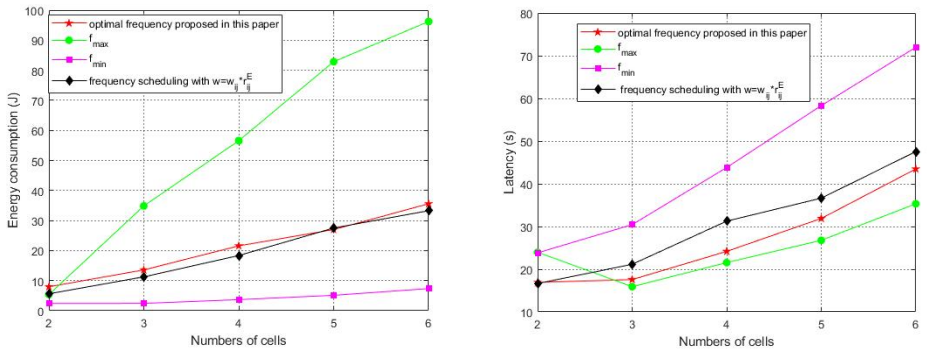

Fig. 2. Influence of the local computing frequency of EDs on (a)energy consumption and (b)latency under different cells

We can observe that in the situation of f_{max}, the energy consumption is much more than that in the situation of f_{min}, and energy consumption of all EDs in the situation of proposed optimal frequency scheduling is much better than that under f_{max} and a little bit higher than that under f_{min}. The latency in the situation of f_{max} is much lower than that under f_{min} with the number of small cells increasing, and the latency under optimal frequency scheduling proposed in this paper is much lower than that under f_{min} and is a little bit higher than that in f_{max}. Thus, the optimal frequency scheduling proposed in this paper can perform better in terms of energy consumption than f_{max} and lower delay than f_{min} to get a tradeoff via local computing scheduling. It is noted that the introduction of weight factor and the proposed frequency scheduling in this paper can achieve a gratifying total cost. We also compare the local frequency scheduling proposed in this paper with proposed scheme in [8] in which the weighting factor is related to the residual energy of smart mobile devices. As we can see that the optimal frequency scheduling proposed in this paper can obtain lower delay than the scheme in [8] but the scheme in [8] can obtain lower energy consumption than scheme

proposed in this paper. In general, the CPU-cycle frequency scheduling proposed in this paper can obtain better performance.

6.3 Analysis of convergence

The effect of variant parameters on the convergence of GPCOA have been investigated in **Fig. 3**, and the comparison of convergence of disparate algorithm is conducted in **Fig. 4**.

There existing some significant parameters in genetic algorithm (i.e., K, p_m and p_c mentioned above). We observe from **Fig. 3 (a)**, the performance of GPCOA is best when K=18. The reason is that when K is big (e.g., K=36), the size of population is so big that algorithm is hard to converge and this situation will contribute to wasting of resource. Another reason is that the diversity of population is not large enough when K is small (e.g., K=9). We can arrive at a conclusion from **Fig. 3 (b)**, the performance of GPCOA drops with p_m increasing. Because the operation of mutation is with randomness, it doesn't make reference to any prior message. p_m is designed to increase the diversity of ethnic, a big probability may violate this principle. As we can observe from **Fig. 3 (c)**, the total cost of proposed GPCOA first rises then drops with the growth of p_c. We can obtain that GPCOA converge fastest in the situation of $p_c = 0.54$. That is because a small p_c can't effectively renew the population, while a big p_c is easy to destroy the favorable mode an, increase the randomness, and miss the optimal individual easily. Hence, the value of k is set to equal 18, $p_c = 0.54$ and $p_m = 0.02$ for proposed GPCOA.

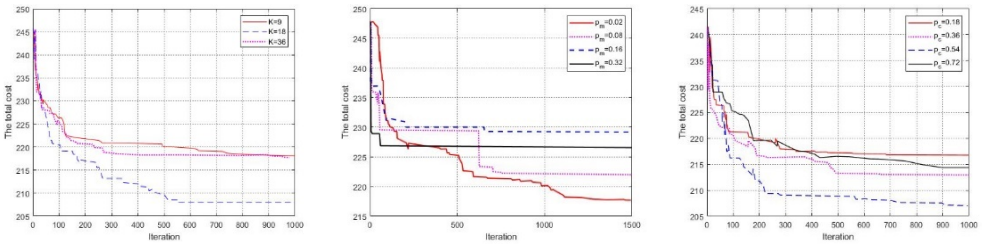

Fig. 3. convergence of GPCOA of variant parameters (a)The total cost versus the Iteration with disparate K.(b) with different crossover probability p_c.(c) with different mutation probability p_m.

The convergence of the proposed GPCOA is proved via being compared to other some baseline, as shown in **Fig. 4**. We can observe that PSO converges fastest while PSOCA is likely to trap into a local optimal solution, then follows GPCOA and GA, and the behavior of GACA lies between PSOCA and GPCOA. It is reasonable, due to that proposed GPCOA combines the advantages of strong global searching ability of GA and the advantages of strong local searching ability of PSO, and PSO in GPCOA can boost the speed of convergence of algorithm. Thus, single PSO or GA is not as convergent as GPCOA.

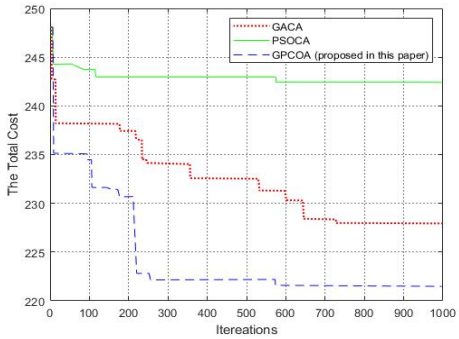

Fig. 4. The convergence comparison of three algorithms

In order to prove the effectiveness of the proposed GPCOA, we compare the proposed GPCOA with verified Brute Force Searching Algorithm (VBFS), which can obtain the optimal solution to show the distance between proposed GPCOA and the optimal solution. As we can observed from **Fig. 5** that the proposed GPCOA has a good convergence and robustness. Firstly, the proposed GPCOA converge quickly, and the performance becomes better along with the number of iteration rounds.Although the ultimate performance of GPCOA is still far way from the optimal performance, the distance is very small. Thus, a trade off between the acceptable performance and the time complexity exits.

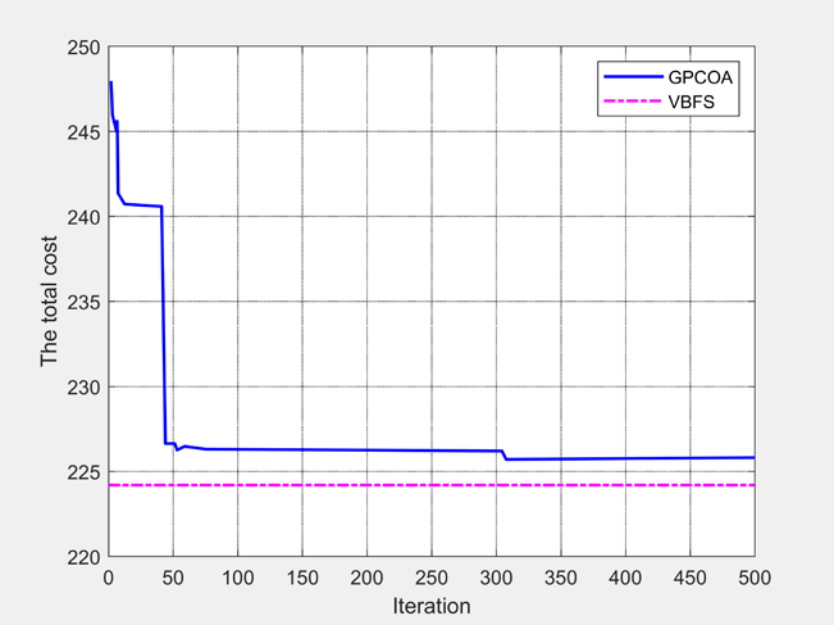

Fig. 5. The performance comparison of GPCOA and VBFS

6.4 Performance Analysis

In this subsection, the proposed GPCOA is testified through being compared with other baseline algorithms under different parameters. The relationship between GPCOA performance and the computation capacity is shown in **Fig. 5**, in this situation, we assume that SBSs have 30 channels. In the wake of the increased computation resources, the total cost

decreasing and the number of offloading EDs increasing. When the computation resource reaches a certain number, both of them become steady. That is because when the limit of computation resource of MEC server is easy to reach, the number of EDs, task of which is offloaded, is small because of the long execution delay, contributing to many EDs executing their computation task locally. Instead, when the limit of computation resource of SBS is hard to get, it is active for EDs to offload to get a low cost. It can be concluded from **Fig. 5** that it is significant to match wireless resource and the computation resource.

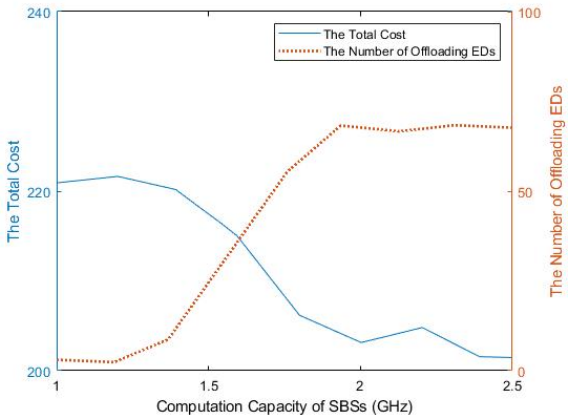

Fig. 5. The behavior of GPCOA versus the limit of computation resource of MEC server.

In order to prove the availability of proposed GPCOA, the performance of GPCOA is compared with other baseline algorithms: GACA, PSOCA, ROA and LCA. **Fig. 6** shows the relationship between the total cost obtained by variant algorithms and the number of EDs from 0 to 300.

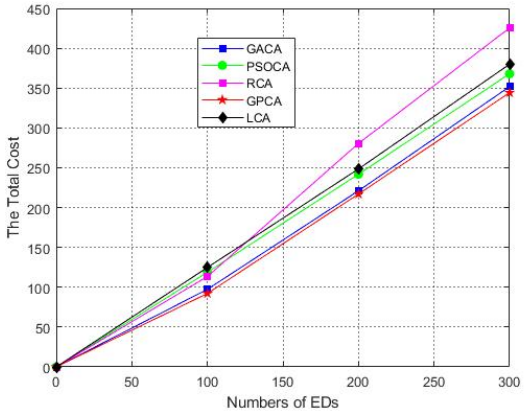

Fig. 6. The total cost versus the number of EDs with variant algorithms.

With the observation from **Fig. 6** that the performance of GPCOA outperforms other baseline algorithms. GPCOA, GACA and PSOCA perform offloading decision making, power control, computation resource assignment, channel allocation together, especially, GPCOA also adopt the optimal local frequency scheduling. Thus, GPCOA outperformances other baseline algorithms. Finally, we investigate the influence of the weight factor defined in system model on performance of GPCOA. The tradeoff between latency of computation task and energy consumption of EDs for disparate numbers of cells is exploited in **Fig. 7**. With the

observation that the weight factor proposed in this paper can save more energy and $w_{i,j} = 0.8$ and can obtain lower delay than all the baseline algorithms except when $w_{i,j} = 0.8$. However, when $w_{i,j} = 0.8$, the energy consumption of EDs is very huge. The observed total cost is also lower than majority of baseline algorithms except when $w_{i,j} = 0.2$.

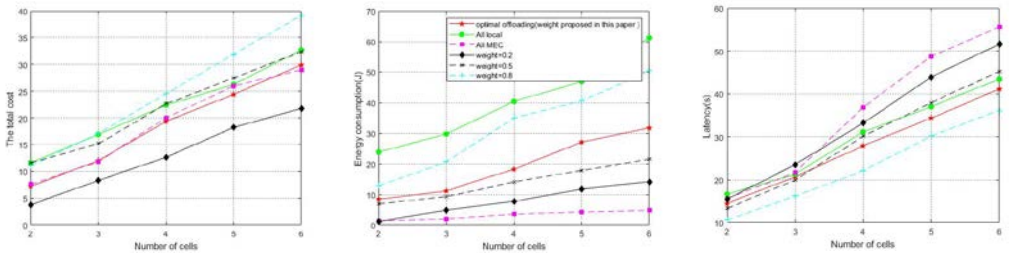

Fig. 7. Tradeoff between energy consumption of EDs and latency of their tasks

7. Conclusion

The tradeoff between energy consumption of EDs and latency of completing the tasks for SDIN-enabled MEC based IIoT is exploited in this paper. To decrease the total cost of EDs, we jointly take the computation offloading, communication and computation resources assignment into consideration, and propose a latency-aware weighting factor based on the predicted latency of computation task of EDs. First, a computation offloading model in SDIN-enabled MEC network is introduced, and we formulate the main problem into a MINLP problem. Then we propose GPCOA to tackle such a complex problem. Ultimately, the convergence and performance of proposed GPCOA have been verified by emulation. Outcomes of simulation reveal that the proposed GPCOA outperforms other baseline algorithms.The generation of individuals in the initial population of most swarm intelligence algorithms is random within a given range, resulting in large randomness and uncertainty of the initial individuals. Therefore, we plan to use tent chaotic map to initialize to achieve the improvement of local development and global exploration capabilities.

References

[1] Abolfazli, S, et al., “Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, 337-368, 2014. Article (CrossRef Link)

[2] F. Tao, et al., "CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1435-1442, May., 2014. Article (CrossRef Link)

[3] R. Chaudhary, et al., "SDN-Enabled Multi-Attribute-Based Secure Communication for Smart Grid in IIoT Environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2629-2640, June 2018. Article (CrossRef Link)

[4] F. Guo, et al., "An Efficient Computation Offloading Management Scheme in the Densely Deployed Small Cell Networks With Mobile Edge Computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651-2664, Dec. 2018. Article (CrossRef Link)

[5] J. Tang et al., "Energy Minimization in D2D-Assisted Cache-Enabled Internet of Things: A Deep Reinforcement Learning Approach," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5412-5423, Aug. 2020. Article (CrossRef Link)

[6] M. Mukherjee et al., "Latency-Driven Parallel Task Data Offloading in Fog Computing Networks for Industrial Applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6050-6058, Sept. 2020. Article (CrossRef Link)
[7] J. Liu, et al., "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. of 2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1451-1455, 2016. Article (CrossRef Link)
[8] J. Zhang, et al., "Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633-2645, Aug. 2018. Article (CrossRef Link)
[9] Y. Bi, et al., "Intelligent Quality of Service Aware Traffic Forwarding for Software-Defined Networking/Open Shortest Path First Hybrid Industrial Internet," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1395-1405, Feb. 2020. Article (CrossRef Link)
[10] Ma H, et al., “Leveraging the power of prediction: Predictive service placement for latency-sensitive mobile edge computing,” *IEEE Transactions on Wireless Communications*, 19(10), 6454-6468, 2020. Article (CrossRef Link)
[11] G. Gui, F, et al., "Flight Delay Prediction Based on Aviation Big Data and Machine Learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140-150, Jan. 2020. Article (CrossRef Link)
[12] J. Zhang et al., "Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633-2645, 2018. Article (CrossRef Link)

Xinchang Zhang received the B.E. degree in Synoptic dynamics from Nanjing Institute of Meteorology, Nanjing, in 1997, and the M.S. and Ph.D. degrees in meteorology from Nanjing university of information science & technology, Nanjing, in 2011 and 2019, respectively. He is currently an associate professor with Nanjing University of Information Science and Technology, China. His research interests include Meteorological change and edge computing.

Changsen Xia received his B.E. degree in computer science from Nanjing University of Information Science and Technology, China, in 2020. Now he is a master student in School of Computer and Software, Nanjing University of Information Science & Technology. His main research interests include internet of mobile communication and edge computing.

Tinghuai Ma received his B.E. degree and M.S. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2003. He is currently Professor of computer sciences with Nanjing University of Information Science and Technology, Nanjing, China. His research interests include data mining, social network analysis, and cloud computing.

Lejun Zhang received the M.S. degree in computer science and technology from the Harbin Institute of Technology and the Ph.D. degree in computer science and technology from Harbin Engineering University. He is currently a Professor with Yangzhou University, China. His research interests include computer networks, social network analysis, and information security.

Zilong Jin received the B.E. degree in computer engineering from Harbin University of Science and Technology, China, in 2009, and the M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, Korea, in 2011 and 2016, respectively. He is currently an associate professor with the School of Software, Nanjing University of Information Science and Technology, China. His research interests include wireless sensor networks, mobile wireless networks, and cognitive radio networks.