Learning-Based Context-Aware Resource Allocation for Edge Computing-Empowered Industrial IoT

Haijun Liao, *Student Member, IEEE,* Zhenyu Zhou, *Senior Member, IEEE,* Xiongwen Zhao, *Senior Member, IEEE,* Lei Zhang, Shahid Mumtaz, *Senior Member, IEEE,* Alireza Jolfaei,
Syed Hassan Ahmed, *Member, IEEE,* and Ali Kashif Bashir, *Senior Member, IEEE*

***Abstract*—Edge computing provides a promising paradigm to support the implementation of industrial Internet of Things (IIoT) by offloading computational-intensive tasks from resource-limited machine-type devices (MTDs) to powerful edge servers. However, the performance gain of edge computing may be severely compromised due to limited spectrum resources, capacity-constrained batteries, and context unawareness. In this paper, we consider the optimization of channel selection which is critical for efficient and reliable task delivery. We aim at maximizing the long-term throughput subject to long-term constraints of energy budget and service reliability. We propose a learning-based channel selection framework with service reliability awareness, energy awareness, backlog awareness, and conflict awareness, by leveraging the combined power of machine learning, Lyapunov optimization, and matching theory. We provide rigorous theoretical analysis, and prove that the proposed framework can achieve guaranteed performance with a bounded deviation from the optimal performance with global state information (GSI) based on only local and causal information. Finally, simulations are conducted under both single-MTD and multi-MTD scenarios to verify the effectiveness and reliability of the proposed framework.**

***Index Terms*—Industrial Internet of Things (IIoT), resource allocation, context awareness, edge computing, machine learning, Lyapunov optimization, matching theory.**

I. Introduction

THE fourth industrial revolution aims to realize interconnected, responsive, intelligent and self-optimizing manufacturing processes and systems through seamless integration of advanced manufacturing techniques with industrial Internet of Things (IIoT) [1]. In this new paradigm, billions of machine-type devices (MTDs) will be deployed in the field for continuously performing various tasks such as monitoring, billing, and protection [2], [3]. Nevertheless, the tension between resource-limited MTDs and computational-intensive tasks has become the bottleneck for reliable service provisioning [4].

Offloading computational-intensive tasks from resource-limited MTDs to powerful servers provides a promising solution for accommodating the fast-growing computational demands. In conventional cloud computing, the remote cloud servers are generally located far away from MTDs, and the long-distance data transmission raises numerous issues including unstable connection, network congestion, and unbearable latency [5]. In comparison, edge computing [6], which shifts the computational capabilities from remote clouds to network edges within radio access network (RAN) [7], is a promising paradigm to reduce latency, relieve congestion, and prolong battery lifetime. It has attracted intensive research efforts from both industry and academia. In [8], Fan *et. al* considered the workload balancing problem in fog computing, and proposed a distributed device association algorithm to minimize the communication latency and the computational latency. They also extended their work to drone-assisted communication networks for IoT [9]. Markakis *et. al* developed a multi-access edge computing based IoT framework for supporting next-generation emergency services, and provided several use cases of remote healthcare monitoring and management [10]. Omoniwa *et. al* proposed an edge computing-based IoT framework to enhance smart grid with improved scalability, security, response and less system cost [11].

Unfortunately, although edge computing provides a promising way to exploit the abundant computational resources of edge servers, its performance gain may be severely compromised due to limited spectrum resources, capacity-constrained batteries, and context unawareness. First, to deliver a large volume of tasks from MTDs to the edge server on a real-time basis, channel selection has to be dynamically optimized in accordance with time-varying context parameters such as channel state information (CSI), energy state information

Manuscript received August 1, 2019; revised September 10, 2019 and October 31, 2018; accepted December 15, 2019; current version December 30, 2019. This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant Number 61971189; the Science and Technology Project of State Grid Corporation of China under Grant Number SGSDDK00KJJS1900405; the Exploration Project of State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources (North China Electric Power University) under Grant Number LAPS2019-12; the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization (COMPETE 2020), Regional Operational Program of the Agarve (2020), Fundao para a ciłncia e Tecnologia, i-Five: Extenso do acesso de espectro dinmico para rdio 5G, POCI-01-0145-FEDER-030500. *(Corresponding author: Z. Zhou)*.

H. Liao, Z. Zhou and X. Zhao are with State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources (North China Electric Power University), and School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China (e-mail: haijun_liao@ncepu.edu.cn, zhenyu_zhou@ncepu.edu.cn, zhaoxw@ncepu.edu.cn).

L. Zhang is with Shandong Electric Power Research Institute for State Grid Corporation of China, Jinan, China (e-mail: 18660130685@163.com).

S. Mumtaz is with the The Instituto de Telecomunicações,1049-001, Aveiro, Portugal (e-mail: smumtaz@av.it.pt).

A. Jolfaei is with the Department of Computing, Macquarie University, Sydney NSW 2113, Australia (e-mail: alireza.jolfaei@mq.edu.au).

S. H. Ahmed is with the Department of Electrical and Computer Science, Georgia Southern University, Statesboro, GA 30460, USA (e-mail: sh.ahmed@ieee.org).

A. K. Bashir is with the Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, U.K (e-mail: dr.alikashif.b@ieee.org).

(ESI), server load, and service reliability requirement. Conventional centralized optimization approaches [12], [13], rely on a common presumption that there exists a central node, e.g., the base station (BS), which has the perfect knowledge of all the context parameters. This presumption is too optimistic in real-world implementation considering the prohibitive cost of signaling overhead to collect information of the entire network. Therefore, a distributed optimization approach where each MTD individually optimizes its channel selection strategy based on only local information is more desirable. However, when the number of MTDs far more exceeds that of available channels, selection conflict will occur frequently if multiple MTDs compete for the same channel, thus making the strategies of channel selection coupled across different MTDs. Second, given the limited battery capacity, a MTD will be out of service when the battery energy is exhausted. As a result, the short-term channel selection strategy also couples the long-term energy budget. Last but not least, industrial applications often require that certain service reliability should be guaranteed [14]. How to meet the stringent reliability requirement with limited resources and information brings another dimension of difficulty.

Matching theory provides a flexible, low-complexity, and efficient tool to solve the combinatorial problem such as channel selection [15], task selection [16], and server selection [17]. However, it requires perfect knowledge of global state information (GSI) to construct the preference list, which specifies the fundamental matching criteria [18]. There exist some research attempts which study the optimization of channel selection based on matching and game theory [19], [20]. However, they rely on the assumption that the uncertain context parameters follow some well-known probability distribution, and may suffer from severe performance loss if the practical probability distributions of uncertain factors disagree from the presumed statistical models.

In this paper, we propose a learning-based context-aware channel selection framework by combining machine learning, Lyapunov optimization, and matching theory. Specifically, we adopt the upper confidence bound (UCB) algorithm [21] to enable a MTD to learn the matching preferences and maximize the long-term optimality performance while maintaining a well-balanced tradeoff between exploitation and exploration. UCB was originally developed to solve the multi-armed bandit (MAB) problem [22], which involves sequential decision making based on only local information. It was designed for the single-player scenario and thereby inevitably leading to selection conflicts in the multi-player scenario where multiple MTDs are prone to select the same channel [23].

We aim at maximizing the long-term network throughput subject to long-term constraints of energy budget and service reliability. The stochastic optimization problem is converted to a series of short-term deterministic problems by leveraging Lyapunov optimization [14]. We start from the simplified single-MTD scenario with perfect GSI, and propose a Service-reliability-aware, Energy-aware, and data-Backlog-aware GSI (SEB-GSI) algorithm for channel selection. Then, we extend SEB-GSI to the nonideal case with only local information, and develop a UCB-based channel selection algorithm named SEB-UCB. It enables the MTD to dynamically balance throughput, energy consumption, and service reliability via online learning. Next, for the multi-MTD scenario with GSI, we formulate the optimization problem of channel selection as a one-to-one matching between MTDs and channels, and propose a matching-based solution named SEB-Matching GSI (SEB-MGSI). Afterwards, we emphasize the multi-MTD scenario with only local information, and develop a matching-learning based context-aware channel selection algorithm named SEB Conflict-aware MUCB (SEBC-MUCB), in which each MTD makes decision and learns the selection conflicts by continuously observing the relationship between matching preferences and matching results.

The main contributions are summarized as follows:

- *Learning-based channel selection:* We propose a learning-based channel selection framework by leveraging the combined power of UCB, Lyapunov optimization and matching theory. It can learn the long-term optimal strategy and achieve guaranteed performance with a bounded deviation while the long-term constraints of energy budget and service reliability are satisfied in a best effort way based on only local and causal information.
- *Context awareness:* The proposed framework can achieve service reliability awareness, energy awareness, and backlog awareness by dynamically adjusting the exploitation weights in accordance with the performances of throughput, energy consumption and service reliability. It can also achieve conflict awareness by continuously learning the difference between matching preference and actual matching result.
- *Multiple deployment scenarios and information availability cases:* The simplified single-MTD scenario is firstly studied to provide some insight. Then, the more complicated multi-MTD scenario where selection conflicts exist is investigated. For both the single-MTD and the multi-MTD scenarios, the ideal case with perfect GSI is firstly studied as the performance benchmark. Then, the analysis is extended to the nonideal case with only local information where learning is considered.
- *Rigorous theoretical analysis and extensive performance evaluation:* We analyze the optimality performance of the proposed framework from the perspective of network throughput and learning regret. We also provide a comprehensive analysis of computational complexity. Extensive simulations are carried out to validate its effectiveness and reliability under various scenarios and parameter settings.

The remaining parts of this paper are organized as follows. The system model and the problem formulation are introduced in Section II. Section III and Section IV describe the learning-based context-aware channel selection for the single-MTD scenario and the multi-MTD scenario, respectively. A performance analysis from the perspective of optimality and complexity is given Section V. Practical implementation considerations and simulation results are provided in Section VI and Section VII. Section VIII concludes this paper.

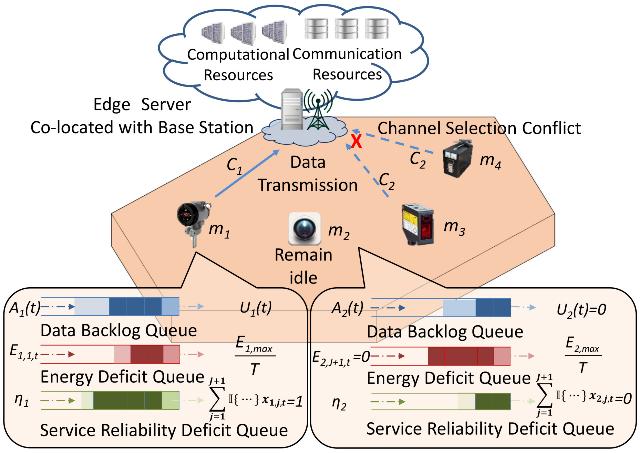


Fig. 1. System model.

II. System Model and Problem Formulation

In this section, the system model and problem formulation are introduced.

A. System Model

As shown in Fig. 1, we consider a single-cell scenario where an edge server is collocated with a BS. The BS provides connection service and the edge server provides computing service for K MTDs within the cell, the set of which is denoted by $\mathcal{M} = \{m_1, \cdots, m_k, \cdots, m_K\}$. There exist J orthogonal subchannels, the set of which is defined as $\mathcal{C} = \{c_1, \cdots, c_j, \cdots, c_J\}$. The bandwidth of subchannel c_j is denoted by B_j. Channel selection conflict occurs when more than one MTDs select the same subchannel at the same time, and only one MTD can succeed to access the subchannel under the coordination of the BS.

A time-slotted model is adopted where the total optimization period is divided into T slots with equal length τ, the set of which is denoted by $\mathcal{T} = \{1, \cdots, t, \cdots, T\}$. In this model, CSI remains unchanged within a slot and varies across different slots. In each slot, each MTD determines its channel selection strategy individually. Particularly, a MTD faces $J+1$ options, i.e., either selecting one of the J subchannels or remaining idle. Fig. 1 shows an example of channel selection with 4 MTDs and 2 subchannels. m_1 selects subchannel c_1 for data transmission while m_2 remains idle. Channel selection conflict occurs between m_3 and m_4 due to the simultaneous selection of subchannel c_2.

In the following, the models of task transmission, energy consumption, delay, and service reliability are introduced.

1) Task Transmission Model: In the t-th slot, $A_k(t)$ new tasks with equal size γ_k arrive at $m_k \in \mathcal{M}$, which are firstly stored in the local buffer and then are transmitted to the edge server. Hence, the total task size is $\gamma_k A_k(t)$. Meanwhile, it has to retransmit $Y_k(t)$ amount of data, which have not been correctly delivered due to bit error. The task data stored in the local buffer of m_k can be modeled as a queue, i.e., queue k. $\gamma_k A_k(t)$ as well as $Y_k(t)$ can be seen as the amount of task data entering the queue and $U_k(t)$ represents the amount of task data leaving the queue. Define $Q_k(1)$ as the initial amount of data backlog. $Q_k(t)$ is the backlog of data queue k in the t-th slot, i.e., an accumulation of data that are yet to be processed. $Q_k(t)$ is dynamically evolved as

$$Q_k(t+1) = \max\{Q_k(t) - U_k(t), 0\} + \gamma_k A_k(t) + Y_k(t+1). \tag{1}$$

The set of channel selection indicators consists of $J+1$ binary elements, which is denoted by $\{x_{k,j,t}\}$, where $x_{k,j,t} \in \{0,1\}$. When $j = 1, 2, \cdots, J$, $x_{k,j,t} = 1$ represents that m_k selects subchannel c_j for data transmission in the t-th slot and when $j = J+1$, $x_{k,j,t} = 1$ represents that m_k remains idle.

Considering the powerful computational capability of the edge server, the objective of each MTD is to offload as many tasks as possible, which equals to maximizing the total amount of task data that can be transmitted, i.e., the throughput.

Uplink transmission is considered here. Denote $H_{k,j,t}$ as the uplink channel gain of subchannel c_j between m_k and the BS. Given $x_{k,j,t}$, the achievable uplink transmission rate is given by

$$R_{k,j,t} = \begin{cases} B_j \log_2(1 + \frac{P_{\text{TX}} H_{k,j,t}}{\delta^2}), & j = 1, 2, \cdots, J \\ 0, & j = J+1 \end{cases}, \tag{2}$$

where δ^2 is the noise power, and P_{TX} is the transmission power. The throughput of m_k in the t-th slot is given by

$$z_{k,j,t} = \min\{Q_k(t), \tau R_{k,j,t}\}. \tag{3}$$

The amount of data transmitted to the edge server can be

$$U_k(t) = \sum_{j=1}^{J+1} x_{k,j,t} z_{k,j,t}. \tag{4}$$

Denote the bit error rate (BER) for m_k transmitting data through subchannel c_j in the t-th slot as $P^e_{k,j,t}$. We consider the noncoherent binary phase shift keying (BPSK) modulation and the corresponding BER [24] of it can be derived as

$$P^e_{k,j,t} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{P_{\text{TX}} H_{k,j,t}}{\delta^2}}\right). \tag{5}$$

Here, BPSK is just used as an example to derive the queue evolution model, which can be naturally extended to other modulation schemes such as quadrature amplitude modulation (QAM) and orthogonal frequency division multiplexing (OFDM).

Therefore, $Y_k(t+1)$, the amount of data that has to be retransmitted in the next slot can be calculated as

$$Y_k(t+1) = U_k(t) P^e_{k,j,t}. \tag{6}$$

2) Energy Consumption Model: In the t-th slot, the energy consumption of m_k for data transmission is the transmission power multiplied by the transmission delay, i.e.,

$$E_{k,j,t} = \begin{cases} P_{\text{TX}} \min\{\frac{Q_k(t)}{R_{k,j,t}}, \tau\}, & j = 1, 2, \cdots, J. \\ 0, & j = J+1. \end{cases} \tag{7}$$

The limited battery capacity exerts a direct impact on the total energy budget of m_k over T slots, which is denoted by

$E_{k,max}$. Therefore, the long-term energy consumption of m_k must satisfy

$$E_k = \sum_{t=1}^{T} \sum_{j=1}^{J+1} x_{k,j,t} E_{k,j,t} \leq E_{k,max}. \tag{8}$$

3) Delay Model: In IIoT, the data size of computational results is generally smaller than that of the computational tasks. Therefore, for the sake of simplicity, we can neglect the downlink transmission delay. Some previous works, e.g., [25]–[27], also ignore the downlink transmission time. On the other hand, our work can be easily extended to the scenario where the downlink transmission time is considered. Therefore, the total offloading delay is the sum of the transmission delay and computational delay, which can be given by

$$d_{k,j,t}^{total} = d_{k,j,t}^{tra} + d_{k,j,t}^{com}. \tag{9}$$

Given $x_{k,j,t}$ and $z_{k,j,t}$, the transmission delay is calculated by dividing throughput $z_{k,j,t}$ with transmission rate $R_{k,j,t}$, i.e.,

$$d_{k,j,t}^{tra} = \begin{cases} \frac{z_{k,j,t}}{R_{k,j,t}} = \min\{\frac{Q_k(t)}{R_{k,j,t}}, \tau\}, & j = 1, 2, \cdots, J. \\ +\infty, & j = J+1. \end{cases} \tag{10}$$

Based on the computational intensity model in [28], assuming that the computational intensity of the task data transmitted by m_k in the t-th slot is $\lambda_{k,t}$ (CPU cycles/bit), it requires $z_{k,j,t}\lambda_{k,t}$ CPU cycles to process the task data. It is noted that although a linear relationship between workload and data size is employed, our work is compatible with other nonlinear models and can be used for different kinds of IIoT applications with different computing intensities. Denoting the available computational resources for m_k in the t-th slot as $\xi_{k,t}$, the computational delay is calculated as

$$d_{k,j,t}^{com} = \begin{cases} \frac{z_{k,j,t}\lambda_{k,t}}{\xi_{k,t}}, & j = 1, 2, \cdots, J. \\ +\infty, & j = J+1. \end{cases} \tag{11}$$

4) Service Reliability Requirement Model: We model the service reliability requirement in terms of delay. Denoting the task delay requirement as $d_{k,t}$, the task offloading is unsuccessful if the offloaded task cannot be processed within the specified delay requirement, i.e., $d_{k,j,t}^{total} > d_{k,t}$. Denote $X_{k,T}$ as the number of successful task offloading for m_k over T slots, which is given by

$$X_{k,T} = \sum_{t=1}^{T} \sum_{j=1}^{J+1} \mathbb{I}\{d_{k,j,t}^{total} \leq d_{k,t}\} x_{k,j,t}. \tag{12}$$

$\mathbb{I}\{x\}$ is an indicator function with $\mathbb{I}\{x\} = 1$ if event x is true and $\mathbb{I}\{x\} = 0$ otherwise. The edge server performs computational resource optimization at the end of each slot and feeds back the result of whether the delay requirement of m_k can be satisfied or not.

The service reliability requirement is defined as

$$\frac{X_{k,T}}{T} \geq \eta_k, \tag{13}$$

where $\eta_k \in (0, 1]$ represents the minimum successful probability of task offloading.

B. Problem Formulation

The objective is to maximize the long-term network throughput under the long-term constraints of energy budget and service reliability. Therefore, network throughput maximization problem is formulated as

$$\begin{aligned} \mathbf{P1}: & \max_{\{x_{k,j,t}\}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{j=1}^{J+1} x_{k,j,t} z_{k,j,t}, \\ \text{s.t. } & C_1: \sum_{k=1}^{K} x_{k,j,t} \leq 1, j = 1, 2, \cdots, J, \forall t \in \mathcal{T}, \\ & C_2: \sum_{j=1}^{J+1} x_{k,j,t} \leq 1, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{T}, \\ & C_3: \sum_{t=1}^{T} \sum_{j=1}^{J+1} x_{k,j,t} E_{k,j,t} \leq E_{k,max}, \forall m_k \in \mathcal{M}, \\ & C_4: \frac{X_{k,T}}{T} \geq \eta_k, \forall m_k \in \mathcal{M}, \end{aligned} \tag{14}$$

where C_1 and C_2 are the channel selection constraints, i.e., in each slot, each subchannel can be selected by at most one MTD, and each MTD can select only one subchannel at most or remains idle. C_3 and C_4 correspond to the constraints of energy consumption and service reliability, respectively. Here, we focus on optimizing channel selection strategy while the optimization of computational resource allocation is left to the future work. The reason is that the proposed algorithm is naturally compatible with any computational resource allocation scheme. Similarly, some previous works also only consider the channel selection problem [28]–[30]. On the other hand, the joint optimization of channel selection and computational resource allocation is a completely different problem, which requires different system modeling, problem formulation, and optimization design. Utilizing learning algorithms to solve the joint optimization problem of integer channel selection and continuous computational resource allocation is also a worthwhile research direction which will be investigated in the future work.

III. Learning-Based Context-Aware Channel Selection for the Single-MTD Scenario

In this section, we consider the single-MTD scenario with only one MTD, e.g., m_k, and propose a learning-based context-aware channel selection algorithm.

A. Problem Transformation

Problem $\mathbf{P1}$ cannot be directly solved due to the long-term optimization objective and constraints. To provide a tractable solution, we leverage Lyapunov optimization to transform a coupled long-term stochastic optimization problem into a series of short-term deterministic problems [31], [32], which can be solved in low complexity while the data backlog, energy consumption, and service reliability are balanced over time.

Based on the concept of virtual queue [33], the long-term energy budget and service reliability constraints, i.e., C_3 and C_4, can be transformed to queue stability constraints. We define a virtual energy deficit queue $N_k(t)$ and a virtual service reliability deficit queue $F_k(t)$, which are evolved as

$$N_k(t+1) = \max\{N_k(t) + \sum_{j=1}^{J+1} x_{k,j,t} E_{k,j,t} - \frac{E_{k,max}}{T}, 0\},$$
$$F_k(t+1) = \max\{F_k(t) + \eta_k - \sum_{j=1}^{J+1} \mathbb{I}\{d_{k,j,t}^{total} \leq d_{k,t}\} x_{k,j,t}, 0\}, \tag{15}$$

with $N_k(1) = F_k(1) = 0$. $N_k(t)$ represents the deviation of current energy consumption from the energy budget, while $F_k(t)$ reflects the deviation of service reliability from the specified requirement. Examples of queue evolution for MTDs are shown in Fig. 1. Taking m_1 as an example, the data queue $Q_1(t)$, the virtual energy deficit queue $N_1(t)$, and the virtual service reliability deficit queue $F_1(t)$ are dynamically updated at each slot based on (1) and (15). Comparing m_1 and m_2, it is noted that the data backlog and the service reliability deficit of m_1 are larger while the energy deficit of m_2 is larger.

Then, $\mathbf{P1}$ can be transformed into a series of short-term optimization subproblems. At each slot, if the energy consumption of m_k until the t-th slot does not exceed the energy budget, an online multi-objective optimization problem is defined to maximize throughput and service reliability while minimizing energy consumption, which is given by

$$\mathbf{P2}: \min_{\{x_{k,j,t}\}} \sum_{k=1}^{K} \sum_{j=1}^{J+1} [-V_k z_{k,j,t} + \alpha_k N_k(t) E_{k,j,t} - \beta_k F_k(t) (\sum_{j=1}^{J+1} \mathbb{I}\{d_{k,j,t}^{total} \leq d_{k,t}\} x_{k,j,t} - \eta_k)],$$
$$\text{s.t. } C_1 \sim C_2. \tag{16}$$

For convenience, we write $\theta_{k,j,t} = -V_k z_{k,j,t} + \alpha_k N_k(t) E_{k,j,t} - \beta_k F_k(t)(\sum_{j=1}^{J+1} \mathbb{I}\{d_{k,j,t}^{total} \leq d_{k,t}\} x_{k,j,t} - \eta_k)$. Here, $\theta_{k,j,t}$ is a weighted sum of throughput, energy consumption and service reliability, where V_k, $\alpha_k N_k(t)$, and $\beta_k F_k(t)$ are the corresponding weights.

$\mathbf{P2}$ and $\mathbf{P1}$ are not equal, and the results of $\mathbf{P2}$ may not be feasible for $\mathbf{P1}$. Nevertheless, we can prove that the results of $\mathbf{P2}$ are within a bounded deviation from the optimal results in Section V. Furthermore, C_3 can be guaranteed by defining that if the energy budget of m_k is exhausted, then it cannot transmit data and is forced to remain idle. In other words, at the t-th slot, $\mathbf{P2}$ will be solved if and only if the energy budget is not exhausted. On the other hand, C_4 is satisfied in a best effort way due to service reliability awareness, i.e., a large deviation from the service reliability requirement will enforces m_k to select the option with higher successful chances of task offloading, thereby trying the best to satisfy C_4. It is noted that C_4 cannot be 100% guaranteed due to the lack of centralized optimization and coordination among all the MTDs.

The local information is referred as the information that can be possessed by m_k without additional information exchange with other entities in the network, e.g., the BS or the other MTDs. The nonlocal information refers to the information that can only be possessed by m_k with additional information exchange. Otherwise, if information exchange is infeasible, nonlocal information is unknown to m_k. Therefore, the information required to solve $\mathbf{P2}$ can be classified into two categories, i.e.,

Algorithm 1 SEB-GSI

```
1:  Input: $V_k$, $\alpha_k$, $\beta_k$.
2:  **Phase 1:** Initialization
3:  Set $Q_k(1)$ as the initial amount of data backlog, $N_k(1) = 0$, $F_k(1) = 0$, $x_{k,j,t} = 0$, $j = 1, 2, \cdots, J+1, \forall t \in \mathcal{T}$.
4:  **Repeat**
5:  **Phase 2:** Decision making
6:  Input: $H_{k,j,t}$, $\delta^2$.
7:  Calculate the accurate value of $\theta_{k,j,t}$ with GSI, $j = 1, 2, \cdots, J+1$ .
8:  Choose $j$ by solving $\mathbf{P2}$.
9:  Observe $z_{k,j,t}$, $E_{k,j,t}$ and whether the delay requirement can be satisfied or not.
10: Update $U_k(t)$ and $Y_k(t+1)$ based on (4) and (6).
11: Update $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (1) and (15).
12: **Until** $t > T$.
```

- **Local Information:** information that can be possessed by m_k without additional information exchange, e.g., the queue backlog $Q_k(t)$, the transmission power P_{TX}, the total energy budget $E_{k,max}$, the computational intensity of task data $\lambda_{k,t}$, the task delay requirement $d_{k,t}$, and the service reliability requirement η_k.
- **Nonlocal Information:** information that cannot be possessed by m_k without additional information exchange, e.g., the uplink channel gain $H_{k,j,t}$ for any subchannel $c_j \in \mathcal{C}$, the available computational resources of the edge server $\xi_{k,t}$, and the channel selection strategies of other MTDs $\{x_{k,j,t}\}$ (only required for the multi-MTD scenario).

For the local information, the time-varying information is denoted by the symbol with subscript t or as a function of t, e.g., $Q_k(t)$, $\lambda_{k,t}$, and $d_{k,t}$. Otherwise, the local information is fixed, e.g., P_{TX}, $E_{k,max}$, and η_k. The nonlocal information is expressed in the same way and all the nonlocal information is time-varying.

Based on whether m_k has the nonlocal information or not, we consider an ideal and nonideal case, respectively. In the ideal case, m_k has the perfect knowledge of GSI, which includes both local and nonlocal information. In the nonideal case, m_k only knows the local information while the nonlocal information is unavailable.

B. The SEB-GSI Algorithm for the Ideal Case

For the ideal case with GSI, we propose a context-aware channel selection algorithm named SEB-GSI with service reliability awareness, energy awareness and backlog awareness. SEB-GSI does not require future non-causal information. The detailed procedures are summarized in Algorithm 1, which consists of two phases, i.e., initialization (Line 2 $\sim$ 3) and decision making (Line 5 $\sim$ 9). Algorithm 1 is provided to demonstrate how to initialize queues, determine the optimal option, and update queues.

In the initialization phase, the initial length of all the queues and initial values of all the selection indicators are set as zero.

Then, the decision making phase is executed in a slot-by-slot fashion. At the beginning of the t-th slot, m_k calculates the value of $\theta_{k,j,t}$ towards option j, $j = 1, 2, \cdots, J+1$, based on the current GSI. The optimum option j can be found by solving $\mathbf{P2}$, which is equivalent to a minimum seeking problem with computational complexity $\mathcal{O}(J)$. Afterwards, m_k sets $x_{k,j,t} = 1$, and updates all queues accordingly. In the next slot, the iteration continues until $t > T$.

The proposed SEB-UCB can adapt to the variations of the amount of data backlog, energy state and the service reliability state due to the endowed context awareness, which is achieved through the dynamic adjustment of channel selection strategy based on the values of $F_k(t)$, $N_k(t)$, and $Q_k(t)$. Details are given as follows:

- **Service reliability awareness:** When the service reliability deviates severely from the service reliability requirement, a large weight $F_k(t)$ will be placed on the service reliability term which enforces m_k to select the option with higher successful chances of task offloading, thereby enabling service reliability awareness.
- **Energy awareness:** When the energy consumption significantly exceeds the current energy budget, a large weight $N_k(t)$ on the energy consumption term will enforce m_k to select the option with less consumption, i.e., remaining idle, thereby enabling energy awareness.
- **Backlog awareness:** A large data backlog $Q_k(t)$ will lead to a large throughput $z_{k,j,t} = \tau R_{k,j,t}$ based on (3), which motivates m_k to choose the subchannel with higher data transmission rate, thereby enabling backlog awareness.

Since $F_k(t)$, $N_k(t)$, and $Q_k(t)$ are updated without requiring future information, SEB-GSI optimizes the balance among throughput performance, energy consumption, and service reliability requirement in an online fashion.

C. The SEB-UCB Algorithm for the Nonideal Case

In the nonideal case where the nonlocal information is unavailable, the proposed SEB-GSI algorithm is infeasible because the accurate value of $\theta_{k,j,t}$ cannot be obtained. To tackle this problem, we modify SEB-GSI based on the UCB1 framework [21], which is a low-complexity learning-based algorithm to deal with the sequential decision-making problem, and develop a learning-based context-aware channel selection algorithm named SEB-UCB. Instead of directly calculating $\theta_{k,j,t}$ in SEB-GSI, SEB-UCB estimates $\theta_{k,j,t}$ based on historical observations while simultaneously taking into account the uncertainty of estimation via confidence bound. It enables m_k to learn the optimal option based only on local information and achieve a bounded deviation from the optimal performance obtained with GSI.

The proposed SEB-UCB algorithm is summarized in Algorithm 2. In each time slot, m_k makes decisions based on only two kinds of local information: $\bar{\theta}_{k,j,t-1}$ and $\hat{x}_{k,j,t-1}$, where $\bar{\theta}_{k,j,t-1}$ represents the empirical estimation of $\theta_{k,j,t-1}$ up to slot t, and $\hat{x}_{k,j,t-1}$ represents the number of times that m_k has selected the j-th option up to slot t. The estimation of m_k towards the option j in the t-th slot is estimated as

Algorithm 2 SEB-UCB

```
Input: $V_k$, $\alpha_k$, $\beta_k$, $\omega$.
**Phase 1:** Initialization
Set $Q_k(1)$ as the initial amount of data backlog, $N_k(1) = 0$, $F_k(1) = 0$, $\bar{\theta}_{k,j,0} = 0$, $\hat{x}_{k,j,0} = 0$ and $x_{k,j,t} = 0$,
    $j = 1, 2, \cdots, J+1, \forall t \in \mathcal{T}$.
**Repeat**
**Phase 2:** Estimation and decision making
Calculate the estimation value of the MTD towards option $j$ as (17).
Select the optimal option $j$ based on (18).
**Phase 3:** Learning
Observe $z_{k,j,t}$, $E_{k,j,t}$ and whether the delay requirement can be satisfied or not.
Update $\bar{\theta}_{k,j,t}$ and $\hat{x}_{k,j,t}$ based on (19) and (20).
Update $U_k(t)$ and $Y_k(t+1)$ based on (4) and (6).
Update $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (1) and (15).
**Until** $t > T$.
```

$$\widetilde{\theta}_{k,j,t} = \bar{\theta}_{k,j,t-1} - \omega \sqrt{\frac{2 \ln t}{\hat{x}_{k,j,t-1}}}, \tag{17}$$

where the first term represents the empirical performance of the option j, and the second term represents the confidence bound, which is designed to balance the tradeoff between exploration and exploitation. On one hand, the first term pushes m_k to select *a priori known optimal option* up to slot t. On the other hand, the second term is inversely proportional to $\hat{x}_{k,j,t-1}$, which allows m_k to explore options with less number of selections in order to improve the accuracy of estimation. Here, ω is the weight of exploration compared with exploitation, i.e., a larger ω represents a higher preference for exploration.

After estimating $\widetilde{\theta}_{k,j,t}$ for all the $J+1$ options, m_k chooses option j with the least estimation value, which is determined as

$$j = \arg\min_{j} \left\{ \widetilde{\theta}_{k,j,t} \right\}. \tag{18}$$

Then, m_k observes the corresponding results $z_{k,j,t}$, $E_{k,j,t}$ associated with $x_{k,j,t} = 1$ and whether the delay requirement can be satisfied or not. Accordingly, $\bar{\theta}_{k,j,t}$ and $\hat{x}_{k,j,t}$ are updated as

$$\begin{aligned} \bar{\theta}_{k,j,t} = & \frac{\bar{\theta}_{k,j,t-1} \hat{x}_{k,j,t-1}}{\hat{x}_{k,j,t-1} + x_{k,j,t}} \\ & + \frac{-V_k z_{k,j,t} x_{k,j,t}}{\hat{x}_{k,j,t-1} + x_{k,j,t}} + \frac{\alpha_k N_k(t) E_{k,j,t} x_{k,j,t}}{\hat{x}_{k,j,t-1} + x_{k,j,t}} \\ & - \frac{\beta_k F_k(t) (\sum_{j=1}^{J} \mathbb{I}\{d_{k,j,t}^{total} \le d_{k,t}\} x_{k,j,t} - \eta_k) x_{k,j,t}}{\hat{x}_{k,j,t-1} + x_{k,j,t}}, \end{aligned} \tag{19}$$

and

$$\hat{x}_{k,j,t} = \hat{x}_{k,j,t-1} + x_{k,j,t}. \tag{20}$$

Algorithm 3 SEB-MGSI

```
1:  Input: $\{V_k\}$, $\{\alpha_k\}$, $\{\beta_k\}$, $\omega$.
2:  Phase 1: Initialization
3:  Set $Q_k(1)$ as the initial amount of data backlog, $N_k(1) = 0$, $F_k(1) = 0$, $\bar{\theta}_{k,j,0} = 0$, and $x_{k,j,t} = 0, j = 1, 2, \cdots, J+1, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{T}$.
4:  Repeat
5:  Phase 2: Preference list construction
6:  Each MTD calculates its preference value towards each option as (21).
7:  Each MTD constructs its preference list $\mathcal{F}_k$ and any $m_k \in \mathcal{M}_t$ transmits $\mathcal{F}_k$ to the edge server for iterative matching.
8:  Phase 3: Iterative matching
9:  Step 1: Initialization
10: Initialize $\phi = \emptyset$, $\Omega = \emptyset$.
11: Step 2: Pricing-based iterative matching
12: if $\exists \phi(m_k) = \emptyset$ then
13:     any $m_k \in \mathcal{M}_t$ selects its most preferred subchannel in $\mathcal{F}_k$.
14:     if any $c_j \in \mathcal{C}$ selected by only one MTD $m_k$ then
15:         $\phi(m_k) = c_j$.
16:     else
17:         Add $c_j$ into $\Omega$.
18:         for $c_j \in \Omega$ do
19:             $c_j$ raises its price $\rho_{k,j}$ as (22).
20:             All the MTDs selecting $c_j$ update their preferences as (21) and renew their selection strategies.
21:         end for
22:     end if
23: end if
24: Observe $z_{k,j,t}$, $E_{k,j,t}$ and whether the delay requirement can be satisfied or not.
25: Update $U_k(t)$ and $Y_k(t+1)$ based on (4) and (6).
26: Update $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (1) and (15).
27: Until $t > T$.
```

Based on $z_{k,j,t}$, $U_k(t)$ and $Y_k(t+1)$ can be calculated based on (4) and (6). Next, the three queues, i.e., $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$, are updated as (1), (15).

Finally, increase t to $t+1$, and repeat lines 5 ∼ 12 until $t > T$.

IV. Learning-Based Context-Aware Channel Selection for the Multi-MTD Scenario

In this section, we consider channel selection under the multi-MTD scenario, where the channel selection strategies of different MTDs are coupled. Both the SEB-GSI and the SEB-UCB algorithms proposed in the previous section are not suitable for this scenario because the coupling among MTDs are not considered. To tackle this problem, we start from the ideal case with perfect GSI, and develop a matching-based context-aware channel selection algorithm named SEB-MGSI. Next, we consider the more practical nonideal case with only local information, and develop a matching-learning based context-aware channel selection algorithm named SEBC-MUCB.

A. The SEB-MGSI Algorithm for the Ideal Case

When K MTDs are competing for the J subchannels, the channel selection problem involves a one-to-one matching between K MTDs and J subchannels. The definition of matching is given by

Definition 1. (Matching): *Denote ϕ as the one-to-one correspondence from set $\mathcal{M} \cup \mathcal{C}$ onto itself. Specifically, $\phi(m_k) = c_j$ indicates that m_k is matched with subchannel c_j, i.e., $x_{k,j,t} = 1, j = 1, 2, \cdots, J$, and $\phi(m_k) = m_k$ indicates that m_k is not matched with any subchannel and has to remain idle, i.e., $x_{k,J+1,t} = 1$.*

Remark 1. $x_{k,J+1,t} = 1$ actually contains two situations, the first of which is that m_k prefers to remain idle, and the second of which is m_k being forced to remain idle due to the shortage of subchannel.

The SEB-MGSI algorithm is developed based on pricing-based matching [16], which is summarized in Algorithm 3. It can be implemented in two phases: initialization (Line 2 ∼ 3), preference list construction (Line 5 ∼ 7) and iterative matching (Line 8 ∼ 22).

1) Initialization: The initial length of all the queues and initial values of all the selection indicators are set as zero.

2) Preference List Construction: In the second phase of preference list construction, Since the preference of m_k towards any option j, $j = 1, \cdots, J+1$, is inversely proportional to $\theta_{k,j,t}$, it can be simply expressed as

$$L_{k,j,t} = \frac{1}{\theta_{k,j,t}} - \rho_{k,j}\mathbb{I}\{j < J+1\}, \tag{21}$$

where $\rho_{k,j}$ represents the cost of matching m_k with c_j, the initial value of which is set as zero.

Denote the preference list of m_k towards all the $J+1$ options as $\mathcal{F}_k$, which is obtained by sorting all the $L_{k,j,t}, j = 1, 2, \cdots, J+1$, in a descending order.

If option $J+1$ ranks the first in $\mathcal{F}_k$, m_k will skip the iterative matching process and remain idle during this slot. Otherwise, any $m_k \in \mathcal{M}_t$ updates $\mathcal{F}_k$ by removing option $J+1$, where $\mathcal{M}_t \subseteq \mathcal{M}$ is the set of MTDs selecting to transmit data in the t-th slot. Then any $m_k \in \mathcal{M}_t$ transmits it to the edge server for resolving matching conflicts based on the following procedures:

3) Iterative Matching: **Step 1**: Initialization

- Initialize $\phi = \emptyset$ and $\Omega = \emptyset$. Here, Ω denotes the conflicting set of subchannels which are selected by more than one MTDs.

Step 2: Pricing-based iterative matching

Repeat

- If $\exists \phi(m_k) = \emptyset$, any $m_k \in \mathcal{M}_t$ selects its most preferred subchannel in $\mathcal{F}_k$.
- For any subchannel $c_j \in \mathcal{C}$, if it is selected by only one MTD, e.g., m_k, then they are directly matched, i.e., $\phi(m_k) = c_j$. Otherwise, add c_j into Ω.
- If $\Omega \neq \emptyset$,
  - Each subchannel $c_j \in \Omega$ raises its price $\rho_{k,j}$ as

$$\rho_{k,j} = \rho_{k,j} + \frac{\Delta\rho_j}{F_k(t)}, \tag{22}$$

Algorithm 4 SEBC-MUCB

```
1: Input: $\{V_k\}$, $\{\alpha_k\}$, $\{\beta_k\}$, $\omega$.
2: Phase 1: Initialization
3: Set $Q_k(1)$ as the initial amount of data backlog, $N_k(1) = 0$, $F_k(1) = 0$, $\bar{\theta}_{k,j,0} = 0$, $\hat{x}_{k,j,0} = 0$, and $x_{k,j,t} = 0$, $j = 1, 2, \cdots, J+1, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{T}$.
4: Temporarily match $\forall m_k \in \mathcal{M}$ with $\forall c_j \in \mathcal{C}$ to observe the performances of throughput, energy consumption and delay.
5: Repeat
6: Phase 2: Pricing-based matching
7: Each MTD calculates its preference value towards each option as (23).
8: Each MTD constructs its preference list $\mathcal{F}_k$ and the $m_k \in \mathcal{M}_k$ transmits $\mathcal{F}_k$ to the edge server for iterative matching.
9: Each MTD performs the corresponding selection based on $\phi(m_k)$.
10: Phase 3: Learning
11: Observe $z_{k,j,t}$, $E_{k,j,t}$ and whether the delay requirement can be satisfied or not.
12: Update $\bar{\theta}_{k,j,t}$ and $\hat{x}_{k,j,t}$ based on (19) and (20).
13: Update $U_k(t)$ and $Y_k(t+1)$ based on (4) and (6).
14: Update $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (1), (15).
15: Until $t > T$.
```

where $\Delta\rho_j$ is the step size for price rising.

- All the MTDs which have selected c_j recalculate their preferences towards c_j based on (21), and renew their selection strategies accordingly. If the cost of c_j is too high, some MTDs will give it up and select other subchannels.
- Repeat the pricing process until only one MTD remains, e.g., m_k. Then, set $\phi(m_k) = c_j$ and remove c_j from Ω.
- If any c_j in $\mathcal{F}_k$ has been matched with other MTDs and is unavailable to m_k, then $\phi(m_k) = m_k$.

Until $\forall \phi(m_k) \neq \emptyset$.

Finally, the MTDs select the subchannels based on the derived ϕ, observe the corresponding results $z_{k,j,t}$, $E_{k,j,t}$ associated with $x_{k,j,t} = 1$, and whether the delay requirement can be satisfied or not. Then, each MTD m_k updates $U_k(t)$, $Y_k(t+1)$, $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (19), (20), (4), (6), (1) and (15). The iterations between the phase of preference list construction and the phase of iterative matching are terminated when $t > T$.

In the proposed pricing-based matching, the price of occupying c_j for m_k is inversely proportional to $F_k(t)$, thereby allowing MTDs with larger service reliability deficit to have a higher probability to be matched with a subchannel, which further enhances service reliability awareness.

B. The SEBC-MUCB Algorithm for the Nonideal Case

In the nonideal case where the nonlocal information required to construct preference lists of MTDs is unavailable, the matching-based SEB-GSI algorithm is infeasible. Following the idea of SEB-UCB developed in subsection III-C, an intuitive solution is to enable a MTD to estimate its preference list via online learning. We augment SEB-UCB by adding conflict awareness into the learning process, and develop the matching-learning based SEBC-MUCB algorithm. In SEBC-MUCB, a MTD can learn the impacts of decision coupling and matching conflicts by continuously observing the difference between its matching preference and actual matching results.

SEBC-MUCB is summarized in Algorithm 4, which consists of three phases, i.e., initialization (Line 2 $\sim$ 4), pricing-based matching (Line 6 $\sim$ 9), and learning (Line 10 $\sim$ 14).

In the first phase of initialization, firstly, the initial length of all the queues and initial values of all the selection indicators are set as zero. Then, for any $m_k \in \mathcal{M}$, it is temporarily matched with every $c_j \in \mathcal{C}$ to observe the performances of throughput, energy consumption and delay.

In the second phase of pricing-based matching, m_k estimates its preference towards the j-th option as

$$\widetilde{L}_{k,j,t} = \frac{1}{\bar{\theta}_{k,j,t-1}} + \omega\sqrt{\frac{2\ln t}{\hat{x}_{k,j,t-1}}} - \rho_{k,j}\mathbb{I}\{j < J+1\}. \quad (23)$$

Here, the preference value of m_k towards an option that has not been selected, e.g., $\hat{x}_{k,j,t-1} = 0$, is defined as $+\infty$ so that each option can be selected by m_k at least once.

Then, based on (23), m_k constructs its preference list $\mathcal{F}_k$ similarly as subsection IV-A and transmits it to the edge server. Next, MTDs are matched with subchannels based on the pricing-based matching. Eventually, each MTD selects the subchannel according to the obtained $\phi(m_k)$.

In the third phase of learning, each MTD m_k observes the corresponding results $z_{k,j,t}$, $E_{k,j,t}$ associated with $x_{k,j,t} = 1$ and whether the delay requirement can be satisfied or not. Then, each MTD m_k updates $\bar{\theta}_{k,j,t}$, $\hat{x}_{k,j,t}$, $U_k(t)$, $Y_k(t+1)$, $Q_k(t+1)$, $N_k(t+1)$, and $F_k(t+1)$ as (19), (20), (4), (6), (1) and (15). The iterations between the phase of pricing-based matching and the phase of learning are terminated when $t > T$.

V. Performance Analysis

In this section, we provide a comprehensive performance analysis of the proposed algorithms from the perspective of optimality and complexity.

A. Optimality

We first present the bounded cumulative throughput performance of SEB-MGSI. Then, we quantify the performance loss due to learning in terms of learning regret and provide its upper bound. Finally, the bounded cumulative throughput performance of SEBC-MUCB is provided.

To provide theoretical upper bound of the cumulative throughput performance, we describe a scenario where MTDs know the GSI for the future T slots. We define $x^*_{k,j,t}$, $z^*_{k,j,t}$ and j^* as the channel selection indicator, throughput and the optimum option derived with T-slot GSI. Specifically, $x^*_{k,j,t} = 1$ is satisfied when event $j = j^*$ is true and $x^*_{k,j,t} = 0$ otherwise. Accordingly, define $\ddot{x}_{k,j,t}$, $\ddot{\theta}_{k,j,t}$, $\ddot{z}_{k,j,t}$ and $\ddot{j}$ as the channel selection indicator, the weighted sum of throughput, energy consumption and service reliability, throughput and the optimum option achieved by the algorithms for the ideal case, i.e., SEB-MGSI for multi-MTD scenario and SEB-GSI for the single-MTD scenario. Define $\breve{x}_{k,j,t}$, $\breve{\theta}_{k,j,t}$, $\breve{z}_{k,j,t}$ and $\breve{j}$ as the channel selection indicator, the weighted sum of throughput, energy consumption and service reliability, throughput and the optimum option achieved by the algorithms for the ideal case, i.e., SEBC-MUCB for multi-MTD scenario and SEB-UCB for the single-MTD scenario.

Theorem 1. *The cumulative throughput achieved by SEB-MGSI is be lower bounded as*

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} x^*_{k,j,t}z^*_{k,j,t} - \frac{KT^2B_{max}}{V_k} \leq \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} \ddot{x}_{k,j,t}\ddot{z}_{k,j,t}, \tag{24}$$

where B_{max} is defined as

$$\begin{aligned} B_{max} &= \frac{1}{2}(\max\{\gamma_k A_k(t) + Y_k(t+1) - U_k(t)\})^2 \\ &+ \frac{1}{2}(\max\{\sum_{j=1}^{J+1} x_{k,j,t}E_{k,j,t} - \frac{E_{k,max}}{T}\})^2 \\ &+ \frac{1}{2}(\max\{\sum_{j=1}^{J+1} \mathbb{I}\{d^{total}_{k,j,t} \leq d_{k,t}\}x_{k,j,t} - \eta_k\})^2. \end{aligned} \tag{25}$$

Proof: See Appendix A. ■

Learning regret represents the expected performance difference between the cumulative weighted sum of throughput, energy consumption and service reliability achieve by SEB-GSI and that achieved by SEBC-MUCB. Given K MTDs and $J+1$ options, the learning regret R is defined as

$$\begin{aligned} R &= \mathbb{E}\{\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1}[\ddot{x}_{k,j,t}\ddot{\theta}_{k,j,t} - \breve{x}_{k,j,t}\breve{\theta}_{k,j,t}]\} \\ &= \mathbb{E}\{\sum_{t=1}^{T}\sum_{k=1}^{K}[x_{k,\ddot{j},t}\theta_{k,\ddot{j},t} - x_{k,\breve{j},t}\theta_{k,\breve{j},t}]\}. \end{aligned} \tag{26}$$

For the purpose of simplicity, we define

$$\Delta\theta_{k,\ddot{j},\breve{j}} = \ddot{\theta}_{k,j} - \breve{\theta}_{k,j}, \tag{27}$$

where $\ddot{\theta}_{k,j} = \mathbb{E}[\theta_{k,\ddot{j},t}]$ and $\breve{\theta}_{k,j} = \mathbb{E}[\theta_{k,\breve{j},t}]$.

Theorem 2. *When $\omega = 1$, the learning regret of the SEBC-MUCB is upper bounded as*

$$\begin{aligned} R &\leq 8(J+1)\sum_{k=1}^{K}(\Delta\theta_{k,\ddot{j},\breve{j}})^3\ln(T) + K(J+1)\Delta\theta_{k,\ddot{j},\breve{j}} \\ &+ (J+1)\sum_{k=1}^{K}\sum_{t=1}^{+\infty}[2t^{-4K+2}\Delta\theta_{k,\ddot{j},\breve{j}}] \end{aligned} \tag{28}$$

Proof: See Appendix B. ■

Based on the definition of learning regret, the cumulative throughput achieved by SEBC-MUCB can be derived as the cumulative throughput achieved by SEB-MGSI minus learning regret.

Theorem 3. *The cumulative throughput achieved by SEBC-MUCB is lower bounded as*

$$\begin{aligned} \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=1}^{J+1}\breve{x}_{k,j,t}\breve{z}_{k,j,t} \geq &\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=1}^{J+1} x^*_{k,j,t}z^*_{k,j,t} - R \\ &- \frac{KT^2B_{max}}{V_k} - \alpha_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{N}_k(t)\ddot{E}_{k,j,t} \\ &- \beta_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{F}_k(t)(\eta_k - \sum_{j=1}^{J} x_{k,\ddot{j},t}). \end{aligned} \tag{29}$$

Proof: See Appendix C. ■

Theorem 1 and theorem 3 indicate that both SEB-MGSI and SEBC-MUCB can achieve a guaranteed throughput performance. Theorem 2 indicates that SEBC-MUCB can achieve a bounded deviation from SEB-MGSI.

Theorem 4. *The cumulative throughput achieved by SEB-GSI is lower bounded as*

$$\sum_{t=1}^{T}\sum_{j=1}^{J+1} x^*_{k,j,t}z^*_{k,j,t} - \frac{T^2B_{max}}{V_k} \leq \sum_{t=1}^{T}\sum_{j=1}^{J+1}\ddot{x}_{k,j,t}\ddot{z}_{k,j,t}. \tag{30}$$

Theorem 5. *When $\omega = 1$, the learning regret of the SEB-UCB is upper bounded as*

$$R \leq 8(J+1)(\Delta\theta_{k,\ddot{j},\breve{j}})^3\ln(T) + \Delta\theta_{k,\ddot{j},\breve{j}}(J+1)(1+\frac{\pi^2}{3}). \tag{31}$$

Theorem 6. *The cumulative throughput achieved by SEB-UCB is lower bounded as*

$$\begin{aligned} \sum_{t=1}^{T}\sum_{j=1}^{J+1}\breve{x}_{k,j,t}\breve{z}_{k,j,t} \geq &\sum_{t=1}^{T}\sum_{j=1}^{J+1} x^*_{k,j,t}z^*_{k,j,t} - R \\ &- \frac{T^2B_{max}}{V_k} - \alpha_k\sum_{t=1}^{T}\ddot{N}_k(t)\ddot{E}_{k,j,t} \\ &- \beta_k\sum_{t=1}^{T}\ddot{F}_k(t)(\eta_k - \sum_{j=1}^{J} x_{k,\ddot{j},t}). \end{aligned} \tag{32}$$

Proof: Theorem 4, Theorem 5, and Theorem 6 can be proved as special cases of Theorem 1, Theorem 2, and Theorem 3, respectively, when $K = 1$. The detailed proof is ignored due to space limitation. ■

Theorem 7. *For SEBC-MUCB, after the initial $\lceil 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2\ln(t)\rceil$ times of selecting a non-optimal option, the probability of selecting a non-optimal option is upper bounded by $2t^{-4K}$. As $t \rightarrow +\infty$, the upper bound converges to 0.*

Proof: See Appendix D. ■

B. Complexity

SEB-GSI: The computational complexity of SEB-GSI consists of four parts. The first part is initialization with the complexity of $\mathcal{O}(J+4)$, and the second part is calculating $\theta_{k,j,t}$ of $J+1$ options with the complexity of $\mathcal{O}(J+1)$. The third part is seeking the minimum $\theta_{k,j,t}$ with the complexity of $\mathcal{O}(J)$ and the fourth part is renewing queues with the complexity of $\mathcal{O}(3)$. Therefore, the computational complexity of FEB-GSI is $\mathcal{O}(J+4)+\mathcal{O}(J+1)+\mathcal{O}(J)+\mathcal{O}(3)$.

SEB-UCB: The computational complexity of SEB-UCB is composed of three parts. The computational complexity of the first phase is $\mathcal{O}(3J+6)$, and that of the second phase is $\mathcal{O}(J+1)+\mathcal{O}(J)$. The complexity of the third phase is $\mathcal{O}(2J+5)$. Therefore, the computational complexity of SEB-UCB is $\mathcal{O}(3J+3)+\mathcal{O}(J+1)+\mathcal{O}(J)+\mathcal{O}(2J+5)$.

SEB-MGSI: The computational complexity of SEB-MGSI is composed of three parts. The first part is initialization with the complexity of $\mathcal{O}(2J+5)$, and the second part is the complexity of pricing-based matching. Assuming the matching conflicts can be resolved within ϖ iterations, the conflict can be solved with the complexity of $\mathcal{O}(J+1)+\mathcal{O}((J+1)\log(J+1))+\mathcal{O}(I\varpi)$ when $K\geq J$. The complexity of renewing queues is $\mathcal{O}(3)$. Therefore, the computational complexity of SEB-MGSI is $\mathcal{O}(2J+5)+\mathcal{O}(J+1)+\mathcal{O}((J+1)\log(J+1))+\mathcal{O}(I\varpi)+\mathcal{O}(3)$.

SEBC-MUCB: The computational complexity of SEBC-MUCB consists of three parts. The complexity of the first and second phases are the same as that of SEB-UCB and SEB-GSI, respectively. The complexity of the third phase if $\mathcal{O}(2J+5)$. Therefore, the complexity of SEBC-MUCB is $\mathcal{O}(3J+6)+\mathcal{O}(J+1)+\mathcal{O}((J+1)\log(J+1))+\mathcal{O}(I\varpi)+\mathcal{O}(2J+5)$ when $K\geq J$.

VI. Implementation Considerations

In real-world implementation, the convergence time and the performance loss due to learning can be further reduced. Theorem 2 indicates that the learning regret is related to both the numbers of MTDs and subchannels to be explored as well as the exploration cost. Two heuristic solutions are provided here, i.e., set division and task division.

A. Set Division

To reduce the numbers of MTDs and subchannels, one heuristic solution is to divide the set of subchannels $\mathcal{C}$ and the set of MTDs $\mathcal{M}$ into several subsets. Then, a subchannel set, e.g., $\mathcal{C}_s\subset\mathcal{C}$, is exclusively licensed to a MTD subset $\mathcal{M}_s\subset\mathcal{M}$, i.e., only MTDs belonging to the subset $\mathcal{M}_s$ are allowed to use subchannels in $\mathcal{C}_s$. As a result, the numbers of competing MTDs and subchannels can be reduced since $\mid\mathcal{C}_s\mid<J$ and $\mid\mathcal{M}_s\mid<K$.

When implementing the set division-based heuristic solution, the BS has to obtain the precise knowledge of the set of subchannels $\mathcal{C}$ and the set of MTDs $\mathcal{M}$. Since the sets of subchannels and MTDs do not vary every slot, the BS only needs to collet this information once at each optimization duration. The BS will collect the information of $\mathcal{C}$ and $\mathcal{M}$, perform set division based on certain optimization rules, and inform the MTDs of the division results, i.e., $\mid\mathcal{C}_s\mid$ and $\mid\mathcal{M}_s\mid$. On the other hand, both $\mid\mathcal{C}_s\mid$ and $\mid\mathcal{M}_s\mid$ should be decided vigilantly. Specifically, $\mid\mathcal{C}_s\mid\ll\mid\mathcal{M}_s\mid$ will lead to severe competition while $\mid\mathcal{C}_s\mid\gg\mid\mathcal{M}_s\mid$ will incur a large exploration cost.

B. Task Division

To increase the convergence speed and reduce the exploration cost, another heuristic solution is to enable MTDs to utilize a smaller task for learning. The MTDs can divide a large task into several smaller tasks. Since the task size is small, both the transmission delay and computational delay can be reduced significantly. Furthermore, with smaller task, each slot can also be divided into several subslots, and in each subslot, the MTDs can make a channel selection decision and perform learning by observing the feedback from the edge server, thereby increasing the total number of exploration in each slot. This can dramatically improve the convergence speed and reduce the convergence time. When implementing the task division-based heuristic solution, a MTD has to divide a large task into several smaller tasks. In other words, the task should be dividable. Both task division and slot division can be performed by the MTD based on local information.

C. Implementation Delay

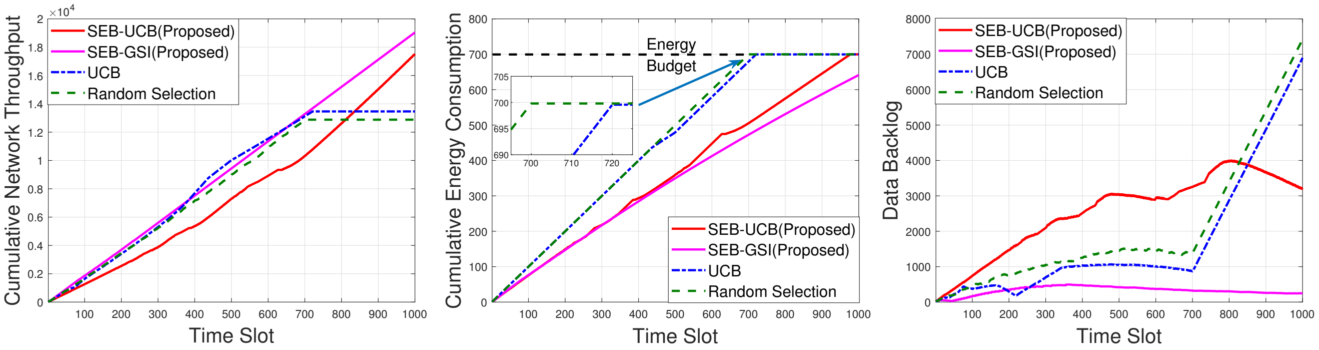
Although the learning-based algorithm involves a lot of iterations, the iteration delay in each slot is negligible. The reason is that during each slot, a MTD only makes one decision and then waits for the reward associated with the decision. In other words, there is only one iteration of channel selection at each slot. Therefore, the delay for processing a task mainly consists of the transmission delay and the computational delay, while the iteration delay can be ignored.

VII. Simulations

In this section, we validate the proposed algorithms via simulations under the scenarios of single-MTD and multi-MTD, respectively.

A. Performance under the Single-MTD Scenario

In the single-MTD scenario, we consider one MTD and three subchannels over a total period of $T=10^3$ time slots, i.e., $K=1$ and $J=3$. We set $\tau=1$ s, $P_{\text{TX}}=1$ W, $E_{k,max}=700$ J. We assume that $A_k(t)$ follows a uniform distribution within the interval $[0.9\bar{A}_k,1.1\bar{A}_k]$ Mbits, where $\bar{A}_k=20$ Mbits represents the time-average amount of collected data. The initial value $Q_k(1)$ is randomly selected within the interval $[0.8\bar{A}_k,1.2\bar{A}_k]$ Mbits. The computational complexity is set as $\lambda_{k,t}=10^3$ CPU cycles/bit. The available computational resource for m_k in the t-th slot $\xi_{k,t}$ is randomly distributed within the interval $[0.9\bar{\xi}_k,1.1\bar{\xi}_k]$ CPU cycles, where $\bar{\xi}_k=18\times10^9$ CPU cycles represents the time-average amount of computational resource. $U_k(t)$ does not need to be initialized, the value of which depends on the selection strategies, CSI as well as local data backlog. The service reliability requirement is set as $\eta_k=0.7$. We set $V_k=1$, $\alpha_k=5$, and $\beta_k=3$ to balance the tradeoff among throughput performance, energy consumption, and service reliability. The achievable transmission rate of subchannel s_j in each slot follows a uniform distribution within the range $[0.8\bar{R}_j,1.2\bar{R}_j]$, where $\bar{R}_j$ represents the average transmission rate. We set $\bar{R}_j=10,20,30$ Mbits when $j=1,2,3$.

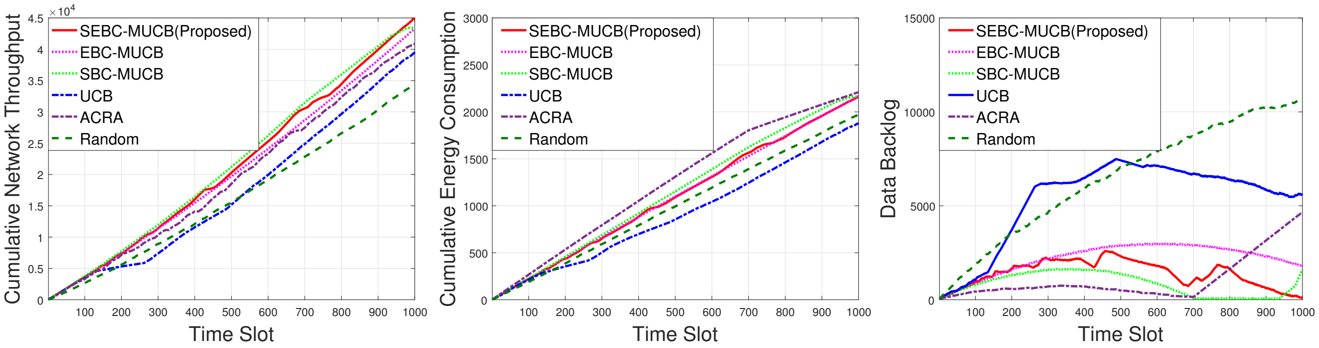


(a) Network throughput versus time slot.

(b) Energy consumption versus time slot.

(c) Data backlog versus time slot.

Fig. 2. Performances under single-MTD scenario.



(a) Network throughput versus time slot.

(b) Energy consumption versus time slot.

(c) Data backlog versus time slot.

Fig. 3. Performances under multi-MTD scenario.

The weight of exploration ω is set as 1. Two heuristic algorithms are used for comparison. One is the conventional UCB algorithm proposed in [34], and the other is the random selection algorithm in which m_k randomly selects a subchannel at each slot. The SEB-GSI with perfect GSI is used as an upper performance benchmark.

Fig. 2(a) and Fig. 2(b) show the cumulative network throughput and cumulative energy consumption performances over a total of 10^3 slots. Compared with UCB and random selection, the proposed SEB-UCB with only local information can improve throughput by 30% and 36% respectively, while satisfying the constraint of energy consumption. Particularly, there exists a performance floor after 700 slots. The reason is demonstrated in Fig. 2(b), which explicitly shows that the two heuristic algorithms use energy more aggressively at the beginning and then run out of the energy at $t = 700$ and $t = 720$, thereby leaving no energy for data transmission. It is noted that the energy consumption of the proposed algorithms will not increase after $t = 1000$ since the energy budget is exactly exhausted at $t = 1000$, i.e., the proposed algorithms can well exploit the available energy during the specified optimization duration compared with other heuristic algorithms. Besides, the SEB-UCB performs just slightly worse than the SEB-GSI algorithm with perfect GSI. The curve trends of both the network throughput and energy consumption performances track those of SEB-GSI strictly.

Fig. 2(c) shows the data backlog performance. Simulation results demonstrate that SEB-UCB can provide bounded data backlog, while the backlogs of UCB and random selection increase linearly with time after 700 slots, which significantly degrades the queue stability performance and may even lead to severe data loss.

B. Performance under the Multi-MTD Scenario

For the multi-MTD scenario, we consider three MTDs and three subchannels, i.e., $K = J = 3$. We set $E_{k,max} = 730$ J, $\eta_k = 0.73$, $V_k = 1$, $\alpha_k = 20$, and $\beta_k = 25, \forall m_k \in \mathcal{M}$. The other simulation parameters remain the same as those in the single-MTD scenario.

Five heuristic algorithms are used for comparison. The first one is the EBC-MUCB algorithm without service reliability awareness, i.e., the service reliability constraint is not considered. The second one is the SBC-MUCB algorithm without energy awareness, i.e., the energy consumption constraint is not considered. The third one is the conventional UCB algorithm, and the fourth one is random selection. The fifth one is the Lyapunov optimization-based access control and resource allocation (ACRA) algorithm developed in [14]. ACRA requires perfect GSI to find the optimum option. Here, we assume that only the CSI of the previous slot is available, i.e., the CSI is outdated information. In other words, optimization

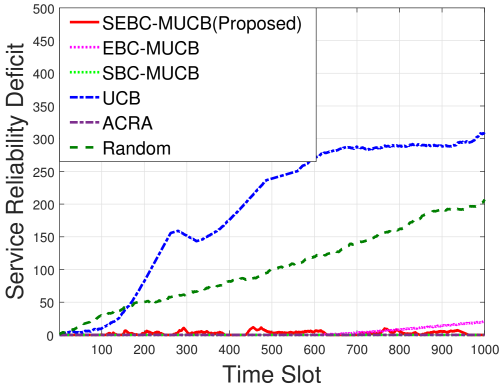


Fig. 4. Service reliability deficit versus time slot.

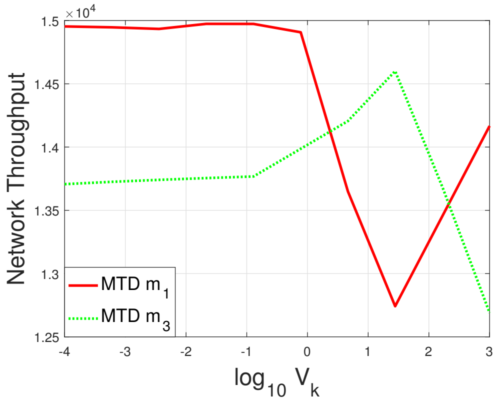


Fig. 5. Impact of V_k.

at the t-th slot is performed based on the CSI of the $(t-1)$-th slot.

Fig. 3(a) shows the cumulative network throughput versus time slot. The proposed SEBC-MUCB outperforms UCB and random selection by 13.7% and 31.2%, respectively. Compared with SBC-MUCB and EBC-MUCB, SEBC-MUCB improves throughput by 3.46% and 3.96%, respectively, due to the additional consideration of energy awareness and service reliability awareness. Taking SBC-MUCB as an example, although it achieves a higher throughout at the beginning, it runs out of energy at $t = 981$ and is forced to be idle for the remaining slots, which significantly degrades the overall throughput performance.

Fig. 3(b) shows the cumulative energy consumption versus time slot. Simulation results show that The energy consumption of SEBC-MUCB and EBC-MUCB algorithms has not exceeded the energy budget due to energy awareness. Different from the scenario of single-MTD, UCB consumes the least energy since the frequent selection conflicts force MTDs to remain idle so that the energy consumption becomes less.

Fig. 3(c) demonstrates that SEBC-MUCB achieves the least data backlog among all the algorithms. In comparison, the data backlog of SBC-MUCB increases dramatically after $t = 981$ due to the ignorance of energy awareness. UCB performs worse than EBC-MUCB and SBC-MUCB since the frequent selection conflicts impede MTDs from data transmission and data backlog becomes very large.

Fig. 4 shows the service reliability deficit versus time slot. The proposed SEBC-MUCB can meet the service reliability requirement and achieve the second least service reliability deficit. Although SBC-MUCB achieves the least service reliability deficit, its throughput and energy consumption performance are worse than SEBC-MUCB because only service reliability awareness is considered. The service reliability deficit of EBC-MUCB increases dramatically after $t = 700$ due to the negligence of service reliability awareness. UCB performs the worst since it has not been endowed with the capability of conflict resolution.

From Fig. 3(a) to Fig. 4, we can find that although the energy consumption and the service reliability deficit of ACRA are nearly the same as those of SEBC-MUCB, the throughput performance and the data backlog performance are worse. SEBC-MUCB outperforms ACRA by 10.58% in terms of throughput, and 4783.76% in terms of data backlog due to the endowed capability of online learning. Particularly, the data backlog performance of ACRA is significantly degraded by employing the outdated CSI for optimization. Therefore, we can conclude that learning plays an important role for backlog reduction under the scenario where perfect GSI is unavailable.

Fig. 5 shows the impact of parameter V_k on the throughput performances of MTDs. Specifically, we set $V_1 = V_2 = 1$ for m_1 and m_2, while V_3 increases from 10^{-4} to 10^3 for m_3. Simulation results demonstrate that as V_3 increases, the throughput of m_3 increases first and then decreases, while the throughput of m_1 shows the opposite trend. The rationale is that when V_3 increases from 10^{-4} to 25 ($\log(V_3)$ increase from -4 to 1.4), m_3 puts a larger weight on the throughput, and becomes more active to explore channels for throughput improvement. This will cause more channel selection conflicts, thereby reducing the throughput of m_1. However, when V_3 is too large ($\log(V_3) > 1.4$), m_3 over-evaluates throughput and has little concern on energy consumption. It will quickly run out of energy and is forced to remain idle, which significantly degrades the throughput performance. Meanwhile, other MTDs such as m_1 can benefit from the idle state of m_3 since the channel selection conflicts is relieved.

VIII. Conclusions

In this paper, we proposed learning-based channel selection which incorporates service reliability awareness, energy awareness and backlog awareness. We started from single-MTD scenario and proposed distributed low-complexity SEB-GSI algorithm with CSI and SEB-UCB algorithm under information uncertainty. Then, we extended it to the multi-MTD scenario and developed SEBC-MUCB algorithm by integrating MAB, Lyapunov optimization and matching theory. Simulation results demonstrate that the proposed SEB-UCB can improve throughput by 30% and 36% compared with UCB and random selection. SEBC-MUCB outperforms UCB and random selection by 13.7% and 31.2% while stabilizing data backlog queue and satisfying energy consumption constraint as well as service reliability requirement. Due to the limited computational capability and battery capacity of MTDs, we only consider the scenario of task offloading, while local computing is ignored. Our future work will focus on the online cross-layer resource optimization including local computation, rate control, channel selection, and resource allocation in the edge server under information uncertainty.

Appendix A Proof of Theorem 1

Define $q_k(t)$, $n_k(t)$ and $f_k(t)$ as

$$\begin{aligned} q_k(t) &= \gamma_k A_k(t) + Y_k(t+1) - U_k(t), \\ n_k(t) &= \sum_{j=1}^{J+1} x_{k,j,t} E_{k,j,t} - \frac{E_{k,max}}{T}, \\ f_k(t) &= \sum_{j=1}^{J+1} \mathbb{I}\{d_{k,j,t}^{total} \leq d_{k,t}\} x_{k,j,t} - \eta_k. \end{aligned} \tag{33}$$

Taking (33) into (1) and (15), we can obtain

$$\begin{aligned} Q_k(t+1) - Q_k(t) &\geq q_k(t), \\ N_k(t+1) - N_k(t) &\geq n_k(t), \\ F_k(t+1) - F_k(t) &\geq f_k(t). \end{aligned} \tag{34}$$

Define the concatenated vector of the data backlog queue and virtual queues as $\mathbf{G}(t) = [\{Q_k(t)\}, \{N_k(t)\}, \{F_k(t)\}]$. Define the Lyapunov function and one-slot Lyapunov drift as

$$L(\mathbf{G}(t)) = \frac{1}{2} \sum_{k=1}^{K} [Q_k^2(t) + N_k^2(t) + F_k^2(t)], \tag{35}$$

$$\Delta_1(\mathbf{G}(t)) = L(\mathbf{G}(t+1)) - L(\mathbf{G}(t)). \tag{36}$$

Define the drift-minus-reward term as

$$DR_1(\mathbf{G}(t)) = \Delta_1(\mathbf{G}(t)) - V_k \sum_{k=1}^{K} \sum_{j=1}^{J+1} z_{k,j,t}. \tag{37}$$

Taking (36) into (35), we can derive

$$\begin{aligned} \Delta_1(\mathbf{G}(t)) = &\frac{1}{2} \sum_{k=1}^{K} [Q_k^2(t+1) - Q_k^2(t)] \\ &+ \frac{1}{2} \sum_{k=1}^{K} [N_k^2(t+1) - N_k^2(t)] \\ &+ \frac{1}{2} \sum_{k=1}^{K} [F_k^2(t+1) - F_k^2(t)]. \end{aligned} \tag{38}$$

Based on (34), we have

$$Q_k^2(t+1) \leq (Q_k(t) + q_k(t))^2. \tag{39}$$

Taking (39) into (52)

$$\begin{aligned} \Delta_1(\mathbf{G}(t)) \leq &\sum_{k=1}^{K} \frac{1}{2} [q_k^2(t) + n_k^2(t) + f_k^2(t)] \\ &+ \sum_{k=1}^{K} [q_k(t) Q_k(t) + n_k(t) N_k(t) + f_k(t) F_k(t)]. \end{aligned} \tag{40}$$

Define B_k as

$$B_k = \max_{t \in \mathcal{T}} \frac{1}{2} [q_k^2(t) + n_k^2(t) + f_k^2(t)]. \tag{41}$$

Taking (40) and (41) into (37)

$$\begin{aligned} DR_1(\mathbf{G}(t)) \leq &\sum_{k=1}^{K} B_k - V_k \sum_{j=1}^{J+1} \sum_{k=1}^{K} z_{k,j,t} \\ &+ \sum_{k=1}^{K} [q_k(t) Q_k(t) + n_k(t) N_k(t) + f_k(t) F_k(t)]. \end{aligned} \tag{42}$$

Similarly as (36), T-slot Lyapunov drift can be defined as

$$\Delta_T(\mathbf{G}(t)) = L(\mathbf{G}(t+T)) - L(\mathbf{G}(t)). \tag{43}$$

Therefore, the sum of $DR_1(\mathbf{G}(t))$ over T slots can be derived as

$$\begin{aligned} &\Delta_T(\mathbf{G}(1)) - V_k \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{j=1}^{J+1} z_{k,j,t} \\ &\leq \sum_{k=1}^{K} T B_k - V_k \sum_{t=1}^{T} \sum_{j=1}^{J+1} \sum_{k=1}^{K} z_{k,j,t} \\ &+ \sum_{t=1}^{T} \sum_{k=1}^{K} [q_k(t) Q_k(t) + n_k(t) N_k(t) + f_k(t) F_k(t)]. \end{aligned} \tag{44}$$

The last term on the right-hand side of (48) satisfies

$$\begin{aligned} &\sum_{t=1}^{T} \sum_{k=1}^{K} [q_k(t) Q_k(t) + n_k(t) N_k(t) + f_k(t) F_k(t)] \\ &= \sum_{t=1}^{T} \sum_{k=1}^{K} [q_k(t) Q_k(1) + n_k(t) N_k(1) + f_k(t) F_k(1)]| \\ &+ \sum_{t=1}^{T} \sum_{k=1}^{K} [(Q_k(t) - Q_k(1)) q_k(t)] \\ &+ \sum_{t=1}^{T} \sum_{k=1}^{K} [(N_k(t) - N_k(1)) n_k(t)] \\ &+ \sum_{t=1}^{T} \sum_{k=1}^{K} (F_k(t) - F_k(1)) f_k(t)]. \end{aligned} \tag{45}$$

Define q_{max}, n_{max} and f_{max} as the maximum positive value for all MTDs to satisfy

$$\begin{aligned} Q_k(t+1) - Q_k(t) &\leq q_{max}, \\ N_k(t+1) - N_k(t) &\leq n_{max}, \\ F_k(t+1) - F_k(t) &\leq f_{max}. \end{aligned} \tag{46}$$

Based on (46), we can obtain

$$\begin{aligned} &(Q_k(t) - Q_k(1)) q_k(t) \\ &= (Q_k(t) - Q_k(t-1)) q_k(t) + (Q_k(t-1) - Q_k(1)) q_k(t) \\ &\leq q_{max} q_k(t) + (Q_k(t-1) - Q_k(1)) q_k(t). \end{aligned} \tag{47}$$

Since $Q_k(1) = N_k(1) = F_k(1) = 0$, taking (47) into (45), we can bound the last term on the right-hand side of (48) as

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{k=1}^{K}[q_k(t)Q_k(t) + n_k(t)N_k(t) + f_k(t)F_k(t)] \\
&\leq \sum_{k=1}^{K}\sum_{t=1}^{T}(t-1)(q_{max}^2 + n_{max}^2 + f_{max}^2) \\
&= \frac{KT(T-1)}{2}(q_{max}^2 + n_{max}^2 + f_{max}^2) \\
&= KT(T-1)B_{max}, \qquad (48)
\end{aligned}$$

where $B_{max} = \frac{1}{2}(q_{max}^2 + n_{max}^2 + f_{max}^2)$.

We can see $B_{max} \geq B_k$ must be satisfied, (48) can thus be bounded as

$$\begin{aligned}
&\Delta_T(\mathbf{G}(1)) - V_k \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} z_{k,j,t} \\
&\qquad \leq KT^2 B_{max} - V_k \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} z_{k,j,t}. \qquad (49)
\end{aligned}$$

We denote $x_{k,j,t}^*$ and $z_{k,j,t}^*$ as the channel selection indicators and throughput performance obtained by employing exhaustive method with T-slot GSI. By applying SEB-MGSI into the left side and considering exhaustive method on the right-hand side, we obtain

$$\begin{aligned}
&\Delta_T(\mathbf{G}(1)) - V_k \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J} \ddot{x}_{k,j,t}\ddot{z}_{k,j,t} \\
&\qquad \leq KT^2 B_{max} - V_k \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} x_{k,j,t}^* z_{k,j,t}^* \qquad (50)
\end{aligned}$$

By dividing both sides of (50) by V_k, we can derive that

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} x_{k,j,t}^* z_{k,j,t}^* - \frac{KT^2 B_{max}}{V_k} \\
&\qquad \leq \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1} \ddot{x}_{k,j,t}\ddot{z}_{k,j,t}. \qquad (51)
\end{aligned}$$

This completes the proof of Theorem 1.

Appendix B
Proof of Theorem 2

At each slot, either the optimal or the non-optimal subchannel will be selected. Denote the number of times that a non-optimal selection for m_k, i.e., option $\breve{j}$, has been selected up to the t-slot as $\chi_{k,\breve{j},t-1}$. If option $\breve{j}$ is selected by m_k in the t-th slot, then $\chi_{k,\breve{j},t} = \chi_{k,\breve{j},t-1} + 1$. The learning regret R can be derived as

$$\begin{aligned}
R &= \mathbb{E}\{\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{J+1}[\ddot{x}_{k,j,t}\ddot{\theta}_{k,j,t} - \breve{x}_{k,j,t}\breve{\theta}_{k,j,t}]\} \\
&= \mathbb{E}\{\sum_{t=1}^{T}\sum_{k=1}^{K}[x_{k,\ddot{j},t}\theta_{k,\ddot{j},t} - x_{k,\breve{j},t}\theta_{k,\breve{j},t}]\} \\
&= \sum_{k=1}^{K}\sum_{\breve{j}=1}^{J+1}\mathbb{E}[\theta_{k,\ddot{j},t} - \theta_{k,\breve{j},t}]\mathbb{E}[\chi_{k,\breve{j},T}] \\
&= \sum_{k=1}^{K}\sum_{\breve{j}=1}^{J+1}\Delta\theta_{k,\ddot{j},\breve{j}}\mathbb{E}[\chi_{k,\breve{j},T}]. \qquad (52)
\end{aligned}$$

Then, we recall the indicator function $\mathbb{I}\{x\}$ where $\mathbb{I}\{x\} = 1$ if event x is true and $\mathbb{I}\{x\} = 0$ otherwise. Besides, we make a crude approximation that the non-optimal selection are made at least m times. If $a_{k,\breve{j},t} = 1$, we should have

$$\widetilde{\theta}_{k,\breve{j},t} \geq \widetilde{\theta}_{k,\ddot{j},t}. \qquad (53)$$

It indicates that the upper confidence bound of the selected option should be larger than that of the optimal option. Therefore, we have

$$\begin{aligned}
\chi_{k,\breve{j},T} &\leq m + \sum_{t=mI+1}^{T}\mathbb{I}\{\widetilde{\theta}_{k,\breve{j},t} \geq \widetilde{\theta}_{k,\ddot{j},t}, \chi_{k,\breve{j},t-1} \geq m\}, \\
&\qquad \forall m_k \in \mathcal{M}, \ddot{j} = 1, 2, \cdots, J+1, \qquad (54)
\end{aligned}$$

where the second item represents the crude approximation made above.

We define the amount of times that option $\ddot{j}$ has been selected by m_k up to slot t as $\hat{x}_{k,\ddot{j},t-1}^{'}$. Denote $B_{k,\breve{j},t}$ as the confidence interval, which can be given as

$$B_{k,\breve{j},t-1} = \sqrt{\frac{2\ln t}{\hat{x}_{k,\breve{j},t-1}}}. \qquad (55)$$

To write the inequality in a nicer form, we make a further approximation as follows,

$$\begin{aligned}
\chi_{k,\breve{j},T} \leq m + \sum_{t=mI+1}^{T}\mathbb{I}\{&\max_{m \leq \hat{x}_{k,\breve{j},t-1} < t}\frac{1 + \breve{F}_k(t)}{\bar{\theta}_{k,\breve{j},t-1}} + B_{k,\breve{j},t-1} \\
&\geq \min_{0 < \hat{x}_{k,\ddot{j},t-1}^{'} < t}\frac{1 + \ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} + B_{k,\ddot{j},t-1}\}, \\
&\forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1, 2, \cdots, J+1. \qquad (56)
\end{aligned}$$

Indeed, there will be at least one pair ($\hat{x}_{k,\breve{j},t-1}$, $\hat{x}_{k,\ddot{j},t-1}^{'}$) that can satisfy the inequality if (56) is satisfied. Therefore, we just need to count the number of such pairs which satisfy (56). That is, we can expand the event above into the double sum which is at least as large:

$$\chi_{k,\breve{j},T} \leq m + \sum_{t=mI+1}^{T} \sum_{\hat{x}_{k,\breve{j},t-1}=m}^{t-1} \sum_{\hat{x}'_{k,\ddot{j},t-1}=1}^{t-1} \mathbb{I}\{\frac{1+\breve{F}_k(t)}{\bar{\theta}_{k,\breve{j},t-1}} + B_{k,\breve{j},t-1} \geq \frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} + B_{k,\ddot{j},t-1}\},$$
$$\forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1,2,\cdots,J+1. \quad (57)$$

We make another odd inequality by increasing the sum to go from $t=1$ to $+\infty$, and we can replace $t-1$ with t as:

$$\chi_{k,\breve{j},T} \leq m + \sum_{t=1}^{+\infty} \sum_{\hat{x}_{k,\breve{j},t-1}=m}^{t} \sum_{\hat{x}'_{k,\ddot{j},t-1}=1}^{t} \mathbb{I}\{\frac{1+\breve{F}_k(t)}{\bar{\theta}_{k,\breve{j},t-1}} + B_{k,\breve{j},t-1} \geq \frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} + B_{k,\ddot{j},t-1}\},$$
$$\forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1,2,\cdots,J+1. \quad (58)$$

Suppose that this event actually happens, $\forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1,2,\cdots,J+1$, at least one of the following cases must be true:

$$\text{(a)}: \frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t},$$
$$\text{(b)}: \frac{1+\breve{F}_k(t)}{\bar{\theta}_{k,\breve{j},t-1}} \geq \frac{1+\breve{F}_k(t)}{\breve{\theta}_{k,j}} + B_{k,\breve{j},t},$$
$$\text{(c)}: \frac{1}{\ddot{\theta}_{k,j}} < \frac{1}{\breve{\theta}_{k,j}} + 2B_{k,\breve{j},t}. \quad (59)$$

Case (a) means that the reciprocal of the optimal option's empirical mean is less than or equal to the lower confidence bound. Case (b) means that the reciprocal of empirical mean of option j is larger than or equal to the upper confidence bound. It can be proved if case (a) and case (b) are false, then case (c) must be true. Case (a) happens with a probability:

$$\mathrm{P}\{\frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t}, \forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1,2,\cdots,J+1\}$$
$$= \mathrm{P}\{\frac{1+\ddot{F}_1(t)}{\bar{\theta}_{1,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_1(t)}{\ddot{\theta}_{1,j}} - B_{1,\ddot{j},t}\}$$
$$\times \mathrm{P}\{\frac{1+\ddot{F}_2(t)}{\bar{\theta}_{2,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_2(t)}{\ddot{\theta}_{2,j}} - B_{2,\ddot{j},t}\}$$
$$\times \cdots \times \mathrm{P}\{\frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t}\}. \quad (60)$$

By applying Chernoff-Hoeffding inequality, when $\omega = 1$, we can derive

$$\mathrm{P}\{\frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t}\}$$
$$\leq e^{-2\hat{x}'_{k,\ddot{j},t-1}(B_{k,\ddot{j},t})^2} = e^{-2\hat{x}'_{k,\ddot{j},t-1}\frac{2\ln t}{\hat{x}'_{k,\ddot{j},t-1}}}$$
$$= e^{-4\ln t} = t^{-4}. \quad (61)$$

Therefore,

$$\mathrm{P}\{\frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t}, \forall m_k \in \mathcal{M}, \breve{j}, \ddot{j} = 1,2,\cdots,J+1\}$$
$$= \prod_{k=1}^{K} \mathrm{P}\{\frac{1+\ddot{F}_k(t)}{\bar{\theta}_{k,\ddot{j},t-1}} \leq \frac{1+\ddot{F}_k(t)}{\ddot{\theta}_{k,j}} - B_{k,\ddot{j},t}\} \leq t^{-4K}. \quad (62)$$

Similarly, the probability of the case (b) is also t^{-4K}. By the union bound, the probability that one of the three cases happens is $2t^{-4K}$ plus whatever the probability of case (c) being true. We can make case (c) always false by a well-chosen m. Note that $\ddot{\theta}_{k,j}$ and $\breve{\theta}_{k,j}$ can be less than 1 through adjusting parameters. Case (c) can then be transformed as

$$\ddot{\theta}_{k,j} - \breve{\theta}_{k,j} - \frac{1}{\sqrt{\frac{8\ln(t)}{m}}} < 0. \quad (63)$$

We can derive that when $m > 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2 \ln(t)$, case (c) is false.

Since the expected value of an event is just its probability of occurrence, the expectation of (58) is

$$\mathbb{E}[\chi_{k,\breve{j},T}] \leq 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2 \ln(t) + \sum_{t=1}^{+\infty} \sum_{\hat{x}_{k,\breve{j},t-1}=m}^{t} \sum_{\hat{x}'_{k,\ddot{j},t-1}=1}^{t} 2t^{-4K}$$
$$\leq 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2 \ln(T) + 1 + \sum_{t=1}^{+\infty} \sum_{\hat{x}_{k,\breve{j},t-1}=1}^{t} \sum_{\hat{x}'_{k,\ddot{j},t-1}=1}^{t} 2t^{-4K}$$
$$= 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2 \ln(T) + 1 + \sum_{t=1}^{+\infty} 2t^{-4K+2}. \quad (64)$$

Taking (64) into (52), we can derive the upper bound of the leaning regret R as

$$R = \sum_{k=1}^{K} \sum_{\breve{j}=1}^{J+1} \Delta\theta_{k,\ddot{j},\breve{j}} \mathbb{E}[\chi_{k,\breve{j},T}]$$
$$\leq 8(J+1) \sum_{k=1}^{K} (\Delta\theta_{k,\ddot{j},\breve{j}})^3 \ln(T) + K(J+1)\Delta\theta_{k,\ddot{j},\breve{j}}$$
$$+ (J+1) \sum_{k=1}^{K} \sum_{t=1}^{+\infty} [2t^{-4K+2} \Delta\theta_{k,\ddot{j},\breve{j}}]. \quad (65)$$

This completes the proof of Theorem 2.

Appendix C
Proof of Theorem 3

From the concept of learning regret, we can derived

$$
\begin{aligned}
&-V_k z_{k,\breve{j},t}+\alpha_k\breve{N}_k(t)E_{k,\breve{j},t}+\beta_k\breve{F}_k(t)(\eta_k-\sum_{j=1}^{J}x_{k,\breve{j},t})\\
&\leq R_{k,t}-V_k z_{k,\ddot{j},t}+\alpha_k\ddot{N}_k(t)E_{k,\ddot{j},t}\\
&+\beta_k\ddot{F}_k(t)(\eta_k-\sum_{j=1}^{J}x_{k,\ddot{j},t}), \qquad (66)
\end{aligned}
$$

where $R_{k,t}$ represents the learning regret of m_k in the t-th slot and it satisfies

$$
\sum_{t=1}^{T}\sum_{k=1}^{K}R_{k,t}=R. \qquad (67)
$$

Then, we can get the following inequality must be satisfied

$$
\begin{aligned}
V_k z_{k,\breve{j},t}\geq V_k z_{k,\ddot{j},t}-R_{k,t}-\alpha_k\ddot{N}_k(t)E_{k,\ddot{j},t}\\
-\beta_k\ddot{F}_k(t)(\eta_k-\sum_{j=1}^{J}x_{k,\ddot{j},t}) \qquad (68)
\end{aligned}
$$

By summing over $k=1,2,\cdots,K$ and $t=1,2,\cdots,T$, (68) can be derived as

$$
\begin{aligned}
&V_k\sum_{k=1}^{K}\sum_{t=1}^{T}x_{k,\breve{j},t}z_{k,\breve{j},t}=V_k\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=1}^{J+1}\breve{x}_{k,j,t}\breve{z}_{k,j,t}\\
&\geq V_k\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=1}^{J+1}\ddot{x}_{k,j,t}\ddot{z}_{k,j,t}-R-\alpha_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{N}_k(t)\ddot{E}_{k,j,t}\\
&-\beta_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{F}_k(t)(\eta_k-\sum_{j=1}^{J}x_{k,\ddot{j},t})\\
&\geq\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=1}^{J+1}x^*_{k,j,t}z^*_{k,j,t}-R\\
&-\alpha_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{N}_k(t)\ddot{E}_{k,j,t}-\frac{KT^2B_{max}}{V_k}\\
&-\beta_k\sum_{k=1}^{K}\sum_{t=1}^{T}\ddot{F}_k(t)(\eta_k-\sum_{j=1}^{J}x_{k,\ddot{j},t}). \qquad (69)
\end{aligned}
$$

This completes the proof of Theorem 3.

Appendix D
Proof of Theorem 7

Based on (64), we can derive that after $\lceil 8(\Delta\theta_{k,\ddot{j},\breve{j}})^2\ln(t)\rceil$ times of selecting a non-optimal option, the probability of selecting a non-optimal option at the t-th slot is upper bounded by $2t^{-4K}$. Specifically, as $t\to+\infty$, the upper bound converges to 0.

This completes the proof of Theorem 7.

References

[1] E. Lohan, M. Koivisto, O. Galinina, S. Andreev, A. Tolli, G. Destino, M. Costa, and K. Leppanen, "Benefits of positioning-aided communication technology in high-frequency industrial IoT," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 142–148, Dec. 2018.

[2] A. Ali, L. Feng, A. Bashir, S. A. El-Sappagh, S. Ahmed, M. Iqbal, and G. Raja, "Quality of service provisioning for heterogeneous services in cognitive radio-enabled Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. PP, no. 99, pp. 1–15, Oct. 2018.

[3] A. Musaddiq, Y. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. Kim, "A survey on resource management in IoT operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, Feb. 2018.

[4] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M. Zhou, "FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4388–4400, Jun. 2019.

[5] M. Islam, A. Taha, and S. Akl, "A survey of access management techniques in machine type communications," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 74–81, Apr. 2014.

[6] Y. Mao, J. Zhang, and K. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[7] A. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, and N. Qureshi, "An optimal multi-tier resource allocation of cloud RAN in 5G using machine learning," *Trans. Emerg. Telecommun. Technol.*, vol. PP, no. 99, pp. 1–12, May. 2019.

[8] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. PP, no. 99, pp. 1–11, Jul. 2018.

[9] ——, "Towards traffic load balancing in drone-assisted communications for IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3633–3640, Apr. 2019.

[10] E. Markakis, I. Politis, A. Lykourgiotis, Y. Rebahi, G. Mastorakis, C. Mavromoustakis, and E. Pallis, "Efficient next generation emergency communications over multi-access edge computing," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 92–97, Nov. 2017.

[11] B. Omoniwa, R. Hussain, M. Javed, S. Bouk, and S. Malik, "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4118–4149, Jun. 2019.

[12] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software defined machine-to-machine communication for smart energy management," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 52–60, Oct. 2017.

[13] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 4–16, Dec. 2013.

[14] Z. Zhou, Y. Guo, Y. He, X. Zhao, and W. Bazzi, "Access control and resource allocation for M2M communications in industrial automation," *IEEE Trans. Ind. Informat*, vol. 15, no. 5, pp. 3093–3103, May 2019.

[15] Y. Yuan, T. Yang, H. Feng, and B. Hu, "An iterative matching-Stackelberg game model for channel-power allocation in D2D underlaid cellular networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7456–7471, Nov. 2018.

[16] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.

[17] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sept. 2019.

[18] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017.

[19] T. Sanguanpuak, S. Guruacharya, N. Rajatheva, M. Bennis, and M. Latva-Aho, "Multi-operator spectrum sharing for small cell networks: A matching game perspective," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3761–3774, Jun. 2017.

[20] X. Chen and J. Huang, "Distributed spectrum access with spatial reuse," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 593–603, Mar. 2013.

[21] R. Sutton and A. Barto, *Reinforcement Learning: A Introduction*. Cambridge, MA: USA:MITP, 2018.

[22] C. Tekin and E. Turay, "Multi-objective contextual multi-armed bandit with a dominant objective," *IEEE Internet Things J.*, vol. 66, no. 14, pp. 3799–3813, Jul. 2018.

[23] Y. Xu, A. Anpalagan, Q. Wu, L. Shen, Z. Gao, and J. Wang, "Decision-theoretic distributed channel selection for opportunistic spectrum access: Strategies, challenges and solutions," *IEEE Commun. Surveys Tuts*, vol. 15, no. 4, pp. 1689–1713, Apr. 2013.

[24] S. Oh and K. Li, "BER performance of BPSK receivers over two-wave with diffuse power fading channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1448–1454, Jul. 2005.

[25] C. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[26] S. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5225–5240, Aug. 2018.

[27] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[28] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[29] X. Liu, M. Jia, X. Zhang, and W. Lu, "A novel multichannel Internet of Things based on dynamic spectrum sharing in 5G communication," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 5962–5970, Aug. 2019.

[30] Y. Li, Q. Yin, L. Sun, H. Chen, and H. Wang, "A channel quality metric in opportunistic selection with outdated CSI over Nakagami-m fading channels," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 1427–1432, Mar. 2012.

[31] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA: USA: Morgan and Claypool, 2010.

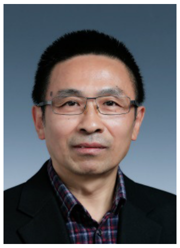
[32] S. Lakshminarayana, M. Assaad, and M. Debbah, "Transmit power minimization in small cell networks under time average QoS constraints," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2087–2103, Oct. 2015.

[33] M. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[34] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, May. 2002.

Haijun Liao is working toward the Ph.D degree at School of Electrical and Electronic Engineering, North China Electric Power University, China. She has participated in several national-level projects and provincial and ministerial level projects, such as the National Natural Science Foundation of China, the Beijing Natural Science Foundation. She served as a reviewer for IEEE Internet of Things Journal, IEEE Access, EURASIP Journal on Wireless Communications and Networking, and other international journals. She was the recipient of the IEEE IWCMC 2019 Best Paper Award. Her research interests mainly focus on resource allocation in machine-to-machine (M2M) communications, smart grid communications, and Internet of things (IoT).

Zhenyu Zhou (M'11-SM'17) received his M.E. and Ph.D degree from Waseda University, Tokyo, Japan in 2008 and 2011 respectively. From September 2012 to April 2019, he was an Associate Professor at School of Electrical and Electronic Engineering, North China Electric Power University, China. Since April 2019, he has been a full professor at the same university. He served as an Associate Editor for IEEE Internet of Things Journal, IEEE Access, EURASIP Journal on Wireless Communications and Networking, and a Guest Editor for IEEE Communications Magazine, IEEE Transactions on Industrial Informatics, and Transactions on Emerging Telecommunications Technologies. He was the recipient of the IET Premium Award in 2017, the IEEE Globecom 2018 Best Paper Award, the IEEE International Wireless Communications and Mobile Computing Conference (IWCMC) 2019 Best Paper Award, and the IEEE Communications Society Asia-Pacific Board Outstanding Young Researcher. His research interests mainly focus on resource allocation in device-to-device (D2D) communications, machine-to-machine (M2M) communications, smart grid communications, and Internet of things (IoT). He is a senior member of IEEE, Chinese Institute of Electronics (CIE), and China Institute of Communications (CIC).

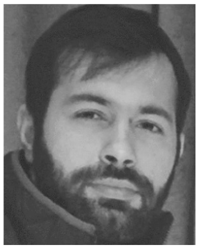


Xiongwen Zhao (SM'06) received the Ph.D. degree (Hons.) from the Helsinki University of Technology (TKK), Finland, in 2002. From 1992 to 1998, he was with the Laboratory of Communications System Engineering, China Research Institute of Radiowave Propagation, where he was a Director and a Senior Engineer. From 1999 to 2004, he was with the Radio Laboratory, TKK, as a Senior Researcher and a Project Manager in the areas of MIMO channel modeling and measurements at 2, 5, and 60 GHz as well as UWB. From 2004 to 2011, he was with Elektrobit Corporation, Espoo, Finland, as a Senior Specialist at EB Wireless Solutions. From 2004 to 2007, he worked in the European WINNER Project as a Senior Researcher in MIMO channel modeling for 4G radio systems. From 2006 to 2008, he also worked in the field of wireless network technologies such as WiMAX and wireless mesh networks. From 2008 to 2009, he worked in mobile satellite communications for GMR-1 3G, DVB-SH RF link budget, and antenna performance evaluations. He is currently a responsibility Professor in Information and Communication Discipline at North China Electric Power University, Beijing, and chairs several projects at the National Science Foundation of China, the Key Program of the Beijing Municipal Natural Science Foundation, Beijing Municipal Science and Technology Commission, and the State Key Laboratories and Industries on radio channel and wireless power communications research. He is a Fellow of the Chinese Institute of Electronics and a Senior Member of IEEE. He was a recipient of the IEEE Vehicular Technology Society Neal Shepherd Memorial Best Propagation Paper Award, in 2014. He served as a TPC Member, the Session Chair, and a Keynote Speaker for numerous international and national conferences. He is a reviewer of the IEEE transactions, journals, letters, and conferences.

Lei Zhang is with with Shandong Electric Power Research Institute for State Grid Corporation of China, Jinan, China. His research interests mainly focus on industrial Internet of things (IoT) and smart grid.

Shahid Mumtaz (SM'16) received the M.Sc. degree in electrical and electronic engineering from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2006, and the Ph.D. degree in electrical and electronic engineering from the University of Aveiro, Portugal, in 2011. He has more than ten years of wireless industry experience. He is currently a Senior Research Scientist with the Instituto de Telecomunicaes, Aveiro, Portugal. Prior to his current position, he was a Research Intern at Ericsson and Huawei Research Labs, Karlskrona, Sweden, in 2005. He has more than 150 publications in international conferences, journal papers, and book chapters. He was a recipient of the Alain Bensoussan Fellowship by ERCIM to pursue research in communication networks for one year at the VTT Technical Research Centre of Finland in 2012. He was nominated as the Vice Chair for the IEEE new standardization on P1932.1: Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks. He is also actively involved in 3GPP standardization on LTE release 12 onwards, along with major manufacturers. He is an ACM Distinguished Speaker.

Alireza Jolfaei received the Ph.D. degree in Applied Cryptography from Griffith University, Gold Coast, Australia. He is an Assistant Professor in Cyber Security at Macquarie University, Sydney, Australia. Prior to this appointment, he worked as an Assistant Professor at Federation University Australia and Temple University in Philadelphia, USA. His current research areas include Cyber Security, IoT Security, Human-in-the-Loop CPS Security, Cryptography, AI and Machine Learning for Cyber Security. He has received multiple awards for Academic Excellence, University Contribution, and Inclusion and Diversity Support. He is a founding member of Federation University IEEE Student Branch. He is a Senior Member of the IEEE and an ACM Distinguished Speaker on the topic of Cyber-Physical Systems Security.



Syed Hassan Ahmed (SM'18) received the bachelors degree in computer science from the Kohat University of Science and Technology, Pakistan, and the master combined Ph.D. degree from the School of Computer Science and Engineering, Kyungpook National University (KNU), South Korea. He is currently an Assistant Professor with the Department of Computer Science, Georgia Southern University, Statesboro, GA, USA. He is also leading the Wireless Internet and Networking Systems Lab. Previously, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Central lorida, Orlando,lorida, Orlando,F FL, USA. In 2015, he was also a Visiting Researcher with Georgia Tech, Atlanta, USA. Overall, he has authored/co-authored over 150 international publications, including journal articles, conference proceedings, book chapters, and three books. His research interests include sensor and ad hoc networks, cyber-physical systems, vehicular communications, and future Internet. In 2016, his work on robust content retrieval in future vehicular networks lead him to win the Qualcomm Innovation Award from KNU. He is currently a member of the Board of Governors and the IEEE VTS liaison to the IEEE Young Professionals Society. Since 2018, he has also been an ACM Distinguished Speaker.

Ali Kashif Bashir received the Ph.D. degree from computer science and engineering from Korea University, Seoul, South Korea, in 2012. He is a Senior Lecturer with Manchester Metropolitan University, Manchester, U.K. In the past, he held appointments with Osaka University, Japan; Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. His research interests include: cloud computing, NFV/SDN, network virtualization, network security, IoT, computer networks, RFID, sensor networks, wireless networks, and distributed computing. Dr. Bashir is the Editor-in-Chief of the IEEE Internet Technology Policy Newsletter and the IEEE Future Directions Newsletter. He is an Editorial Board Member of journals, such as the IEEE Access, the Journal of Sensor Networks, and the Data Communications.