



# Ubiquitous Power Internet of Things-Oriented Low-Latency Edge Task Scheduling Optimization Strategy

Yu Liang<sup>1,2\*</sup> and Taoshen Li<sup>3,4\*</sup>

<sup>1</sup>School of Electrical Engineering, Guangxi University, Nanning, Guangxi, <sup>2</sup>School of Arts and Sciences, Guangxi Open University, Nanning, Guangxi, <sup>3</sup>School of Computer, Electronic and Information, Guangxi University, Nanning, Guangxi, <sup>4</sup>College of Information Engineering, Nanning University, Nanning, Guangxi

## OPEN ACCESS

### Edited by:

Dongliang Xiao,  
South China University of Technology,  
China

### Reviewed by:

Jinshuo Su,  
University of Macau/Taipa, China  
Shangshi Huang,  
Tsinghua University Beijing, China

### \*Correspondence:

Taoshen Li  
tshli@gxu.edu.cn  
Yu Liang  
1329102831@qq.com

### Specialty section:

This article was submitted to  
Smart Grids,  
a section of the journal  
Frontiers in Energy Research

**Received:** 18 May 2022

**Accepted:** 30 May 2022

**Published:** 22 June 2022

### Citation:

Liang Y and Li T (2022) Ubiquitous  
Power Internet of Things-Oriented  
Low-Latency Edge Task Scheduling  
Optimization Strategy.  
Front. Energy Res. 10:947298.  
doi: 10.3389/fenrg.2022.947298

Internet of things of cloud computing offers high-performance computing, storage and networking services, but there are still suffers from a high transmission and processing latency, poor scalability and other problems. Internet of things of edge computing can better meet the increasing requirements of electricity consumers for service quality, especially the increasingly stringent need for low delay. On the other hand, edge intelligent network technology can offers edge smart sensing while significantly improve the efficiency of task execution, but it will lead to a massive collaborative task scheduling optimization problem. In order to solve this problem, This paper studies an ubiquitous power internet of things (UPIoT) smart sensing network edge computing model and an improved multi node cluster cooperative scheduling optimization strategy. The cluster server is added to the edge aware computing network, and an improved low delay edge task collaborative scheduling algorithm (LLETCS) is designed by using the vertical cooperation and multi node cluster collaborative computing scheme between edge aware networks. Then the problem is transformed based on linear reconstruction technology, and a parallel optimization framework for solving the problem is proposed. The simulation results suggest that the proposed scheme can more effectively reduce the UPIoT edge computing latency, and improve the quality of service in UPIoT smart sensing networks.

**Keywords:** ubiquitous power internet of things, edge computing, smart sensing, low latency, linear reconstruction

## 1 INTRODUCTION

The ubiquitous power Internet of Things (UPIoT), which has an electrical system at its core, works with technologies such as smart terminal sensors, communication networks and artificial intelligence to link all parts of the power system (Wang and Wang, 2018). The UPIoT is a special IoT, and the power grid is a specific application object of IoT technology, which covers all aspects of generation, transmission, distribution, transformation and consumption for power or even energy networks; the connected objects range from electrical equipment to household appliances. The UPIoT integrates IoT technology with power systems to realize various types of information sensing devices and the sharing of communication information resources (Reka and Dragicevic, 2018). The UPIoTs can actualize holistic awareness, automatic control and smart decision-making of smart power grids by

deploying hundreds of millions of sensing devices and smart terminals, which is a key direction for current smart grid development.

Once massive power IoT nodes (e.g., power terminals, smart homes, other energy system equipment) are connected, the sensed data volume and data type of UPIoTs will increase sharply (Ge et al., 2016; Wang et al., 2018; Yang et al., 2019); the demand of new electricity users for service will grow continually; and the requirements for service quality, especially the need for low latency, will become increasingly stringent. The high processing delays resulting from the long distance from cloud computing data centres to end users bring about long transmission delays, which impairs the performance of latency-sensitive applications (Pan and McElhannon, 2018); this is a serious problem for smart grids.

In recent years, researchers have begun to study the combination of edge computing and power systems (Okay and Ozdemir, 2016; Shi et al., 2016; Li et al., 2018; Sun et al., 2019). Reference (Jiang et al., 2013) presented a cloud/edge collaboration architecture designed for advanced measurement systems, which stores and analyses power status data (e.g., voltage, current, phase angle) through a three-layer configuration (edge device-fog-cloud), in which fog-layer devices implement all or part of the processing features, thereby effectively enhancing the computational performance of analytical processing of power system data. References (Hagen and Kemper, 2010; Mori et al., 2017) studied the task assignment strategies for the edge computing of power systems. Reference (Hagen and Kemper, 2010) investigated the low-latency task assignment method for power systems; the tasks for centre and edge nodes were dynamically assigned based on the current computing power and task load to effectively improve the computational performance of the system. However, the aforementioned studies focused on edge-edge or edge-cloud resource sharing without considering the smart collaborative computing of multiple nodes in the system. The increasingly enhanced computing power of UPIoT sensing terminal devices can be used to perform complex computing tasks (Mori et al., 2017). However, terminal sensing devices are more resource-constrained than edge servers because they have limited computing resources and battery capacity (Chen et al., 2016). Individual sensing devices struggle to efficiently process large amounts of data; single edge computing nodes may fall short of the requirements for task instantaneity in certain scenarios. Furthermore, the user demand for resources is clearly heterogeneous; for example, some devices are underresourced, while others may be heavily overresourced. Fulfilling computing tasks through the collaboration of multiple nodes of smart sensing networks is a relatively promising approach. Moreover, the increase in the number of sensing devices and edge servers in power IoTs may bring about a massive collaborative task scheduling optimization problem. Therefore, designing a real-time effective, low-complexity task collaborative scheduling and resource optimization strategy for massive edge sensing tasks is absolutely essential.

Based on the analysis above, this paper studies an edge computing architecture for big data sensing by UPIoTs in

response to big data scenarios. In the architecture, edge computing is used for UPIoT smart sensing network edge nodes, including sensing devices and edge server clusters, to construct an “edge computing” layer; migrating the processing of UPIoT-sensed big data computing services to edge computing devices reduces the UPIoT service processing latency, relieves the computing load of cloud servers, and improves the overall network robustness. To further optimize the service processing latency of the above-noted network architecture, this paper studies the collaborative task scheduling optimization and resource allocation strategies of edge sensing nodes to minimize the average task time, thereby balancing the UPIoT smart sensing network edge load and improving the efficiency of task execution. The goal of our research is to solve the problem of limited network resources and insufficient computing power at the edge of ubiquitous power Internet of things, effectively reduce the task processing delay, and reduce the complexity of multi node task intelligent collaborative scheduling optimization.

## 2 RELATED WORK

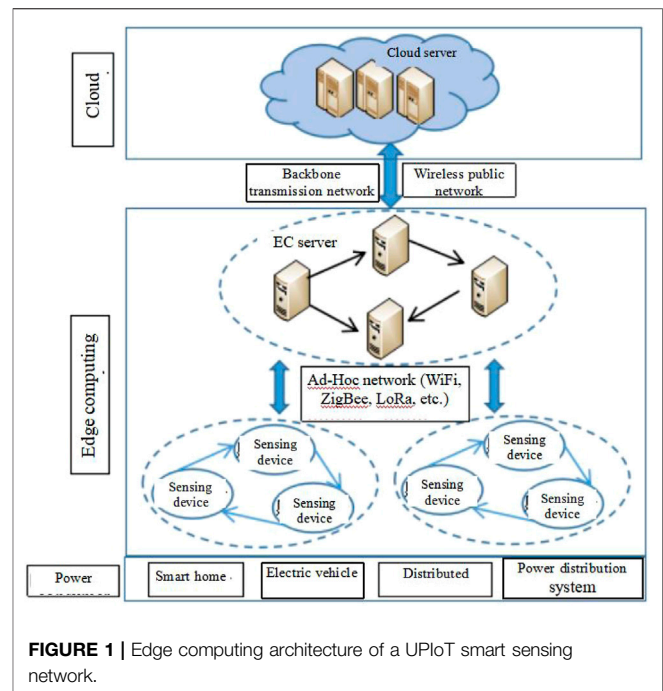
Task scheduling mechanisms are a key technology for computing systems; traditional algorithms applied to task scheduling in local computing include the heterogeneous earliest finish time (HEFT) algorithm, the critical path on a processor (CPOP) algorithm (Zhang and Wen, 2018), the STARS algorithm (Chen et al., 2017), the MOWS-DTM algorithm (Zhou et al., 2018), the artificial fish swarm (AFSA) algorithm (Zhang et al., 2017), the sensing cache task scheduling algorithm (Li et al., 2019), and the SA-DVSA algorithm (Hu et al., 2017a); the optimization objectives of these algorithms include reduction of the task scheduling time, latency optimization, resource conservation, scheduling efficiency improvement, and time and energy consumption constraint integration. However, these algorithms are not satisfactorily applicable to the collaborative task scheduling service between edge computing nodes because of instantaneity and other factors.

Many researchers have studied collaborative task scheduling for edge computing. Reference (Tran et al., 2017) summarized the collaboration between edge computing servers and mobile devices with respect to mobile edge coordination, collaborative cache processing and multilayer interference cancellation and proposed a real-time context-aware edge computing collaboration framework. Reference (Cao et al., 2019) proposed a method for constructing a collaborative service for mobile edge computing systems, constructed collaborative communication links through user nodes, collaboration nodes and access points connected to the edge server (AP) nodes, optimized the allocation of computing and communication resources of user and collaboration nodes, and minimized the total energy consumption under given computing latency constraints. Reference (Hu et al., 2018) proposed a new edge computing collaboration approach and a two-stage approach for the offloading realized by constructing relay cooperative communication links through APs and mobile devices, optimized offloading decisions and saved energy, and minimized the total energy emissions of APs. Considering the

collaborative compatibility between executors, Reference (Hu et al., 2017b) proposed a task assignment model based on collaborative compatibility and load balancing, improved the execution efficiency of the entire process instance and kept the loads balanced for executors using the task assignment method based on multiobjective joint optimization. Reference (Chai et al., 2007) proposed the design of a new cluster-architecture edge streaming media server and proposed the MCLBS cache replacement algorithm for adaptive optimization of edge server loads against the load balancing in cluster servers to reduce the service bandwidth consumption. In response to the unbalanced loads resulting from the skewed distribution of data and the instantaneity, dynamics and unpredictability of the data stream in distributed parallel processing architectures, Reference (Fang et al., 2017) proposed a lightweight balance adjustment method through Key granular migration and tuple granularity split to ensure a balanced system load.

References (Chai et al., 2007; Hu et al., 2017b; Fang et al., 2017; Tran et al., 2017; Hu et al., 2018; Cao et al., 2019) studied edge computing networks consisting of certain devices and edge nodes. Some studies have considered more realistic task scheduling problems in three-tier computing networks, which comprise terminal devices, edge computing servers and cloud servers. Given the finiteness of edge device resources, Xu (Xu et al., 2019) employed edge devices and cloud devices to reduce the waiting time for tasks and to minimize the latency by finding the best scheduling destinations. To address the host selection problem in task deployment of the joint edge computing centre and cloud data centre for an overall prolonged load balance, Dong et al. (Dong et al., 2019) proposed a deployment strategy based on an analysis of the heuristic task clustering method and the firefly swarm optimization algorithm; that is, joint load balancing was realized between the edge computing centre and the cloud computing centre by locating the best host for the task to be handled in the set. Huang et al. (Huang et al., 2020) used the software-defined network (SDN) technology to propose a service orchestration-based cloud mobile edge computing collaborative task offloading solution, which solves key problems in task offloading decisions and specific offload terminal selection; the multiaccess edge computing (MEC) server that minimizes the weighted sum of energy consumption and latency was chosen as the offload terminal based on the task latency and energy estimation for MEC servers, thereby achieving a balanced load and communication load optimization. The aforementioned studies suggested that three-tier task processing architectures offer more computing resources for multiuser edge computing offload systems, but three-tier processing architectures are confronted with more complex computing offload decisions and resource allocation problems. When load balancing is considered in three-tier computing networks, each task has more offload options and the resource management of multiple edge nodes affects other nodes, making it even more challenging to implement load balancing for three-tier computing networks.

Based on the aforementioned studies, this paper proposes a collaborative task scheduling and optimization assignment scheme in a multitier computing network, considering the



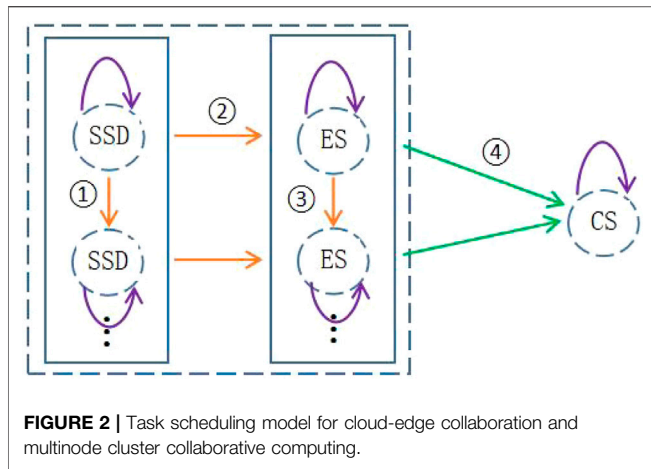
**FIGURE 1** | Edge computing architecture of a UPIoT smart sensing network.

characteristics of big data sensing by UPIoTs; the proposed scheme works with optimized scheduling decisions and computing resource allocation to minimize the total task time and improve the QoS of UPIoT edge networks. The research of this paper is of great significance to promote the development of edge computing technology of ubiquitous power Internet of things, and to guide the practical application.

### 3 SYSTEM ARCHITECTURE

Typical scenarios for the smart sensing of UPIoTs include sensing and diagnostics of electricity anomalies at low-voltage residential users, smart homes, and electrical appliances, escrow services for smart communities and smart buildings, optimization of electric vehicle (EV) charging and switching, and integrated metering for multienergy systems. Smart sensing terminals generate massive amounts of sensing data every day, featuring diversified and complex types, large volumes, a high heterogeneity and complex processing; as a result, traditional power systems struggle to store and process relevant data in a timely and efficient manner.

Most of the smart sensing terminal-sensed data are basic small data, which are of tremendous value. For instance, the vast amount of sensed data produced by smart terminals such as EVs, distributed photovoltaic (PV) cells, smart homes, and power distribution terminals on distribution and user access sides enable grid companies to gain a good understanding of user characteristics and provide these companies with novel means of technical support for peak load shaving, grid utilization improvement, energy conservation and consumption reduction, power stealing prevention, low-voltage O&M, and



system planning. However, the large volumes and types of edge sensed data and the limited computing and storage capacities of the edge network sensing devices that may be in use pose stringent requirements for service processing times, especially for latency-sensitive services.

This paper studied an edge computing architecture for UPIoT smart sensing networks based on the IoT edge computing reference model (Cisco Systems Inc, 2014), with consideration of the aforementioned characteristics of UPIoT-sensed big data. The architecture is divided into three layers from the top down: the traditional power big data cloud computing layer, the cluster edge server layer with certain computing, storage and communication capabilities in power communication systems, and the smart sensing terminal layer with edge computing capabilities, as shown in Figure 1. The cloud computing layer deals with highly complex non-real-time global data services; the edge server layer supports small real-time local data services, which makes it economical without incurring high device costs in terms of computing and storage. The smart sensing terminal layer consists of a variety of different smart sensing devices deployed where power is consumed, e.g., various sensors, smart concentrators, and robots, and principally collects given status parameters. This layer produces a large number of pending tasks and can judge, process or transfer tasks based on certain scheduling strategies.

Cluster servers and smart sensing terminals constitute the edge-aware computing network of this architecture. Due to the limited computing and storage resources for edge-aware computing network devices, individual computing devices cannot handle all big data tasks independently. Therefore, to fulfil different tasks in real time, the architecture adopts edge-cloud collaboration and the multinode cluster collaborative computing scheme between edge sensing networks. The multinode cluster collaborative processing strategy is intended to reduce the data processing latency through the collaboration of multiple nodes of IntelliSense networks, so as to improve the QoS and effectively handle latency-sensitive services. This scheme chooses different scheduling strategies depending on the various types of tasks produced by smart sensing devices (SSDs). Communication-intensive tasks can be fulfilled in real

time autonomously using the SSD resources. Tasks that cannot be fulfilled on their own are assigned to other SSDs with idle resources for resource sharing and rational scheduling between sensing devices. Computationally intensive tasks are dispatched to the edge server layer or remote cloud computing layer for execution. General types of tasks decide whether to execute autonomously or to schedule further depending on the load on the sensing device, which is dependent on the computing power of the device. The task scheduling decision in the edge server layer is dependent on the computing and memory resources of the server. The detailed collaborative task scheduling model is shown in Figure 2, where the orange solid line represents the multinode cluster collaborative scheduling between edge sensing networks. The green solid line represents edge-cloud collaborative scheduling, and the serial numbers represent the task decision scheduling sequence.

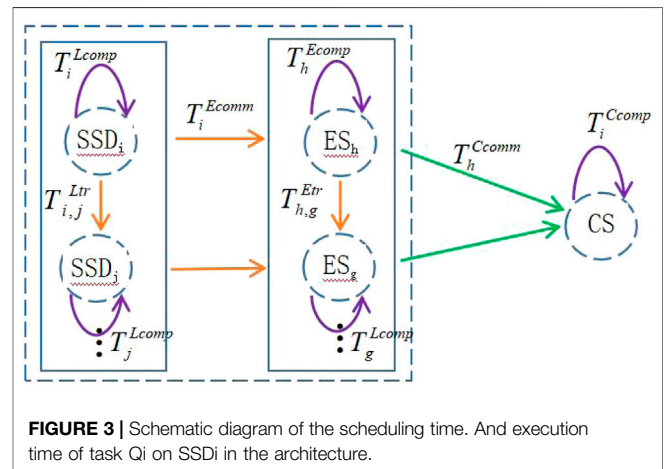
Multinode collaborative computing is a way to break the bottleneck of edge terminal computing; dynamically organized lightweight edge computing clusters can efficiently handle complex analytical tasks, with consideration of the service logic and event input of UPIoTs within the diffusion range with the computing target point as the *in situ* physical centre. Dynamically delineating the smart devices and servers involved in computing based on task granularity and complexity during task scheduling enables load balancing of the computing power, which contributes to more effective handling of latency-sensitive services.

## 4 SYSTEM MODELLING

In this section, the system latency problem is modelled, and then the LLETCS algorithm is described.

### 4.1 Problem Modelling

In UPIoT scenarios, edge computing-based smart sensing networks are composed of  $M$  edge servers (ES),  $N$  SSDs, and one cloud computing centre (CC). The edge server set and the smart sensing device set are denoted as  $M' = \{1, 2, \dots, M\}$  and  $N' = \{1, 2, \dots, N\}$ , respectively. Each SSD has a computing task that is



either processed locally or dispatched to another SSD or ES for processing. The ES may further dispatch it to the remote cloud computing centre for processing depending on the circumstances. The set tasks on SSD<sub>i</sub> are not separable and are expressed as Q<sub>i</sub>=(L<sub>i</sub>,K<sub>i</sub>), where L<sub>i</sub> represents the data size of task Q<sub>i</sub> (unit: Bit), and K<sub>i</sub> denotes the number of CPU cycles required to fulfil task Q<sub>i</sub> (unit: CPU cycle). Based on cloud-edge collaboration and the task scheduling model of multinode cluster collaborative computing, the schematic diagram of the scheduling time and execution time of task Q<sub>i</sub> on SSD<sub>i</sub> in the architecture is shown in **Figure 3**.

### 4.1.1 Edge Network Multinode Cluster Collaborative Computing Model

The computational latency of task Q<sub>i</sub> processing by SSD<sub>i</sub> on its own is

$$T_i^{Lcomp} = K_i / f_i^L, \text{ where } i \in N' \quad (1)$$

where  $f_i^L$  represents the computing power of SSD<sub>i</sub>. If SSD<sub>i</sub> has limited computing resources of its own ( $f_i^L < K_i$ ), the task can be scheduled to another SSD<sub>j</sub> with sufficient idle resources. It is assumed here that each SSD is aware of the available CPU resource information for other devices. Let  $T_{i,j}^{Ltr}$  denote the average round-trip time for task transfers between SSD<sub>i</sub> and SSD<sub>j</sub>. When  $i = j$ ,  $T_{i,j}^{Ltr} = 0$ ; where  $i, j \in j \in N'$

Therefore, the total latency of task Q<sub>i</sub> processing at the smart sensing device layer includes the transmission latency from SSD<sub>i</sub> to SSD<sub>j</sub> and the computing latency of SSD<sub>j</sub>, which can be expressed as

$$T_{Q_i}^L = T_{i,j}^{Ltr} + T_j^{Lcomp} \quad (2)$$

If all SSD resources cannot process task Q<sub>i</sub> independently, the task will be scheduled to ES<sub>h</sub>. Let the total computing power of ES<sub>h</sub> be expressed as F<sub>h</sub>; then  $f_{i,h}$  represents the available CPU resources allocated by ES<sub>h</sub> to task Q<sub>i</sub>. Therefore, the computational latency of task Q<sub>i</sub> being processed by ES<sub>h</sub> is

$$T_{i,h}^{Ecomp} = K_i / f_{i,h} \quad (3)$$

When the task is scheduled to ES, SSD<sub>i</sub> first sends the task Q<sub>i</sub> to the associated server ES<sub>m<sub>i</sub></sub> for execution; the task can also be scheduled to other ESs with sufficient idle resources if the conditions are not met. Here,  $m_i \in M'$  represents the edge server associated with SSD<sub>i</sub>. At this point, the total latency of scheduling task Q<sub>i</sub> to the edge server includes the transmission latency from SSD<sub>i</sub> to ES<sub>m<sub>i</sub></sub>, the transmission latency from ES<sub>m<sub>i</sub></sub> to ES<sub>g</sub>, and the computing latency of ES<sub>g</sub>, and can be expressed as

$$T_{Q_i}^E = T_{i,m_i}^{Ecomm} + T_{m_i,g}^{Etr} + T_{i,g}^{Ecomp} \quad (4)$$

When SSD<sub>i</sub> is associated with server ES<sub>m<sub>i</sub></sub>, the uplink transmission rate on its wireless link can be calculated as

$$R_{i,m_i} = \frac{\vec{B}_{m_i}}{U_{m_i}} \eta_i \quad (5)$$

where the bandwidth resource of ES<sub>h</sub> is  $\vec{B}_h$  Hz, and  $\eta_i$  represents the uplink spectral effect between SSD<sub>i</sub> and the associated ES<sub>m<sub>i</sub></sub>.  $U_{m_i} = \sum_{k \in N'_{m_i}} (1 - A_k)$  represents the number of SSDs associated with ES<sub>m<sub>i</sub></sub>; and  $N_j = \{i: i \in N', m_i = h\}$  denotes the SSD set associated with ES<sub>h</sub>. Then, the transmission latency of scheduling task Q<sub>i</sub> to ES<sub>m<sub>i</sub></sub> is

$$T_{i,m_i}^{Ecomm} = \frac{L_i}{\vec{R}_{i,m_i}} = \frac{L_i}{\vec{B}_{m_i} \eta_i} \sum_{k \in N'_{m_i}} (1 - A_k) \quad (6)$$

### 4.1.2 Cloud Collaborative Computing Model

When the edge computing network cannot handle task Q<sub>i</sub>, the task is scheduled to the cloud computing centre. The average round-trip time of transmission between the edge server and the cloud computing centre is denoted as  $T^{Ccomm}$ . Let  $f_i^c$  represent the task-performable cloud computing power, and  $f_i^c > f_{i,h}^E$ . Then, the cloud computing latency of task Q<sub>i</sub> is  $T_i^{Ccomp} = K_i / f_i^c$ .

Then, the total latency of scheduling task Q<sub>i</sub> to the remote cloud computing centre includes the transmission latency from SSD<sub>i</sub> to the edge server, the transmission latency from the edge node to the cloud computing centre, and the computing latency of the cloud computing centre, and can be expressed as

$$T_{Q_i}^C = T_{i,h}^{Ecomm} + T_h^{Ccomm} + T_i^{Ccomp} \quad (7)$$

where  $i \in N'$ , and  $h \in M'$

Multiple tasks are generated at a time in this system architecture; it is hypothesized that an SSD and an ES can each handle only one task at a time. Binary variables A<sub>i</sub>, B<sub>i,h</sub> and C<sub>i</sub> are defined, where A<sub>i</sub>, B<sub>i,h</sub>, C<sub>i</sub> ∈ {0,1}, which indicate whether task Q<sub>i</sub> is processed in an SSD, an ES or a cloud computing centre, respectively. For the final successful execution of task Q<sub>i</sub>, the following should be satisfied

$$A_i + \sum_{h=1}^M B_{i,h} + C_i = 1, \text{ where } i \in N', h \in M' \quad (8)$$

According to **Eqs 1)–(8)**, the total processing latency of task Q<sub>i</sub> in the edge computing architecture of the UPIoT smart sensing network can be expressed as

$$T_{Q_i} = A_i T_{Q_i}^L + \sum_{h=1}^M B_{i,h} T_{Q_i}^E + C_i T_{Q_i}^C, \text{ Where } i \in N', h \in M' \quad (9)$$

### 4.1.3 Combinatorial Optimization-Based Problem Modelling

This study modelled the problem based on the combinatorial optimization of task scheduling and resource allocation to minimize the total latency for all tasks. First, the scheduling decision vector of SSD<sub>i</sub> is denoted as D<sub>i</sub> = {A<sub>i</sub>, B<sub>i,1</sub>, B<sub>i,2</sub>, ..., B<sub>i,M</sub>, C<sub>i</sub>}, and the scheduling decision of all SSDs is expressed as D = {D<sub>i</sub>, i ∈ N'}. The computing resource allocation vector of all SSDs is expressed as F = {f<sub>i</sub>, j, i ∈ N', j ∈ N'}. Then, the joint optimization objective function can be expressed as

$$P: \min_{D,F,F'} \sum_{i=1}^N T_{Q_i} \quad (10)$$

$$s.t. \sum_{i=1}^N f_{i,h} \leq F_h, h \in M' \quad (11)$$

$$A_i + \sum_{h=1}^M B_{i,h} + C_i = 1, i \in N' \quad (12)$$

$$A_i, B_{i,h}, C_i \in \{0, 1\}, i \in N', h \in M' \quad (13)$$

Constraint (11) states that the amount of computing resources provided by the edge computing network cannot exceed the total computing load of all the devices on each layer; constraint (12) restricts each sensing task to be processed at only one computing node.

Based on the above, the total processing latency of SSD<sub>i</sub> for task Q<sub>i</sub> in the edge computing architecture of the UPIoT smart sensing network can be expressed as

$$\begin{aligned} T_i &= A_i + \sum_{h=1}^M B_{i,h} T_{Q_i}^E + C_i T_{Q_i}^C \\ &= A_i T_{i,j}^{Ltr} + A_i T_j^{Lcomp} + \frac{L_i(1-A_i)}{B_{m_i} \eta_i} \sum_{k \in N'_{m_i}} (1-A_k) + \sum_{h=1}^M B_{i,h} \frac{K_i}{f_{i,h}} \\ &\quad + \sum_{h=1}^M B_{i,h} T_{m_i,h}^{Etr} + C_i (T_h^{Ccomm} + T_i^{Ccomp}) \end{aligned} \quad (14)$$

There are discrete variables and continuous variables, as well as nonlinear objective functions and constraints in the equation above; hence, optimization problem P is a mixed integer nonlinear optimization problem.

## 4.2 Low-Latency Edge Tasks Collaborative Scheduling (LLETCS) Algorithm

### 4.2.1 Linear Reconstruction Technology-Based Problem Transformation

To handle the product term in (14), an additional variable  $u_{i,h}$  is set up; then, let  $u_{i,h} = \frac{1}{f_{i,h}}$ . Two auxiliary variables, i.e.,  $G_i$  and  $V_{i,h}$ , are defined and respectively denoted as

$$G_i = (1 - A_i) \sum_{k \in N'_{m_i}} (1 - A_k), i \in N', \quad (15)$$

$$V_{i,h} = B_{i,h} u_{i,h}, i \in N', h \in M', \quad (16)$$

After new variables  $u_{i,h}$ ,  $G_i$  and  $V_{i,h}$  are introduced, objective function (10) is re-expressed as

$$\begin{aligned} \psi(D, G, V) &= \sum_{i=1}^N A_i T_{i,h}^{Ltr} + \sum_{i=1}^N A_i T_j^{Lcomp} + \sum_{i=1}^N \frac{L_i G_i}{B_{m_i} \eta_i} + \sum_{i=1}^N \sum_{h=1}^M K_i V_{i,h} + \sum_{i=1}^N \sum_{h=1}^M B_{i,h} T_{m_i,h}^{tr} \\ &\quad + \sum_{i=1}^N C_i (T_h^{Ccomm} + T_i^{Ccomp}) \end{aligned} \quad (17)$$

The computing resource constraints are translated into

$$\sum_{i=1}^N \frac{1}{u_{i,h}} \leq F_h, h \in M' \quad (18)$$

Problem P is translated into

$$P1: \min_{D,u,G,V} \psi(D, G, V) \quad (19)$$

s.t. (12), (13), (15), (16), (18)

Let Eqs 15, 16 be respectively equivalent to the following constraints, based on the linear reconstruction technology (Hou et al., 2009):

$$\left\{ \begin{aligned} G_i &\geq 0, & i \in N'; \\ G_i &\leq N_{m_i} (1 - A_i), & i \in N'; \\ G_i &\leq \sum_{k \in N'_{m_i}} (1 - A_k), & i \in N'; \\ G_i &\geq \sum_{k \in N'_{m_i}} (1 - A_k) + N_{m_i} (1 - A_i) - N_{m_i}, & i \in N'. \end{aligned} \right. \quad (20)$$

$$\left\{ \begin{aligned} V_{i,h} &\geq \frac{1}{F_h} B_{i,h}, & i \in N', h \in M'; \\ V_{i,h} &\leq \frac{1}{F_{\min}} B_{i,h}, & i \in N', h \in M'; \\ V_{i,h} &\leq u_{i,h} + \frac{1}{F_h} B_{i,h} - \frac{1}{F_h}, & i \in N', h \in M'; \\ V_{i,h} &\geq u_{i,h} + \frac{1}{F_{\min}} B_{i,h} - \frac{1}{F_{\min}}, & i \in N', h \in M'. \end{aligned} \right. \quad (21)$$

where  $F_{\min}$  is a normal value tending towards 0. Therefore, problem P1 has the following equivalent representation

$$P2: \min_{D,u,G,V} \psi(D, G, V) \quad (22)$$

s.t. (12), (13), (18), (20), (21)

When all binary variables are fixed, P2 is a convex problem.

### 4.2.2 ADMM-Based Parallel Optimization Algorithm

The ADMM-based parallel optimization algorithm is achieved through the sequential iteration of global variables, local variables, and antithetic variables. First, the global variables are optimized by the minimized augmented Lagrangian function. Then, the local variables are updated; and finally, the antithetic variables are updated.

Step I: Build the augmented Lagrangian function for the target problem.

A coupling relationship exists between optimization variables  $A_i$ ,  $B_{i,h}$  and  $C_i$  in constraint (12); hence, to break down problem P<sub>2</sub>, variables  $A_i$ ,  $B_{i,h}$  and  $C_i$  are first copied as variables  $A'_i$ ,  $B'_{i,h}$  and  $Z'_i$  so that

$$\left\{ \begin{aligned} A'_i &= A_i, & i \in N'; \\ B'_{i,h} &= B_{i,h}, & i \in N', h \in M'; \\ C'_i &= C_i, & i \in N'. \end{aligned} \right. \quad (23)$$

Then, coupling constraint (12) is equivalent to the following conditions

$$A'_i + \sum_{h=1}^M B'_{i,h} + C'_i = 1, \quad i \in N' \quad (24)$$

$D' = \{A'_i, B'_{i,1}, B'_{i,2}, \dots, B'_{i,M}, C'_i, i \in N'\}$  represents the copied variable vectors of all smart sensing terminals. Then, problem P2 is equivalent to

$$P3: \min_{D,u,G,V,D'} \psi(D, G, V)$$

s.t. (13), (18), (20), (21), (23), (24).

Therefore, the augmented Lagrangian function of P3 can be expressed as (Boyd et al., 2010)

$$\begin{aligned} \ell_\rho(D, G, V, D', \lambda, \mu, \gamma) = & \psi(D, G, V) + \sum_{i=1}^N \lambda_i (A_i - A'_i) \\ & + \sum_{i=1}^N \sum_{h=1}^M \mu_{i,h} (B_{i,h} - B'_{i,h}) + \sum_{i=1}^N \gamma_i (C_i - C'_i) + \frac{\rho}{2} \sum_{i=1}^N (A_i - A'_i)^2 \\ & + \frac{\rho}{2} \sum_{i=1}^N \sum_{h=1}^M (B_{i,h} - B'_{i,h})^2 + \frac{\rho}{2} \sum_{i=1}^N (C_i - C'_i)^2 \end{aligned} \quad (25)$$

where  $\lambda = \{\lambda_i\}_{i \in N'}$ ,  $\mu = \{\mu_{i,h}\}_{i \in N', h \in M'}$  and  $\gamma = \{\gamma_i\}_{i \in N'}$  are Lagrangian multipliers of the three constraints, respectively.  $\rho > 0$  is an augmented Lagrangian parameter.

Step II: Optimization of global variables.

First, the augmented Lagrangian function is minimized; let  $\bar{G}_h = \{G_i, i \in N'\}$ ,  $\bar{V}_h = \{V_{i,h}, i \in N'\}$ , and  $\bar{u}_h = \{\mu_{i,h}, i \in N'\}$ . Then

$$P4: \min_{\{D,u,\bar{G},\bar{V}\}} \ell_\rho(D, \bar{G}, \bar{V}, D'^{[t]}, \lambda^{[t]}, \mu^{[t]}, \gamma^{[t]}) \quad (26)$$

s.t. (13), (18), (20), (21), (23), (24). where the superscript [t] represents the iteration number.

Then, let  $\bar{A}_h = \{A_i, i \in N'\}$  denote the local scheduling decision vector for all SSDs associated with ES<sub>h</sub>;  $\bar{B}_h = \{B_{i,h}, i \in N'\}$  represents the decision vector that schedules tasks to ES<sub>h</sub>;  $\bar{C} = \{C_i, i \in N'\}$  stands for the decision vector that schedules tasks to the cloud server. P4 is equivalent to

$$\begin{aligned} \ell_\rho(D, \bar{G}, \bar{V}, D'^{[t]}, \lambda^{[t]}, \mu^{[t]}, \gamma^{[t]}) = & \sum_{h=1}^M f_h(\bar{A}_h, \bar{G}_h) + \sum_{h=1}^M g_h(\bar{B}_h, \bar{V}_h) \\ & + w(\bar{C}) \end{aligned} \quad (27)$$

At the (t+1)-th iteration, global variable  $\{D, u, \bar{G}, \bar{V}\}$  can be updated by solving optimization problem P4 (Nguyen et al., 2017; Prak and Boyd, 2017).

The optimization problem is broken down and the subproblems are solved separately below to improve the computing speed through the parallel processing of such subproblems. Eq. 27 clearly show that problem P4 can be equivalently broken down into three subproblems  $P_1, P_2, P_3$ , which are denoted respectively as

$$P_1: \min_{\bar{A}_h, \bar{G}_h} f_h(\bar{A}_h, \bar{G}_h) \quad (28)$$

S.t. (20),  $A_i \in \{0, 1\}, i \in N'$

$$p_2: \min_{\bar{B}_h, \bar{V}_h, \bar{u}_h} g_h(\bar{B}_h, \bar{V}_h) \quad (29)$$

S.t. (18), (21),  $B_{i,h} \in \{0, 1\}, i \in N'$

$$p_3: \min_{\bar{C}} w(\bar{C}) \quad (30)$$

S.t.  $C_i \in \{0, 1\}, i \in N'$

For  $P_1$ , the binary constraints are equivalently expressed as follows:

$$\sum_{i \in N'} (A_i - A_i^2) \leq 0, \quad (31)$$

$$0 \leq A_i \leq 1, \quad i \in N'_h \quad (32)$$

Because (31) is a nonconvex constraint,  $P_1$  can be converted into

$$P'_1: \min_{\bar{A}_h, \bar{G}_h} f_h(\bar{A}_h, \bar{G}_h) + \zeta \sum_{i \in N'_h} (A_i - A_i^2) \quad (33)$$

s.t. (20), (35). where  $\zeta > 0$  is a penalty factor; according to Reference (Che et al., 2014), when the  $\zeta$  component is large, problem  $P_1$  is equivalent to problem  $P'_1$ . Let  $\psi_h(\bar{A}_h, \bar{G}_h) = f_h(\bar{A}_h, \bar{G}_h) + \zeta \sum_{i \in N'_h} A_h$  and  $\varphi_h(\bar{A}_h) = \zeta \sum_{i \in N'_h} A_h^2$ ; then,  $P_1$  can be written as the difference between the two convex functions  $\psi_h(\bar{A}_h, \bar{G}_h)$  and  $\varphi_h(\bar{A}_h)$ . Therefore,  $P'_1$  is a convex difference optimization problem; the locally optimal solution can be obtained by the sequential convex approximation method (Horst and Thoai, 1999).

Subproblem  $P_2$  is a linear function, but the binary constraint is also convex, and the same method for solving subproblem  $P_1$  is used. Subproblem  $P_3$  is an unconstrained binary optimization problem; with the candidate solution comparison method, the solution to  $P_3$  can be expressed as

$$C_i = \begin{cases} 0 & \frac{\rho}{2}(1 - 2C_i^{[t]}) + T_h^{C_{comm}} + T_i^{C_{comp}} + \gamma_i^{[t]} \geq 0 \\ 1 & \frac{\rho}{2}(1 - 2C_i^{[t]}) + T_h^{C_{comm}} + T_i^{C_{comp}} + \gamma_i^{[t]} < 0 \end{cases} \quad (34)$$

Step III: Update of local variable.

Value  $\{D, \bar{G}, \bar{V}\}^{[t+1]}$  of the global variable is achieved at the (t+1)-th iteration, and problem P4 can be expressed as

$$p5: \min_{D'} \ell_\rho(D^{[t+1]}, \bar{G}^{[t+1]}, \bar{V}^{[t+1]}, D', \lambda^{[t]}, \mu^{[t]}, \gamma^{[t]}) \quad (35)$$

S.t. (24),  $A'_i, B'_{i,h}, C'_i \in \{0, 1\}, i \in N', h \in M'$ ,

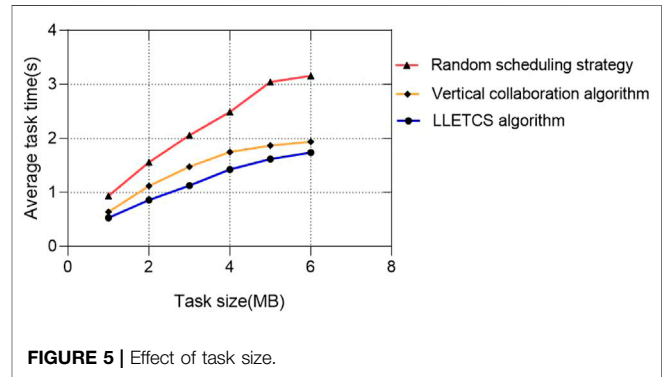
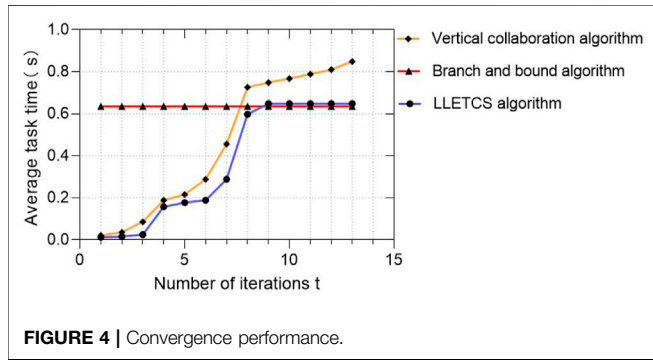
Local variable  $D'$  is updated by solving optimization problem P5. First, the constant term is removed from the function, and P5 is equivalent to

$$\min_{D'} \sum_{i=1}^N W'_i(D'_i) \quad (36)$$

S.t. (24),  $A'_i, B'_{i,h}, C'_i \in \{0, 1\}, i \in N', h \in M'$ , where the function  $W'_i(D'_i)$  is denoted as

**TABLE 1** | Description of simulation experiment parameters.

The Bandwidth	Transmission Power of Terminal	Noise Power Spectral Density	The Computing Power of Edge Server $f_t$	The Computing Power of Edge Smart Sensing Terminal	The Input Data Size per Task	The Number of CPU Cycles Required to Fulfil Task $Q_i$ (K <sub>i</sub> )	Average Backhaul Link Delay $T^{Comm}$ (s)
15 MHz	26 dBm	-174 (dBm/Hz)	20 Gigacycles/s	0.6 Gigacycles/s	N (6000, 1000)	[2,5] gigacycles	4



$$W'_i(D'_i) = \left[ \frac{\rho}{2} A_i'^2 - (\rho A_i^{[t+1]} + \lambda_i^{[t]}) A_i' \right] +$$

$$\sum_{h=1}^h \left[ \frac{\rho}{2} B_{i,h}^2 - (\rho B_{i,h}^{[t+1]} + \mu_i^{[t]}) B_{i,h}' \right] + \left[ \frac{\rho}{2} C_i'^2 - (\rho C_i^{[t+1]} + \gamma_i^{[t]}) C_i' \right] \quad (37)$$

According to Eq. 24, the optimum solution to P5 is as follows:

$$\begin{cases} A_i' = 1, B_{i,h \in M'} = 0, C_i' = 0 & \text{if } c_{i,0} = \vec{C}_{i,\min}, i \in N' \\ A_i' = 0, B_{i,h \neq k} = 1, B_{i,k} = 0, C_i' = 0 & \text{if } c_{i,h} = \vec{C}_{i,\min}, i \in N' \\ A_i' = 0, B_{i,h \in M'} = 0, C_i' = 1 & \text{if } c_{i,M+1} = \vec{C}_{i,\min}, i \in N' \end{cases} \quad (38)$$

where  $\vec{C}_{i,\min} = \min\{c_{i,0}, c_{i,h \in N'}, c_{i,M+1}\}$  represents the minimum value of all feasible solutions to SSD<sub>i</sub>, while  $c_{i,0} = \frac{\rho}{2} - \rho A_i^{[t+1]} - \lambda_i^{[t]}$ ,  $c_{i,h} = \frac{\rho}{2} - \rho B_{i,h}^{[t+1]} - \mu_{i,h}^{[t]}$ ,  $c_{i,M+1} = \frac{\rho}{2} - \rho C_i^{[t+1]} - \gamma_i^{[t]}$ .

Step III: Update of antithetic variable.

The antithetic variable is updated as follows at the (t+1)-th iteration

$$\lambda_i^{[t+1]} = \lambda_i^{[t]} + \rho(A_i^{[t+1]} - A_i'^{[t+1]}), \quad i \in N' \quad (39)$$

$$\mu_{i,h}^{[t+1]} = \mu_{i,h}^{[t]} + \rho(B_{i,h}^{[t+1]} - B_{i,h}'^{[t+1]}), \quad i \in N', h \in M' \quad (40)$$

$$\gamma_i^{[t+1]} = \gamma_i^{[t]} + \rho(C_i^{[t+1]} - C_i'^{[t+1]}), \quad i \in N' \quad (41)$$

## 5 ANALYSIS OF SIMULATION RESULTS

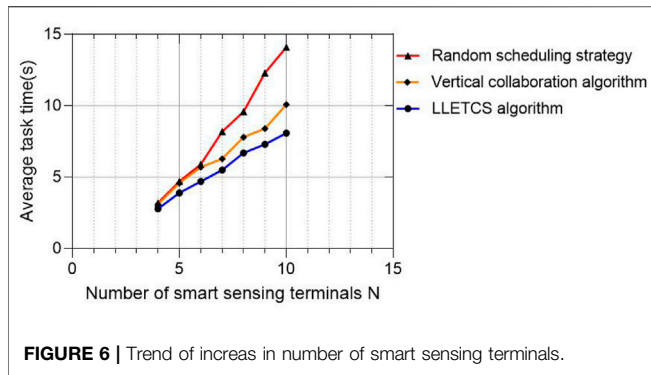
This section simulates the proposed algorithm for verification. The experiment used the MATLAB platform to simulate this process in an IoT scenario, and the results were compared with the performance of a “vertical collaboration scheme only” and a

“random scheduling strategy” under different task computing loads and smart sensing terminal computing powers. “vertical collaboration scheme only” means that the tasks of the terminal can be processed by the terminal, the associated base station or the ECS in this scheme. However, the horizontal cooperation between the terminal and the base station is not considered, and the task cannot be forwarded to other adjacent terminals or base stations. The “random scheduling strategy” means that the tasks of the terminal can only be unloaded to the associated base station for processing.

An IoT edge computing network, consisting of 10 smart sensing terminals and four edge servers, was considered. The computing power of each edge server was 20 Gigacycles/s, and the computing power of each edge smart sensing terminal was 0.6 Gigacycles/s (Chen et al., 2018). The input data size per task was subject to the Gaussian distribution of  $d_i \sim N(6000, 1000)$ . The number of CPU cycles required for task calculation follows the uniform distribution of [2,5] gigacycles. See Table 1 for our simulation experiment parameters.

Figure 4 shows the convergence performance of the proposed algorithm. Here we compare the performance of the proposed scheme with that of the branch and bound method and “vertical cooperation only scheme”. The branch and bound method is a classical method to solve discrete combinatorial optimization problems. For a given small parameter value  $\epsilon \geq 0$ , able to achieve one- $\epsilon$  Optimal performance. We set  $\epsilon = 0.001$ , therefore, a solution regarded as an approximate optimal value can be obtained from the beginning based on the branch and bound method. The ADMM optimization framework adopted in our proposed scheme makes the algorithm converge rapidly when the number of iterations  $t = 10$ , and the performance obtained is close to that of the branch and bound method. Therefore, an approximate optimal solution can also be obtained.



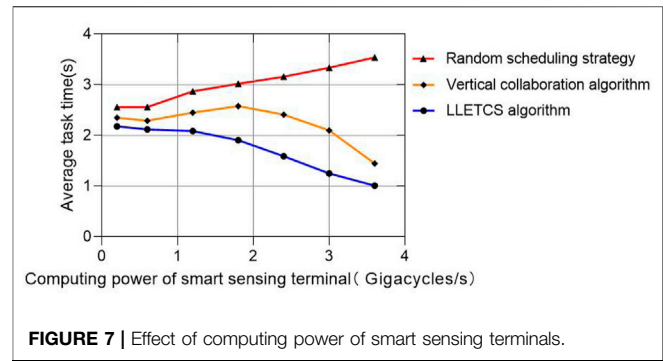


**Figure 5** shows the effect of task size on latency. It can be seen that for the three schemes, the greater the amount of calculation, the greater the average task duration. However, compared with the other two schemes, the proposed scheme has a smaller average task duration under the same task size.

On the other hand, it can be seen that when the amount of computation is small, the performance of the proposed scheme and the vertical collaboration only scheme increase significantly. Because the task processing time is less than the long-distance transmission time to the ECS, the task is more likely to be processed by the edge server, which increases the load on the edge. When the amount of computing increases, the computing time becomes the main component of the task duration. In this case, more tasks will be processed by the ECS, thus reducing the load on the edge and the task processing duration. Therefore, the task scheduling strategy adopted in the proposed scheme makes the system bring better performance gains when the number of tasks processed is moderate.

**Figure 6** illustrates the growth trend of the average runtime of the proposed scheme as the number of smart sensing terminals increased. It can be seen that with the increase of the number of terminals, the average running time of the random scheduling strategy increases the most, while the running time of our scheme increases slowly. This is because although the increase in the number of terminals may be the increase in the number of tasks, the edge network multi node cluster cooperative processing strategy adopted in our scheme can reduce the processing delay of the system to a certain extent, thus weakening the growth of the average running time.

**Figure 7** shows the effect of the smart sensing terminal computing power on the average task duration. We note that The performance difference between the proposed scheme and the vertical collaboration scheme decreased as the computing power of the smart sensing terminal increased. This is because when the computing power of smart sensing terminals increases, the multi node collaborative task scheduling model under the proposed architecture makes the tasks tend to be handled by the terminals themselves, which reduces the need for scheduling to edge servers and the cloud, thereby maximizing the utilization of edge resources.



The numerical results suggest that the proposed architecture can bring significant gains to the cooperative processing of edge networks when the terminal computing power is small, the transmission time is long, and the number of tasks is moderate. This is because in these cases, upiot aware big data computing services tend to process in the edge computing network, reducing the processing delay of upiot services. At the same time, the low delay edge task collaborative optimization scheduling algorithm proposed by us can better balance the edge load of upiot intelligent sensing network, maximize the utilization of edge resources, improve task execution efficiency, and better improve the quality of service (QoS) in upiot intelligent sensing network.

## 6 SUMMARY

The existing power resource management research focuses on the resource sharing between the edge or the edge cloud, does not consider the intelligent collaborative computing of multiple nodes in the system, and the two-tier processing architecture is in the majority. A new scheduling strategy and resource allocation scheme for the edge computing problem in multinode cluster collaboration in UPIoTs were proposed in this article. The scheme adopts a scheduling scheme combining edge cloud cooperation and multi node cluster cooperation between edge aware networks to obtain higher edge resource utilization and effectively reduce task processing delay. To minimize the average task time of UPIoT smart sensing networks, a sensing collaborative task scheduling model was proposed to model the task processing latency of the multilayer computing structure in the architecture. Then, a low-latency edge task collaborative scheduling algorithm was proposed for linear reconstruction technology-based problem transformation, and an ADMM-based parallel optimization framework for solving this problem was presented. The simulation results suggest that the proposed optimization strategy performed well in reducing the UPIoT edge computing latency and maximizing the edge resource utilization, thereby effectively ameliorating the QoS in the UPIoT smart sensing network.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary materials, further inquiries can be directed to the corresponding author.

## REFERENCES

- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *FNT Mach. Learn.* 3, 1–122. doi:10.1561/2200000016
- Cao, X., Wang, F., Xu, J., Zhang, R., and Cui, S. (2019). Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing. *IEEE Internet Things J.* 6, 4188–4200. doi:10.1109/jiot.2018.2875246
- Chai, Y., Gu, L., and Li, S. L. (2007). Cluster-based Edge Streaming Server with Adaptive Load Balance in Mobile Grid. *J. Comput. Res. Dev.* 44, 2136–2142. doi:10.1360/crad20071221
- Che, E., Tuan, H. D., and Nguyen, H. H. (2014). Joint Optimization of Cooperative Beamforming and Relay Assignment in Multi-User Wireless Relay Networks. *IEEE Trans. Wirel. Commun.* 13, 5481–5495. doi:10.1109/twc.2014.2324588
- Chen, H. K., Zhu, J. H., Zhu, X. M., Ma, M., and Zhang, Z. (2017). Resource-delay-aware Scheduling for Real-Time Tasks in Clouds. *J. Comput. Res. Dev.* 54, 446–456. doi:10.7544/issn1000-1239.2017.20151123
- Chen, M.-H., Dong, M., and Liang, B. (2018). Resource Sharing of a Computing Access Point for Multi-User Mobile Cloud Offloading with Delay Constraints. *IEEE Trans. Mob. Comput.* 17, 2868–2881. doi:10.1109/tmc.2018.2815533
- Chen, X., Jiao, L., Li, W., and Fu, X. (2016). Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw.* 24, 2795–2808. doi:10.1109/tnet.2015.2487344
- Cisco Systems Inc (2014). The Internet of Things Reference Model. in *2014 Internet of Things World Forum* (Chicago, IL: Cisco Systems Inc), 1–12.
- Dong, Y., Xu, G., Ding, Y., Meng, X., and Zhao, J. (2019). A 'Joint-Me' Task Deployment Strategy for Load Balancing in Edge Computing. *IEEE Access* 7, 99658–99669. doi:10.1109/access.2019.2928582
- Fang, J. H., Wang, X. T., Zhang, R., and Zhou, A. (2017). High-performance Data Distribution Algorithm on Distributed Stream Systems. *J. Softw.* 28, 563–578. doi:10.13328/j.cnki.jos.005168
- Ge, L., Wang, S., and Qu, H. (2016). Design of Storage Framework for Big Data of SPDU. *Electr. Power Autom. Equip.* 36, 194–202.
- Hagen, S., and Kemper, A. (2010). "Model-based Planning for State-Related Changes to Infrastructure and Software as a Service Instances in Large Data Centers," in 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), Miami, FL, USA, 05–10 July 2010 (IEEE), 11–18. doi:10.1109/cloud.2010.14
- Horst, R., and Thoai, N. V. (1999). DC Programming: Overview. *J. Optim. Theory Appl.* 103, 1–43. doi:10.1023/a:1021765131316
- Hou, Y. T., Shi, Y., and Sherali, H. D. (2009). *Applied Optimization Methods for Wireless Networks*. Cambridge, MA: Cambridge University Press.
- Hu, H. Y., Ji, C. P., and Hu, H. (2017). Method for Optimizing Task Allocation in Workflow System Based on Cooperative Compatibility. *J. Comput. Res. Dev.* 54, 872–885. doi:10.7544/issn1000-1239.2017.20151174
- Hu, H. Y., Liu, R. H., and Hu, H. (2017). Multi-objective Optimization for Task Scheduling in Mobile Cloud Computing. *J. Comput. Res. Dev.* 54, 1909–1919.
- Hu, X., Wong, K.-K., and Yang, K. (2018). Wireless Powered Cooperation-Assisted Mobile Edge Computing. *IEEE Trans. Wirel. Commun.* 17, 2375–2388. doi:10.1109/twc.2018.2794345
- Huang, M., Liu, W., Wang, T., Liu, A., and Zhang, S. (2020). A Cloud-MEC Collaborative Task Offloading Scheme with Service Orchestration. *IEEE Internet Things J.* 7, 5792–5805. doi:10.1109/jiot.2019.2952767
- Jiang, Z., Chan, D. S., Prabhu, M. S., Natarajan, P., Hao, H., and Bonomi, F. (2013). "Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture," in 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, San Francisco, CA, USA, 25–28 March 2013 (IEEE), 25–28. doi:10.1109/sose.2013.73
- Li, C., Tang, J., Tang, H., and Luo, Y. (2019). Collaborative Cache Allocation and Task Scheduling for Data-Intensive Applications in Edge Computing Environment. *Future Gener. Comput. Syst.* 95, 249–264. doi:10.1016/j.future.2019.01.007
- Li, W., Yang, T., Delicato, F. C., Pires, P. F., Tari, Z., Khan, S. U., et al. (2018). On Enabling Sustainable Edge Computing with Renewable Energy Resources. *IEEE Commun. Mag.* 56, 94–101. doi:10.1109/mcom.2018.1700888
- Mori, T., Utsunomiya, Y., Tian, X., and Okuda, T. (2017). "Queueing Theoretic Approach to Job Assignment Strategy Considering Various Inter-arrival of Job in Fog Computing," in 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), South Korea, 27–29 September 2017 (IEEE), 151–156. doi:10.1109/apnoms.2017.8094195
- Nguyen, H. K., Zhang, Y., Chang, Z., and Han, Z. (2017). Parallel and Distributed Resource Allocation with Minimum Traffic Disruption for Network Virtualization. *IEEE Trans. Commun.* 65, 1162–1175. doi:10.1109/tcomm.2017.2650994
- Okay, F. Y., and Ozdemir, S. (2016). "A Fog Computing Based Smart Grid Model," in 2016 International Symposium on Networks, Computers and Communications (ISNCC), 11–13 May 2016 (Hammamet: ISNCC), 1–6. doi:10.1109/isncc.2016.7746062
- Pan, J., and McElhannon, J. (2018). Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* 5, 439–449. doi:10.1109/jiot.2017.2767608
- Prak, J., and Boyd, S. (2017). General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming. *Optim. Control*, 2017, 63, arXiv:1703.07870.
- Reka, S. S., and Dragicevic, T. (2018). Future Effectual Role of Energy Delivery: A Comprehensive Review of Internet of Things and Smart Grid. *Renew. Sustain. Energy Rev.* 91, 90–108. doi:10.1016/j.rser.2018.03.089
- Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet Things J.* 3, 637–646. doi:10.1109/jiot.2016.2579198
- Sun, H., Zhang, J., Wang, P., Lin, J., Guo, S. H., Chen, L., et al. (2019). Edge Computing Technology for Power Distribution IoTs. *Power Syst. Technol.*
- Tran, T. X., Hajisami, A., Pandey, P., and Pompili, D. (2017). Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges. *IEEE Commun. Mag.* 55, 54–61. doi:10.1109/mcom.2017.1600863
- Wang, Q., and Wang, Y. G. (2018). "Research on Power Internet of Things Architecture for Smart Grid Demand," in 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018 (IEEE), 431–438. doi:10.1109/ei2.2018.8582132
- Wang, Z., Wang, Y., and Wang, W. (2018). Smart Internet of Things Oriented Dynamic Load Forecasting Based on Quantum Evolution. *Comput. Integr. Manuf. Syst.* 24, 111–121.
- Xu, X., Li, D., Dai, Z., Li, S., and Chen, X. (2019). A Heuristic Offloading Method for Deep Learning Edge Services in 5G Networks. *IEEE Access* 7, 67734–67744. doi:10.1109/access.2019.2918585
- Yang, Q., Sun, S., Sima, W., He, Y., Chen, Y., Jin, Y., et al. (2019). Progress of Advanced Voltage/current Sensing Techniques for Smart Grid. *High. Volt. Eng.* 45, 349–367. doi:10.13336/j.1003-6520.hve.20190130002
- Zhang, H., Guo, J., Yang, L., Li, X., and Ji, H. (2017). "Computation Offloading Considering Fronthaul and Backhaul in Small-Cell Networks Integrated with MEC," in 2017 IEEE Conference on Computer Communications Workshops

## AUTHOR CONTRIBUTIONS

YL completed the theoretical research, technical method design, experiment and analysis, and the writing and revision of the paper. TL conducted research guidance, technical method demonstration, paper writing and review, etc.

- (INFOCOM WKSHPS), Atlanta, GA, 01-04 May 2017 (IEEE), 115–120. doi:10.1109/infcomw.2017.8116362
- Zhang, W., and Wen, Y. (2018). Energy-efficient Task Execution for Application as a General Topology in Mobile Cloud Computing. *IEEE Trans. Cloud Comput.* 6, 708–719. doi:10.1109/tcc.2015.2511727
- Zhou, Y. M., Li, Z. J., Ge, J. D., Li, C. Y., Zhou, X. Y., Luo, B., et al. (2018). Multi-objective Workflow Scheduling Based on Delay Transmission in Mobile Cloud Computing. *J. Softw.* 29, 3306–3325. doi:10.13328/j.cnki.jos.005479

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

*Copyright © 2022 Liang and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.*