A Cloud-Edge Collaborative Computing Task Scheduling and Resource Allocation Algorithm for Energy Internet Environment

Xin Song, Yue Wang*, Zhigang Xie, Lin Xia
Northeastern University
Shenyang 110819,China
[e-mail:1871649@stu.neu.edu.cn]
*Corresponding author: Yue Wang

Abstract

To solve the problems of heavy computing load and system transmission pressure in energy internet (EI), we establish a three-tier cloud-edge integrated EI network based on a cloud-edge collaborative computing to achieve the tradeoff between energy consumption and the system delay. A joint optimization problem for resource allocation and task offloading in the three-tier cloud-edge integrated EI network is formulated to minimize the total system cost under the constraints of the task scheduling binary variables of each sensor node, the maximum uplink transmit power of each sensor node, the limited computation capability of the sensor node and the maximum computation resource of each edge server, which is a Mixed Integer Non-linear Programming (MINLP) problem. To solve the problem, we propose a joint task offloading and resource allocation algorithm (JTOARA), which is decomposed into three subproblems including the uplink transmission power allocation sub-problem, the computation resource allocation sub-problem, and the offloading scheme selection sub-problem. Then, the power allocation of each sensor node is achieved by bisection search algorithm, which has a fast convergence. While the computation resource allocation is derived by line optimization method and convex optimization theory. Finally, to achieve the optimal task offloading, we propose a cloud-edge collaborative computation offloading schemes based on game theory and prove the existence of Nash Equilibrium. The simulation results demonstrate that our proposed algorithm can improve output performance as comparing with the conventional algorithms, and its performance is close to the that of the enumerative algorithm.

Keywords: Energy internet, Cloud-Edge integrated EI network, Resource allocation, Task offloading

1. Introduction

In preparation for the fossil fuels such as coal, oil and natural gas in the future, renewable energy will be used in the energy market to a greater extent and the whole energy system has undergone great changes, so it is necessary to solve the problem of effective utilization of renewable resource. Supported by the Internet, the energy trading model is evolving. Energy Internet (EI) is an emerging energy industrial development system that integrate new energy technology with information technology [1].With the emergence of energy internet, massive intelligent sensors generate huge volumes of data on key parameters, such as monitoring data, production capacity data, and energy consumption data [2]-[3].Taking electric energy as an example, it has gone through multiple links from production to consumption, each of which involves a large amount of data analysis and processing. To address these challenges, cloud computing come into existence, which can analyze and process a large amount of data uploaded from sensors to cloud servers [4]. However, computing massive data generated from EI increases tremendous burden on cloud servers, which also takes pressure on the backbone network [5]. Moreover, lots of data generated by the connected sensors from EI will be on the order of zettabytes in the near future. Hence, uploading a large amount of data to the remote cloud for future analysis and processing in EI can aggravate the backbone network congestion [6]. Additionally, this can result in unbearable transmission and computation delay which would further affect the quality of service (QoS) for different terminal applications in EI.

Because of the limitations of cloud computing in EI, edge computing defined by the Cisco attracts more and more attentions in academic and industrial circle [7]. Edge computing in EI is the sinking of mobile computing, network control, and storage functions to the edge of the network, which can reduce transmission delay and relieve the pressure on the core network of EI [8]. Recently, many researches involved in edge computing focus on the resource allocation and task offloading [9]-[13]. In [9], a task offloading scheme is proposed, which minimizes the system delay in edge computing network. To minimize the total energy consumption of the computing offloading system, an optimization problem is formulated in [10], which jointly optimizes the computing task offloading scheme and communication resource allocation. Considering the fix CPU frequency and elastic CPU frequency for the mobile device, a semidefinite relaxation is adopted to efficiently find the optimal solutions [11]. In [12], an energy-efficient resource allocation problem is discussed to obtain the optimal solution. To decrease delay and meet the situational awareness requirements of emerging applications, a communication and computation resource allocation based on edge computing is proposed in [13].

Based on the above researches, most of literatures are carried out under the edge computing. However, edge computing has some limitations in EI such as its limited computing resources. Besides, edge devices in EI may consume a large amount of energy which would further impact the quality of service (QoS) for different terminal applications in EI. Meanwhile, only applying cloud computing in EI would result in intolerable delay. Thus, it is inappropriate to individually apply edge computing or cloud computing in EI, which in turn affect the QoS of the energy applicants. Since edge computing is seen as a complementation technology of cloud computing rather than as a competition technology, the collaboration and integration of the edge computing and cloud computing can help to reduce energy consumption and delay so that maintain the QoS for different energy applicants in EI [14].Therefore, in this paper, we apply the cloud-edge collaborative computing in EI to improve energy efficiency and the QoS of the energy applicants.

Currently, several papers have studied the jointly computation offloading and resource allocation under the collaboration of edge computing and cloud computing [15]-[18]. To optimize the total delay and the energy consumption in the cloud-edge network, a computation offloading method is proposed to reduce the system cost in [15]. By using alternating direction method of multipliers (ADMM), an optimization framework is proposed to jointly optimize the computation offloading scheme and resource allocation [16]. Similarly, edge and cloud computing with non-orthogonal multiple access (NOMA) is applied in [17], which utilizes effectively alternating direction method of multipliers (ADMM) to get the optimal computation offloading scheme and the computation and communication resource allocation. Differently, a game-theoretic collaborative computation offloading strategy is proposed in [18] to minimize the users' energy consumption. On the other side, divisible computation tasks generated from EI can be partially processed at the terminals, edge servers and cloud servers [19]-[20]. In [19], to improve edge server with limited computation resource efficiency, a joint communication and computation resource allocation problem is proposed to get the optimal task offloading ratio by utilizing the convex optimization theory. In [20], the pipeline-based offloading scheme is proposed. Besides, by using the classic successive convex approximation, a minimization problem of sum latency is perfectly solved.

However, most of the researches based on cloud edge collaborative computing are carried out under the background of mobile communication network. In contrast, the cloud-edge collaborative technology is applied to the energy internet in this paper. Energy Internet system is a typical multi-node distributed network. The energy has gone through multiple links from production to consumption, and each link involves a large amount of data analysis and calculation. This will cause high network delay due to the insufficient bandwidth of the core network. At the same time, the system response delay may lead to decision-making errors, resulting in heavy losses. These problems can be solved by combining the cloud-edge collaborative technology with energy internet. On the other hand, compared with several existing papers that investigate the collaborative cloud-edge task offloading, to improve the computation resource utilization of the edge server, the competition of the sensor nodes for computation resource of the edge server is considered in this paper. Meanwhile, to make full use of the advantages of the terminals, the edge server, and the cloud server, a three-layer cloud-edge collaborative computing network architecture is established, which studies the multi-terminal edge computing framework that includes multi-dimensional resource integration such as computation resource and transmission resource and proposes three task offloading modes. This architecture can balance and reasonably schedule the computation tasks according to terminals' offloading requirements and the characteristics of the three task offloading modes, so as to improve the utilization of computing and communication resources in the system. The main contributions of this paper are summarized as follows:

(1) An integrated three-tier cloud-edge collaboration network in EI is proposed, which include cloud center, edge server, and the sensor nodes. Each sensor node has an indivisible task that can be executed locally, at the edge server or at the remote cloud server cooperatively.

(2) A joint communication and computation resource allocation and task offloading scheme is proposed to minimize the total system cost, which is defined as a line combination of the energy consumption of the sensor nodes and the system delay. The optimization problem is formulated to investigate the tradeoff between energy consumption of the sensor nodes and the system delay. This optimization problem is a NP-hard problem, which jointly optimizes the task offloading decisions, the transmission power of the sensor nodes, and the resource allocation under the limited communication and computation resource constrains.

(3) Given the NP-hardness of the proposed optimization problem, we decompose the original problem into the uplink transmission power allocation sub-problem, the computation resource allocation sub-problem and computation offloading scheme selection sub-problem.

(4) We obtain the power allocation of the sensor nodes by utilizing bisection method. Meanwhile, we address computation resource allocation of the sensor nodes using line optimization. Then, applying convex optimization techniques, we address the computation resource allocation of the edge server. At last, we propose a cloud-edge collaborative computation offloading scheme based on game theory and prove the existence of Nash equilibrium.

The rest of the papers is organized as follows. In section 2, we set a three-tier cloud-edge integrated EI network architecture. Section 3 formulate the optimization problem to minimize the total system cost. In section 4, we introduce power allocation algorithm, computation resource allocation algorithm, and the optimal task offloading scheme. Section 5 shows the simulation results. Finally, conclusion is given in section 6.

2. System Model

As illustrated in **Fig. 1**, to satisfy the QoS of the applications in EI, we introduce a three-tier cloud-edge integrated EI network architecture.
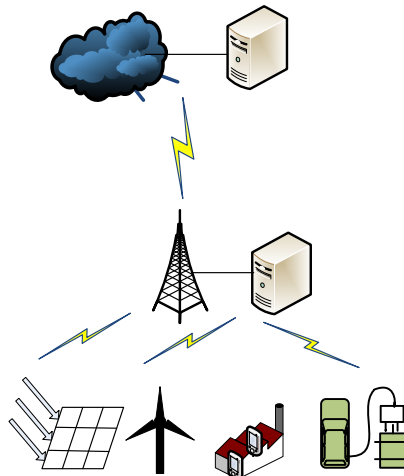
Fig. 1. The architecture of a three-tier cloud-edge heterogeneous network [21]

The three-tier cloud-edge heterogeneous network architecture includes three layers: the terminal equipment layer, the edge server layer and the remote cloud server layer. The terminal equipment layer is the staple-basic element of the architecture, which is mainly composed of power generation, oil and gas extraction, and electric vehicles. The sensing terminal mainly includes measurement sensors, acquisition sensors, and monitoring sensors, which can realize the state awareness, quantity and value transmission, environment monitoring and behavior tracking in EI. As the underlying perception unit of EI, it realizes energy scheduling, protection measurement and control, security operation and maintenance, online monitoring and interconnection in EI, thus generating a large number of computation tasks that need to be dealt with.

The edge computing layer includes the base station (BS) equipped with the edge server, which is called the edge node. The edge node interacts with the remote cloud to obtain relevant

information according to the requirements of computing tasks, analyzes and processes the key product information transmitted by the edge node, and uploads the production status information that cannot be processed to the remote cloud to realize data sharing. What's more, different type of data (such as power generation data, oil production data, customer demand data, and measured data) can be executed in the edge server layer to reduce latency.

The remote cloud service layer includes remote cloud server, which is mainly for storing data from EI [22]. In the three-tier cloud-edge integrated EI network architecture, to enhance the computation power of the cloud-edge network architecture, the remote cloud server is also regarded as a computing node. Meanwhile, setting the cloud-edge architecture network is an effective method to enhance the QoS for the device applicants.

Taking the power line real-time monitoring in EI as an example, a large amount of data needs to be processed in the process of the real-time monitoring. We analyze the data processing in the cloud-edge integrated EI network. First, the sensors in the camera capture the pictures data or video stream data in real time. And then the collected real-time monitor data is transmitted to the edge nodes and the remote cloud server for rapid computation to detect the potential threats. Finally, the monitor data results will be sent to the sensor nodes. Refer to the above task, we consider a three-tier cloud-edge integrated EI network, which consists of N sensor nodes denoted by a set of $\mathcal{N} = \{1, 2, ..., N\}$, one edge node, and a remote cloud center. Assume each sensor node has an indivisible real-time monitoring task in EI to be executed locally, by edge server or by cloud server. Each computation indivisible task is denoted as $S_i = \{V_i, Q_i\}$, which V_i denotes the input data size of the task S_i and Q_i stands for the total number of CPU cycles that required to complete the task S_i.

2.1 Communication Model

In this section，we discuss the communication model in the three-tier cloud-edge integrated EI network. Assume that the sensor nodes are connected to the cloud-edge integrated EI network via orthogonal frequency division multiple access (OFDMA), thus interference signals between the sensor nodes are ignored. We define the signal noise ratio (SNR) between the sensor node i and the edge node as

$$\gamma_i = \frac{p_i h_i}{\sigma^2} \tag{1}$$

where h_i is the channel gain between the sensor node i and the edge node, p_i is the transmission power of the sensor node i, and σ^2 is the noise power.

Further, as the picture data or video stream data of the sensor node i is transmitted to the edge node, the uplink transmission rate of the sensor node i in EI is

$$R_i = w \log_2 \left(1 + \frac{p_i h_i}{\sigma^2}\right) \tag{2}$$

where w is the bandwidth.

After the sensor in the camera captures the pictures data or video stream data in real time, the transmission delay that uploading the real-time monitoring task to the edge node in EI is given by

$$T_i^{trans} = \frac{V_i}{R_i} \tag{3}$$

The energy consumption of the sensor node i during the process of transmitting the real-time monitor data to the edge nodes in EI is expressed by,

$$E_i^{trans} = p_i T_i^{trans} = p_i \frac{V_i}{R_i} \qquad (4)$$

2.2 Computing Model

In this subsection, the computing model is mainly consisted of local computing model, edge computing model, and cloud computing model.

1) Local computing: As the real-time monitor task in EI of the sensor node i is executed locally, the delay of completing the task can be written as

$$T_i^L = \frac{Q_i}{C_i^l} \qquad (5)$$

where C_i^l is the computation ability of the sensor node i.

In addition, the energy consumption of the sensor node i is obtained as

$$E_i^L = k \left(C_i^l \right)^2 Q_i \qquad (6)$$

where k is a coefficient that hinges on the chip architecture.

2) Edge computing: After the sensor in the camera captures the pictures data or video stream data in cloud-edge integrate EI network, the collected real-time monitor data is transmitted to the edge node. The delay of computing task occurred in the BS is given by

$$T_i^e = \frac{Q_i}{C_i^f} \qquad (7)$$

where C_i^f is the computation ability of the edge server.

The total edge computing delay for offloading task from the sensor node i to the edge node mainly includes the transmission delay and the computing delay of the edge server, i.e.

$$T_i^E = T_i^{trans} + T_i^e \qquad (8)$$

3) Cloud computing: If the real-time monitoring task from EI cannot be performed on the edge server, it will be transferred to the cloud server. The delay of transmitting the computation task from the edge nodes to the remote cloud is calculated by

$$T_i^{bc} = \frac{V_i}{R_1^b} \qquad (9)$$

where R_1^b is the data rate of the backhaul link between the edge nodes and the center cloud.

The delay of cloud computing for computing task is calculated by

$$T_i^c = \frac{Q_i}{C_i^c} \qquad (10)$$

where C_i^c is the computation ability of the remote cloud.

Further, the total delay of the cloud processing in the cloud-edge integrated EI network consists of three parts: the transmission delay from the sensor node i to the edge nodes, the transferred delay from the edge nodes to the center cloud, and the computing delay of the remote cloud server, i.e.

$$T_i^C = T_i^{trans} + T_i^{bc} + T_i^c \qquad (11)$$

3. Problem Formulation

Based on the system model, in the three-tier cloud-edge integrated EI network, a task offloading and resource allocation scheme is proposed, which can balance the tradeoff between the energy consumption of the sensor nodes and the total delay of the system. On the other hand, the real-time data monitoring and communication is an important basis for energy internet system optimization. To improve the efficiency of EI and the QoS of the energy applications, we introduce the problem formulation under the necessary system constrains.

3.1 System Constrains

1) Task scheduling constrains: For the real-time monitor computation task from EI, it can be executed locally, by the edge server, or by the remote cloud. To ensure the efficient operation of the real-time data monitoring and communication in EI, we introduce three binary indicators a_i^L, a_i^E and a_i^C to represent the task offloading decision. $a_i^L = 1$ represent the task is processed locally, otherwise $a_i^L = 0$. $a_i^E = 1$ represent the task is executed by the edge node, otherwise $a_i^E = 0$. Similarly, $a_i^C = 1$ represent the task is processed by the remote cloud, otherwise $a_i^C = 0$. The constrains can be obtained as

$$C1: a_i^L + a_i^E + a_i^C = 1$$
$$C2: a_i^L, a_i^E, a_i^C \in \{0,1\} \quad \forall i \in \mathcal{N} \tag{12}$$

where the constrain $C1$ implies each indivisible computation task can only be completed by one way.

2) Transmission power constraint: Considering that the energy goals of EI include energy saving analysis, we set a limit on the transmitting power of the terminal sensor. The transmission power of the sensor node can't exceed the maximum power, and the uplink transmission power allocation constraint is given by

$$C3: 0 \le p_i \le P_{\max} \quad \forall i \in \mathcal{N} \tag{13}$$

3) Computational capacity constraints for terminals: The computation capability of the sensor nodes in EI is restricted into a finite set of values, i.e.

$$C4: C_{\min}^l \le C_i^l \le C_{\max}^l \quad \forall i \in \mathcal{N} \tag{14}$$

4) Computational capacity constraints for the edge server: Since energy management needs to analyze a large amount of energy data in EI, including basic data collection, enterprise data analysis, energy cost analysis and energy efficiency management, in order to avoid the edge server working overload, we need to restrict the computation capacity of the edge server

$$C5: \sum_{i=1}^{N} C_i^f \le C_{max}^f \quad \forall i \in \mathcal{N} \tag{15}$$

3.2 Problem Formulation

As mentioned above, in the three-tier cloud-edge integrated EI network, to tradeoff between the energy consumption of the sensor nodes and the system delay and improve the QoS of the energy applicants in EI, we formulate the optimization problem in the three-tier cloud-edge integrated EI network to minimize the total cost of the system as follows.

$$\min_{p_i, C_i^f, C_i^l, a_i^L, a_i^E, a_i^C} U = \sum_{i=1}^{N} \left[\begin{array}{c} a_i^L \left(w_i^{t'} T_i^L + w_i^{e'} \beta E_i^L \right) + a_i^E \left(w_i^{t'} T_i^E + w_i^{e'} \beta E_i^{trans} \right) \\ + a_i^C \left(w_i^{t'} T_i^C + w_i^{e'} \beta E_i^{trans} \right) \end{array} \right]$$

$$s.t. \quad C1: a_i^L + a_i^E + a_i^C = 1$$
$$C2: a_i^L, a_i^E, a_i^C \in \{0,1\} \quad \forall i \in \mathcal{N}$$
$$C3: 0 \le p_i \le P_{\max} \quad \forall i \in \mathcal{N} \tag{16}$$
$$C4: C_{\min}^l \le C_i^l \le C_{\max}^l \quad \forall i \in \mathcal{N}$$
$$C5: \sum_{i=1}^N C_i^f \le C_{max}^f \quad \forall i \in \mathcal{N}$$

where $w_i^{e'}$ and $w_i^{t'}$ is the weighting factor, which is applied to achieve the tradeoff between the energy consumption of the sensor node and the system delay. To implement the dimensionless combination between delay and energy consumption, β is introduced as the normalizing factor, which is seen as the ratio of average system delay to average energy consumption of the sensor nodes.

Let $w_i^t = w_i^{t'}$ and $w_i^e = \beta w_i^{e'}$ then they represent the weightings of system delay and energy consumption respectively. Hence, the (16) can be represented as

$$\min_{p_i, C_i^f, C_i^l, a_i^L, a_i^E, a_i^C} U = \sum_{i=1}^N \left[\begin{array}{l} a_i^L \left(w_i^t T_i^L + w_i^e E_i^L \right) + a_i^E \left(w_i^t T_i^E + w_i^e E_i^{trans} \right) \\ + a_i^C \left(w_i^t T_i^C + w_i^e E_i^{trans} \right) \end{array} \right]$$
$$s.t. \text{ C1-C5} \tag{17}$$

The optimization problem (17) is a mix integer non-convex problem, which is a NP-hard and extremely hard to solve. The relevant reason is that the feasible region of problem (17) and the objective function is not convex as the variables a_i^L, a_i^E and a_i^C are binary variables.

To solve this optimization problem, we propose a distributed joint task offloading and resource allocation algorithm (JTOARA) in the following section.

4. Problem Decomposition and solution

In this section, we propose a joint task offloading and resource allocation algorithm in the three-tier cloud-edge integrated EI network, which respectively decomposes the optimization problem into the uplink transmission power allocation sub-problem, the computation resource allocation sub-problem and the offloading scheme selection sub-problem under the edge computation offloading mode, cloud computation offloading mode and the local computing mode. Then, we adopt the bisection search algorithm to obtain the optimal uplink transmission power allocation of the sensors. According to the monotony analysis of the function, the computation resource allocation of the sensor nodes is obtained by line optimization method. And the computation resource allocation of the edge server is acquired by the convex optimization theory. Finally, we propose a game-theoretic collaborative computation offloading schemes to obtain the optimal task offloading scheme and prove the existence of Nash Equilibrium.

4.1 Edge Computing Mode

In edge computation offloading mode that satisfies $a_i^E = 1$, $a_i^C = 0$, $a_i^L = 0$, according to (3), (4), (7) and (8), the problem (17) can be rewritten as

$$\min_{p_i,C_i^f} \sum_{i=1}^{N} \left\{ \left[w_i^t \left(\frac{V_i}{R_i} + \frac{Q_i}{C_i^f} \right) + w_i^e p_i \frac{V_i}{R_i} \right] \right\}$$

$$\text{s.t. } C3 : 0 \le p_i \le P_{\max} \quad \forall i \in \mathcal{N} \tag{18}$$

$$C5 : \sum_{i=1}^{N} C_i^f \le C_{max}^f \quad \forall i \in \mathcal{N}$$

1) Power allocation

Decomposed from problem (17), the power allocation sub-problem is expressed as

$$\min_{p_i} \sum_{i=1}^{N} \left\{ \left[w_i^t \left(\frac{V_i}{w\log_2\left(1+\frac{p_i h_i}{\sigma^2}\right)} \right) + w_i^e p_i \frac{V_i}{w\log_2\left(1+\frac{p_i h_i}{\sigma^2}\right)} \right] \right\} \tag{19}$$

$$s.t. \, C3 : 0 \le p_i \le P_{\max} \quad \forall i \in \mathcal{N}$$

Firstly, the derivative of this objective function (19) is calculated as

$$G(p_i) = \frac{w_i^e V_i \left[w\log_2\left(1+\frac{p_i h_i}{\sigma^2}\right) \right] - \frac{w\frac{h_i}{\sigma^2}\left(w_i^t V_i + w_i^e p_i V_i\right)}{\left(1+\frac{p_i h_i}{\sigma^2}\right)\ln 2}}{\left[w\log_2\left(1+\frac{p_i h_i}{\sigma^2}\right) \right]^2} \tag{20}$$

It can be verified that $G(p_i) = 0$ when

$$F(p_i) = w_i^e V_i \left[w\log_2\left(1+\frac{p_i h_i}{\sigma^2}\right) \right] - \frac{w\frac{h_i}{\sigma^2}\left(w_i^t V_i + w_i^e p_i V_i\right)}{\left(1+\frac{p_i h_i}{\sigma^2}\right)\ln 2} = 0 \tag{21}$$

Further, the derivation of $F(p_i)$ is calculated as

$$\frac{\partial F(p_i)}{\partial p_i} = \frac{\left(\frac{h_i}{\sigma^2}\right)^2 \cdot \ln 2 \cdot w \cdot \left(w_i^t V_i + w_i^e p_i V_i\right)}{\left[\left(1+\frac{p_i \cdot h_i}{\sigma^2}\right)\ln 2\right]^2} \tag{22}$$

We can see that both the denominator and the numerator are greater than 0. Thus, we have

$$\frac{\partial F(p_i)}{\partial p_i} = \frac{\left(\frac{h_i}{\sigma^2}\right)^2 \cdot \ln 2 \cdot w \cdot \left(w_i^t V_i + w_i^e p_i V_i\right)}{\left[\left(1+\frac{p_i \cdot h_i}{\sigma^2}\right)\ln 2\right]^2} > 0$$ and $$F(0) = -\frac{w\frac{h_i}{\sigma^2} w_i^t V_i}{\ln 2} < 0 .$$ Introduced by the

above analysis, we can conclude that $F(p_i)$ is a monotonic increasing function. Because of $F(0) < 0$, by using the bisection method, the optimal power allocation is achieved. The bisection method is summarized in Algorithm 1.

In Algorithm 1, the value of $F(P_{max})$ is calculated by using (21). If $F(P_{max}) \leq 0$, the objective function (19) is monotonically decreasing. When the uplink transmission power of the sensor nodes increases gradually, the value of the objective function (19) decreases. Thus, the optimal transmission power is the maximum uplink transmission power. There is another scenario that $F(P_{max}) > 0$, where the objective function (19) decreases and then increases monotonically in the feasible region. To solve the optimal power, the power is initialized, which is $p_i' = 0$ and $p_i'' = P_{max}$. Meanwhile, p_i^* that is the median value of the power is taken. If $F(p_i^*) > 0$, let $p_i'' = p_i^*$. If $F(p_i^*) \leq 0$, let $p_i' = p_i^*$. When the terminal condition that is represented as $p_i'' - p_i' \leq \varepsilon$ is met, the cycle is terminated and the optimal power distribution scheme is obtained.

Algorithm 1 Binary Search Algorithm for Uplink Power Allocation
1: Calculate $F(P_{max})$ using (21)
2: **if** $F(P_{max}) \leq 0$, **then**
3:     $p_i^* = P_{max}$
4: **else**
5:     Set tolerance $\varepsilon > 0$
6:     Initialize $p_i' = 0$ and $p_i'' = P_{max}$
7:     **repeat**
8:         set $p_i^* = \dfrac{p_i' + p_i''}{2}$
9:         **if** $F(p_i^*) \leq 0$ **then**
10:             Set $p_i' = p_i^*$
11:         **else**
12:             Set $p_i'' = p_i^*$
13:         **end if**
14:     **until** $p_i'' - p_i' \leq \varepsilon$
15:     Set $p_i^* = \dfrac{p_i' + p_i''}{2}$
16: **end if**

2) Resource allocation scheme

The objective function of resource allocation scheme is separated from the problem (18), and is expressed as

$$\min_{C_i^f} \sum_{i=1}^{N} w_i^t \left(\frac{Q_i}{C_i^f} \right)$$

$$s.t.\ C5: \sum_{i=1}^{N} C_i^f \leq C_{max}^f \quad \forall i \in \mathcal{N} \tag{23}$$

It is obvious that the constraint C5 in (23) is convex. Denote that the objective function in (23) is $\Omega(C_i^f)$. The second derivatives of the objective function are represented as

$$\frac{\partial^2 \Omega\left(C_i^f\right)}{\partial C_i^{f\,2}} = \frac{2 w_i^t Q_i}{\left(C_i^f\right)^3} > 0 \tag{24}$$

$$\frac{\partial^2 \Omega\left(C_i^f\right)}{\partial C_i^f \partial C_j^f} = 0, \ \forall i \neq j \tag{25}$$

According to the formula (24) and the formula (25), the Hessian matrix of the objective function in (23) is a symmetric positive definite matrix. Thus, the original optimization problem (23) is a convex optimization problem, which can be solved by the Karush-Kuhn-Tucker (KKT) conditions [23]. The Lagrangian function of the optimization problem (23) can be obtained as follow

$$L\left(C_i^f, \ \mu\right) = \sum_{i=1}^{N} \left(w_i^t \frac{Q_i}{C_i^f} \right) + \mu \left(\sum_{i=1}^{I_1} C_i^f - C_{max}^f \right) \tag{26}$$

By calculating the derivative of the Lagrangian function (26), the derivation is set to be zero.

$$\frac{\partial L}{\partial C_i^f} = -\frac{w_i^t Q_i}{\left(C_i^f\right)^2} + \mu = 0 \tag{27}$$

According to (27), we can conclude that

$$C_i^f = \sqrt{\frac{w_i^t Q_i}{\mu}} \tag{28}$$

Then, substituting (28) into the (26), we can achieve that

$$V\left(\mu\right) = 2 \sum_{i=1}^{N} \sqrt{w_i^t Q_i \mu} - \mu C_{max}^f \tag{29}$$

Next, we take the derivative of (29), and is followed as

$$\frac{\partial V\left(\mu\right)}{\partial \mu} = \frac{\sum\limits_{i=1}^{N} \sqrt{w_i^t Q_i}}{\sqrt{\mu}} - C_{max}^f \tag{30}$$

The value of formulate (30) is set to zero, we can derive that

$$\sqrt{\mu} = \frac{\sum\limits_{i=1}^{N} \sqrt{w_i^t Q_i}}{C_{max}^f} \tag{31}$$

By substituting (31) in to (28), the computation resource of the edge server is followed as,

$$C_i^f = \frac{\sqrt{w_i^t Q_i}}{\sum\limits_{i=1}^{N} \sqrt{w_i^t Q_i}} C_{max}^f \tag{32}$$

4.2 Cloud Computing Mode

In cloud computation offloading mode that is $a_i^L = 0$, $a_i^E = 0$, $a_i^C = 1$, after receiving offloading request from the sensor nodes, the cloud server has to perform offloading decision and allocate computation and communication resource to these sensor nodes. The objective function of power allocation scheme is separated from the problem (17), and is expressed as

$$\min_{p_i, C_i^f} \sum_{i=1}^{N} \left\{ \left[w_i^t \left(\frac{V_i}{R_i} + \frac{V_i}{R_1^b} + \frac{Q_i}{C_i^c} \right) + w_i^e p_i \frac{V_i}{R_i} \right] \right\} \tag{33}$$

$$\text{s.t. } C3 : 0 \le p_i \le P_{\max} \quad \forall i \in \mathcal{N}$$

Since the form of the objective function (33) in the cloud computing model is the same as that in the edge computing model, the optimal uplink transmission power in these two modes is solved by the same manner. The bisection method is also summarized in Algorithm 1.

4.3 Local Computing Mode

In local computing mode that is $a_i^L = 1$, $a_i^E = 0$, $a_i^C = 0$, decoupled from the problem (17), the CRAP is expressed as,

$$\min_{C_i^l} \sum_{i=1}^{N} \left(w_i^t \frac{Q_i}{C_i^l} + w_i^e k \left(C_i^l \right)^2 Q_i \right) \tag{38}$$

$$\text{s.t. } C4 : C_i^{\min} \le C_i^l \le C_i^{\max}$$

The first derivative of the objective function (38) and set it to be zero, and is obtained as

$$H\left(C_i^l \right) = -w_i^t Q_i + 2 w_i^e k \left(C_i^l \right)^3 Q_i = 0 \tag{39}$$

Then, the solution of the equation (39) is calculated as

$$C_i^{l*} = \sqrt[3]{\frac{w_i^t}{2 w_i^e k}} \tag{40}$$

For $C_i^l > C_i^{l*}$, the objective function monotonously increases with the increase of C_i^l. Otherwise, it monotonously decreases with the increase of C_i^l. Thus, combined with the constraints $C4$, when $C_i^{l*} \le C_i^{\min}$, the objective function is monotonously increasing with the increase of C_i^l. Thus, the optimal solution is $C_i^{\min}$. Instead, the objective function is monotonously decreasing with the increase of C_i^l first and then decreasing gradually. Thus, the optimal solution is $C_i^{\max}$. Specially, for $C_i^{\min} < C_i^{l*} \le C_i^{\max}$, the objective function is monotonously decreasing and then increasing gradually with the increase of C_i^l. Therefore, the optimal solution is C_i^{l*}. the optimal solution C_i^{opt} can be calculated as

$$C_i^{opt} = \begin{cases} C_i^{\min}, C_i^{l*} \le C_i^{\min} \\ C_i^{l*}, C_i^{\min} < C_i^{l*} \le C_i^{\max} \\ C_i^{\max}, C_i^{l*} > C_i^{\max} \end{cases} \tag{41}$$

4.4 Offloading Scheme Selection

In this subsection, the offloading scheme selection scheme is proposed by leveraging game theory [24]. When the uplink transmission power allocation and computation resource allocation are obtained, the objective function of offloading scheme selection scheme is expressed as

$$\min_{a_i^L, a_i^E, a_i^C} U = \sum_{i=1}^{N} U_i$$
$$s.t. \ C1: a_i^L + a_i^E + a_i^C = 1$$
$$C2: a_i^L, a_i^E, a_i^C \in \{0,1\} \ \forall i \in \mathcal{N} \tag{42}$$

According to the formula (17), the cost of each sensor node is expressed as,

$$U_i = a_i^L \left[w_i^t T_i^L + w_i^e E_i^L \right] + a_i^E \left[w_i^t T_i^E + w_i^e E_i^{trans} \right] + a_i^C \left[w_i^t T_i^C + w_i^e E_i^{trans} \right] \tag{43}$$

Let $\mathbf{A}_{-i}^o$ be matrix of the offloading scheme of all sensor nodes except sensor node i. The matrix $\mathbf{A}_{-i}^o$ has $N-1$ rows and three columns. Each row of $\mathbf{A}_{-i}^o$ respresents the offloading vector of a sensor node. The matrix $\mathbf{A}_{-i}^o$ is denoted as

$$\mathbf{A}_{-i}^o = \begin{pmatrix} a_{11}^L & a_{12}^E & a_{13}^C \\ \vdots & \vdots & \vdots \\ a_{i-1,1}^L & a_{i-1,2}^E & a_{i-1,3}^C \\ a_{i+1,1}^L & a_{i+1,2}^E & a_{i+1,3}^C \\ \vdots & \vdots & \vdots \\ a_{N1}^L & a_{N2}^E & a_{N3}^C \end{pmatrix} \tag{44}$$

Given the offloading scheme $\mathbf{A}_{-i}^o$, the sensor node i would like to select a proper offloading scheme $\mathbf{a}_i^o$ to minimize its own cost in the competitive environment. And the whole system computation cost for each sensor node is written as

$$\min_{a_i^L, a_i^M, a_i^C} U_i \left(\mathbf{a}_i^o, \mathbf{A}_{-i}^o \right) \ \forall i \in \mathcal{N} \tag{45}$$

The offloading selection scheme is regarded as a potential game $\Gamma = \left\{ \mathcal{N}, \left(\mathbf{a}_i^o \right)_{i \in \mathcal{N}}, \left(U_i \right)_{i \in \mathcal{N}} \right\}$. As it is known, the potential game includes players, strategies, and cost function.

Players. Each sensor node is a player and there are N participants that select local computing, edge offloading, or the remote cloud offloading.

Strategies. $\mathbf{a}_i^o = \left(a_i^L, a_i^E, a_i^C \right)$ is the computation offloading decision for the sensor node i, and $\mathcal{A}$ is the offloading scheme for all sensor nodes.

Cost Function. The overhead of the sensor node i is $U_i \left(\mathbf{a}_i^o, \mathbf{A}_{-i}^o \right)$.

Then, the solution for the potential game model is Nash Equilibrium and the concept is denoted as:

Definition 1: A computation offloading strategy profile $\mathcal{A}^* = \left(\mathbf{a}_1^{o*}, \mathbf{a}_2^{o*}, ..., \mathbf{a}_n^{o*} \right)$ is Nash Equilibrium if no sensor node can reduce its cost by changing its offloading decision at the equilibrium $\mathcal{A}^*$, i.e.

$$U_i\left(\mathbf{a}_i^{o'},\mathbf{A}_{-i}^{o'}\right)\le U_i\left(\mathbf{a}_i^{o},\mathbf{A}_{-i}^{o}\right)\ \forall \mathbf{a}_i^{o'}\in\mathcal{A}\ ,i\in\mathcal{N} \tag{46}$$

We then prove the existence of Nash Equilibrium. To proceed, we first introduce the definition of the potential game.

Definition 2: If a potential function H exists, the game is called potential game. The function is expressed as,

$$U_i\left(\mathbf{a}_i^{o},\mathbf{A}_{-i}^{o}\right)-U_i\left(\mathbf{a}_i^{o'},\mathbf{A}_{-i}^{o'}\right)= H_i\left(\mathbf{a}_i^{o},\mathbf{A}_{-i}^{o}\right)-H_i\left(\mathbf{a}_i^{o'},\mathbf{A}_{-i}^{o'}\right) \tag{47}$$

The model has remarkable self-stability, which makes the sensor nodes in the equilibrium state obtain a mutually satisfactory solution, and there is no motive for the sensor node to deviate. This property is important for non-cooperative computation offloading problems, because the sensor nodes are selfish to act in their own interest. Meanwhile, according to the nature of potential game, there must be a pure strategic Nash equilibrium solution and it has the property of finite improvement.

Lemma 1. The following function for all sensor nodes is a potential function.

$$H\left(\mathbf{a}_i^{o},\mathbf{A}_{-i}^{o}\right)=\left(1-a_i^{E}\right)\left(\,(\sum_{j\in\mathcal{N},\ j\neq i}U_j^{E})+a_i^{L}U_i^{L}+a_i^{C}U_i^{C}\right)+a_i^{E}\sum_{i\in\mathcal{N}}U_i^{E} \tag{48}$$

Proof: There are three offloading modes for each sensor node. We first discuss the scheme of switching the offload decision between the cloud computing mode and the edge computing offloading mode. Assuming that the sensor node i chooses the edge computing mode, and then switch the offloading mode to the remote cloud computing mode.

Based on (16), we obtain the following equation.

$$U_i\left((0,1,0),\mathbf{A}_{-i}^{o}\right)-U_i\left((0,0,1),\mathbf{A}_{-i}^{o'}\right)=U_i^{E}-U_i^{C} \tag{49}$$

Based on (48), the potential function in this case are as follows

$$H\left((0,1,0),\mathbf{A}_{-i}^{o}\right)=\sum_{i\in\mathcal{N}}U_i^{E} \tag{50}$$

$$H\left((0,0,1),\mathbf{A}_{-i}^{o}\right)=(\sum_{j\in\mathcal{N},\ j\neq i}U_j^{E})+U_i^{C} \tag{51}$$

From (50) and (51), we can get

$$H\left((0,1,0),\mathbf{A}_{-i}^{o}\right)-H\left((0,0,1),\mathbf{A}_{-i}^{o}\right)=U_i^{E}-U_i^{C} \tag{52}$$

From (49) and (52), we can derive

$$U_i\left((0,1,0),\mathbf{A}_{-i}^{o}\right)-U_i\left((0,0,1),\mathbf{A}_{-i}^{o'}\right)=H\left((0,1,0),\mathbf{A}_{-i}^{o}\right)-H\left((0,0,1),\mathbf{A}_{-i}^{o}\right) \tag{53}$$

From (53), we can see that the difference between the potential function in the edge offloading mode and the cloud offloading mode is equal to that of the overhead function. Therefore, we prove the existence of Nash Equilibrium in the potential game. The offloading selection algorithm in the cloud-edge integrated EI network is proposed in Algorithm 2.

Algorithm 2: The distributed JTOARA algorithm

Input: N : number of the sensor nodes;

l : the index of iteration times;

$A_i, Q_i, w_i^t, w_i^e, w, h_i$.

Output: $\left\{ \mathcal{A}^*, \mathcal{C}_f^*, \mathcal{P}^* \right\}$:optimal resource allocation

1: Initialize: each node chooses edge offloading mode $\mathcal{A}_o$

2: for $i = 1$ to N do

3:   compute the total overhead of three offloading mode

4:   update the offloading mode with the minimum overhead

5: end for

6: update $\mathcal{A}$.

7: while $\mathcal{A} \neq \mathcal{A}_o$ do

8:     $\mathcal{A}_o = \mathcal{A}$ and $l = l + 1$

9:     for $i = 1$ to N do

10:       $a_i^M = 1$ and update $\mathcal{A}, \mathcal{P}, \mathcal{C}_f$

11:     compute the total overhead of three offloading mode

12:     update the offloading mode with the minimum overhead

13:     end for

14: end while

15: Output the optimal offloading scheme and resource allocation.

In Algorithm 2, firstly, all sensor nodes offload the computation tasks to the edge server. Then, we separately compute the local overhead, edge computing overhead and cloud computing overhead. After comparing the overhead of the three offloading modes, the offloading matrix $\mathcal{A}$ is updated. The updated matrix $\mathcal{A}$ is regarded as the new initial matrix $\mathcal{A}_o$ .Next, each sensor node is gone through and switch the offloading mode to edge computing mode. Meanwhile, each sensor node updates the computation offloading decision. When the offloading matrix $\mathcal{A}$ updated, the uplink transmission power of the sensor nodes and the computation resource of the edge server are recalculated. Then, each sensor node computes the total overhead of three offloading mode to update the offloading mode with the minimum overhead. After several iterations, the task offloading mode selection algorithm will reach Nash Equilibrium state.

5. Simulation Results

Table 1. Simulation parameters values

Parameters	Values
System bandwidth	10MHz
Noise power density	-100dBm/Hz
The maximum power of sensor nodes	300mW
Data size of the task	[0.1-1] MB
The number of require CPU cycle for task	[0.1-0.9] GHz
The maximum CPU rate of edge server	20GHz
The maximum CPU rate of cloud server	40GHz

In this section, simulation results are presented to evaluate the performance of the proposed algorithm. In the three-tier cloud-edge integrated EI network, terminal users are sensor nodes equipped with energy internet device. We consider one base station and 10 sensor nodes. The wireless channel gain is chosen as follows: $h = 127 + 30\log d$, where d represents the distance between the sensor node and the base station. The simulation parameters are summarized in **Table 1**.

To evaluate the performance of the JTOARA, we compare the proposed JTOARA with four other base algorithms in their respective framework.

1) Local computing. It means all real-time monitor computation tasks from EI are decided to executed locally.

2) Edge computing. Each sensor node from EI determines executed their real-time monitor computation task on the edge node.

3) Cloud computing. All real-time monitor computation tasks from EI are offloaded to the remote cloud.

4) Enumerative algorithm. This is a brute-force method to find the optimal offloading scheduling solution via searching 2^n possible decisions through the exhaustive analysis.
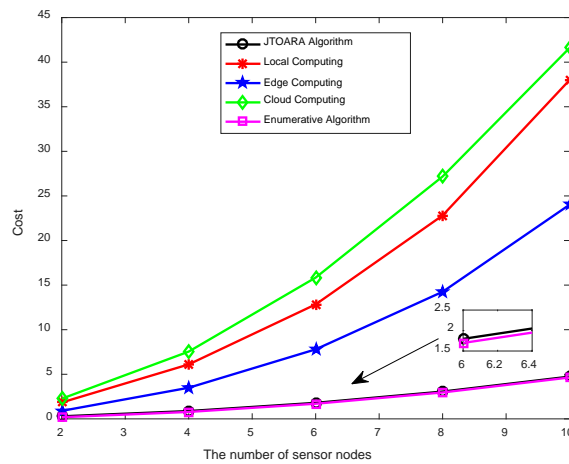
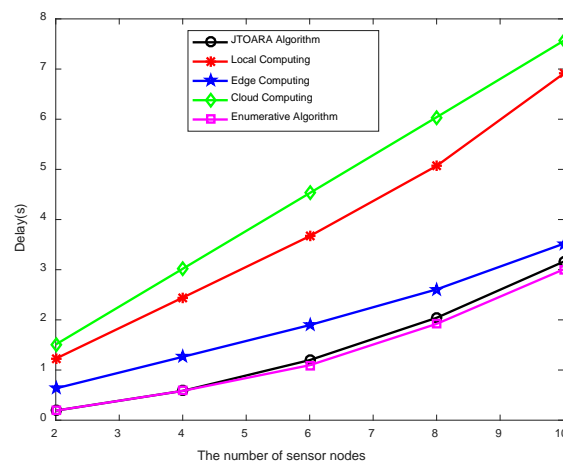**Fig. 2.** The system cost versus the number of sensor nodes

Fig. 2 displays the total cost of five algorithms versus the number of sensor nodes in the cloud-edge integrated EI network. From **Fig. 2**, it is seen that as the number of senor nodes increases, the total cost continues to increase. By comparison with local computing, edge computing, and cloud computing, the total cost of the JTOARA is minimum. For example, when the number of the sensor node is 6, it can be noted that the total cost of the JTOARA algorithm is 1.797, which is lower than local computing, edge computing and cloud computing. Then, the total cost of our proposed algorithm is a little higher than but close to the enumerative algorithm.
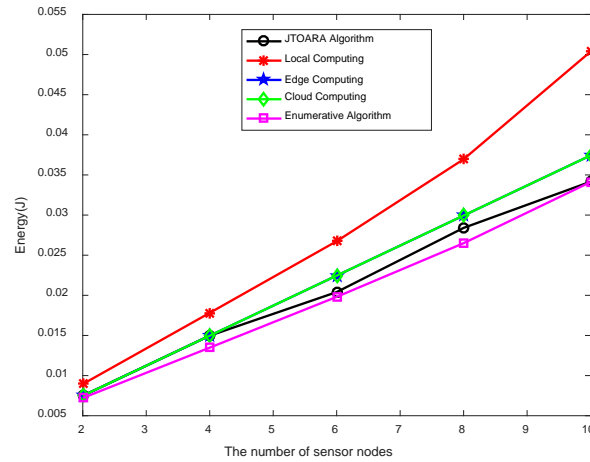
Fig. 3. The energy consumption versus the number of sensor nodes.
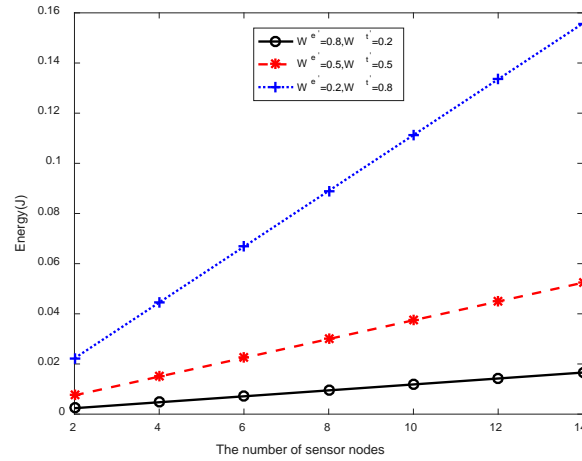
Fig. 3 shows the energy consumption versus the number of sensor nodes in the cloud-edge integrated EI network. From **Fig. 3**, we can find that as the number of sensor nodes increases, the energy consumption of the sensor nodes from five schemes increases. We can observe that the energy consumption for "local computing" is the highest. However, the energy consumption of the sensor nodes in cloud offloading mode is equal of that in edge offloading mode. The reason is that, the energy consumption of the sensor node is the transmission energy, which is the same in both cloud and edge computing modes.

Fig. 4 depicts the impact of the number of sensor nodes on the system delay in the cloud-edge integrated EI network. From **Fig. 4**, we can find that the system delay increases with the number of sensor nodes. As the cloud server is far from the sensor nodes, the system delay of cloud computing is the highest. The delay of our proposed algorithm is close to the enumerative algorithm, which has lower time complexity. Hence, our proposed optimal offloading scheme has better performance compared with other schemes.
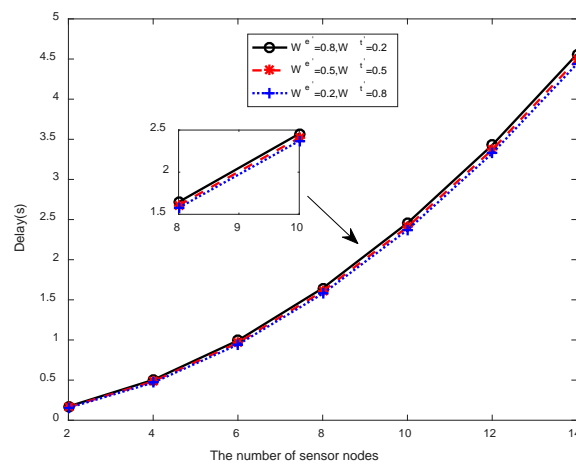

Fig. 4. The system delay versus the number of sensor nodes.

Fig. 5 displays the energy consumption versus the number of sensor nodes under different weighting factor in the cloud-edge integrated EI network. From **Fig. 5**, we can observe that as the sensor nodes increase, the energy consumption of the sensor nodes increases. Besides, as the energy weighting factor value increases, the energy consumption of sensor nodes decrease. The reason is that as the weighting on energy consumption increases, in order to reduce the total cost of the system, the energy consumption will be correspondingly reduced.


Fig. 5. The energy of sensor nodes versus the number of sensor nodes.

Fig. 6 shows the system delay versus the number of sensor nodes under different weighting factor in the cloud-edge integrated EI network. From **Fig. 6**, we can observe that as the sensor nodes increase, the system delay increases. Besides, as the delay weighting factor value increases, the system delay decreases. The reasons are as follows: as the weighting on system delay increases, in order to reduce the total cost of the system, the system delay will be correspondingly reduced.


Fig. 6. The system delay versus the number of sensor nodes

Fig. 7 shows the cost versus the number of required CPU cycles per task in the cloud-edge integrated EI network. From **Fig. 7**, it is seen that as the number of required CPU cycles per task increases, the total costs in all schemes increase too. And our scheme has better performance than other three baseline algorithms. The reason is that both the local completion time, edge server execution time, and remote cloud execution time increase as the CPU cycles increases. Besides, compared with the stochastic algorithm and the average algorithm, the cost of the JTOARA has better performance.
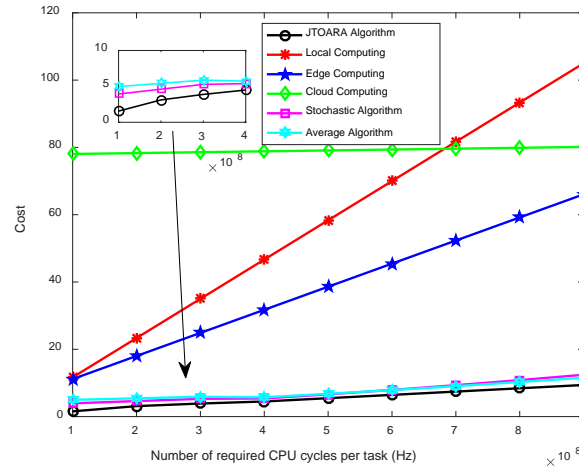
Fig. 7. The system cost verse the number of required CPU cycles per task.

Fig. 8 depicts the cost versus the data size of the task in the cloud-edge integrated EI network. From **Fig. 8**, for all schemes except local computing, we can note that as the increase of data size of the task, the total cost increases. Moreover, our proposed scheme has the best performance than other three baseline schemes. The reason for this is that as the data size of the computation task increases, the delay and energy consumption for offloading computation task becomes higher. Besides, compared with the stochastic algorithm and the average algorithm, the cost of the JTOARA has better performance.
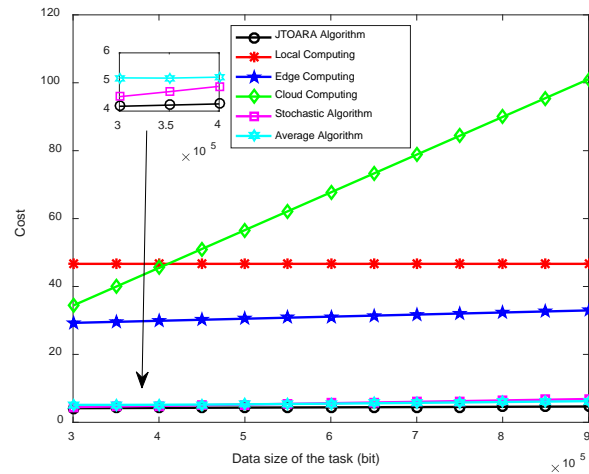
Fig. 8. The system cost verse the data size of the task

6. Conclusion

In this paper, we propose a cloud-edge collaborative computing task scheduling and resource allocation algorithm, which minimizes the total system cost. The system model including cloud center, edge server and the sensor nodes is built for the cloud-edge integrated EI network, where each sensor node has an indivisible task that can be executed locally, at the edge node, or at a remote cloud cooperatively. To improve the efficiency of EI and the QoS of the energy applicants, we propose a joint task offloading and resource allocation scheme under the limited communication and computation resource constrains, in which the optimization problem is a NP-hard problem, which is difficult to solve. To obtain the optimal solution, the optimization problem is divided into the power allocation sub-problem, the computation resource allocation sub-problem and the offloading scheme selection sub-problem. A bisection search algorithm is developed to obtain the optimal power allocation for each sensor node. Then, we derive the optimal computation resource allocation of the edge server by KKT condition and convex optimization theory. Furthermore, we establish a game model to obtain the optimal task offloading scheme. The simulation results show that the proposed algorithm can significantly reduce the total system cost, has a fast convergence rate, and decrease the communication and computation delay compared with conventional approaches.

Reference

[1] L. Cheng, N. Qi, F. Zhang, H. Kong and X. Huang, "Energy Internet: Concept and practice exploration," in *Proc. of 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1-5, Nov. 2017. Article (CrossRef Link)

[2] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng and S. J. Dale, "The Future Renewable Electric Energy Delivery and Management (FREEDM) System: The Energy Internet," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 133-148, Jan. 2011. Article (CrossRef Link)

[3] Zhou, Kaile, Shanlin Yang, and Zhen Shao, "Energy internet: the business perspective," *Applied Energy*, vol. 178, pp. 212-222, Sep. 2016. Article (CrossRef Link)

[4] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu, "Mobile cloud computing: A survey," *Future generation computer systems*, vol. 29, no.1, pp. 84-106, Jan. 2013. Article (CrossRef Link)

[5] Cheraghlou, Mehdi Nazari, Ahmad Khademzadeh, and Majid Haghparast, "New fuzzy-based fault tolerance evaluation framework for cloud computing," *Journal of Network and Systems Management*, vol 27, no.4, pp. 930-948, Feb. 2019. Article (CrossRef Link)

[6] Muthulakshmi, B., and K. Somasundaram, "A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment," *Cluster Computing*, vol 22, no.5, pp. 10769-10777, Sep. 2019.

[7] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018. Article (CrossRef Link)

[8] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, Aug. 2017. Article (CrossRef Link)

[9] M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587-597, Mar. 2018. Article (CrossRef Link)

[10] K. Zhang et al., "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," *IEEE Access*, vol. 4, pp. 5896-5907, Aug. 2016. Article (CrossRef Link)

[11] T. Q. Dinh, J. Tang, Q. D. La and T. Q. S. Quek, "Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571-3584, Aug. 2017. Article (CrossRef Link)

[12] C. You, K. Huang, H. Chae and B. Kim, "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397-1411, Mar. 2017. [Article (CrossRef Link)](#)

[13] Sun J, Gu Q, Zheng T, et al, "Joint communication and computing resource allocation in vehicular edge computing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, pp. 1-13, Mar. 2019. [Article(CrossRef Link)](#)

[14] J. Pan and J. McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439-449, Feb. 2018. [Article (CrossRef Link)](#)

[15] Xu, Xiaolong, et al, "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol.95, pp.522-533, Jan. 2019. [Article (CrossRef Link)](#)

[16] Y. Wang, X. Tao, X. Zhang, P. Zhang and Y. T. Hou, "Cooperative Task Offloading in Three-Tier Mobile Computing Networks: An ADMM Framework," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2763-2776, Mar. 2019. [Article (CrossRef Link)](#)

[17] Y. Liu, F. R. Yu, X. Li, H. Ji and V. C. M. Leung, "Distributed Resource Allocation and Computation Offloading in Fog and Cloud Networks with Non-Orthogonal Multiple Access," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12137-12151, Dec. 2018. [Article (CrossRef Link)](#)

[18] H. Guo and J. Liu, "Collaborative Computation Offloading for Multiaccess Edge Computing Over Fiber–Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4514-4526, May 2018. [Article (CrossRef Link)](#)

[19] J. Ren, G. Yu, Y. He and G. Y. Li, "Collaborative Cloud and Edge Computing for Latency Minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031-5044, May 2019. [Article (CrossRef Link)](#)

[20] C. Kai, H. Zhou, Y. Yi and W. Huang, "Collaborative Cloud-Edge-End Task Offloading in Mobile-Edge Computing Networks with Limited Communication Capability," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 324-364, Aug. 2020. [Article (CrossRef Link)](#)

[21] Zhang, Huaguang, et al, "Distributed optimal energy management for energy internet," *IEEE Transactions on Industrial Informatics*, vol.13, no. 6, pp. 3081-3097, Jun. 2017. [Article (CrossRef Link)](#)

[22] Yang T, Guo Q, Tai X, et al, "Applying blockchain technology to decentralized operation in future energy internet," in *Proc. of IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1-5, Jan. 2017. [Article (CrossRef Link)](#).

[23] Stephen Boyd L V, Stephen Boyd L V, "Convex optimization," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1859-1859, 2006. [Article (CrossRef Link)](#)

[24] W. Saad, Z. Han, M. Debbah, A. Hjorungnes and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 77-97, Sep. 2009. [Article (CrossRef Link)](#)

Xin Song was born in Jilin, China, in 1978. She received her Ph.D. degree in Communication and Information System in Northeastern University in China in 2008. She is now a teacher working in Northeastern University at Qinhuangdao, China. Her research interests are in the area of robust adaptive beam-forming and wireless communication.

Yue Wang was born in Shandong Province, China, in 1995. She received the B.S degree from Shandong Normal University, Jinan, in 2017. She is currently pursuing the M.S degree with Engineering Optimization and Smart Antenna Institute from Northeastern University. Her research interests include resource allocation in mobile edge computing.

Zhigang Xie received the B.E. degree in communications engineering from the Northeastern University at Qinhuangdao, Qinhuangdao, China in 2012 and the M.E. degree in Software Engineering from Northeastern University, Shenyang, China in 2016. He is currently working towards his Ph.D. degree in Communication and Information System at Northeastern University, Shenyang, China. His research interests include wireless communication networks, Internet of Things, mobile edge computing, and green wireless communications.

Lin Xia received the B.E. degree in Electronics and Information Engineering from the University of Science and Technology Liaoning, China, 2018. She is currently pursuing the M. S. degree in Electronic and communications engineering with Northeastern University (NEU), Shenyang, China. Her current research interests include full-duplex device-to-device communications, physical layer security and wireless resource allocation.