Task Allocation Mechanism of Power Internet of Things Based on Edge Routing Optimization

Cheng Zhong[1(✉)], Yi Li[2], Sujie Shao[2], Shaoyong Guo[2], and Qianjun Wang[2]

[1] Xiong'an New District Power Supply Company of State Grid Hebei Electric Power Co., Ltd., Xiong'an 071600, Hebei, China
1094642873@qq.com

[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
buptssj@bupt.edu.cn

Abstract. With large number of smart devices accessing the power Internet of Things, cooperative edge computing has become a new computing paradigm for delay sensitive business. However it still faces some challenges today: 1) task allocation in edge networks, 2) route selection during data transmission. This paper proposes a cooperative computing mechanism based on edge routing optimization for power Internet of Things to solve these problems. Firstly, establish a task allocation model based on multi-edge-node cooperation in which the data routing process includes three stage: sending route, merging route and returning route. The model is designed to minimize the average task completion delay. Secondly, Improved Biogeography-based Optimization in Task Allocation and Sending Route Selection (IBBO-TASRS) algorithm is proposed to solve the task allocation and route selection. The simulation results show that the proposed algorithms can reduce the service delay well.

Keywords: Power Internet of Things · Cooperative edge computing · Task allocation · Route optimization

1 Introduction

With the development of the power Internet of Things [1], many new business terminals are emerging and connected to the sensor network [2]. For example, the remote monitoring business uses cameras to collect video data of power scenes and perform real-time intelligent analysis. Smart Augmented Reality (AR) glasses sense the operation and maintenance site and guide the operation. Unmanned Aerial Vehicles (UAV) conduct all-round inspections of remote

Supported by the Science and Technology Project of State Grid Corporation of China: Research and application of key technologies for wireless communication network coverage of urban power underground pipe gallery (Grant No. 5700-202113189A-0-0-00).

F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 118–131, 2022.
https://doi.org/10.1007/978-3-031-23141-4_9

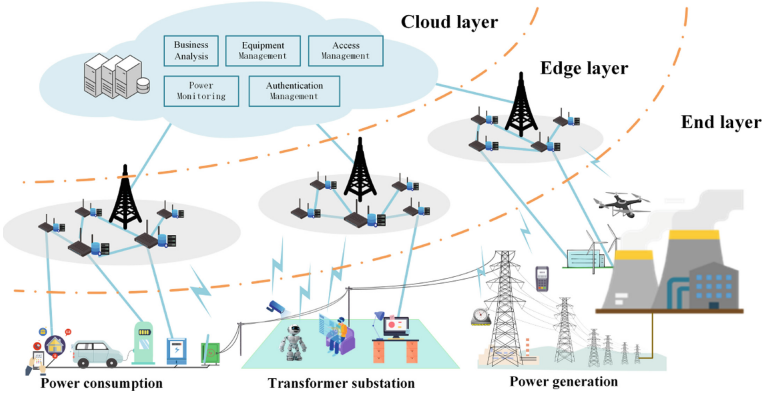


Fig. 1. The power Internet of Things architecture based on edge computing

transmission lines. The extensive deployment of a large number of sensing devices and control systems has improved the holographic perception capabilities of the power grid and the operating efficiency of power equipment. However, the reliability and safety of these service terminals have aroused widespread concern, and it is necessary to prevent attackers from eavesdropping and interception data and ensure work efficiency in complex environments. At the same time, these applications usually require a lot of computing resources and require high response delays. Wearable devices such as AR glasses usually require 10–50 ms. Today, edge computing [3], as an extension of cloud computing [4], has become an effective solution for delay and resource-sensitive services. Figure 1 shows the architecture of the power Internet of Things based on edge computing. The end layer is divided into power consumption area, power transformation area and power generation area, covering video monitoring, equipment detection, charging pile, intelligent patrol inspection, smart home and other businesses. The edge layer deploys edge nodes with computing and storage resources, such as wireless access points, routers, edge servers, etc. Clouds far away from business terminals have a large amount of computing resources, while providing business analysis, equipment management and other functions. Power business terminals are connected to edge nodes by means of wired, Wi-Fi, micro-power wireless, 4 G/5 G, and low-power wide area networks, etc. The computing tasks are uploaded to edge nodes to be completed nearby, which reduce network delay and cloud center load.

Considering the limited computing and storage resources of edge nodes, it is usually difficult for a single node to meet the demanding resource and delay requirements of service terminals. Therefore, it is necessary to adopt the cooperative edge computing [5] mode to calculate tasks initiated by service terminals. [6] has shown that cooperative edge computing is better than single node computing. However, most of the existing cooperative edge computing literature only considers the problem of task allocation without route optimization into the research [7], such as data sending route, returning route, etc. However, most of the literatures on collaborative edge computing do not consider the opti-

mization of data routing in edge networks, such as task sending routing and computation result aggregation routing, and different routing choices will have a great impact on the delay of task transmission. Therefore, in collaborative edge computing, not only the choice of computing nodes should be considered, but also the routing scheme should be paid attention to in order to reduce the delay in both transmission and computation processes. This complicates the problem of collaborative computing.

To solve the problem of task allocation and routing selection in cooperative edge computing, this paper proposes a cooperative computing mechanism based on edge routing optimization for power Internet of Things, which optimizes task allocation and route selection at the same time, and minimizes the average task completion delay while meeting business resource requirements and delay requirements. The specific contributions are as follows:

1) In order to reduce the delay of business completion in the power Internet of Things scenario, a task allocation model based on multi-edge-node cooperation is proposed. In the model, business request and resource models, delay models and data route models are established respectively. The data route model includes three stages: sending route, merging route and returning route. Finally, the problem of minimizing the average task completion delay under the condition of satisfying the business resource demand and delay demand is put forward.
2) The IBBO-TASRO algorithm is proposed to solve the task allocation and route selection. In IBBO-TASRO the elite retention strategy and improved migration probability are used to improve the biogeographic algorithm to avoid the problem of task assignment getting into local optimum.

2 Related Work

With the development of the Internet of Things in the future, the number of terminals will show explosive growth while creating more data. The cloud computing model will bring about big data transmission, privacy and energy consumption issues [8,9]. Cooperative edge computing uses multiple edge nodes to contribute resources to participate in task computing, solving the problem of insufficient resources of a single node and high latency of cloud computing. The current research on cooperative edge computing includes task allocation, route selection and other issues.

The task allocation mainly studies how tasks are distributed among the edge nodes and cloud centers to achieve optimal latency or energy consumption. [10] studied how the task request initiated by the business terminal is mapped to the edge node problem and the resource allocation problem to minimize the service delay, and finally proposes the BRT algorithm to solve the optimal task allocation and resource allocation decisions. [11] proposed a cooperative computing offloading framework based on three-tier mobile edge computing networks. At the same time, it also considered horizontal offloading between edge nodes. Joint

optimization of offloading decision-making and computing resource allocation is used to minimize average delay.

The route selection problem mainly studies the transmission route problem of tasks in the edge network. [12] studied the energy management problem of MEC on power-constrained equipment. In order to solve the trade-off between data processing capacity and energy efficiency, a sustainable strategic intelligence drives edge routing algorithm is proposed. [13] proposed a temporary on-demand multi-path distance vector routing protocol based on link lifetime and energy consumption prediction to save routing discovery when routing fails cost (Table 1).

3 Task Allocation Model Based on Multi-edge-node Cooperation

3.1 Business Requests and Resources

Edge nodes (EN) with computing and storage capabilities are deployed in the network, represented by set $\mathcal{M} = \{1, 2, ..., M\}$, and there are a large number of user end (UE) accessing the network, represented by set $\mathcal{N} = \{1, 2, ..., N\}$.

The task sent by UE i is represented by $\mathcal{T}_i$, which includes more than one subtasks represented by $w_{ij} = (c_{ij}, e_{ij}, t_{ij}, d_{ij}, \lambda_{ij})$. c_{ij} represents computing resource requirement, e_{ij} represents storage resource requirement, d_{ij} represents the amount of input data. t_{ij} represent the processing delay when the resource requirements are met. λ_{ij} represents the ratio of the result data amount to the input data amount. There is no temporal dependency between subtasks. $\hat{t}_i$ represents the delay constraint of task $\mathcal{T}_i$.

Because of the resource heterogeneity of EN, containers and virtualization technologies are used to support the realization of EN resource allocation, and the amount of resources required by subtasks is represented by the number of virtual resource units. It is assumed that the task requests of all UEs are sent at the same time. The task allocation decision $X = \{x_{ijk}\}$ is specified as follows,

$$x_{ijk} = \begin{cases} 1 & \text{if subtask } j \text{ of UE } i \text{ is allocated to EN } k \\ 0 & \text{else} \end{cases} \tag{1}$$

Considering that each subtask must be calculated by one node, the following constraints need to be met:

$$\sum_{k \in \mathcal{M}} x_{ijk} = 1 \tag{2}$$

The resource of EN k is represented by (C_k, E_k). C_k means the number of virtual computing resource units. E_k means the number of virtual storage resource units. Therefore, the task allocation decision X has the following constraints,

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{T}_i} c_{ij} x_{ijk} \leq C_k, k \in \mathcal{M} \tag{3}$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{T}_i} e_{ij} x_{ijk} \leq E_k, k \in \mathcal{M} \tag{4}$$

Table 1. Definition of variables

Symbol	Description
N	the number of UEs
M	the number of ENs
$\mathcal{N}$	UE set
$\mathcal{M}$	EN set
R_i	EN set for completing subtasks of UE i
$\mathcal{T}_i$	subtask set requested by UE i
w_{ij}	subtask j of UE i
c_{ij}	the computing resource requirement of w_{ij}
e_{ij}	the storage resource requirement of w_{ij}
d_{ij}	the amount of input data of w_{ij}
t_{ij}	the computing delay of w_{ij} if the resource requirement is met
$\hat{t}_i$	the delay constraint of $\mathcal{T}_i$
λ_{ij}	the ratio of the result data amount to the input data amount
C_k	the number of virtual computing units of EN k
E_k	the number of virtual storage unit of EN k
x_{ijk}	whether subtask j of UE i is allocated to EN k
$y_{i\overline{k}}$	whether the result of task $\mathcal{T}_i$ is merged on EN $\overline{k}$
u_i	the access point of UE i
$\mathcal{N}_k$	the UE set associated with EN k
B_k	the bandwidth of EN k
p_i	the transmission power of UE i
h_{i,u_i}	the UE i and ENu_i channel gain
σ^2	the additive Gaussian white noise power
y_i	the signal-to-noise ratio of UE i and EN u_i
$f_{iju_ikg}^{send}$	whether the gth routing path from u_i to k is selected for input data sending of w_{ij}
$f_{ijk\overline{k}g}^{merge}$	whether the gth routing path from k to $\overline{k}$ is selected for result data merging of w_{ij}
$f_{i\overline{k}u_ig}^{return}$	whether the gth routing path from $\overline{k}$ to u_i is selected for result data of returning of w_{ij}
$W_{kk'g}$	whether there is a gth path between EN k and EN k'
$H_{kk'g}$	number of hops of gth routing path between EN k and EN k'
$R_{kk'g}$	number of transmission bandwidth of gth routing path between EN k and EN k'
$T_{i,r_{i1},r_{i2}}^{finish}$	the cooperative completion delay of subtask $\mathcal{T}_i$ by r_{i1}, r_{i2}
$T_{i,r_{i1},r_{i2}}^{comm}$	the data transmission delay of subtasks of UE i from EN r_{i1} to EN r_{i2}
$T_{i,r_{i1}}^{comp}$	the computing delay of the subtasks of UE i on EN r_{i1}

3.2 Delay

Each UE connect to the nearest EN. Assume the access point of UE i is EN $u_i \in \mathcal{M}$. The UE set associated with EN k can be represented as $\mathcal{N}_k = \{i : i \in \mathcal{N}, u_i = k\}$.

The bandwidth resource of EN k is B_k Hz. We defaults that the UEs associated with EN allocate the EN bandwidth resources evenly. The signal-to-noise ratio of UE i and EN u_i is $y_i = \frac{p_i h_{i,u_i}}{\sigma^2}$, where p_i represents UE i transmission power, h_{i,u_i} represents UE i and ENu_i channel gain, and σ^2 represents additive Gaussian white noise power. Therefore when the UE i accesses the EN u_i, the uplink data transmission rate of the UE i expressed as follow.

$$v_i = \frac{B_{u_i}}{|\mathcal{N}_{u_i}|} log_2(1 + y_i) \tag{5}$$

The network delay from UE i to EN u_i includes the transmission delay caused by the port rate and propagation delay. c represent the propagation speed of the wired or wireless channel, and D_{i,u_i} represent the physical distance between the UE i and EN u_i. Therefore, the network delay of task input data transmission to access point EN can be expressed as,

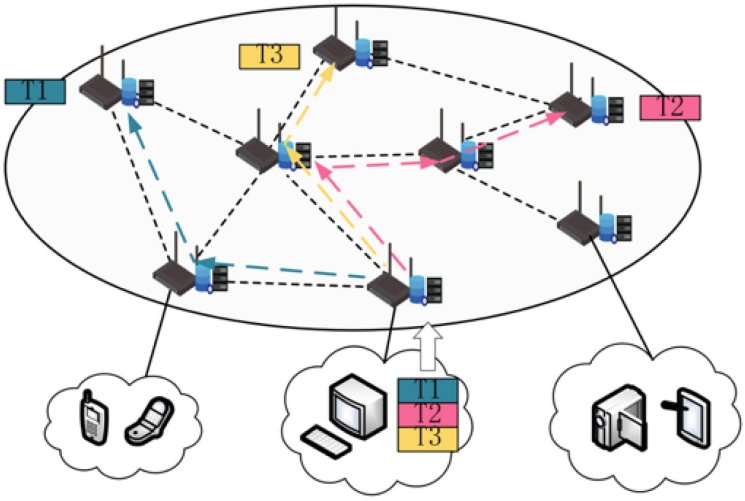
$$T^{up}_{i,u_i} = \frac{\sum_{j \in \mathcal{T}_i} d_{ij}}{v_i} + \frac{D_{i,u_i}}{c} \tag{6}$$

Similar to [14,15], the downlink bandwidth of the UE is much higher than the uplink bandwidth, and data size after task processing is usually much smaller than before, so we ignore the downlink transmission delay of sending the task results from EN to UE.

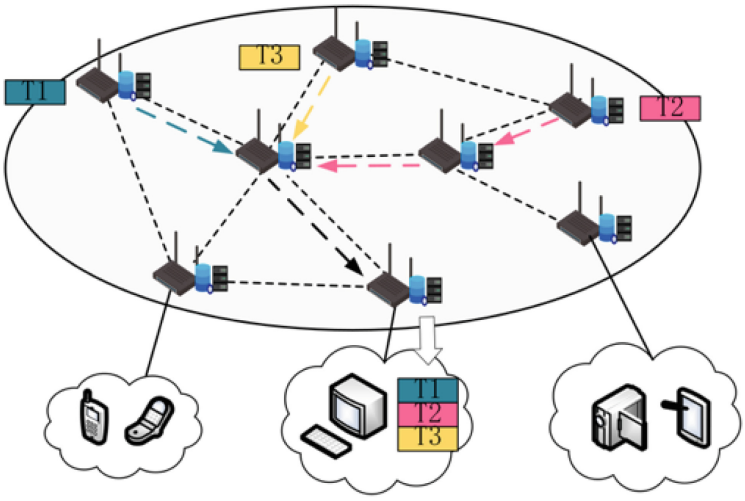
The computing task sent by the UE arrives at the computing node through multiple EN forwarding during transmission in the edge network. Assume that there are at most G routing paths available between any two EN, and $W_{kk'g}$ represent if there is a gth path between EN k and EN k'. Number of hops and transmission bandwidth of gth routing path between EN k and EN k' is represented by $H_{kk'g}$ and $R_{kk'g}$. For convenience, we assume that queuing delays on EN are not taken into account when data transferred over the network. We also assume that there will be no device or link failures in the network.

3.3 Data Route

The cooperative computing delay of task includes computing delay in edge node and data transmission delay in the edge network. Considering that the delay of task computation is fixed, it is only necessary to optimize the task allocation and route selection. The multi-edge-node cooperative computing process is shown in Fig. 2. Firstly, each subtask sent by UE starts from the access point and is transmitted to the corresponding computing node through the sending route, as shown in Fig. 2(a). After the subtask is completed, all the processing results of



(a) Data sending route and task computing



(b) Result merging and returning route

Fig. 2. The process of multi-edge-node cooperative computing

each subtask are sent to a EN to merge, and finally returned to the access point, as shown in Fig. 2(b).

According to the location of task data in the network, the route process in cooperative computing is divided into three stages: sending route, merging route, and returning route. The starting point of the sending route is the UE's access point, and the end point is the computing node, which is determined by the task allocation decision X. After the subtask is completed the start point of merging route is the computing node, and the end point is the merging node. The starting

point in the returning route is the merging node, and the end point is the access point. It can be seen that the selection of different merging nodes will have an impact on the delay of merging route and returning route. We set the merging node decision as $Y = \{y_{i\overline{k}}\}$, $i \in \mathcal{N}$, $\overline{k} \in \mathcal{M}$. Its value specification is as follows:

$$y_{i\overline{k}} = \begin{cases} 1 & \text{if merging node of task } \mathcal{T}_i \text{ is } \overline{k} \\ 0 & \text{else} \end{cases} \tag{7}$$

Since each task needs to select a merging node, there are the following constraints:

$$\sum_{\overline{k} \in \mathcal{M}} y_{i\overline{k}} = 1, i \in \mathcal{N} \tag{8}$$

We use $F^{send} = \{f^{send}_{iju_ikg}\}$, $F^{merge} = \{f^{merge}_{ijk\overline{k}g}\}$, $F^{return} = \{f^{return}_{i\overline{k}u_ig}\}$ to represent the route selection strategy for the three stages. $f^{send}_{iju_ikg}$ indicates whether to choose the gth path when the input data of subtask w_{ij} are transmitted to EN k in sending stage. $\{f^{merge}_{ijk\overline{k}g}\}$ and $\{f^{return}_{i\overline{k}u_ig}\}$ are the same as above. The value specification of $F^{send} = \{f^{send}_{iju_ikg}\}$ is as follows:

$$f^{send}_{iju_ikg} = \begin{cases} 1 & \text{The } g\text{th routing path from } u_i \text{ to } k \text{ is selected for} \\ & \text{data } w_{ij} \\ 0 & \text{else} \end{cases} \tag{9}$$

The value specification of F^{send} and F^{return} is the same as above.

Because the start point of the sending route is the UE's access point, the end point is determined by X, and only one path between them can be selected, so the constraints of F^{send} are as follows:

$$\sum_{k \in \mathcal{M}} (x_{ijk} \sum_{g=1}^{G} W_{u_ikg} f^{send}_{iju_ikg}) = 1, \forall i \in \mathcal{N}, j \in \mathcal{T}_i \tag{10}$$

$$\sum_{k \in \mathcal{M}} \sum_{g=1}^{G} f^{send}_{iju_ikg} = 1, \forall i \in \mathcal{N}, j \in \mathcal{T}_i \tag{11}$$

In the same way, the start point of the merging route is determined by the task allocation decision X, and the end point is determined by Y, and only one path between them can be selected. So the constraints of F^{merge} are as follows:

$$\sum_{k \in \mathcal{M}} (x_{ijk} \sum_{\overline{k} \in \mathcal{M}} (y_{i\overline{k}} \sum_{g=1}^{G} W_{k\overline{k}g} f^{merge}_{ijk\overline{k}g})) = 1, \forall i \in \mathcal{N}, j \in \mathcal{T}_i \tag{12}$$

$$\sum_{k \in \mathcal{M}} \sum_{k \in \mathcal{M}} \sum_{g=1}^{G} f^{merge}_{ijk\overline{k}g} = 1, \forall i \in \mathcal{N}, j \in \mathcal{T}_i \tag{13}$$

The constraints of $F^{return}_{ij\overline{k}u_ig}$ are as follows:

$$\sum_{\overline{k}\in\mathcal{M}}(y_{i\overline{k}}\sum_{g=1}^{G}W_{\overline{k}u_ig}f^{return}_{i\overline{k}u_ig})=1,\forall i\in\mathcal{N} \tag{14}$$

$$\sum_{\overline{k}\in\mathcal{M}}\sum_{g=1}^{G}f^{return}_{i\overline{k}u_ig}=1,\forall i\in\mathcal{N} \tag{15}$$

The delay calculation process of cooperative computing is as follows:

Sending Delay. The network transmission delay of the subtasks w_{ij} from the access point u_i to the computing node is:

$$T^{send}_{ij}=\sum_{k\in\mathcal{M}}\sum_{g=1}^{G}x_{ijk}W_{u_ikg}f^{send}_{iju_ikg}H_{u_ikg}\frac{d_ij}{R_{u_ikg}},\forall i\in\mathcal{N},j\in\mathcal{T}_i \tag{16}$$

Computing Delay. Under the condition that the requirements of computing resource and storage resource are met, the computing delay of each subtask is $T^{comp}_{ij}=t_{ij}$

Result Merging Delay. After the subtask w_{ij} has been processed, the delay for sending the result to the merging node is:

$$T^{merge}_{ij}=\sum_{k\in\mathcal{M}}\sum_{\overline{k}\in\mathcal{M}}\sum_{g=1}^{G}x_{ijk}y_{i\overline{k}}W_{k\overline{k}g}f^{merge}_{ijk\overline{k}g}H_{k\overline{k}g}\frac{d_{ij}}{R_{k\overline{k}g}},\quad \forall i\in\mathcal{N},j\in\mathcal{T}_i \tag{17}$$

Result Returning Delay. After the results of $\mathcal{T}_i$ all arrive at the merging node, the delay of data transmission to the access point u_i is:

$$T^{return}_{i}=\sum_{\overline{k}\in\mathcal{M}}\sum_{g=1}^{G}y_{i\overline{k}}W_{\overline{k}u_ig}f^{return}_{ij\overline{k}u_ig}H_{\overline{k}u_ig}\frac{\sum_{j\in\mathcal{T}_i}\lambda_{ij}d_{ij}}{R_{\overline{k}u_ig}},\forall i\in\mathcal{N}$$

Considering that the merging time depends on the latest subtask in $\mathcal{T}_i$ that arrives at the merging node, the cooperative completion delay of $\mathcal{T}_i$ is:

$$T^{coop}_{i}=\max_{j\in\mathcal{T}_i}\{T^{send}_{ij}+T^{comp}_{ij}+T^{merge}_{ij}\}+T^{return}_{i} \tag{18}$$

3.4 Problem Model

Combining Sects. 3.1, 3.2, and 3.3, the total delay T_i for task $\mathcal{T}_i$ from sending request to receiving the result can be obtained as

$$T_i = T_{i,u_i}^{up} + T_i^{coop} + T_{i,u_i}^{down} \tag{19}$$

T_{i,u_i}^{down} represents the delay of the result from the EN u_i to the UE i. As described in Sect. 3.2, the delay is small and ignored. At the same time, the delay of generating task allocation and route selection decisions are ignored. Therefore, the task completion delay of the UE i is

$$T_i = T_{i,u_i}^{up} + T_i^{coop} \tag{20}$$

The average completion delay of all UE tasks is

$$\overline{T} = \frac{1}{N} \sum_{i \in \mathcal{N}} T_i \tag{21}$$

We describes task allocation and route selection as a problem of minimizing the average completion delay of tasks, as shown below:

$$\text{P1:} \min_{X,Y,F^{send},F^{merge},F^{return}} \overline{T}$$
$$\text{s.t. constraints (1)–(4),(7)–(15)}$$

4 Algorithm

In this section, we propose an Improved Biogeography-based Optimization Algorithm in Task Allocation and Sending Route Optimization (IBBO-TASRO) to solve the problem of task allocation X and route selection F^{send}.

4.1 Task Allocation and Sending Route Strategy

Decisions in P1 includes task allocation X, three-stage route selection F^{send}, F^{merge}, F^{return} and merging node Y. It is difficult to solve for these variables at the same time. The solution space is relatively large and it is difficult to meet real-time requirements. Considering that in the case of meeting service resource requirements, the closer the node participating in the task computing and the result merging is to the access point, the shorter the data transmission path in the network, and the smaller the average task completion delay. Based on this idea, we initialize the merging node decision, set the merging node of each task at the access node EN u_i, $\overline{Y} = \{y_{ik} | \forall i \in \mathcal{N}, y_{iu_i} = 1, y_{ik} = 0, k \in \mathcal{M}/\{u_i\}\}$. The delay of returning route is 0. We convert problem P1 to P2:

$$\text{P2:} \min_{X,F^{send},F^{merge}} \overline{T}$$
$$Y = \overline{Y}$$
$$\text{s.t. constraints (1)–(4),(7)–(15)}$$

The IBBO-TASRO algorithm is used to solve P2. The IBBO-TASRO algorithm flow is as follows.

Algorithm 1. IBBO-TASRO

Require: $L, S_{max}, I, E, m_{max}, \Omega, \mathcal{T}_i, i \in \mathcal{N}, C_k, E_k, k \in \mathcal{M}$.
Ensure: X^*, F^{send*}

Initialize task allocation decisions X^l for each habitat.
for $n = 1$ to ω **do**
  for $l = 1$ to L **do**
    for $i = 1$ to N **do**
      for j to $\mathcal{T}_i$ **do**
        calculate $F^{send,l}, F^{merge,l}$ according to Eq.(X)
      end for
    end for
  end for
  calculate HIS according to $\overline{T}(X, \widetilde{Y}, F^{send}, F^{merge})$
  select the best top 2 habitat as collection ψ
  for $l = 1$ to L **do**
    if $l \in \psi$ **then**
      continue
    end if
    if rand(0,1) $<\lambda'(s_l)$ **then**
      for $l'1 = 1$ to L **do**
        if rand(0,1) $<\mu(s_l)$ **then**
          Generate a variable i randomly in range$[0, D]$
          $x_i^l \longleftarrow x_i^{l'}$
          if X^l doesn't meet constraint (3) or (4) **then**
            replace X^l with a new habitat within bounds.
          end if
        end if
      end for
    end if
  end for
  for $l = 1$ to L **do**
    if $l \in \psi$ **then**
      continue
    end if
    if $rand(0, 1) < m(s_i)$ **then**
      $x_i^l \longleftarrow 1 + rand(M)$
    end if
    if X^l doesn't meet C1 C2 then **then**
      replace X^l with a new habitat within bounds.
    end if
  end for
end for

5 Experimental Results and Analysis

In order to verify the superiority of the joint optimization algorithm of task assignment and routing proposed in this paper, we conduct a simulation exper-

iment. The simulation environment was run on a 64 bit MacOS Sierra system with a 2.8 GHz Intel Core i7 processor and 16 GB of memory.

It is assumed that the simulation environment is a 1 km × 1 km square area which contains 10 EN and 60 UE. EN and UE are randomly deployed in this area. The CPU frequency(GHz) and storage size(GB) of each edge node obey normal distribution, which are $\tilde{N}_1(10, 2 \times 10^2)$, $\tilde{N}_1(10^2, 3 \times 10^2)$. We set virtual computing unit to be 0.2 Ghz and virtual storage unit to be 0.6 GB. The number of subtasks of each UE follows a uniform distribution $U(2, 7)$. The requirements of virtual computing resource unit and virtual storage unit follow poisson distribution. The mean values of poisson distribution are $\lambda_1 = 9, \lambda_2 = 11, \lambda_3 = 50$. The data transmission rate(KB/s) between two edge nodes follows the normal distribution $\tilde{N}_3 = (3 \times 10^3, 10^3)$.

Figure 3 shows the convergence effect of the four algorithms. The comparison algorithms are Biogeography-based Optimization for Task Allocation and Sending Route Selection(BBO-TASRO), Genetic Algorithm for Task Allocation and Sending Route Selection(GA-TASRO) and Particle Swarm Optimization for Task Allocation and Sending Route Selection(PSO-TASRO). The four algorithms begin to converge after 100 iterations, and finally converge to 108, 122, 134, and 151 ms respectively at 500 iterations. It can be seen that IBBO-TASRO has the fastest convergence speed and the lowest average task completion delay. IBBO-TASRO speeds up the search rate and avoids the algorithm from falling into a local optimal solution through the elite retention strategy and the improved migration rate.

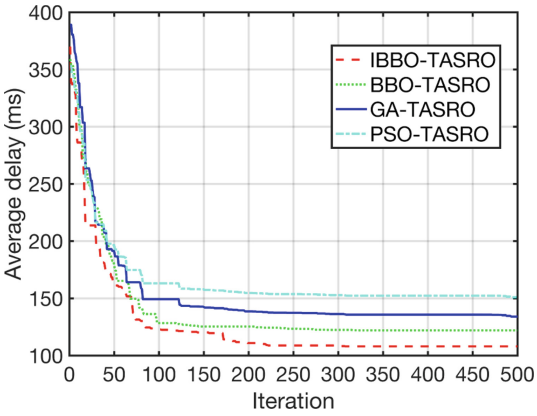


Fig. 3. Convergence performance of IBBO-TASRO and comparison with other algorithms

Figure 4 shows the average delay of the IBBO-TASRO under different UE numbers comparison with other algorithms. As the number of UE increases, the average delay of the four algorithms shows an upward trend. When the number of UE is 90, the average delays of the IBBO-TASRO, BBO-TASRO, PSO-TASRO,

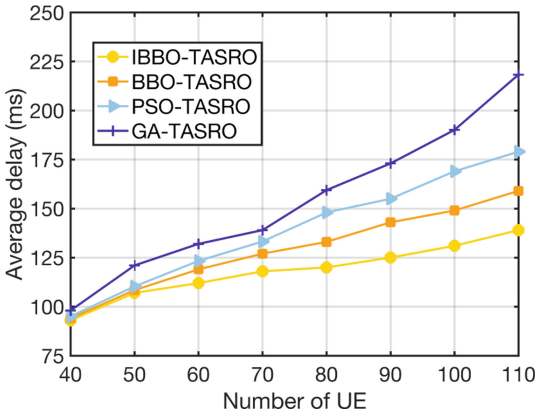


Fig. 4. Average delay of IBBO-TASRO with different number of UE

and GA-TASRO algorithm are 125, 146, 148, 173 respectively. Compared with BBO-TASRO, the delay of IBBO-TASRO is reduced by 1.4%. IBBO-TASRO uses the elite retention strategy in each algorithm iteration, avoiding the destruction of high-quality candidate solutions in the migration and mutation operation, and use the improved migration rate to enhance the stability of the algorithm convergence.

6 Conclusion

This paper proposes a cooperative computing mechanism based on edge routing optimization for power Internet of Things to minimize the service delay. First, a task allocation model based on multi-edge-node cooperation is establish, and the model is transformed into the problem of minimizing the average task completion delay. Then the IBBO-TASRO algorithm is proposed to solve the task allocation and route selection. Finally the experimental results show that compared with other algorithms, the proposed algorithm has lower delay and can meet the resource requirements of a large number of business terminals at the same time.

References

1. Ou, Q., Zhen, Y., Li, X, et al.: Application of internet of things in smart grid power transmission. In: 2012 third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, pp. 96–100. IEEE (2012)
2. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)
3. Shi, W., Cao, J., Zhang, Q., et al.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
4. Hayes, B.: Cloud computing (2008)

5. Gong, C., Lin, F., Gong, X., et al.: Intelligent cooperative edge computing in internet of things. IEEE Internet Things J. **7**(10), 9372–9382 (2020)
6. Tran, T.X., Hajisami, A., Pandey, P., et al.: Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges. IEEE Commun. Mag. **55**(4), 54–61 (2017)
7. Zhu, S., Gui, L., Chen, J., Zhang, Q., Zhang, N.: Cooperative computation offloading for UAVs: a joint radio and computing resource allocation approach.: In: 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, pp. 74–79 (2018). https://doi.org/10.1109/EDGE.2018.00017
8. Sun, P.J.: Privacy protection and data security in cloud computing: a survey, challenges, and solutions. IEEE Access **7**, 147420–147452 (2019)
9. Jararweh, Y, Doulat, A, AlQudah, O., et al.: The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In: 2016 23rd International Conference on Telecommunications (ICT), pp. 1–5. IEEE (2016)
10. Niu, X., Shao, S., Xin, C., et al.: Workload allocation mechanism for minimum service delay in edge computing-based power Internet of Things. IEEE Access **7**, 83771–83784 (2019)
11. Wang, Y., Tao, X., Zhang, X., et al.: Cooperative task offloading in three-tier mobile computing networks: an ADMM framework. IEEE Trans. Veh. Technol. **68**(3), 2763–2776 (2019)
12. Trinh, H., Calyam, P., Chemodanov, D., et al.: Energy-aware mobile edge computing and routing for low-latency visual data processing. IEEE Trans. Multimedia **20**(10), 2562–2577 (2018)
13. Zhang, D.G., Chen, L., Zhang, J., et al.: A multi-path routing protocol based on link lifetime and energy consumption prediction for mobile edge computing. IEEE Access **8**, 69058–69071 (2020)
14. Du, J., Zhao, L., Feng, J., et al.: Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. IEEE Trans. Commun. **66**(4), 1594–1608 (2017)
15. Chen, M., Hao, Y.: Task offloading for mobile edge computing in software defined ultra-dense network. IEEE J. Sel. Areas Commun. **36**(3), 587–597 (2018)