Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000
DOI: xxx/xxxx

ARTICLE TYPE

Joint Energy and Latency Optimization for Upstream IoT Offloading Services in Fog Radio Access Networks

Duc-Nghia Vu | Nhu-Ngoc Dao | Yongwoon Jang | Woongsoo Na | Young-Bin Kwon | Hyunchul Kang | Jason J. Jung | Sungrae Cho*

School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea

Correspondence
*Sungrae Cho, School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea. Email: srcho@cau.ac.kr

Present Address
221 Heukseok-dong, Dongjak-gu, Seoul 06974, Republic of Korea.

Abstract

Recently, the emergence of fog computing and big Internet of things (IoT) data have been considered as the main representatives identifying fifth generation (5G) mobile networks. In 5G, cloudization is extended from the core to the access tiers, referred to as fog radio access networks (FRANs). FRANs provide ultralow-latency offloading services to a massive number of IoT devices in their proximity. In this paper, we propose a joint energy and latency optimization (JELO) scheme for upstream IoT offloading services in FRANs. JELO scheme controls the offloaded task assignment among fog-enabled eNodeBs (FNs) with strict consideration of the systematic resources and specific characteristics of individual tasks. The joint objective function aims at optimizing the energy consumption and offload latency for entire networks. Simulation results demonstrate that the JELO scheme outperforms existing approaches in terms of the energy consumption and load balancing while maintaining IoT service satisfaction.

KEYWORDS:
joint optimization, energy-aware, latency-aware, upstream offloading, fog radio access network

1 | INTRODUCTION

Dealing with the rapid emergence of the Internet of things (IoT) has been considered a big challenge in fifth generation (5G) mobile networks. As recently reported by Gartner[1], nearly 20 billion IoT-connected devices are estimated to be online in 2020. IoT categories involve diverse applications in all of the cross-industry, vertical-specific, and consumer segments. Despite of the heterogeneity in application, IoT devices are characterized by a low energy consumption and limited computational capability while the IoT services increasingly require low latency, complex execution, and big data analysis[2,3]. This advanced IoTization forces the 5G networking infrastructure to integrate novel technologies for satisfying a massive number of connections and a huge volume of offloaded IoT traffic. In such a context, the computing capability has been extended from the cloud to the 5G eNodeBs at the edge (a.k.a. fog-enabled eNodeBs), resulting in fog radio access networks (FRANs)[4,5]; see Fig. 1. FRANs provide ultralow-latency offloading services to a massive number of IoT devices in their proximity, which is especially beneficial for time-sensitive applications[6,7]. The FRANs integrate fog computing into radio access networks (RANs) by adding a substantial amount of storage, communication, and computation resources. In FRANs, high power nodes (HPNs) are deployed to provide a wide-area coverage and execute the control operations. The HPNs are interconnected via crosshaul links and coordinated by orchestrators. In a fog computing's perspective, the macro eNodeBs (MeNBs), pico eNodeBs (PeNBs), and extended remote radio heads (eRRHs) are generally considered as fog nodes (FNs). The FNs may be equipped with local caches, where the

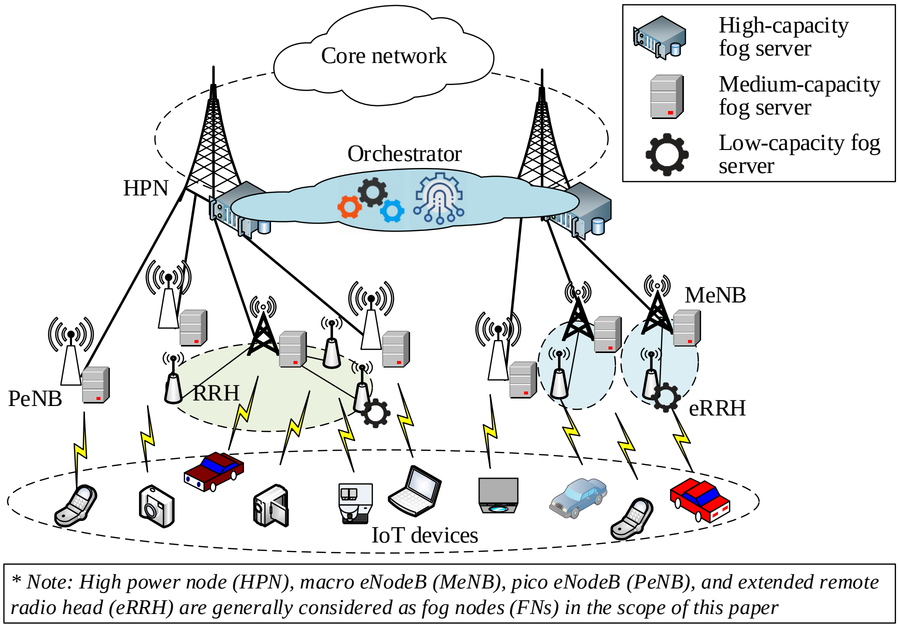

FIGURE 1 IoT offloading services in fog radio access network.

interesting contents can be stored, as well as computing processors for offloaded service handling. These FNs are controlled by HPNs.

In order to minimize the task offloading latency for time-sensitive IoT applications, IoT devices typically intend to associate with the highest-capacity fog-enabled eNodeBs (FNs) in terms of the CPU power as well as wireless channel quality. These behaviors may lead the FNs to be overloaded owing to the massive number of requests. In this circumstance, there exist unfair conditions among FNs [8,9]. As a result, forthcoming IoT devices possibly suffer from a low quality of service (QoS) problem compared to the previous ones. Finally, the total latency and energy efficiency for offloaded activities may not be optimal from a network-wide perspective since some FNs are free, while others are over capacity. Especially, these problems are severe in the 5G paradigm, where green and real-time communication has been considered one of the pilot features [10,11].

Our literature review discovers that the joint latency and energy consumption problems have not attracted sufficient attention form the research community [12,13]. According our taxonomy, existing studies can be classified into either intrinsic or extrinsic approaches. Intrinsic approaches mainly consider the internal features of FNs, such as the CPU frequency, cache, temporary buffer, communication resource, and battery, to develop objectives and solutions. On the other hand, extrinsic approaches focus on the distinguished characteristics of the served entities (i.e., IoT devices and applications) to design appropriate schemes for satisfying them effectively. Although these two types of approaches introduced improvements in the computational performance, they lack a comprehensive consideration that balances both intrinsic and extrinsic features to provide a flexible optimization solution for energy consumption and latency balancing.

In this paper, we propose a joint energy and latency optimization (referred to as JELO) scheme for upstream IoT offloading services in FRANs. A balance between the energy consumption and the offload latency, which are generated by task transmission and execution, has been developed as an objective function. Distinguished from the existing approaches, both systematic resources (CPU frequency, buffer size, and quality of wireless channels) and task characteristics (size, complexity, and execution deadline) are strictly considered as the main constraints to design the final solution. In order to reduce the complexity of the optimization problem, a Lagrangian relaxation method is used to separate the problem into two cooperative subfunctions, which are resolved by the *M 0/1-knapsack* and semi-assignment schemes, respectively. As a result, the contributions of this paper can be summarized as follows:

- First, JELO provides energy-efficient end-to-end offloading for time-sensitive IoT services while ensuring controllable latency thresholds. In addition, IoT device associations is driven by the computational performance of the networks to better adapt task assignments within various network environments.

- Second, JELO enables relaxation for the objective function, which is an integer linear programming problem, in order to achieve an approximation of the optimal result with reasonable costs.
- Third, comprehensive simulations and analyses have been performed to demonstrate the computational improvement of the JELO scheme compared to existing schemes in terms of the energy consumption and offloading QoS.

The rest of this paper is organized as follows. Section 2 reviews the existing works in the literature. Section 3 analyzes the problem in detail. Section 4 describes our proposed JELO scheme along with the mathematical expressions and their proofs. Then, Section 5 presents the simulation results. Finally, Section 6 concludes the paper.

2 | LITERATURE REVIEW

Resource management techniques have been typically taken into account as a major research issues in FRANs for either energy efficiency or offload latency purposes. There is a significant number of researchers tackling this problem[14,15,16]. Unfortunately, there have been few proposals to resolve the joint optimization between the energy efficiency and the offload latency[12,13]. As mentioned in Section 1, our taxonomy consists of two main categories: intrinsic and extrinsic approaches.

As an example of intrinsic approaches, Guan *et al.*[17] formulated the joint optimization of the computation and resource management in FNs, aiming at providing energy-efficient offloading services. The objective function is addressed by using an iterative algorithm and a matching-based sub-optimal algorithm. On the contrary, Zhang *et al.*[18] considered the energy capacity and sensitive latency to develop an energy-aware offloading scheme. Since the problem is a mixed-integer nonlinear problem, an iterative search algorithm combined with an interior penalty policy has been proposed to find the optimal value. Considering a small-cell 5G network model, Yang *et al.*[19] formulated an offloading function, which targets the energy consumption of the system at all system entities. An artificial fish swarm algorithm (AFSA)-based scheme has been proposed to solve the optimization problem. On the other hand, Jeong *et al.*[20] considered the trust issues in fog computation by using a Vickrey-Clarke-Groves (VCG)-auction-based hierarchical trust computing algorithm to manage resource allocation. Considering the resource limitations and the quality of service in FRANs, Chabbouh *et al.*[21] proposed joint service offloading and scheduling to handle the offloaded tasks within a reduction in the execution cost. An energy-efficient multisite offloading policy (EMOP) using a Markov decision[22] process has been proposed to balance the task processing among FNs in FRANs. The network is transformed into a graph model considering a delay constraint.

Adopting extrinsic approaches, Munoz *et al.*[23] assumed that multiple antennas are available simultaneously at the user devices and FNs. A partial closed-form expression of the transmission features such as the transmission power, precoder, and rate in both the uplink and downlink was developed in order to obtain the optimal energy and latency tradeoff for application offloading. A simple 1-D convex numerical optimization technique was used to resolve the objective function. Focusing on other scenarios where offloading services are provided by multiple virtual machines through backhaul links[24], Lagen *et al.*[25] optimized the offloading strategy at the user devices in order to minimize the energy consumption subject to latency constraint. The channel conditions at the air interface and the backhaul link capacities are considered in the development of the solution. On the other hand, a Lyapunov optimization-based dynamic computation offloading (LODCO) algorithm[26] has been proposed to balance the energy consumption and offloading latency for user devices. Lyu *et al.*[27] proposed asymptotically optimal task admission for delay-sensitive applications in fog computing. The mixed-integer programming of task admission is transformed into an integer programming (IP) problem with the optimal substructure by pre-admitting resource-restrained devices. Utilizing network-assisted D2D collaboration[28,29], Pu *et al.*[30] proposed efficient task scheduling policies for energy-efficient and incentive-aware offloading in a D2D fogging framework. The policies adapt to various features of the task type, the user amount, and the task generation frequency.

Despite the merits of the two types of approaches, both intrinsic and extrinsic features have not been comprehensively considered in order to provide an adaptive joint optimization solution for the energy consumption and latency.

3 | PROBLEM STATEMENT

We consider an FRAN system model including M FNs that are geographically deployed according to the IoT traffic intensity and service requirements. The FNs are equipped with multiple antennas, local caches, and possible fog-based computation that might provide computation offloading services to the resource-constrained IoT devices. Without loss of generality, we assume

TABLE 1 Notation Definitions.

Symbol	Definition
Ω	A set of FNs
M	Number of FNs
$\Gamma[t]$	A set of tasks generated by IoT devices at timeslot t
$\Gamma_i[t]$	A set of tasks assigned to the *i-th* FN at timeslot t
N	Number of tasks generated by IoT devices at timeslot t
υ_j	Desired rate of the *j-th* IoT device
r_{ij}	Number of resource blocks required from the *i-th* FN to satisfy the data rate υ_j of the *j-th* IoT device
C_i	Radio resource capacity of the *i-th* FN according to the RB unit
F_i	Computation capability of the *i-th* FN
w_j	Task generated by the *j-th* IoT device
u_j	Upload data size of task w_j
d_j	Response data size of task w_j
c_j	Average computational complexity of task w_j
τ_j	Execution deadline to accomplish task w_j
E_j^c	Energy consumption of the *i-th* FN for computation of task w_j
E_j^t	Energy consumption for data transmission of task w_j
E_j	Total energy consumption of the IoT device and FNs when task w_j is offloaded to the FNs
L_j^c	Computation latency for executing task w_j
L_j^t	Transmission latency to accomplish task w_j
L_j	Total latency in the network when task w_j is offloaded to the FNs
λ_j	Amount of computation specified by the computing cycles to execute the task w_j of the *j-th* IoT device
κ	The coefficient denoting the energy consumed per CPU cycle
P	Power consumption when using 1 RB for transmitting data
$Q[t]$	Computing queue of the *i-th* FN at timeslot t
α	A balance coefficient for energy and latency in the network

that there are N IoT devices that generate their tasks and offload them to the FNs. The IoT tasks are processed and responded to the IoT devices on-demand. The notation used in this paper is summarized in Table 1.

We assume that the *j-th* IoT device assigns a task w_j to the FNs and receives response data with the same required data rate υ_j due to the trivial quality of the signal change for uplink and downlink transmission[31]. The number of resource blocks (RBs) r_{ij} that the *i-th* FN must assign to the *j-th* IoT device is derived as

$$r_{ij} = \left\lceil \frac{\upsilon_j}{\Delta f \log_2(1 + \mathrm{SINR}_{ij})} \right\rceil , \tag{1}$$

where $r_{ij} \in \mathbb{N}$, Δf is the bandwidth that 1 RB utilizes during 1 ms (i.e., 180 KHz[32]), and SINR_{ij} is the signal-to-interference-plus-noise ratio on the data channel between the *i-th* FN and the *j-th* IoT device.

In terms of task execution offloading, task w_j of the *j-th* IoT device is characterized by a four dimensional feature vector given by

$$\overrightarrow{w_j} \triangleq \left[u_j, c_j, d_j, \tau_j\right] , \tag{2}$$

where u_j and d_j are respectively defined as the upload data size and response data of the task according to the bit unit, c_j is average computational complexity of the task, and τ_j is the execution deadline to accomplish the task.

When a task is assigned to the FNs, the FNs perform task execution. Let λ_j be amount of computation that is determined by the computing cycle unit to execute w_j. λ_j is obtained by

$$\lambda_j = u_j c_j. \tag{3}$$

Accordingly, the energy consumption E_j^c of the *i-th* FN for task execution w_j [33] is given by

$$E_j^c = \kappa F_i^2 \lambda_j, \tag{4}$$

where κ is the coefficient denoting the energy consumed per CPU cycle, and F_i is the CPU computation capability of the *i-th* FN. Meanwhile, the energy consumption E_j^t for the data transmission of task w_j to the *i-th* FN in the network is derived as

$$E_j^t = r_{ij} P, \tag{5}$$

where P is power consumption when using 1 RB for transmitting data[2]. The total energy consumption E_j for the IoT device and FNs when task w_j is offloaded to *i-th* FN is given by

$$E_j = E_j^c + E_j^t, \tag{6}$$

In order to ensure service quality requirements (e.g., the data rate, and the latency), each task w_j is required to execute in a time duration of τ_j. The computation latency L_j^c for executing task w_j at the *i-th* FN is determined by

$$L_j^c = \frac{Q_i[t] + \lambda_j}{F_i}, \tag{7}$$

where $Q_i[t]$ denotes the computing queue of the *i-th* FN at timeslot t. Similarly, we can obtain the transmission latency L_j^t for uploading the input data and receiving the response data of task w_j from the *i-th* FN by

$$L_j^t = \frac{u_j + d_j}{v_j}, \tag{8}$$

Accordingly, total latency in the network when task w_j is offloaded to the FNs is derived as

$$L_j = L_j^c + L_j^t. \tag{9}$$

In the context of the IoT era, the devices are required to be operated under rigorous requirements featuring a high precision, real-time responses, and high automation. However, almost all IoT devices are lightweight embedded platforms with the restricted computing resources, storage, and power transmission. Hence, it is inappropriate to execute real-time services using the constrained resources of IoT devices. In FRANs, the heavy tasks (i.e., high λ_j) and time-sensitive services (i.e., very low τ_j) are offloaded to the FNs which are equipped with processors with a high processing power. A problem emerges when the enormous number of tasks of the IoT devices is offloaded to the FNs. This consumes a large amount of FN resources for handling these tasks. If these tasks are not assigned optimally to the FNs, this might lead to an overloading issue due to inefficient resource utilization. In the energy-aware approach, the IoT devices prefer to connect and assign tasks to the FNs that have a high SINR. This is because these devices will consume a lower transmission energy since a lower number of the RBs r_{ij} are used. Meanwhile, the latency-aware approach drives IoT tasks to FNs that have a high processing power to reduce the computation latency L_j^c. As a consequence of this approach, these FNs reach overcapacity owing to a very large number of assigned tasks. These two approaches cause an imbalanced load among FNs, resulting in increases in the energy consumption and latency. This motivates the combined optimization for the energy and latency of the whole network. The problem that jointly considers the energy and latency for uploading the tasks of the IoT devices to FNs is formulated as

$$(\mathcal{P}) \quad \min \sum_{j=1}^{N} \left[\alpha E_j + (1-\alpha) L_j\right], \tag{10}$$

where α is a balance coefficient. This primarily effects the joint energy and latency optimization problem. If α is large enough, this means that the network energy consumption minimization is more important. Otherwise, a small α means that the network latency optimization is mainly considered. An appropriate value of α results in the optimal energy and latency for entire networks. In this paper, we propose the JELO algorithm to address problem $\mathcal{P}$ in order to achieve the network energy and latency minimization. The JELO scheme is presented in the next section.

4 | JOINT ENERGY AND LATENCY OPTIMIZATION

With the aim of achieving the optimal task assignments, we address problem $\mathcal{P}$ to obtain the combined optimization of the energy consumption and latency for whole networks. Problem $\mathcal{P}$ is concretely reformulated as

$$(\mathcal{P}) \quad \min \sum_{i=1}^{M}\left[\alpha\left(\kappa F_i^2 \sum_{j=1}^{N} x_{ij}\lambda_j + \sum_{j=1}^{N} x_{ij} r_{ij} P\right) + (1-\alpha)\left(\sum_{j=1}^{N} x_{ij}\frac{Q_i[t]+\lambda_j}{F_i} + \sum_{j=1}^{N}\frac{u_j+d_j}{v_j}\right)\right] \tag{11}$$

$$\text{s.t.}\ \overrightarrow{w_j}\left(u_j, c_j, d_j, \tau_j\right) \in \Gamma[t], \forall j = 1,2,\cdots,N, \tag{12}$$

$$\sum_{j=1}^{N} x_{ij} r_{ij} \le C_i, \forall i = 1,2,\cdots,M, \tag{13}$$

$$\sum_{i=1}^{M}\left(x_{ij}\frac{Q_i[t]+\lambda_j}{F_i} + \frac{u_j+d_j}{v_j}\right) \le \tau_j, \forall j = 1,2,\cdots,N, \tag{14}$$

$$\sum_{i=1}^{M} x_{ij} = 1, \forall j = 1,2,\cdots,N, \tag{15}$$

$$x_{ij} \in \{0,1\}, \forall i = 1,2,\cdots,M, \forall j = 1,2,\cdots,N. \tag{16}$$

The constraint in (13) ensures that the number of RBs allocated to IoT devices does not exceed the capacity of the FNs. The constraint in (14) guarantees the completion deadline of a task when it is assigned to the FNs. The constraint in (15) ensures that one task is assigned to only one FN. It is observed that problem $\mathcal{P}$ can be transformed into an integer Programming problem as follows:

$$(\mathcal{P}) \quad \min \sum_{i=1}^{M}\left[\sum_{j=1}^{N} x_{ij}\left(\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + (1-\alpha)\frac{Q_i[t]+\lambda_j}{F_i}\right) + (1-\alpha)\sum_{j=1}^{N}\frac{u_j+d_j}{v_j}\right] \tag{17}$$

$$\text{s.t.}\ \sum_{i=1}^{M} x_{ij}\frac{Q_i[t]+\lambda_j}{F_i} \le \tau_j - \frac{u_j+d_j}{v_j}, \forall j = 1,2,\cdots,N, \tag{18}$$

$$(12), (13), (15), (16).$$

For a given α, the value of $\sum_{i=1}^{M}\left[(1-\alpha)\sum_{j=1}^{N}\frac{u_j+d_j}{v_j}\right]$ is constant for each timeslot. Hence, problem $\mathcal{P}$ is equivalent to

$$(\mathcal{P}) \quad \min \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\left[\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + (1-\alpha)\frac{Q_i[t]+\lambda_j}{F_i}\right] \tag{19}$$

$$\text{s.t.}\ (12), (13), (15), (16), (18).$$

In order to address problem $\mathcal{P}$, we formulate problem $\mathcal{P}'$ as

$$(\mathcal{P}') \quad \min \quad \gamma\sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\left[\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right] + \delta\sum_{i=1}^{M}\sum_{j=1}^{N} y_{ij}\left[\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right] \tag{20}$$

$$\text{s.t.}\ \sum_{i=1}^{M} y_{ij} = 1, \forall j = 1,2,\cdots,N, \tag{21}$$

$$x_{ij} = y_{ij}, \forall i = 1,2,\cdots,M, j = 1,2,\cdots,N, \tag{22}$$

$$y_{ij} \in \{0,1\}, \forall i = 1,2,\cdots,M, j = 1,2,\cdots,N, \tag{23}$$

$$(12), (13), (16), (18).$$

where γ and δ are positive parameters. We define $v(\cdot)$ as the optimal objective function value of problem $(\cdot)$. It is observed that problem $\mathcal{P}$ is related to problem $\mathcal{P}'$ in the following way

$$v(\mathcal{P}') = (\gamma+\delta)v(\mathcal{P}), \tag{24}$$

It is recognized that if $\gamma + \delta = 1$, then $v(\mathcal{P}) = v(\mathcal{P}')$. By using the Lagrangian multiplier method[34] for relaxing the constraint in (22), the problem $\mathcal{P}'$ is relaxed as follows:

$$\begin{aligned}(\mathcal{L}(\varepsilon)) \quad \min \quad \mathcal{F}(x,y) = {} & \gamma \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\left[\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right] \\ & + \delta \sum_{i=1}^{M}\sum_{j=1}^{N} y_{ij}\left[\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right] + \sum_{i=1}^{M}\sum_{j=1}^{N}\varepsilon_{ij}(y_{ij}-x_{ij}) \\ \text{s.t. } & (12), (13), (16), (18), (21), (23).\end{aligned} \tag{25}$$

where ε_{ij} is the Lagrangian multiplier for the constraint $x_{ij} = y_{ij}$. The Lagrangian relaxation $\mathcal{L}(\varepsilon)$ can be separated into two problems. One problem is related to the x variables, denoted as $LX(\varepsilon)$. Another problem is related to the y variables, defined as $LY(\varepsilon)$. The problems $LX(\varepsilon)$ and $LY(\varepsilon)$ are derived as

$$\begin{aligned}(LX(\varepsilon)) \quad \min \quad & \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\left[\gamma\left(\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right) - \varepsilon_{ij}\right] \\ \text{s.t. } & \sum_{j=1}^{N} x_{ij} r_{ij} \le C_i, \forall i = 1,2,\cdots,M, \\ & \sum_{i=1}^{M} x_{ij}\frac{Q_i[t]+\lambda_j}{F_i} \le \tau_j - \frac{u_j+d_j}{v_j}, \forall j = 1,2,\cdots,N, \\ & x_{ij} \in \{0,1\}, \forall i = 1,2,\cdots,M, \forall j = 1,2,\cdots,N.\end{aligned} \tag{26}$$

$$\begin{aligned}(LY(\varepsilon)) \quad \min \quad & \sum_{i=1}^{M}\sum_{j=1}^{N} y_{ij}\left[\delta\left(\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right) + \varepsilon_{ij}\right] \\ \text{s.t. } & \sum_{i=1}^{M} y_{ij} = 1, \forall j = 1,2,\cdots,N, \\ & y_{ij} \in \{0,1\}, \forall i = 1,2,\cdots,M, j = 1,2,\cdots,N.\end{aligned} \tag{27}$$

Accordingly, we obtain $v(\mathcal{L}(\varepsilon)) = v(LX(\varepsilon)) + v(LY(\varepsilon))$. We observe that problem $LX(\varepsilon)$ can be separated into M $0/1$-knapsack problems[35] with N variables, where each $KP_i(\varepsilon)$ is defined as

$$\begin{aligned}(KP_i(\varepsilon)) \quad \min \quad & \sum_{j=1}^{N} x_{ij}\left[\gamma\left(\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right) - \varepsilon_{ij}\right] \\ \text{s.t. } & \sum_{j=1}^{N} x_{ij} r_{ij} \le C_i, \\ & x_{ij}\frac{Q_i[t]+\lambda_j}{F_i} \le \tau_j - \frac{u_j+d_j}{v_j}, \forall j = 1,2,\cdots,N, \\ & x_{ij} \in \{0,1\}, \forall j = 1,2,\cdots,N.\end{aligned} \tag{28}$$

The knapsack problem $KP_i(\varepsilon)$ can be efficiently addressed by using dynamic programming. Then, we can derive $v(LX(\varepsilon)) = \sum_{i=1}^{M} v(KP_i(\varepsilon))$. Meanwhile, $LY(\varepsilon)$ is a simple semi-assignment problem. This can be separated into N trivial generalized upper bound (GUB) problems. We directly obtain solution by

$$v(LY(\varepsilon)) = \sum_{j=1}^{N}\left[\delta\left(\alpha\left(\kappa F_{i_j}^2\lambda_j + r_{i_j j}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_{i_j}}\right) + \varepsilon_{i_j j}\right] \tag{29}$$

where $i_j \in \operatorname{argmin}_{i=1,\cdots,M}\left[\delta\left(\alpha\left(\kappa F_i^2\lambda_j + r_{ij}P\right) + \frac{(1-\alpha)(Q_i[t]+\lambda_j)}{F_i}\right) + \varepsilon_{ij}\right]$, and $y_{i_j} = 1, \forall j = 1,\cdots,N$.

Given a value of ε, we can derive the optimal value $v(\mathcal{L}(\varepsilon))$ for problem $\mathcal{P}'$ by solving problems $LX(\varepsilon)$ and $LY(\varepsilon)$. We need to find the optimal ε^* to obtain the final optimal value $v(\mathcal{L}(\varepsilon))$. A subgradient algorithm[36] is proposed to address problem $\mathcal{P}'$, resulting in the optimal solution $v(\mathcal{P})$ for problem $\mathcal{P}$. Particularly, a subgradient for the function $v(\mathcal{L}(\varepsilon))$ at the point ε^k is any

Algorithm 1 Joint Energy and Latency Optimization Algorithm.

Input: Set of FNs Ω, $\Gamma(t)$, α, γ, δ
Output: Set of tasks assigned to the *i-th* FN $\Gamma_i(t), \forall i = 1, 2, \cdots, M$

• Initialization
x^k,y^k, and ε^k are x,y, and ε at the *k-th* step, respectively
ε^0 is the starting point
ξ is the tolerance, φ is the step size
$\nabla\mathcal{L}(\varepsilon)$ is the gradient of $\mathcal{L}(\varepsilon)$ with respect to x and y
repeat
  Given ε^k, solve M *knapsack* problems in (28) to derive x^k and $\upsilon(LX(\varepsilon^k))$
  Solve the semi-assignment problem in (27) to derive y^k and $\upsilon(LY(\varepsilon^k))$
  $\upsilon(\mathcal{L}(\varepsilon^k)) = \upsilon(LX(\varepsilon^k)) + \upsilon(LY(\varepsilon^k))$
  $\varepsilon^{k+1} = \varepsilon^k - \varphi\nabla\mathcal{L}(\varepsilon^k)$
  $k = k + 1$
until $||\nabla\mathcal{L}(\varepsilon^k)|| \leq \xi$
$x^* = x^k$, $y^* = y^k$, and $\varepsilon^* = \varepsilon^k$
From x^*, obtain $\Gamma_i(t)$

vector Θ such that

$$\upsilon(\mathcal{L}(\varepsilon)) \leq \upsilon(\mathcal{L}(\varepsilon^k)) + \Theta(\varepsilon - \varepsilon^k) \quad \text{for all} \quad \varepsilon \in \mathrm{R}^{m\times n}. \tag{30}$$

For each *k-th* step of iteration, we solve $\upsilon(\mathcal{L}(\varepsilon^k))$ and then get the solution (x^k, y^k). The iteration stops when we can obtain the approximate optimal value ε^*, resulting in the optimal values x^* and y^*. The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 addresses the Lagrangian relaxation problem $\mathcal{L}(\varepsilon)$ using the subgradient method, thus obtaining the optimal solution for the equivalent problem $\mathcal{P}'$. With the given γ and δ, the optimal objective value for problem $\mathcal{P}$ can be easily derived following (24). In the proposed scheme, we first arbitrarily initialize the starting point value for the Lagrange multiplier ε^0. For each *k-th* step, we solve problem $\mathcal{L}(\varepsilon)$, aiming to derive x^k and y^k by addressing subproblems $LX(\varepsilon^k)$ and $LY(\varepsilon^k)$. The solution of problem $LX(\varepsilon^k)$ can be obtained by solving M knapsack problem in (28) using dynamic programming, resulting in x^k. Meanwhile, $LY(\varepsilon)$ is the semi-assignment problem. We can derive the optimal solution following (29). The multiplier ε^{k+1} for step $k+1$ will be updated following the gradient $\nabla\mathcal{L}(\varepsilon)$ by $\varepsilon^{k+1} = \varepsilon^k - \varphi\nabla\mathcal{L}(\varepsilon^k)$, where φ is the step size. According to the subgradient method, the value of the objective problem will decrease after each iteration. This means that the result will converge to optimal solution after some number of iterations. Iteration stops when the magnitude of the gradient $||\nabla\mathcal{L}(\varepsilon)||$ is less than a very small tolerance ξ. Then, we obtain the optimal solution x^* for problem $\mathcal{P}$. In other words, we derive the set of optimal tasks $\Gamma_i(t)$ assigned to the *i-th* FN at timeslot t.

5 | PERFORMANCE EVALUATION

5.1 | Simulation Settings and Methodology

In this section, we present a series of simulation studies to evaluate the performance of our proposed JELO scheme. We consider a network model including 20 FNs that are deployed randomly over an area of 500×500 square meters. FNs are equipped with various computing processors with CPU frequencies of $\{1.5, 3.0, 6.0, 12.0, 24.0\}$ GHz. We suppose that there are 300 IoT devices served in the network. With the diversity of IoT services such as sensor reading, motion detection, and video surveillance, each IoT device requires a transmission data rate varying in the range of $[512, 2048]$ kbps with computational complexities of $\{10, 50, 100, 500, 1000\}$ computing cycles/bit[16]. The task execution deadline to satisfy the IoT service requirement is assumed in the range of $[0.1, 1]$ s. Note that the proposed scheme is performed by the orchestrator with an algorithmic execution threshold θ of 5 ms. For convenience, the threshold θ is included into the task execution deadline. In addition, network slicing techniques[37] are utilized in order to separate the uploaded IoT services into groups, which are characterized by latency and/or energy requirements' levels. The optimization schemes are provided for each group accordingly. The detailed simulation parameters are summarized in Table 2.

TABLE 2 Simulation Parameters.

Parameter	Value
Network area	500m×500m
Number of FNs	20
CPU frequency of the FN processors	{1.5, 3.0, 6.0, 12.0, 24.0} GHz
Number of IoT devices	300
Transmission data rate of the IoT devices	512–2048 kbps
Computational complexity of the tasks	{10, 50, 100, 500, 1000} computing cycles/bit
Execution deadline	0.1–1 s
Timeslot duration	0.1 s

In order to demonstrate the performance of the proposed JELO scheme, we compare our scheme with three task assignment schemes including the pattern-identified online task scheduling (PIOTS) [16], offline Hungarian task assignment (OHTA) [38], and online greedy task assignment (OGTA) [39] algorithms. In the PIOTS scheme, offline task scheduling among the FNs is performed on the set of all self-organizing maps (i.e., task patterns) using the Hungarian method to obtain the expected optimal task assignments while minimizing the latency. In a real-time context, the arriving tasks are assigned to FNs on the basic of the expected task assignment. Meanwhile, the OHTA scheme collects all arriving tasks at the local buffer of the FNs during one timeslot. At the end of each timeslot, the optimal task assignment is determined using a repeated Hungarian method for addressing problem $\mathcal{P}$. However, it cannot ensure the resource and execution latency constraints in $\mathcal{P}$. The OGTA scheme aims to minimize the energy and latency for each task. This means that each arrived task is assigned to the FNs that provide the joint energy and latency optimization. For the evaluation metrics, the performance of our proposed scheme is analyzed in terms of the latency, energy consumption, objective function, and task execution rate.

5.2 | Numerical Result Analysis

The balance coefficient α represents the priorities of the latency and energy in the network. In other words, the balance coefficient α harmonizes the task assignment to achieve joint latency and energy optimization. Fig. 2 shows the total network latency and energy consumption of the proposed JELO scheme according to the balance coefficient α. In general, the latency and energy consumption gradually increase when the arrival workload λ (i.e., arriving tasks) increases and $\lambda < 8$ gigacycles. This is reasonable because the enormous number of tasks arriving at the networks lead to an increase in transmission as well as a buffering latency for the IoT tasks at the FNs since the FNs may not be able to provide prompt execution owing to resource constraints. Meanwhile, the numerous tasks also consume a large of amount of energy for the transmission and execution processes. By controlling the factor α, we can orchestrate the energy and latency. It is observed that a higher α leads to a higher network latency and lower network energy consumption since the higher α means that the energy aspect is mainly prioritized. When the arrival workload λ increases to more than 8 gigacycles, the energy and latency irregularly change. Particularly, with $\alpha = 0.1$, the latency decreases rapidly while the energy increases when the workload reaches 10 gigacycles. The reason is because the JELO scheme jointly balances the energy and latency by addressing problem $\mathcal{P}$. When the latency reaches the allowable threshold, the incoming tasks are preferably assigned to FNs that have a large number of computational resources in the processor and/or a lower occupied buffer to process aiming at guaranteeing the task execution deadlines, resulting in an increase in the energy consumption. As a result, it is recognized that we can obtain the best performance in terms of the latency with $\alpha = 0.1$ and the energy with $\alpha = 0.7$. In this paper, we configure $\alpha = 0.1$ as the simulation parameter to target real-time IoT services that require a very low latency.

Fig. 3 depicts the optimal value of the objective function $\mathcal{P}$ of the proposed JELO scheme compared to other task assignment schemes. The simulation results reveal that the objective function values of the JELO and OGTA schemes slightly increase as following the arrival workload increases. It is observed that our proposed JELO scheme obtains the best optimal value for the objective function $\mathcal{P}$. This is because the JELO scheme thoroughly addresses the task assignment problem for joint energy and latency minimization by relaxing the original problem $\mathcal{P}$ into two subproblems to achieve global optimality. On the other hand, the OGTA scheme assigns the incoming tasks to the FNs which provide a minimization of the energy and latency to them.

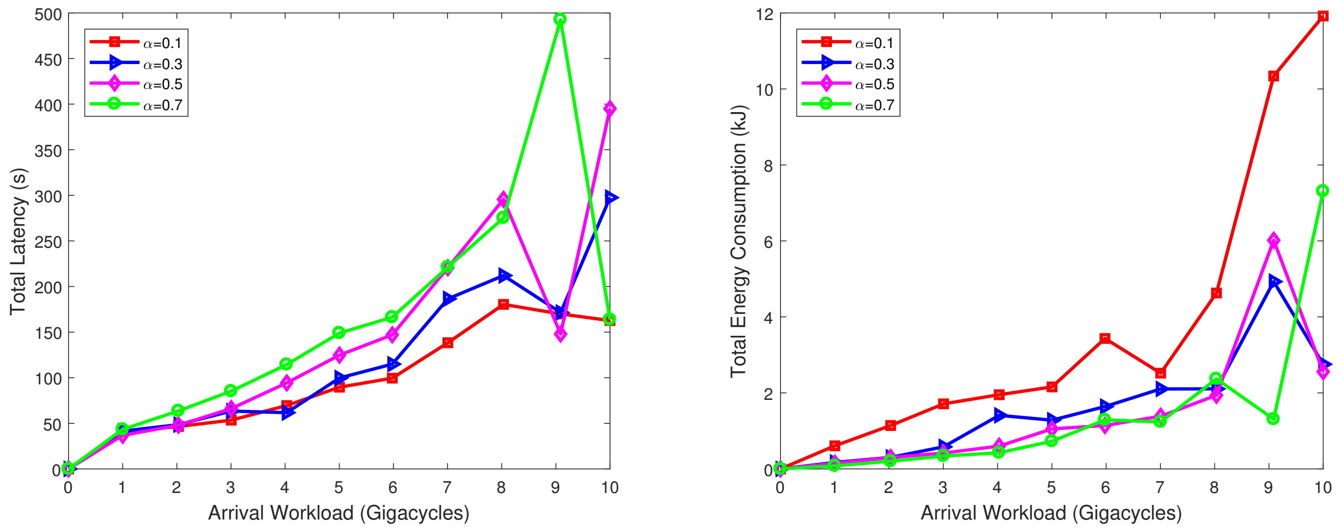

FIGURE 2 Total network latency and energy consumption of the JELO scheme depending on the balance coefficient α.

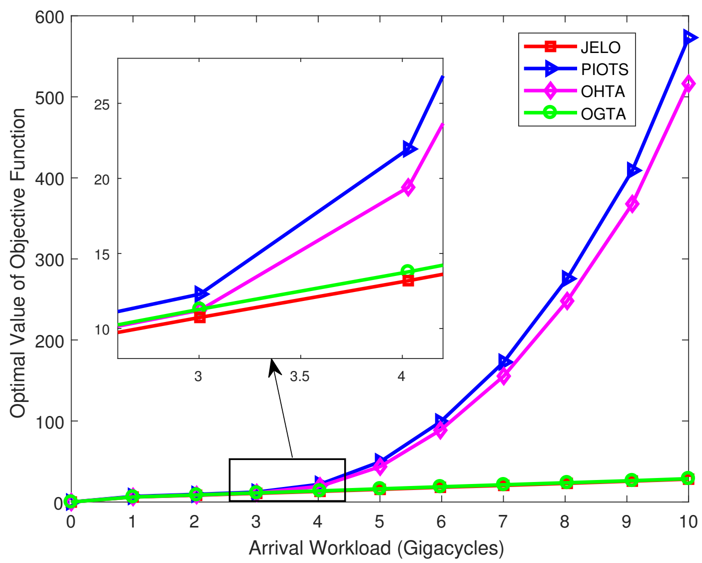

FIGURE 3 Optimal value of the objective function $\mathcal{P}$ in a comparison of four task assignment schemes.

Hence, this algorithm only derives a local optimal value, resulting in an approximate value compared to the JELO algorithm. Meanwhile, the objective function values of the OHTA and PIOTS schemes increase rapidly when the arrival workload increases. The OHTA scheme assigns tasks to the FNs at the end of each timeslot by using the repeated Hungarian algorithm to solve problem $\mathcal{P}$. Although this algorithm might better adapt to the varying task arrivals, it cannot obtain the exact optimal assignment since the Hungarian algorithm is not a completely exact algorithm for problem $\mathcal{P}$. This method tries to assign equal numbers of tasks to the FNs without considering the computation volume λ of each task. This causes unfair workloads among the FNs, and some FNs reach overcapacity, resulting in a higher buffering latency and energy consumption. Similarly, the PIOTS scheme uses the Hungarian method for determining the expected optimal task assignment. However, it focuses on minimizing the network latency while the energy is not considered. Thus, it has the worst performance with regards to the objective function value.

The network latency of the proposed JELO scheme in a comparison of three assignment schemes is illustrated in Fig. 4. Generally, the network latency increases when the workload arriving at the network increases. It seems that there is no difference

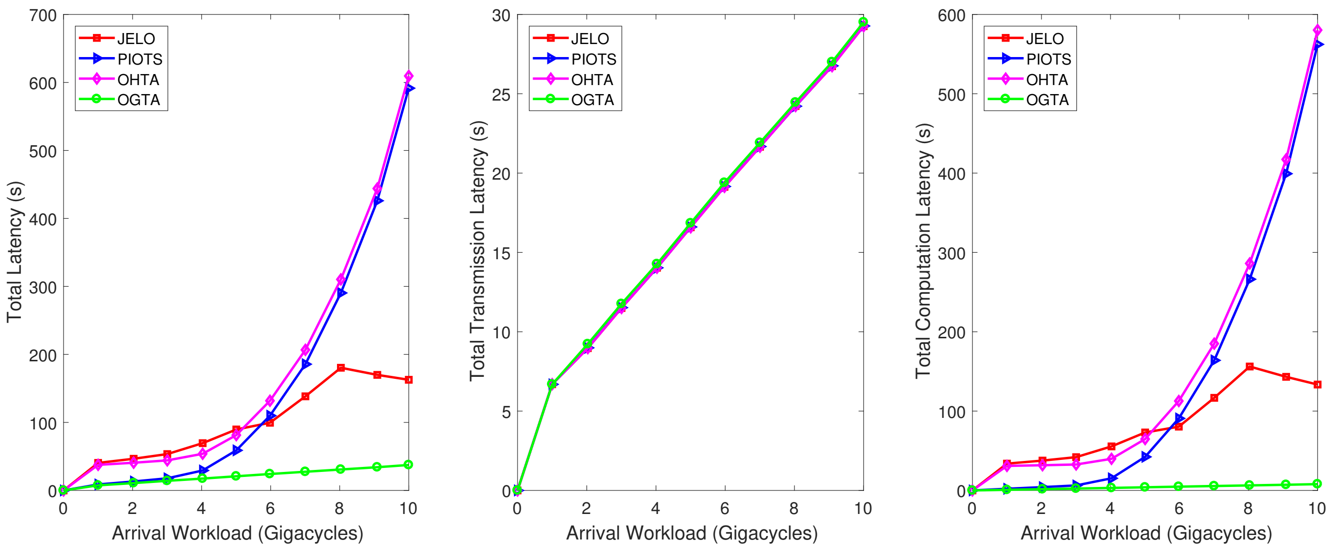

FIGURE 4 Network latency of four task assignment schemes.

in the total transmission latency since the desired data transmission rates of the IoT devices are satisfied. The difference in the latencies of the schemes is almost due to the task computation latency, resulting in a different total network latency. It is recognized that the JELO scheme obtains lower performance in terms of the latency compared to that of the PIOTS and OGTA schemes when the arrival workload is low (less than 4 gigacyles). This is understandable because the PIOTS and OGTA schemes are online task assignment algorithms. This means that the arriving tasks are assigned immediately to the FNs for execution. Meanwhile, the JELO and OHTA schemes need to gather the arriving tasks and wait until the end of each timeslot to determine the optimal assignment. This increases the buffering latency. It is observed that when the volume arriving tasks is larger than 5 gigacycles, the JELO scheme demonstrates its effectiveness. The JELO scheme provides a better total latency compared to the PIOTS and OGTA algorithms. The Hungarian method aims to assign equal numbers of tasks to the FNs. In other words, the Hungarian method ensures that each FN has an equal number of tasks with the aim of minimizing the joint energy and latency. The non-exact optimal solution from the repeated Hungarian method leads to incomplete optimal task assignment. Hence, some FNs might be burdened with a large workload when the number of arriving tasks increases, resulting in increases in the computation and buffering latencies at these FNs. Although the JELO scheme cannot obtain the latency optimization as in the OGTA scheme, it can derive energy and latency balancing. The greedy behavior of the OGTA scheme allows it to achieve the better latency. However, this consumes a larger energy for computation.

Fig. 5 shows the performance of the JELO scheme in terms of the energy consumption compared to that of the PIOTS, OHTA, and OGTA schemes. According to the simulation results, the PIOTS scheme has the largest transmission energy consumption. The PIOTS scheme assigns tasks based on the predetermined pattern-identified tasks. This is not a real incoming task set. Thus, the arriving tasks might be assigned to worse FNs that have to allocate a large number of RBs for transmission. This results in a large transmission energy. Although the energy performance of the JELO scheme is less that that of the OHTA scheme, it provides a lower energy consumption compared to the OGTA scheme. This demonstrates the harmonization between the energy and latency of the JELO scheme. It is worthwhile to note that the aggregate transmission energy is much smaller than the computation energy. This means that the aggregate energy for the entire network is mainly consumed during task processing. It is found that the OGTA scheme has the worst energy performance since it consumes a large computation energy for targeting the latency optimization. Meanwhile, the PIOTS and OHTA schemes cannot derive the exact-optimal task assignment. A number of incoming tasks might be assigned to FNs that have a low processing power. As a consequence, these tasks are not to executed immediately and are then located in the buffer. This does not consume much computation energy but effects the buffering latency. The JELO scheme adjusts the energy and latency. It does not consume too much energy but still ensures optimization of the latency.

The execution error rate is utilized to evaluate the serviceability of the network. It is defined by the percentage of deadline-violated tasks of the IoT devices per the total number of tasks arriving at the networks at each timeslot. Fig. 6 presents the task

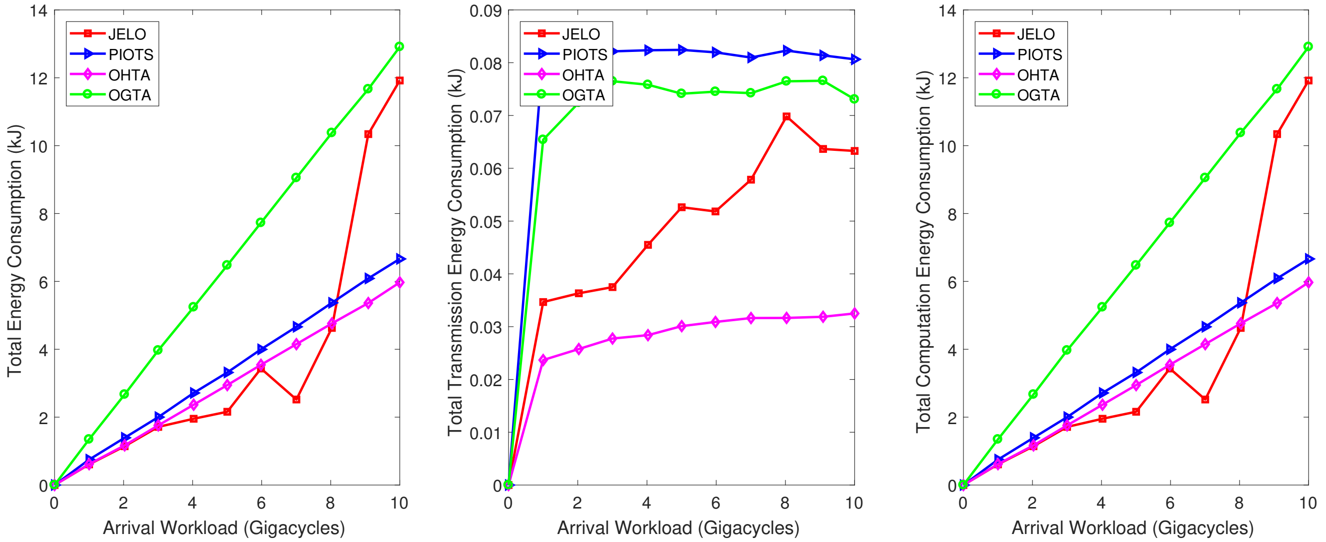

FIGURE 5 Network energy consumption of four task assignment schemes.

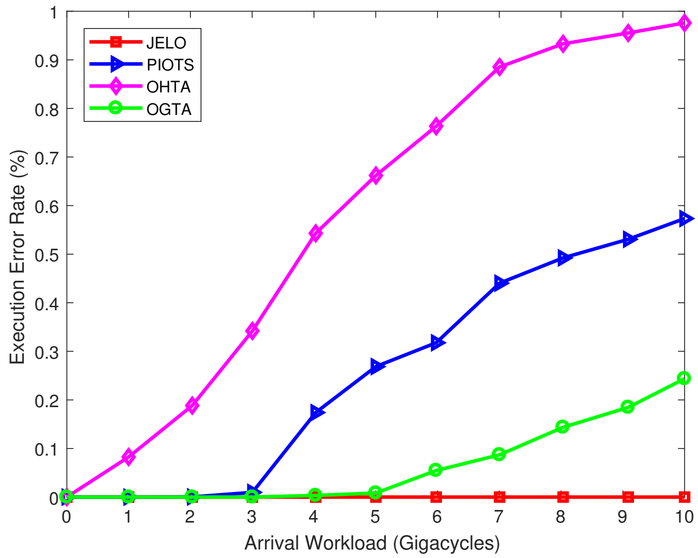

FIGURE 6 Task execution error rate.

execution error rate of the four assignment algorithms. The simulation results reveal that the JELO scheme achieves the best performance when it has no error tasks during the execution period. This is because the JELO scheme addresses the optimal task assignment while considering the execution deadline constraints of each task. In other words, the JELO scheme ensures that each task is executed within the required time. Meanwhile, the OGTA scheme assigns the incoming tasks to the FNs with the aim of minimizing the latency and energy of each task. It can optimize the local latency and provide better performance compared to that of the PIOTS and OHTA algorithms. However, the execution deadline of each task is not guaranteed, resulting in a number of deadline-violated tasks. Similarly, the OHTA and PIOTS schemes use the Hungarian method for optimal task assignment without considering the deadline constraints of the tasks. This leads to a large number of overdue tasks. However, the PIOTS scheme achieves better performance compared to the OHTA scheme since it can assign a task promptly, thus reducing the buffering latency.

6 | CONCLUDING REMARKS

In this paper, we propose the joint energy and latency optimization (JELO) algorithm for offloading upstream time-sensitive IoT services in FRANs. In JELO, the optimal task assignment problem for balancing the energy consumption and offload latency, which are generated by task transmission and execution, is developed as the objective function. The systematic resources and specific characteristics of individual tasks are strictly considered as main constraints to design the final solution. In order to reduce the complexity, we relax the optimization problem into two subproblems, which are M 0/1 knapsack and semi-assignment problems. By addressing these sub-problems, we can derive the final optimal solution using a subgradient method with reasonable costs. The experimental results reveal that our proposed JELO scheme overcomes existing approaches in terms of the energy consumption, latency, and load balancing. For our future researches, the joint energy and latency optimization for the upstream IoT offloading services with different requirements (i.e., both low latency and low energy) will be considered.

ACKNOWLEDGMENTS

Part of this work was presented at the 32nd International Conference on Information Networking (ICOIN) at Chiang Mai, Thailand, Jan. 10–12, 2018.

Financial disclosure

This research was supported (in part) by Chung-Ang University research grant in 2017 and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science ICT under Grant NRF-2017R1A4A1015675.

Conflict of interest

The authors declare no potential conflicts of interests.

References

1. Gartner, Inc. . *Gartner says 8.4 billion connected "Things" will be in use in 2017, up 31 percent from 2016.* http://www.gartner.com/newsroom/id/3598917. Accessed January 10, 2018; 2017.

2. Dao NN, Park M, Kim J, Cho S. Adaptive MCS selection and resource planning for energy-efficient communication in LTE-M based IoT sensing platform. *PloS one.* 2017;12(8):e0182527.

3. Munir A, Kansakar P, Khan SU. IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of things. *IEEE Consumer Electronics Magazine.* 2017;6(3):74–82.

4. Xiang H, Zhou W, Daneshmand M, Peng M. Network slicing in fog radio access networks: Issues and challenges. *IEEE Communications Magazine.* 2017;55(12):110–116.

5. Vu DN, Dao NN, Cho S. Downlink sum-rate optimization leveraging Hungarian method in fog radio access networks. In: IEEE International Conference on Information Networking (ICOIN):1–1; January 10–12, 2018; Chiang Mai, Thailand.

6. Peng M, Yan S, Zhang K, Wang C. Fog-computing-based radio access networks: Issues and challenges. *IEEE Network.* 2016;30(4):46–53.

7. Dao NN, Lee Y, Cho S, Kim E, Chung KS, Keum C. Multi-tier multi-access edge computing: The role for the fourth industrial revolution. In: IEEE International Conference on Information and Communication Technology Convergence (ICTC):1280–1282; October 18–20, 2017; Jeju, South Korea.

8. Taleb T, Ksentini A, Jantti R. "Anything as a Service" for 5G mobile systems. *IEEE Network.* 2016;30(6):84–91.

9. Chih-Lin I, Han S, Xu Z, Wang S, Sun Q, Chen Y. New paradigm of 5G wireless internet. *IEEE Journal on Selected Areas in Communications.* 2016;34(3):474–482.

10. Akyildiz IF, Nie S, Lin SC, Chandrasekaran M. 5G roadmap: 10 key enabling technologies. *Computer Networks.* 2016;106:17–48.

11. Carvalho GHS, Woungang I, Anpalagan A, Jaseemuddin M, Hossain E. Intercloud and HetNet for mobile cloud computing in 5G systems: Design issues, challenges, and optimization. *IEEE Network.* 2017;31(3):80–89.

12. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials.* 2017;19(3):1657–1681.

13. Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials.* 2017;19(3):1628–1656.

14. Rahman GS, Peng M, Zhang K. Radio resource allocation for achieving ultra-low latency in fog radio access networks. *IEEE Access.* 2018;6(PP):1–1.

15. Shih YY, Chung WH, Pang AC, Chiu TC, Wei HY. Enabling low-latency applications in fog-radio access networks. *IEEE Network.* 2017;31(1):52–58.

16. Dao NN, Vu DN, Lee Y, Cho S, Cho C, Kim H. Pattern-identified online task scheduling in multitier edge computing for industrial IoT services. *Mobile Information Systems.* 2018;2018:2101206.

17. Guan M, Bai B, Wang L, Jin S, Han Z. Joint Optimization for Computation Offloading and Resource Allocation in Internet of Things. In: IEEE 86th Vehicular Technology Conference (VTC-Fall):1–5; 2017.

18. Zhang J, Hu X, Ning Z, et al. Energy-latency Trade-off for Energy-aware Offloading in Mobile Edge Computing Networks. *IEEE Internet of Things Journal.* 2017;99:PP–PP.

19. Yang L, Zhang H, Li M, Guo J, Ji H. Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G. *IEEE Transactions on Vehicular Technology.* 2018;99:PP–PP.

20. Jeong S, Na W, Kim J, Cho S. Internet of Things for Smart Manufacturing System: Trust Issues in Resource Allocation. *IEEE Internet of Things Journal.* 2018;99:PP–PP.

21. Chabbouh O, Rejeb SB, Choukair Z, Agoulmine Na. A strategy for joint service offloading and scheduling in heterogeneous cloud radio access networks. *EURASIP Journal on Wireless Communications and Networking.* 2017;2017(1):196.

22. Terefe MB, Lee H, Heo N, Fox GC, Oh S. Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing. *Pervasive and Mobile Computing.* 2016;27:75–89.

23. Munoz O, Pascual-Iserte A, Vidal J. Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. *IEEE Transactions on Vehicular Technology.* 2015;64(10):4738–4755.

24. Dao NN, Lee J, Vu DN, et al. Adaptive resource balancing for serviceability maximization in fog radio access networks. *IEEE Access.* 2017;5:14548–14559.

25. Lagen S, Pascual-Iserte A, Muñoz-Medina O, Vidal J. Energy Efficiency in Latency-Constrained Application Offloading from Mobile Clients to Multiple Virtual Machines. *IEEE Transactions on Signal Processing.* 2018;66(4):1065–1079.

26. Mao Y, Zhang J, Letaief KB. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications.* 2016;34(12):3590–3605.

27. Lyu X, Tian H, Ni W, Zhang Y, Zhang P, Liu RP. Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing. *IEEE Transactions on Communications.* 2018;99:PP–PP.

28. Dao NN, Park M, Kim J, Paek J, Cho S. Resource-aware relay selection for inter-cell interference avoidance in 5G heterogeneous network for Internet of things systems. *Future Generation Computer Systems.* 2018;99:PP–PP.

29. Dao NN, Kim Y, Jeong S, Park M, Cho S. Achievable Multi-Security Levels for Lightweight IoT-Enabled Devices in Infrastructureless Peer-Aware Communications. *IEEE Access.* 2017;5:26743–26753.

30. Pu L, Chen X, Xu J, Fu X. D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE Journal on Selected Areas in Communications.* 2016;34(12):3887–3901.

31. Bai Z. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures. *3GPP, Sophia Antipolis, Technical Specification 36.213 v. 11.4. 0.* 2013;.

32. ETSI . Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation. *ETSI TS 136.211 V9.* ;.

33. Chen X. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems.* 2015;26(4):974–983.

34. Cao Z, Guo H, Zhang J, Niyato D, Fastenrath U. Improving the efficiency of stochastic vehicle routing: A partial lagrange multiplier method. *IEEE Transactions on Vehicular Technology.* 2016;65(6):3993–4005.

35. Pferschy U, Scatamacchia R. Improved dynamic programming and approximation results for the knapsack problem with setups. *International Transactions in Operational Research.* 2018;25(2):667–682.

36. Nesterov Y. Primal-dual subgradient methods for convex problems. *Mathematical programming.* 2009;120(1):221–259.

37. Zhang H, Liu N, Chu X, Long K, Aghvami A-H, Leung VCM. Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. *IEEE Communications Magazine.* 2017;55(8):138–145.

38. Li S, Ni Q, Sun Y, Min G, Al-Rubaye S. Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era. *IEEE Transactions on Industrial Informatics.* 2018;PP(99):1–1.

39. To H, Fan L, Tran L, Shahabi C. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In: IEEE International Conference on Pervasive Computing and Communications (PerCom):1–8; Mar. 14-18, 2016; Sydney, Australia.