PAPER

# Cloud-Edge-End Collaborative Multi-Service Resource Management for IoT-Based Distribution Grid

Feng WANG[†a)], Xiangyu WEN[†], Lisheng LI[†], Yan WEN[††], Shidong ZHANG[†], *and* Yang LIU[†], *Nonmembers*

**SUMMARY** The rapid advancement of cloud-edge-end collaboration offers a feasible solution to realize low-delay and low-energy-consumption data processing for internet of things (IoT)-based smart distribution grid. The major concern of cloud-edge-end collaboration lies on resource management. However, the joint optimization of heterogeneous resources involves multiple timescales, and the optimization decisions of different timescales are intertwined. In addition, burst electromagnetic interference will affect the channel environment of the distribution grid, leading to inaccuracies in optimization decisions, which can result in negative influences such as slow convergence and strong fluctuations. Hence, we propose a cloud-edge-end collaborative multi-timescale multi-service resource management algorithm. Large-timescale device scheduling is optimized by sliding window pricing matching, which enables accurate matching estimation and effective conflict elimination. Small-timescale compression level selection and power control are jointly optimized by disturbance-robust upper confidence bound (UCB), which perceives the presence of electromagnetic interference and adjusts exploration tendency for convergence improvement. Simulation outcomes illustrate the excellent performance of the proposed algorithm.

***key words:*** *IoT, cloud-edge-device collaboration, distribution grid, sliding window pricing matching, disturbance-robust UCB, resource management*

## 1. Introduction

Distribution grid is the key pillar of smart grid which serves as the bridge between high-voltage power grid and low-voltage energy consumer. Massive internet of things (IoT) devices are deployed in distribution power grid to collect the operation data of major electric equipment [1]. These collected data are uploaded, processed, and mined to support novel multi-services such as source-grid-load-storage physical collaboration, power supply and utilization resource collaboration, operation and control service collaboration, as well as ecological collaboration of external and inner power grid. However, conventional cloud computing cannot fulfill the demands to reduce latency and energy consumption due to long data backhaul distance and communication bottleneck [2].

Cloud-edge-end collaboration is a novel distributed computing model that integrates the advantages of edge computing and cloud computing. It enables efficient resource allocation by leveraging the distributed computing capabilities of the cloud, edge, and end sides. By effectively combining the powerful computing and data analysis capabilities of the cloud with the high real-time performance advantages of the edge, cloud-edge-end collaboration achieves more efficient computing and data processing capabilities while reducing network latency [3]. The major concern of cloud-edge-end collaboration lies on management of multi-dimensional sensing, communication, and computing resources [4], [5]. For example, edge gateways optimize device scheduling, data compression level selection, and transmission power control to reduce data queuing delay, transmission redundancy, and energy consumption. There exist various studies attempt to investigate cloud-edge-end collaborative resource management. In [6], Fang et al. employed deep reinforcement learning assisted cooperative caching and task unloading algorithms to optimize energy consumption of cloud-edge-end cooperative networks. Yang et al. proposed the novel cloud-edge-end orchestrated computing scheme for reducing energy consumption and obtaining optimal policies by repairing missing values in [7]. In [8], Fan et al. proposed the task unloading and resource scheduling plan to realize the overall task handling latency optimization in cloud-edge-end network. However, despite the rapid progress achieved by aforementioned studies, several open issues remain unaddressed, which are summarized as below.

First, multi-service resource management in IoT-based distribution grid involves the coordinated scheduling of sensing, communication, and computing resources. Different optimization variables have different appropriate timescales. For instance, to avoid large communication overheads caused by frequent switching of device, device scheduling could be optimized at large timescale [9]. On the other hand, compression level selection, power control, and computing resource allocation should be jointly optimized at small timescale to accommodate time-varying communication quality as well as queuing backlog [10]. Second, the device scheduling at large timescale depends on decisions of resource management at small timescale, and vice versa. The specific preference of IoT device towards device scheduling depends on both energy consumption and queuing delay performances, which are not a priori until the small-timescale resource management decisions are made and executed [11]. Finally, electromagnetic interference caused by large-scale deployment of power electronics equipment such as distributed photovoltaic (PV) inverter, electric vehicle, reactive power compensation capacitor, and insulated gate bipolar transis-

Manuscript received February 1, 2024.
Manuscript revised March 13, 2024.
Manuscript publicized May 13, 2024.

[†]Shandong Smart Grid Technology Innovation Center, State Grid Shandong Electric Power Research Institute, Jinan 250003, China.

[††]Economic and Technical Research Institute of State Grid Shandong Electric Power, Jinan 250022, China.

a) E-mail: Wangfengscu_1990@163.com

DOI: 10.1587/transfun.2024EAP1011

tor (IGPT) has a strong disturbance on channel gain and data transmission. Conventional compression level selection and power control joint optimization algorithms cannot rapidly explore alternative optimal strategy under impulse electromagnetic interference with intermittent characteristics and suffer from slow convergence and large fluctuation [12].

Matching algorithm is a method used to address combinatorial optimization problem among multiple sides [13], [14]. The elements from one side is matched with elements of the other side based on preference ranking. Matching algorithm has been widely used in resource scheduling. In [15], Wang et al. adopted the stable matching method for data offloading and payment decision between mobile tasks and edge nodes for edge computing. In [16], Zhao et al. employed a novel matching approach where the stations and resources are matched to decide their desired allocation policy for performance enhancement. In [17], Rahim et al. presented a matching method-assisted channel selection algorithm to meet the differentiated requirement of different users. However, the above literatures need perfect global knowledge to establish the required matching preferences, which is mainly designed for single-timescale resource management optimization. For the problem considered in this work, the large-timescale matching preferences depends on the decisions of small-timescale resource management, which is not a priori when establishing matching preferences. In addition, since the information of channel gain and electromagnetic interference occurrence timing is undertrain, conventional matching algorithm relying on deterministic knowledge is no longer suitable.

Several studies attempt to combine matching with reinforcement learning to address multi-timescale resource management problem with stochastic characteristics [18]. Upper confidence bound (UCB) possess the advantages of strong adaptability and flexibility under various environments. UCB has been widely utilized in addressing resource management problems under uncertain information [19]. In [20], Deng et al. presented the UCB-assisted resource scheduling capacity optimization method that improves resource utilization by considering the priority of microservices, achieving more rational exploration. In [21], Qin et al. suggested a UCB-assisted learning approach for matching, using UCB to learn and establish a matching model to estimate the performance of each task offloading candidate, achieving deterministic task offloading selection under certainty. In [22], Qin et al. proposed a machine learning-based conflict and volatility aware UCB matching solution, utilizing UCB to address information uncertainty and realize efficient information resource management. However, conventional UCB adjusts exploration based solely on selection times, which cannot rapidly enhance exploration under strong and impulsive interference. It requires a long time to overcome the performance disturbance and reach convergence. In [23], Zafar et al. proposed a multi-timescale voltage stability constraint V/VAR optimization method, utilizing the multi-timescale optimization method to improve the steady state stability and security of the smart distribution grid. In [24], Li et al. proposed a cloud-edge-device collaborative high-concurrency access management algorithm based on multi-timescale joint optimization of channel pre-allocation and load balancing degree to meet the requirements of high concurrent access for massive IoT devices. However, these methods of jointly optimizing multiple optimization variables on multi-timescale do not take into account the electromagnetic interference and cannot effectively avoid falling into non-optimal solutions, requiring a long time to overcome the performance disturbance and reach convergence.

To address these issues, a cloud-edge-end collaborative multi-timescale multi-service resource management algorithm for IoT-based distribution grid is proposed. First, we develop system models of end-edge data uploading, cloud-side computing resource allocation, and queuing delay constraint. Next, we formulate a multi-timescale joint optimization problem aiming to minimize the weighted sum of total energy consumption and queuing delay with the long-term constraints on device-side and cloud-side queuing delay and occurrence probability of extreme events. Finally, by introducing virtual deficit queues, the formulated multi-timescale optimization problem is developed to a device scheduling optimization problem at large timescale and a joint optimization problem of compression level selection optimization, transmission power control, and computing resource allocation at small timescale, which are addressed by the proposed algorithm. Three major contributions are presented as follows.

- **Low-Latency and Low-energy-consumption Resource Management based on sensing, communication, and computing integration:** We propose a low-latency and low-energy-consumption resource management framework by integrating sensing, communication, and computing, where sensing refers to data acquisition, communication refers to data transmission, and computing refers to computing resource allocation. Specifically, compression level selection (sensing), device scheduling and transmission power control (communication), and computing (computing resource allocation) are jointly optimized, which reduces the weighted sum of energy consumption and queuing delay.
- **Large-timescale Device Scheduling based on Sliding Window Pricing Matching:** We propose a large-timescale device scheduling scheme based on sliding window pricing matching. It utilizes historical performances of energy consumption and queuing delay to estimate the device scheduling preferences. The sliding time window which determines the estimation timespan is utilized to remove outdated historical performances and improve estimation accuracy. Next, matching price rising scheme based on queue backlog and service priority is developed to eliminate matching conflicts caused by device scheduling quota.
- **Small-timescale Compression Level Selection and Power Control based on disturbance-robust UCB:** We develop a disturbance-robust UCB scheme which determines the existence of electromagnetic interference by comparing the performance difference between two consecutive slots. Then, based on the presence of electromagnetic interference,

the weight of confidence interval is dynamically increased to promote the tendency of exploration. This can effectively avoid being trapped in non-optimal solutions, increase the convergence speed, and reduce performance fluctuation.

The remaining part is arranged below. Section 2 describes the system model. Section 3 describes the problem formulation and decomposition. Cloud-edge-end collaborative multi-timescale multi-service resource management algorithm for IoT-based distribution grid is proposed in Sect. 4. Section 5 and Sect. 6 give the simulation results and conclusion.

## 2. System Model

Figure 1 illustrates the cloud-edge-end collaborative multi-service resource management framework for IoT-based distribution grid. The device layer contains IoT devices, which are scheduled to collect key state data of electric equipment and monitor distribution grid operation. The collected data are compressed and uploaded to the edge layer. The edge layer is composed of several edge gateways responsible for resource management of device scheduling, compression level selection, and transmission power control. The data are further uploaded from edge gateway to the cloud layer via 5G. Define the set of $N$ edge gateways as $\mathcal{S} = \{s_1, \cdots, s_n, \cdots, s_N\}$, where $s_n$ represents the $n$-th edge gateway. The set of devices in range of scheduling management of $S_n$ can be denoted by $\mathcal{K}_n = \{k_1^n, \cdots, k_m^n, \cdots, k_{M_n}^n\}$, where $\forall \mathcal{K}_n \cap \mathcal{K}_{n'} = \varnothing$, $n \neq n'$. The cloud layer contains a cloud server denoted as $s_0$, which allocates computing resources to process the data of devices to support novel multi-service applications including source-grid-load-storage physical collaboration, power supply and utilization resource collaboration, operation and control service collaboration, as well as ecological collaboration of external and inner power grid.

To avoid large communication overhead caused by frequent switching of devices, device scheduling is optimized in each period, i.e, large timescale, while compression level selection, transmission power control, as well as computing resource allocation are jointly optimized in each time slot, i.e., small timescale. Total optimization duration contains $I$ periods which is denoted by $\mathcal{I} = \{1, \cdots, i, \cdots, I\}$. One period is composed of $T_0$ slots having same slot length $\tau$. The slots set in periods $i$ is $T(i) = \{(i-1)T_0+1, (i-1)T_0+2, \cdots, iT_0\}$. The total slot set is $\mathcal{T} = \{1, \cdots, t, \cdots, T\}$, and $T = IT_0$. Define large-timescale device scheduling variable as $x_m^n(i) \in \{0, 1\}$. If $x_m^n(i) = 1$, it represents that device $k_m^n$ is scheduled to collect data and upload them to edge gateway $s_n$ in period $i$, and otherwise, $x_m^n(i) = 0$. Due to communication resource limitation, a device scheduling constraint is imposed where at most $q$ devices can be scheduled by $s_n$ simultaneously, which can be expressed as

$$\sum_{m=1}^{M_n} x_m^n(t) \leq q, \forall s_n \in \mathcal{S}, \forall t \in \mathcal{T}. \tag{1}$$

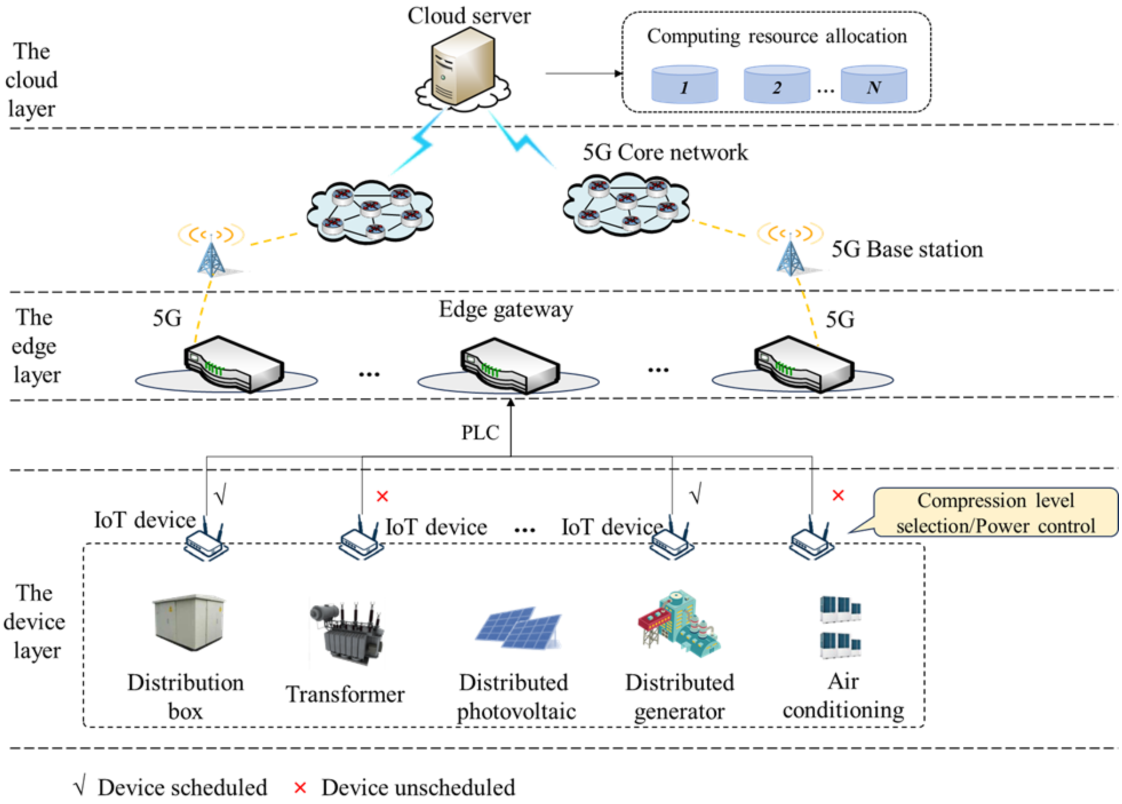

**Fig. 1** Cloud-edge-end collaborative multi-service resource management framework for IoT-based distribution grid.

Define $a_{m,l}^n(t)$, $P_m^n(t)$, and $\varphi_m^n(t)$ as the small-timescale variables of compression level selection, transmission power control, and computing resource allocation in slot $t$, respectively. Among them, compression is utilized to reduce data redundancy and improve transmission efficiency. Assume there exist a total of $L$ compression levels which can be represented as $\mathcal{L} = \{1, \cdots, l, \cdots, L\}$. The compression rate and compression ratio of the $l$-th compression level are denoted as $R_l$ and $\theta_l$, respectively. Thus, after compression, the size of data to be transmitted is reduced to $\frac{1}{\theta_l}$ of the original data size. If $a_{m,l}^n(t) = 1$, it represents that the $l$-th compression level is selected for device $k_m^n$ to compress data within slot $t$, else $a_{m,l}^n(t) = 0$. The transmission power control is introduced in 2.1 and the computing resource allocation is introduced in 2.2. The goal is to achieve low-latency and low-energy-consumption multi-service resource management for distribution grid by jointly optimizing large-timescale device scheduling as well as small-timescale compression level selection, transmission power control, and computing resource allocation.

### 2.1 End-Edge Data Uploading Model

Denote $R_m^n(t)$ as the transmission rate between $k_m^n$ and $s_n$ in slot $t$, which can be calculated as

$$R_m^n(t) = B_m^n \log_2\left(1 + SINR_m^n(t)\right), \tag{2}$$

where $B_m^n$ is the available bandwidth between device $k_m^n$ and edge gateway $s_n$. $SINR_m^n(t)$ represents the signal to inference plus noise ratio (SINR), which can be calculated as

$$SINR_m^n(t) = \frac{P_m^n(t) h_m^n(t)}{N_{m,n}^{\mathrm{EMI}}(t) + N_0}, \tag{3}$$

where $P_m^n(t)$ denotes the device $k_m^n$'s transmission power. $N_0$ is Gaussian white noise power. $h_m^n(t)$ is channel gain between device $k_m^n$ and edge server $s_n$. $N_{m,n}^{\mathrm{EMI}}(t)$ is the electromagnetic interference power.

For simplicity purpose, transmission power $P_m^n(t)$ is discretized into $J$ levels. The set of the transmission power is $\mathcal{P} = \left\{P_{\min}, \cdots, P_{\min} + \frac{(j-1)(P_{\max}-P_{\min})}{J-1}, \cdots, P_{\max}\right\}$, and $P_m^n(t) \in \mathcal{P}$. $P_{\max}$ and $P_{\min}$ represent upper and lower bounds of transmission power.

When device $k_m^n$ selects compression level $l$ for data compression, i.e., $a_{m,l}^n(t) = 1$, the actual transmission rate between $k_m^n$ and $s_n$ is expressed as

$$R_{m,n}^C(t) = \min(R_l, \theta_l R_m^n(t)). \tag{4}$$

where $R_l$ represents the compression rate of the $l$-th compression level.

Then, the amount of data that device $k_m^n$ can upload to edge gateway$s_n$ is expressed as

$$A_m^n(t) = \min\left\{Q_m^n(t), \tau R_{m,n}^C(t)\right\}. \tag{5}$$

The collected data are stored on IoT device and maintained as a data queue. The data queue backlog evolution of $k_m^n$ in slot $t$ can be expressed as

$$Q_m^n(t+1) = \max\{Q_m^n(t) - x_m^n(i) A_m^n(t), 0\} + U_m^n(t), \tag{6}$$

where $U_m^n(t)$ represents collected data size serving as the queue input. $x_m^n(t)A_m^n(t)$ denotes data volume uploaded into edge gateway, which serves as the queue output.

Therefore, the device-side queuing delay of device $k_m^n$ is expressed as

$$\tau_{m,n}^{\mathrm{qu,de}}(t) = \frac{Q_m^n(t)}{\tilde{U}_m^n(t)}. \tag{7}$$

$\tilde{U}_m^n(t)$ represents the average data input rate of $k_m^n$ until slot $t$, which is expressed as

$$\tilde{U}_m^n(t) = \frac{\sum_{d=1}^{t-1} U_m^n(d)}{(t-1)\tau}. \tag{8}$$

The data transmission delay and data transmission energy consumption of $k_m^n$ in slot $t$ are given by

$$\tau_{m,n}^{\mathrm{Tx}}(t) = \frac{A_m^n(t)}{R_{m,n}^C(t)}, \tag{9}$$

$$E_{m,n}^{\mathrm{Tx}}(t) = \tau_{m,n}^{\mathrm{Tx}}(t) P_m^n(t). \tag{10}$$

The edge gateway $s_n$ receives the data uploaded by $k_m^n$ and further transmits to the cloud.

### 2.2 Cloud-Side Computing Resource Allocation Model

The cloud server maintains a data queue for each device. The evolution of the data queues on the devices and cloud is shown in Fig. 2. The data queue backlog corresponding to device $k_m^n$ cached on the cloud can be calculated as

$$W_m^n(t+1) = \max\left\{W_m^n(t) - Z_m^n(t), 0\right\} + \frac{x_m^n(i) A_m^n(t)}{\theta_l}, \tag{11}$$

where $\frac{x_m^n(t)A_m^n(t)}{\theta_l}$ represents the data uploaded by $k_m^n$, serving as the queue input. $Z_m^n(t)$ serves as the queue output, indicating data volume processed by the cloud.

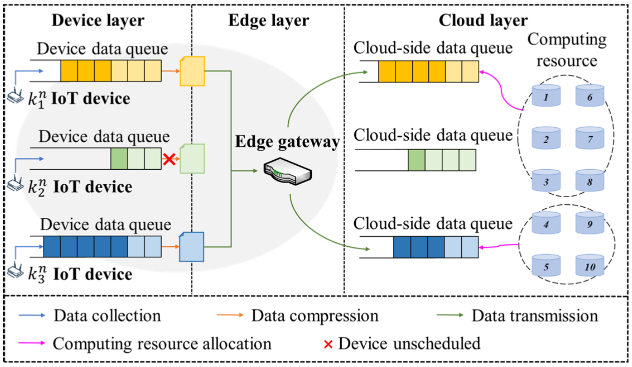

**Fig. 2** The evolution of the data queues on the devices and cloud.

Defining the computing resources (cycles/s) allocated by the cloud server for processing data of $k_m^n$ as $\varphi_m^n(t)$, $Z_m^n(t)$ is calculated as

$$Z_m^n(t) = \min\left\{\frac{\tau\varphi_m^n(t)}{\beta_m^n}, W_m^n(t)\right\}, \tag{12}$$

where $\beta_m^n$ represents the data computational complexity (cycles/bit) of device $k_m^n$.

Therefore, the cloud-side queuing delay of device $k_m^n$ is expressed as

$$\tau_{m,n}^{\text{qu,cl}}(t) = \frac{W_m^n(t)}{\tilde{A}_m^n(t)}. \tag{13}$$

$\tilde{A}_m^n(t)$ represents the average data input rate of $W_m^n(t)$ until slot $t$, which is expressed as

$$\begin{aligned}\tilde{A}_m^n(t) =& \frac{\sum_{e=1}^{i-1}\sum_{d=(e-1)T_0+1}^{eT_0} x_m^n(e)A_m^n(d)}{(t-1)\theta_l} \\ &+ \frac{\sum_{d=(i-1)T_0+1}^{t-1} x_m^n(i)A_m^n(d)}{(t-1)\theta_l}.\end{aligned} \tag{14}$$

The data processing delay and energy consumption for cloud server processing $k_m^n$ data are expressed as

$$\tau_{m,n}^{\text{pro}}(t) = \frac{Z_m^n(t)\beta_m^n}{\varphi_m^n(t)}, \tag{15}$$

$$E_{m,n}^{\text{pro}}(t) = \varpi_{m,n}\left[\varphi_m^n(t)\right]^3 \tau_{m,n}^{\text{pro}}(t), \tag{16}$$

where $\varpi_{m,n}$ is the energy consumption factor.

## 2.3 Long-Term Queuing Delay Constraint

### 2.3.1 Device-Side Long-Term Queuing Delay Constraint

Considering the low-latency requirement for services such as source-grid-load-storage physical collaboration, long-term queuing delay constraints are imposed on device-side and cloud-side data queues [25].

The device-side long-term average queuing delay constraint of device $k_m^n$ can be expressed as

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\tau_{m,n}^{\text{qu,de}}(t) \le \tau_{m,\max}^{n,\text{de}}, \tag{17}$$

where $\tau_{m,\max}^{n,\text{de}}$ represents the maximum tolerable device-side queuing delay for $k_m^n$.

### 2.3.2 Cloud-Side Long-Term Queuing Delay Constraint

The cloud-side long-term average queuing delay constraint of device $k_m^n$ can be expressed as

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\tau_{m,n}^{\text{qu,cl}}(t) \le \tau_{m,\max}^{n,\text{cl}}, \tag{18}$$

where $\tau_{m,\max}^{n,\text{cl}}$ represents the maximum tolerable cloud-side queuing delay for $k_m^n$.

### 2.3.3 Device-Side Extreme Event Constraint

As devices have limited data stored capacity, an extreme event occurs when the device-side data queue backlog exceeds a certain threshold. The extreme event may lead to data loss and data overwriting, resulting in incomplete multi-service information, inefficient service operations, and even grid operation risks. Therefore, we impose a occurrence probability constraint of the extreme events, which is expressed as

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\Pr\left(Q_m^n(t) > Q_{m,\max}^n\right) \le \chi_m^n, \tag{19}$$

where $Q_{m,\max}^n$ is the device-side data queue backlog threshold for device $k_m^n$, and $\chi_m^n$ is the maximum tolerable occurrence probability of extreme events for $k_m^n$.

## 2.4 Total Energy Consumption and Queuing Delay Model

The total queuing delay of device $k_m^n$ includes the device-side and cloud-side queuing delay, which is given by

$$\tau_{m,n}^{\text{qu,total}}(t) = \tau_{m,n}^{\text{qu,de}}(t) + \tau_{m,n}^{\text{qu,cl}}(t). \tag{20}$$

The total energy consumption of device $k_m^n$ includes the data transmission and data processing energy consumption. It can be expressed as

$$E_{m,n}^{\text{total}}(t) = E_{m,n}^{\text{Tx}}(t) + E_{m,n}^{\text{pro}}(t). \tag{21}$$

# 3. Problem Formulation and Decomposition

## 3.1 Problem Formulation

To realize low-latency and low-energy consumption multi-service resource management for IoT-based distribution grid, the objective is to minimize average weighted sum of total energy consumption and queuing delay of all devices over $T$ slots. The large-timescale device scheduling as well as small-timescale compression level selection, transmission power control, and computing resource allocation are jointly optimized with the long-term constraints on queuing delay and occurrence probability of the extreme events. Define the weighted sum of total energy consumption and queuing delay as

$$\zeta(t) = \tau_{m,n}^{\text{qu,total}}(t) + V_E E_{m,n}^{\text{total}}(t), \tag{22}$$

where $V_E$ is a weight that balances the energy consumption and queuing delay. Hence, the optimization problem is given by

$$\mathbf{P1}: \min_{\{x_m^n(i), P_m^n(t), a_{m,l}^n(t), \varphi_m^n(t)\}} \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M_n}\zeta(t)$$

$$\begin{aligned}
\text{s.t.}\ \ & C_1 : x_m^n(i) \in \{0,1\}, \forall s_n \in \mathcal{S}, \forall k_m^n \in \mathcal{K}_n, \forall i \in \mathcal{I},\\
& C_2 : (1), \forall s_n \in \mathcal{S}, \forall i \in \mathcal{I},\\
& C_3 : P_m^n(t) \in \mathcal{P}, \forall s_n \in \mathcal{S}, \forall k_m^n \in \mathcal{K}_n, \forall t \in \mathcal{T},\\
& C_4 : a_{m,l}^n(t) \in \{0,1\}, \forall s_n \in \mathcal{S}, \forall k_m^n \in \mathcal{K}_n,\\
& \qquad\qquad\qquad\qquad \forall t \in \mathcal{T}, \forall l \in \mathcal{L},\\
& C_5 : \sum_{l=1}^{L} a_{m,l}^n(t) = 1, \forall s_n \in \mathcal{S}, \forall k_m^n \in \mathcal{K}_n, \forall t \in \mathcal{T},\\
& C_6 : \sum_{n=1}^{S_n} \sum_{m=1}^{M_n} \varphi_m^n(t) \leq \varphi_{\max}(t), \forall s_n \in \mathcal{S}, \forall t \in \mathcal{T},\\
& C_7 : (17) \sim (19), \forall s_n \in \mathcal{S}, \forall k_m^n \in \mathcal{K}_n. \qquad (23)
\end{aligned}$$

$C_1$ and $C_2$ are the device scheduling constraints. $C_3$ indicates transmission power constraint. $C_4$ as well as $C_5$ are the compression level selection constraints, indicating that a device can only choose a single compression level per slot. $C_6$ indicates computing resource constraints of the cloud server, where $\varphi_{\max}(t)$ means maximum usable computing resources for cloud server. $C_7$ includes long-term device-side and cloud-side queuing delay constraints as well as occurrence probability constraint of the extreme events.

### 3.2 Problem Decomposition

It is difficult to settle **P1** on account of contradiction of long-term constraints and short-term optimization. Therefore, we introduce virtual queue to decompose the optimization problem [26]. For the long-term device-side queuing delay constraint, define device-side queuing delay deficit virtual queue as

$$D_m^n(t+1) = \max\{D_m^n(t) + \tau_{m,n}^{\text{qu,de}}(t) - \tau_{m,\max}^{n,\text{de}}, 0\}. \quad (24)$$

$D_m^n(t)$ represents the deviation degree between the current device-side queuing delay and the constraint.

Similarly, for the long-term cloud-side queuing delay constraint, define the cloud-side queuing delay deficit virtual queue as

$$O_m^n(t+1) = \max\{O_m^n(t) + \tau_{m,n}^{\text{qu,cl}}(t) - \tau_{m,\max}^{n,\text{cl}}, 0\}. \quad (25)$$

For the long-term occurrence probability constraint on the extreme events, define the extreme event occurrence probability deficit virtual queue as

$$Y_m^n(t+1) = \max\{Y_m^n(t) + \mathbb{I}(Q_m^n(t+1) > Q_{m,\max}^n) - \chi_m^n, 0\}. \quad (26)$$

$\mathbb{I}\{\bullet\} = 1$ indicates the event is true, or $\mathbb{I}\{\bullet\} = 0$.

Based on the Lyapunov optimization [27], the original long-term optimization problem is developed to several short-term optimization problems per slot. The optimization objective of the decomposed problem is given by

$$\begin{aligned}
\zeta'(t) = {} & \tau_{m,n}^{\text{qu,total}}(t) + V_E E_{m,n}^{\text{total}}(t) - V_Q Q_m^n(t) x_m^n(t) A_m^n(t)\\
& + V_Y Y_m^n(t) \mathbb{I}(Q_m^n(t+1) > Q_{m,\max}^n)\\
& + V_D D_m^n(t) \tau_{m,n}^{\text{qu,de}}(t) - V_W W_m^n(t) Z_m^n(t)\\
& + V_O O_m^n(t) \tau_{m,n}^{\text{qu,cl}}(t), \qquad (27)
\end{aligned}$$

where $V_Q$, $V_Y$, $V_D$, $V_W$, and $V_O$ are weights of $Q_m^n(t)$, $Y_m^n(t)$, $D_m^n(t)$, $W_m^n(t)$, and $O_m^n(t)$, respectively.

Therefore, the decomposed optimization problem is given by

$$\begin{aligned}
\textbf{P2}: & \min_{\{x_m^n(i), P_m^n(t), a_{m,l}^n(t), \varphi_m^n(t)\}} \zeta'(t)\\
\text{s.t.}\ & C_1 \sim C_6. \qquad (28)
\end{aligned}$$

Based on optimization variables, decomposed optimization problem is separated as two optimization subproblems.

The first optimization subproblem aims to minimize the device-side queuing delay, data transmission energy consumption, and device-side queue fluctuation by jointly optimizing large-timescale device scheduling as well as small-timescale compression level selection and transmission power control, i.e.,

$$\begin{aligned}
\textbf{SP1}: & \min_{\{x_m^n(i), P_m^n(t), a_{m,l}^n(t)\}} \zeta_{m,n}^{\text{SP1}}(t)\\
& = \tau_{m,n}^{\text{qu,de}}(t) + V_E E_{m,n}^{\text{Tx}}(t)\\
& - V_Q Q_m^n(t) x_m^n(i) A_m^n(t) + V_D D_m^n(t) \tau_{m,n}^{\text{qu,de}}(t)\\
& + V_Y Y_m^n(t) \mathbb{I}(Q_m^n(t+1) > Q_{m,\max}^n)\\
\text{s.t.}\ & C_1 \sim C_5. \qquad (29)
\end{aligned}$$

The second optimization subproblem aims to minimize the cloud-side queuing delay, data processing energy consumption, and cloud-side queue fluctuation by optimizing the computing resource allocation, i.e.,

$$\begin{aligned}
\textbf{SP2}: & \min_{\{\varphi_m^n(t)\}} \zeta_{m,n}^{\text{SP2}}(t)\\
& = \tau_{m,n}^{\text{qu,cl}}(t) + V_E E_{m,n}^{\text{pro}}(t) - V_W W_m^n(t) Z_m^n(t)\\
& + V_O O_m^n(t) \tau_{m,n}^{\text{qu,cl}}(t)\\
\text{s.t.}\ & C_6. \qquad (30)
\end{aligned}$$

## 4. Cloud-Edge-End Collaborative Multi-Timescale Multi-Service Resource Management Algorithm for IoT-Based Distribution Grid

A cloud-edge-end collaborative multi-timescale multi-service resource management algorithm for IoT-based distribution grid is proposed. It consists of a large-timescale device scheduling algorithm based on sliding window pricing matching, and a small-timescale algorithm based on disturbance-robust UCB. Fig. 3 illustrates the detailed algorithm principle which is described below.

### 4.1 Large-Timescale Device Scheduling Based on Sliding Window Pricing Matching

Matching theory is used to address the large-timescale device

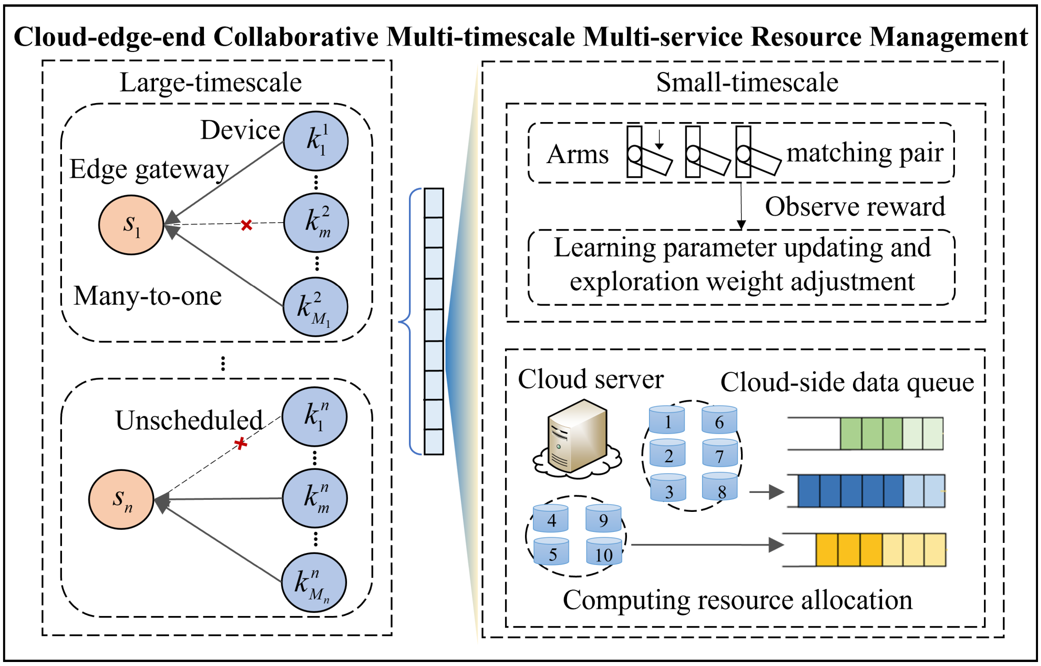

**Fig. 3** Framework of cloud-edge-end collaborative multi-timescale multi-service resource management.

scheduling problem. However, conventional matching algorithms require perfect global knowledge to construct preferences, which is not a priori and depends on small-timescale optimization decisions. To address this issue, we improve conventional matching by developing a sliding window price matching approach. First, the matching preferences of device scheduling is constructed based on historical performances already known, and a sliding window is utilized to remove outdated historical performances from preference estimation to improve estimation accuracy and increase convergence speed. Second, a price rising mechanism based on device-side queue backlog and service priority is developed to eliminate matching conflict caused by device scheduling constraint defined in (1). The procedure of sliding window-based matching preference list construction and price rising-based matching competition elimination are introduced as follows.

### 4.1.1 Sliding Window Based Matching Preference List Construction

Model device scheduling problem as a group of parallel many-to-one matching, while the $n$-th matching is a mapping from the device set $\mathcal{K}_n$ to the edge gateway $s_n$. Define $F$ as the sliding window size in terms of slots. To achieve the goal of minimizing total energy consumption and queuing delay, the matching preference of device scheduling is designed to be negatively related with performance of energy consumption and queuing delay over previous $F$ slots. The preference of device $k_m^n$ towards scheduling in period $i$ is given by

$$\Theta_m^{n,\mathrm{con}}(i) = \frac{1}{\frac{1}{F}\sum_{d=0}^{F-1}\zeta_{m,n}^{\mathrm{SP1}}((i-1)T_0-d)} - \kappa_m^{n,\mathrm{con}}(i), \quad (31)$$

where $\kappa_m^{n,\mathrm{con}}(i)$ is matching price used to eliminate conflict, and its initial value is set as zero.

On the other hand, an unscheduled device saves energy consumption. Therefore, the preference of $k_m^n$ towards unscheduling is given by

$$\Theta_m^{n,\mathrm{dis}}(i) = \frac{V_\Theta}{F}\sum_{d=0}^{F-1} V_E E_{m,n}^{\mathrm{Tx}}((i-1)T_0-d), \quad (32)$$

where $V_\Theta$ represents the adjustment parameter used to uniform order of magnitude between $\Theta_m^{n,\mathrm{con}}(i)$ and $\Theta_m^{n,\mathrm{dis}}(i)$.

For device $k_m^n$, if $\Theta_m^{n,\mathrm{con}}(i) > \Theta_m^{n,\mathrm{dis}}(i)$, it sends an access request to edge gateway $s_n$. Otherwise, it does not send access request and stays unscheduled in this period. Afterwards, if the number of access requests received by each edge gateway $s_n$ exceeds $q$, the constraint defined in (1). We develop a price rising mechanism with service priority awareness to eliminate conflict, which is introduced in the next subsection.

### 4.1.2 Service Priority-Aware Matching Price Rising

First, all devices competing to access are added in a set $\Omega_n$. Then, increase the matching prices of all the devices in set $\Omega_n$ as

$$\kappa_m^{n,\mathrm{con}}(i) = \kappa_m^{n,\mathrm{con}}(i) + \frac{\Delta\kappa}{p_m^n}, \quad (33)$$

where $p_m^n$ represents the service priority of device $k_m^n$. In

**Algorithm 1** Large-timescale Device Scheduling based on Sliding Window Pricing Matching

```
Initialization:
for $i = 1, 2, \cdots, I$ do
    Initialize $\Omega_n = \mathcal{K}_n$, $\kappa_m^{n,\mathrm{con}}(i) = 0$ and $\Delta\kappa = 0.1$.
Sliding Window based Matching Preference List Construction:
    for $n = 1, 2, \cdots, N$ do
        while $|\Omega_n| > q$ do
            for $m = 1, 2, \cdots, M_n$ do
                Calculate $\Theta_m^{n,\mathrm{con}}(i)$ and $\Theta_m^{n,\mathrm{dis}}(i)$ based on (31), (32).
                if $\Theta_m^{n,\mathrm{con}}(i) > \Theta_m^{n,\mathrm{dis}}(i)$ then
                    Device $k_m^n$ send an access request to edge gateway $s_n$.
                else
                    $\Omega_n = \Omega_n \backslash k_m^n$.
                end if
            end for
Service Priority-aware Matching Price Rising:
            if The number of requesting devices is less than or equal to $q$ then
                The edge gateway directly matches with all requesting devices.
            else
                Update $\kappa_m^{n,\mathrm{con}}(i)$ based on (33).
            end if
        end while
    end for
end for
```

other words, the price rising step correlates negatively with service priority, which ensures devices supporting high-priority services have a slower price rising step and a larger tendency to be matched, i.e., scheduled.

Next, update the matching preference based on (33). Due to service priority-aware price rising, the preferences of devices with smaller priories are reduced rapidly. If $\Theta_m^{n,\mathrm{con}}(i) < \Theta_m^{n,\mathrm{dis}}(i)$, namely, the benefit of being unscheduled is larger than that of being scheduled, device $k_m^n$ is forced to give up and removed from the set $\Omega_n$. Thus, the price rising procedure is repeated to eliminate matching conflict until the number of competing devices is less than $q$.

The specific implementation procedures of sliding window pricing matching are summarized in Algorithm 1.

### 4.2 Small-Timescale Compression Level Selection and Power Control Based on Disturbance-Robust UCB

UCB is an effective method for solving the multi-armed bandit (MAB) problem. It considers both the average gain of the arms and the upper bound of the confidence level to strike a balance between exploration and exploitation. However, the traditional UCB algorithm uses a fixed exploration weight in the solution process, which cannot dynamically adjust the weight of the confidence interval when the burst electromagnetic interference occurs. As a result, the algorithm lacks the ability to explore new optimal solutions after the channel environment changes and is prone to converging to non-optimal solutions. Therefore, we propose a disturbance-robust UCB scheme that determines the existence of electromagnetic interference by comparing the performance difference between two consecutive slots. Then, based on the presence of electromagnetic interference, the weight of the confidence interval is dynamically increased to promote the tendency of exploration. This can effectively avoid being trapped in non-optimal solutions, increase the convergence speed, and reduce performance fluctuation, thereby achieving disturbance robustness under the presence of burst electromagnetic interference.

**Algorithm 2** Small-timescale Compression Level Selection and Power Control based on Disturbance-robust UCB

```
for $t = 1, 2, \ldots, T$ do
    Phase 1: Initialization
    Device $k_m^n$ iterates through all $L \times J$ arms to obtain initial reward values $\Upsilon_{m,n}^{l,j}(0)$.
    Phase 2: Exploration and exploitation balanced arm selection
    Calculate the upper confidence bound for each arm based on (34).
    Select the arm with the largest upper confidence bound as the optimal solution.
    Phase 3: Learning parameter updating
    Calculate the reward $\Upsilon_{m,n}^{l,j}(t)$ and update $\overline{\Upsilon}_{m,n}^{l,j}(t)$ and $v_{m,n}^{l,j}(t)$.
    Phase 4: Disturbance-robust exploration weight adjustment
    Detect the presence of interference based on (37).
    if $\Delta\overline{\Upsilon}_m^n(t) > \Delta\overline{\Upsilon}_{m,n}^{\mathrm{th}}$ then
        Update the exploration factor based on (38).
    else
        Let $\omega(t+1)$ be $\omega_0$.
    end if
end for
```

Upon obtaining the large-timescale device scheduling decision based on sliding window pricing matching, we proceed to optimize the small-timescale compression level selection and power control using the proposed disturbance-robust UCB to increase the convergence speed, and reduce performance fluctuation. First, we model the optimization problem of compression level selection and power control as a MAB problem, and the details are as follows.

Arms: Arms refer to the candidate solutions of compression level selection and transmission power control. To reduce the solution space, one compression level and one transmission power level are combined to form a candidate arm. Since there exist $L$ data compression levels and $J$ transmission power levels, the total number of arms is $L \times J$. Arm $\pi_{l,j}$ represents the matching pair of the $l$-th compression level and the $j$-th power level.

Actions: Actions are the decisions of selecting which arm to reduce the reward.

Rewards: Reward is defined as the benefit perceived after selecting and executing an action. The reward corresponding to selecting arm $\pi_{l,j}$ can be denoted by the reciprocal of the optimization goal, which is $\Upsilon_{m,n}^{l,j}(t) = \frac{1}{\zeta_{m,n}^{\mathrm{SP1}}(t)}$.

The concrete realization of small-timescale compression level selection and power control based on disturbance-robust UCB is concluded in Algorithm 2, which is introduced below.

Step 1: Initialization. Device $k_m^n$ iterates through all $L \times J$ arms to obtain initial reward values $\Upsilon_{m,n}^{l,j}(0)$. Define the arm selection indicator variable as $g_{m,n}^{l,j}(t)$. When device $k_m^n$ selects arm $\pi_{l,j}$ in slot $t$, $g_{m,n}^{l,j}(t) = 1$. Average reward value for device $k_m^n$ for $\pi_{l,j}$ can be given by $\overline{\Upsilon}_{m,n}^{l,j}(t)$, and the number of times that $\pi_{l,j}$ has been selected up to slot $t$ is denoted as $\upsilon_{m,n}^{l,j}(t)$. The exploration weight of confidence interval is denoted as $\omega(t)$. Initialize $\overline{\Upsilon}_{m,n}^{l,j}(0) = \Upsilon_{m,n}^{l,j}(0)$ and $\upsilon_{m,n}^{l,j}(0) = 1$. Define $\omega(1) = \omega_0$, where $\omega_0$ represents the initial value of the exploration weight.

Step 2: Exploration and exploitation balanced arm selection. Based on $\overline{\Upsilon}_{m,n}^{l,j}(t)$ and $\upsilon_{m,n}^{l,j}(t)$, device $k_m^n$ calculates the upper confidence bound for each arm as

$$\tilde{\Upsilon}_{m,n}^{l,j}(t) = \overline{\Upsilon}_{m,n}^{l,j}(t-1) + \omega(t)\sqrt{\frac{\ln(t)}{\upsilon_{m,n}^{l,j}(t-1)}}, \tag{34}$$

where $\overline{\Upsilon}_{m,n}^{l,j}(t-1)$ is average reward for device $k_m^n$ for selecting arm $\pi_{l,j}$ up to slot $(t-1)$.

The second term allows device to balance exploration and exploitation based on the number of arm selection times. If $\upsilon_{m,n}^{l,j}(t-1)$ is large which indicates that arm $\pi_{l,j}$ has already been selected many times, the second term of exploration becomes smaller to enforce exploitation with larger average reward. Conversely, if $\upsilon_{m,n}^{l,j}(t-1)$ is smaller, the second term becomes dominant to enforce exploration of arms with potential better rewards.

Upon obtaining $\tilde{\Upsilon}_{m,n}^{l,j}(t)$, the arm having the largest upper confidence bound is selected as optimal solution in slot $t$.

Step 3: Learning parameter updating. After executing selected compression level and power level, the average reward, i.e., $\overline{\Upsilon}_{m,n}^{l,j}(t)$, and arm selection time, i.e., $\upsilon_{m,n}^{l,j}(t)$, are updated based on observed reward $\Upsilon_{m,n}^{l,j}(t)$, which are given by

$$\overline{\Upsilon}_{m,n}^{l,j}(t) = \frac{\overline{\Upsilon}_{m,n}^{l,j}(t-1)\,\upsilon_{m,n}^{l,j}(t-1) + g_{m,n}^{l,j}(t)\,\Upsilon_{m,n}^{l,j}(t)}{\upsilon_{m,n}^{l,j}(t-1) + g_{m,n}^{l,j}(t)}, \tag{35}$$

$$\upsilon_{m,n}^{l,j}(t) = \upsilon_{m,n}^{l,j}(t-1) + g_{m,n}^{l,j}(t). \tag{36}$$

Step 4: Disturbance-robust exploration weight adjustment. The presence of interference is detected by comparing the deviation of rewards between adjacent slots, and can be expressed as

$$\begin{aligned}\Delta\Upsilon_m^n(t) = &\sum_{l=1}^{L}\sum_{j=1}^{J} g_{m,n}^{l,j}(t)\,\Upsilon_{m,n}^{l,j}(t)\\ &-\sum_{l=1}^{L}\sum_{j=1}^{J} g_{m,n}^{l,j}(t-1)\,\Upsilon_{m,n}^{l,j}(t-1).\end{aligned} \tag{37}$$

If $\Delta\overline{\Upsilon}_m^n(t)$ is greater than deviation threshold $\Delta\overline{\Upsilon}_{m,n}^{\text{th}}$, electromagnetic interference is indicated. In this case, the exploration factor is rapidly increased to facilitate faster convergence. If $\Delta\overline{\Upsilon}_m^n(t)$ is less than or equal to $\Delta\overline{\Upsilon}_{m,n}^{\text{th}}$, let $\omega(t+1)$ remain $\omega_0$. Therefore, the exploration weight is adjusted as

$$\omega(t+1) = \begin{cases} \omega(t) + V_\Upsilon \frac{\Delta\Upsilon_m^n(t)}{\Delta\Upsilon_{m,n}^{\text{th}}}, \Delta\Upsilon_m^n(t) > \Delta\Upsilon_{m,n}^{\text{th}} \\ \omega_0, \Delta\Upsilon_m^n(t) \le \Delta\Upsilon_{m,n}^{\text{th}} \end{cases} \tag{38}$$

The proposed exploration weight adjustment mechanism can significantly reduces convergence time and performance degradation by exploring alternative arms with potentially better rewards, thereby achieving disturbance robustness under the presence of burst electromagnetic interference.

**Algorithm 3** Cloud-side Greedy-based Computational Resource Allocation Algorithm

```
Input: $W_m^n(t)$, $\tau_{m,n}^{\text{qu,cl}}(t)$ $E_{m,n}^{\text{pro}}(t)$ $Z_m^n(t)$, $O_m^n(t)$.
Initialize $\mathcal{K}_n' = \{k_m^n \in \mathcal{K}_n | W_m^n(t) > 0\}$, and $\Delta\varphi_{\max}(t) = \varphi_{\max}(t)$.
while $\mathcal{K}_n' \neq \varnothing$ and $\Delta\varphi_{\max}(t) > 0$ do
    $\varphi_m^n(t) = \min\{\Delta\varphi_{\max}(t), \frac{\beta_m^n}{\tau}\left[W_m^n(t)\right]\}$.
    $k_{m^*}^n = \text{argmax}_{k_m^n \in \mathcal{K}_n} \zeta_{m,n}^{\text{SP2}}(t)$.
    $\Delta\varphi_{\max}(t) = \Delta\varphi_{\max}(t) - \varphi_{m^*}^n(t)$.
    $\mathcal{K}_n' = \mathcal{K}_n' \backslash k_{m^*}^n$.
end while.
```

### 4.3 Cloud-Side Greedy-Based Computational Resource Allocation Algorithm

A low-complexity heuristic approach for addressing cloud-side computing resource allocation problem is presented based on min-max theory. The main concept is to aggressively allocate computing recourse to the device data queue with the worst performance, i.e., the maximum objective value defined in terms of $\zeta_{m,n}^{\text{SP2}}(t)$. The implementation procedure is summarized in Algorithm 3.

First, initialize the set of devices with computing resource requirements as $\mathcal{K}_n' = \{k_m^n \in \mathcal{K}_n | W_m^n(t) > 0\}$ and the available computing resources as $\Delta\varphi_{\max}(t) = \varphi_{\max}(t)$. Then, calculate the objective value $\zeta_{m,n}^{\text{SP2}}(t)$ achieved by processing the data of each device $k_m^n$ and the corresponding required amount of computing resources $\varphi_m^n(t)$. Select the device $k_{m^*}^n$ with the largest value $\zeta_{m^*,n}^{\text{SP2}}(t)$, and allocate the required computing resources $\varphi_{m^*}^n(t)$ to process its data. Afterwards, remove $k_{m^*}^n$ from the set $\mathcal{K}_n'$ and update $\Delta\varphi_{\max}(t) = \Delta\varphi_{\max}(t) - \varphi_{m^*}^n(t)$.

The computing resource allocation iteration terminates once all device data have been processed, meaning that $\mathcal{K}_n' = \varnothing$, or no computing resources remain free, meaning that $\Delta\varphi_{\max}(t) = 0$.

## 5. Simulation Result

We verify the effective performance of the proposed algorithm through simulations. A multi-service resource management scenario for IoT-based distribution grid is considered. The scenario includes 1 cloud server, 10 edge gateways, and 400 IoT devices. Each edge gateway manages the scheduling of 40 IoT devices. Other relevant simulation parameters are specified in Table 1. The data compression sensing learning based information aggregation algorithm (DCSL) [28] and the UCB and virtual queue theory based

**Table 1** Simulation parameters.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $N$ | 10 | $M_n$ | 40 |
| $T$ | 100 | $T_0$ | 10 |
| $F$ | 4 | $I$ | 10 |
| $\tau$ | 100 ms | $q$ | 25 |
| $B_m^n$ | [1,2] MHz | $U_m^n(t)$ | [1.2,1.8] Mbits |
| $p_m^n$ | [1,2,3] | $P_m^n(t)$ | [0.1,0.2,0.3,0.4,0.5] W |
| $L$ | 3 | $N_0$ | -114 dBm |
| $V_E$ | $2 \times 10^3$ | $\chi_m^n$ | 0.5 |
| $\tau_{m,\max}^{n,\mathrm{de}}$ | 100 ms | $\tau_{m,\max}^{n,\mathrm{cl}}$ | 120 ms |

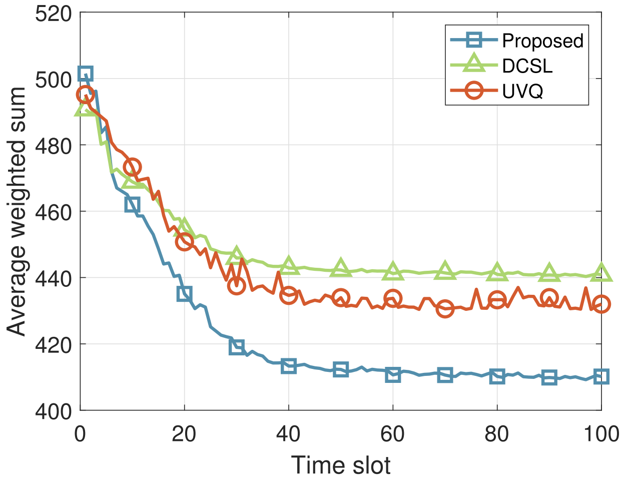

**Fig. 4** Average weighted sum of energy consumption and queuing delay versus time slot.

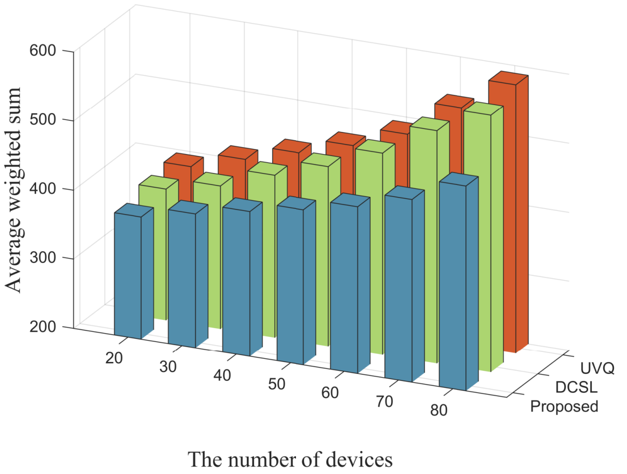

**Fig. 5** Average weighted sum of energy consumption and queuing delay ($M_n$ = 20 ~ 80).

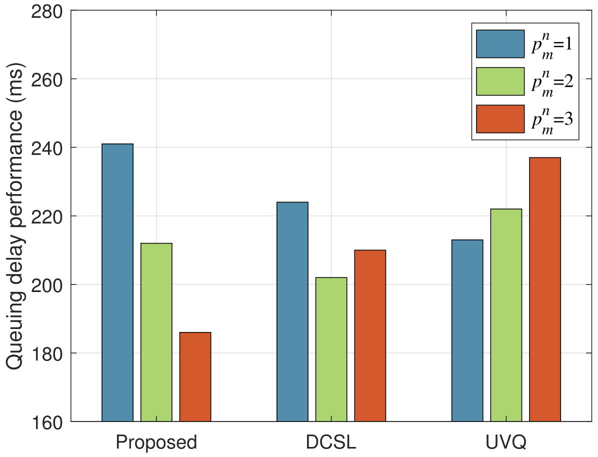

**Fig. 6** The queuing delay performance and energy consumption performance of services with differentiated priorities.

device scheduling algorithm (UVQ) [29] are used for comparison. DCSL schedules the device randomly and optimizes data compression level and transmission power control by using the epsilon greedy algorithm. UVQ optimizes device scheduling based on UCB and utilizes a constant compression level. Then UVQ constructs a virtual queue of energy consumption constraints to adjust transmission power dynamically. However, both DCSL and UVQ process the arrived data in serial order without considering service priority and computing resource allocation. None of them considers the solution to address the performance fluctuation caused by electromagnetic interference and guarantee constraint of extreme event occurrences.

Figure 4 illustrates the average weighted sum of energy consumption and queuing delay versus time slot. The proposed algorithm has the lowest average sum. When $t$=100, the weighted sum of the proposed algorithm is decreased by 6.96% and 5.05% compared to DCSL and UVQ. The rationale behind this is that the proposed algorithm not only optimizes large-timescale device scheduling but also optimizes small-timescale data compression level, transmission power control and computing resource allocation, which is adaptable to time-varying channel state conditions. In addition, it combines interference-robust weight adjustment and virtual deficit queue backlog-aware extreme event occurrence constraint guarantee mechanisms to improve convergence and optimality performances. Details are explained in other figures.

Figuer 5 illustrates the average weighted sum of energy consumption and queuing delay versus the number of devices $M_n$. When increasing $M_n$ from 20 to 80, the competition among devices becomes intense, and the performances of all the algorithms degrade. However, under 80 devices, the average weighted sum achieved by the proposed algorithm is 13.11% and 15.62% less than those of DCSL and UVQ, respectively. The rationale behind this is that the sliding window size can be flexibly adjusted based on practical needs to remove outdated historical performances for realizing more accurate preference estimation. In addition, price rising based on service priority and queuing backlog effectively eliminates device competition. Particularly, devices with lower service priority and smaller queuing backlog are forced to wait for the next period.

Figure 6 shows the queuing performance under differentiated service priorities. It is obvious that service priority awareness is achieved by the proposed algorithm. The queu-

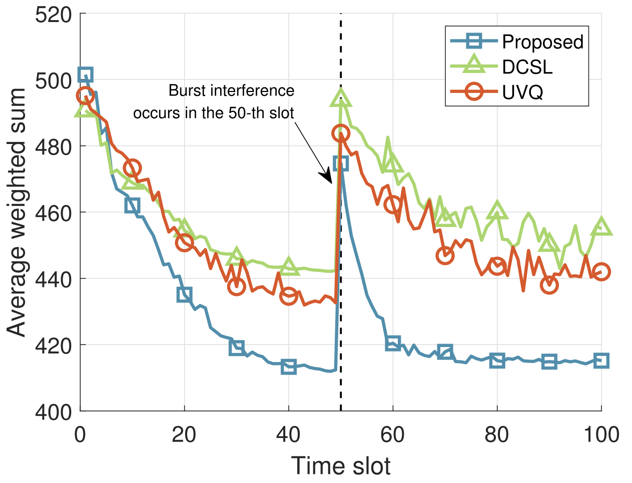

**Fig. 7** Average weighted sum of energy consumption and queuing delay under sudden interference.

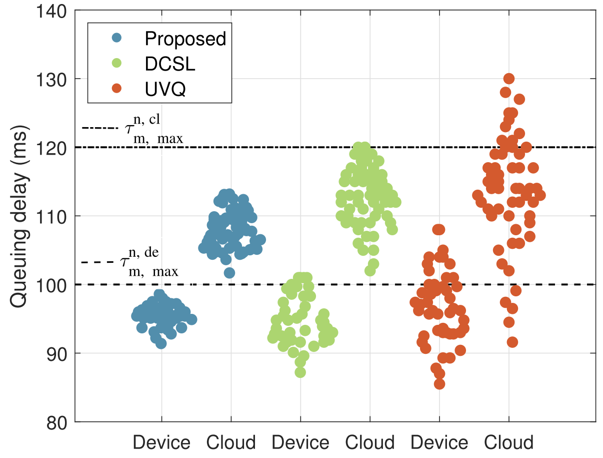

**Fig. 8** Scatter plot of queuing delay performances on the device side and cloud side.

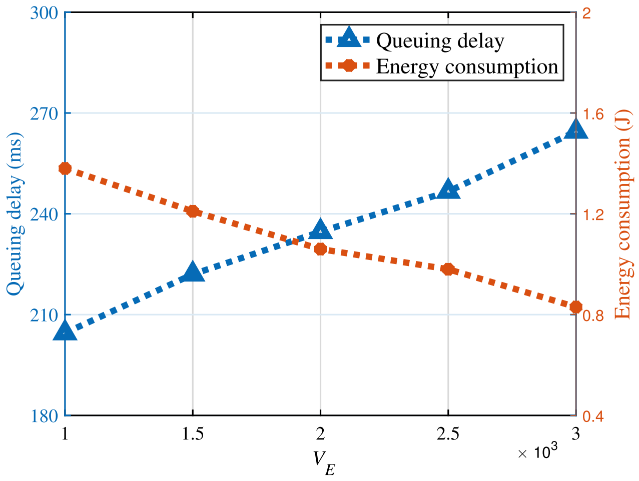

**Fig. 9** Total queuing delay and total energy consumption versus $V_E$.

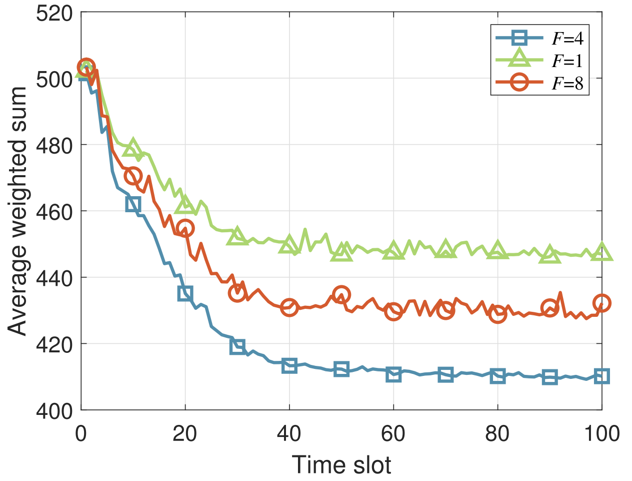

**Fig. 10** Average weighted sum of energy consumption and queuing delay versus $F$.

ing delay of the highest-priority service is 22.82% lower than that of the lowest-priority service. In comparison, neither DCSL nor UVQ demonstrates clear queuing delay performance improvement for high-priority service. This is due to the consideration of service priority in the matching price rising mechanism.

Figure 7 shows the average weighted sum of energy consumption and queuing delay under sudden interference. A bust electromagnetic interference occurs at the 50-th slot, and the performances of all the three algorithms become degraded instantaneously. However, the proposed algorithm converges fastest to the smallest weighted sum. Compared to DCSL and UVQ, the average weighted sum increase is reduced by 14.97% and 21.85%. This gain of interference robustness originates from the careful detection of interference existence by comparing performance degradations between consecutive slots. In addition, the exploration weight of UCB is rapidly increased to find other better alternative strategy in the presence of interference.

Figure 8 shows the scatter plot of queuing delay performances on the device side and cloud side. Compared to the DCSL and UVQ, the proposed algorithm not only has the lowest queuing delay, but also guarantees queuing delay constraints. The reasons are three folds. First, virtual deficit queue backlogs corresponding to queuing delay constraints are considered to adjust optimization objective. Second, the virtual deficit queue backlog corresponding to constraint of extreme event occurrence is also considered to avoid extreme large queue backlog. Finally, exploration weight is rapidly improved to reduce convergence duration and queuing delay fluctuation caused by burst electromagnetic interference.

Figure 9 shows the tradeoff between the total energy consumption and the total queuing delay under different weight values of $V_E$. It is shown that as the weight increases from $1 \times 10^3$ to $3 \times 10^3$, the total energy consumption is reduced by 39.86% at the cost of increased total queuing delay, i.e., an increment of 29.34%. Thus, by adjusting the weight value, the proposed algorithm can effectively quantifies the complex tradeoff between energy consumption and queuing delay, which sheds insight into network design. Therefore, the value of $V_E$ should be carefully determined following multi-services of distribution grid to meet practical needs.

Figure 10 shows the average weighted sum of energy consumption and queuing delay under different sizes of sliding time window $F$. The figure shows that when $t$=100, the proposed algorithm with window size $F$=4 has the best performance. The average weighted sum is 8.91% and 4.87% lower than those of $F$=1 and $F$=8, respectively. This is because a too small window size is insufficient to explore historical performance, while a too large window size incorporates numerous outdated data which significantly reduces estimation accuracy of matching preferences.

## 6. Conclusion

We presented a cloud-edge-end collaborative multi-timescale multi-service resource management algorithm to realize low-delay and low-energy consumption resource management. Specifically, the sliding window pricing matching was proposed to solve large-timescale device scheduling, and the disturbance-robust UCB and greedy algorithm was developed to solve small-timescale compression level selection, power control, and computing resource allocation. Compared to DCSL and UVQ, the proposed algorithm reduces the average weighted sum of total energy consumption and queuing delay by 6.96% and 5.05%. Moreover, simulation results demonstrate that it can adaptively achieve service priority awareness and reduce performance fluctuation caused by burst electromagnetic interference. Future work will concentrate on the joint optimization of edge-side multi-dimensional resources in multi-service scenarios with different delay requirements.

## Acknowledgments

This work was supported by National key research and development program of China "Key Technologies for Coordination and Interoperation of Power Distribution Service Resource" (2021YFB2401300).

### References

[1] H. Liao, Z. Zhou, Z. Jia, Y. Shu, M. Tariq, J. Rodriguez, and V. Frascolla, "Ultra-low AoI digital twin-assisted resource allocation for multi-mode power IoT in distribution grid energy management," IEEE J. Sel. Areas Commun., vol.41, no.10, pp.3122–3132, 2023.

[2] S.S. Shinde and D. Tarchi, "Collaborative reinforcement learning for multi-service internet of vehicles," IEEE Internet Things J., vol.10, no.3, pp.2589–2602, 2023.

[3] H. Huang, Y. Xue, J. Wu, Y. Tao, and M. Hu, "Temporal computing resource allocation scheme with end device assistance," IEEE Internet Things J., vol.9, no.18, pp.16884–16896, 2022.

[4] S. Zhang, Z. Yao, H. Liao, Z. Zhou, Y. Chen, and Z. You, "Endogenous security-aware resource management for digital twin and 6G edge intelligence integrated smart park," China Commun., vol.20, no.2, pp.46–60, 2023.

[5] Z. Hu, C. Fang, Z. Wang, S.-M. Tseng, and M. Dong, "Many-objective optimization-based content popularity prediction for cache-assisted cloud–edge–end collaborative IoT networks," IEEE Internet Things J., vol.11, no.1, pp.1190–1200, 2024.

[6] C. Fang, X. Meng, Z. Hu, F. Xu, D. Zeng, M. Dong, and W. Ni, "AI-driven energy-efficient content task offloading in cloud-edge-end cooperation networks," IEEE Open J. Comput. Soc., vol.3, pp.162–171, 2022.

[7] X. Yang, X. Guan, N. Wang, Y. Liu, H. Wu, and Y. Zhang, "Cloud-edge-end intelligence for fault-tolerant renewable energy accommodation in smart grid," IEEE Trans. Cloud Comput., vol.11, no.2, pp.1144–1156, 2023.

[8] W. Fan, X. Liu, H. Yuan, N. Li, and Y. Liu, "Time-slotted task offloading and resource allocation for cloud-edge-end cooperative computing networks," IEEE Trans. Mobile Comput., pp.1–16, 2024.

[9] G. Shuai, Y. Xiaoyu, C. Wanwan, and Z. Min, "Research on unified access technology of multi-service and multi-scenario power IoT terminals," 2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE), pp.1196–1200, 2023.

[10] Y. Fan, L. Wang, W. Wu, and D. Du, "Cloud/edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach," IEEE Trans. Comput. Social Syst., vol.8, no.2, pp.451–463, 2021.

[11] L. Feng, Q. Yang, K. Kim, and K.S. Kwak, "Two-timescale resource allocation for wireless powered D2D communications with self-interested nodes," IEEE Access, vol.7, pp.10857–10869, 2019.

[12] R. Grootjans and N. Moonen, "Design of cost-effective power quality and EMI sensor for multinode network," IEEE Lett. Electromagn. Compat. Practice Appl., vol.5, no.4, pp.131–136, 2023.

[13] D. Muthirayan, M. Parvania, and P.P. Khargonekar, "Online algorithms for dynamic matching markets in power distribution systems," IEEE Control Syst. Lett., vol.5, no.3, pp.995–1000, 2021.

[14] J. Liu, G. Wu, S. Xiao, X. Zhou, G.Y. Li, S. Guo, and S. Li, "Joint power allocation and user scheduling for device-to-device-enabled heterogeneous networks with non-orthogonal multiple access," IEEE Access, vol.7, pp.62657–62671, 2019.

[15] X. Wang, J. Wang, X. Zhang, X. Chen, and P. Zhou, "Joint task offloading and payment determination for mobile edge computing: A stable matching based approach," IEEE Trans. Veh. Technol., vol.69, no.10, pp.12148–12161, 2020.

[16] J. Zhao, Y. Liu, K.K. Chai, A. Nallanathan, Y. Chen, and Z. Han, "Spectrum allocation and power control for non-orthogonal multiple access in HetNets," IEEE Trans. Wireless Commun., vol.16, no.9, pp.5825–5837, 2017.

[17] M. Rahim, A.S. Alfakeeh, R. Hussain, M.A. Javed, A. Shakeel, Q.U. Hasan, A. Israr, A.O. Alsayed, and S.A. Malik, "Efficient channel allocation using matching theory for QoS provisioning in cognitive radio networks," Sensors, vol.20, no.7, p.1872, 2020.

[18] Z. Wang, Z. Zhou, H. Zhang, G. Zhang, H. Ding, and A. Farouk, "AI-based cloud-edge-device collaboration in 6G space-air-ground integrated power IoT," IEEE Wireless Commun., vol.29, no.1, pp.16–23, 2022.

[19] H. Wu, Y. Chen, L. Wang, and G. Ma, "$\varepsilon$-Ride: An adaptive event-driven windowed matching framework in ridesharing," IEEE Access, vol.10, pp.43799–43811, 2022.

[20] X. Deng, B. Li, X. Li, Z. Wu, and Z. Yang, "Container and microservice-based resource management for distribution station area," 2023 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), pp.578–581, 2023.

[21] P. Qin, Y. Fu, G. Tang, X. Zhao, and S. Geng, "Learning based energy efficient task offloading for vehicular collaborative edge computing," IEEE Trans. Veh. Technol., vol.71, no.8, pp.8398–8413, 2022.

[22] P. Qin, M. Wang, Z. Cai, R. Ding, X. Zhao, Y. Fu, and X. Wu, "Optimal resource allocation for AGIN 6G: A learning-based three-sided matching approach," IEEE Trans. Netw. Sci. Eng., vol.11, no.2, pp.1553–1565, 2024.

[23] R. Zafar, J. Ravishankar, J.E. Fletcher, and H.R. Pota, "Multi-timescale voltage stability-constrained volt/VAR optimization with battery storage system in distribution grids," IEEE Trans. Sustain. Energy, vol.11, no.2, pp.868–878, 2020.

[24] S. Li, X. You, S. Zhang, M. Fang, and P. Zhang, "Cloud-edge-device

collaborative high concurrency access management for massive IoT devices in distribution grid," IEICE Trans. Fundamentals, 2023.

[25] H. Liao, Z. Zhou, N. Liu, Y. Zhang, G. Xu, Z. Wang, and S. Mumtaz, "Cloud-edge-device collaborative reliable and communication-efficient digital twin for low-carbon electrical equipment management," IEEE Trans. Ind. Inform., vol.19, no.2, pp.1715–1724, 2023.

[26] H. Jiang, T. Wang, and S. Wang, "Multi-scale hierarchical resource management for wireless network virtualization," IEEE Trans. Cogn. Commun. Netw., vol.4, no.4, pp.919–928, 2018.

[27] M. Wasim and D.S. Naidu, "Lyapunov function construction using constrained least square optimization," IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, pp.1–5, 2022.

[28] B. Li, Z. Shi, and X. Wang, "Information aggregation and data compression based on PLC for distributed photovoltaic integration," Alexandria Engineering Journal, vol.85, pp.286–293, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016823010311

[29] J. Zhao, Y. Ni, and H. Zhu, "Multi-armed bandit based device scheduling for crowdsensing in power grids," Frontiers in Energy Research, vol.11, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenrg.2023.1141954

**Feng Wang** received his B.S. degree from Sichuan University, China in 2013 and Ph.D. degree from Sichuan University, China in 2017, majoring in power systems and automation. Since July 2017, he has been working in the State Grid Shandong Electric Power Research Institute. He is engaged in research of distribution automation, energy internet, power system stability and control. In recent years, he has undertaken more than 10 national and state grid corporation level scientific and technological projects as the main completion person, and published many academic papers and patents. He has won the first prize of science and technology innovation from the China Electric Power Enterprise Federation, as well as several other awards. He is the deputy secretary-general of Shandong Electrotechnical Society.

**Xiangyu Wen** received his B.S. degree from Qingdao University, China in 2017, and his M.S. degree from University of Chinese Academy of Sciences, China in 2021, majoring in computer application technology. Since August 2021, he has been working in the State Grid Shandong Electric Power Research Institute as an engineer. His main research direction is power distribution Internet of Things and distribution network digitization. In recent years, as the main project completion person, he has undertaken a number of national and State Grid Corporation level science and technology projects, published many SCI/EI papers and applied for a number of invention patents. Make and revise a number of various standards.

**Lisheng Li** received his B.S. degree from North China Electric Power University, China in 2009, and his M.S. degree from Xi'an Jiaotong University, China in 2013, majoring in power systems and automation. Since July 2018, he has been working in the State Grid Shandong Electric Power Research Institute. He is a senior engineer, a young and middle-aged expert with outstanding contributions in Shandong Province, a standing director of DC Distribution Network Sub-committee of IEEEPES DC Power System technical committee, and a supervisor of Shandong Electrotechnical Society. His research interests include smart distribution network, distribution automation, distribution network operation and maintenance, and distributed energy grid connection technology. He has undertaken 30 science and technology projects of state grid Corporation and provincial company. He has won the first prize of Science and Technology Progress of State Grid Corporation and Shandong Province, obtained more than 10 authorized invention patents in the field of distribution network and distributed power generation, and published more than 20 papers in core journals and above.

**Yan Wen** is a master candidate and a senior engineer. She is currently the Party secretary and vice president of Economic and Technical Research Institute of State Grid Shandong Electric Power Company. She is mainly engaged in the research of power distribution Internet of Things, distributed power consumption and microgrid, and power big data analysis. She has been awarded the honorary titles of Professional Leading Talent of Shandong Electric Power Company and Top Ten Outstanding Youth of Shandong Electric Power Company. As the main completion person of the project, she has undertaken a number of national and state grid corporation level science and technology projects, and published a number of academic papers and patents. She has won 1 first prize of Beijing Science and Technology Award, 1 second prize of Shandong Electric Power Science and Technology Award, 1 third prize of China Electric Power Science and Technology Award, 1 second prize of Liaoning Science and Technology Progress Award, etc.

**Shidong Zhang** received his B.S. degree from China University of Petroleum, China in 2006, and his Ph.D. degree from Beijing University of Posts and Telecommunications, China in 2012, majoring in computer science and technology. From July 2012 to present, He has been working in the State Grid Shandong Electric Power Research Institute. He is a senior engineer, a master's enterprise supervisor of the School of Electrical Engineering of Shandong University, and a member of the Standardization Committee of Shandong Electric Power Company. He is mainly engaged in the research of smart distribution network technology, power Internet of Things and other technical directions. He has published more than 20 academic papers, more than 10 of which are included in EI. As the main writer, he has published 2 monographs and obtained more than 10 authorized patents, and won 1 second prize of Shandong Province Science and Technology Progress Award, 1 third prize of China Electric Power Science and Technology Progress Award, 1 third prize of State Grid Patent award, 3 second prize and 1 third prize of Shandong Province Electric Power Science and Technology Progress Award, and 3 second prize of Shandong Province Electric Power Staff technical Achievements award.

**Yang Liu** received his Ph.D. in Electrical Engineering from Huazhong University of Science and Technology in 2016. Since August 2016, he has been working in the State Grid Shandong Electric Power Research Institute, where he currently serves as the head of the Distribution Network Technology Department and a senior engineer. He is a board member of the DC Distribution Network Sub-Committee of the IEEE PES DC Power System Technology Committee. His primary research areas include distribution network operation control, distribution energy internet, and energy storage applications. He has led over 10 national and State Grid-level technology projects in these fields, published over 30 papers, obtained more than 20 authorized invention patents, and participated in the formulation of 5 national standards, group standards, and enterprise standards.