Energy-Efficient Resource Allocation for Federated Learning in NOMA-Enabled and Relay-Assisted Internet of Things Networks

Mohammed S. Al-Abiad, *Member, IEEE*, Md. Zoheb Hassan, *Student Member, IEEE*, and Md. Jahangir Hossain, *Senior Member, IEEE*

***Abstract*—Distributed machine learning (ML) algorithms are imperative for the next-generation Internet of Things (IoT) networks, thanks to preserving the privacy of users' data and efficient usage of the communication resources. Federated learning (FL) is a promising distributed ML algorithm where the models are trained at the edge devices over the local data sets, and only the model parameters are shared with the cloud server (CS) to generate global model parameters. Nevertheless, due to the limited battery life of the edge devices, improving the energy-efficiency is a prime concern for FL. In this work, we investigate a resource allocation scheme to reduce the overall energy consumption of FL in the relay-assisted IoT networks. We aim at minimizing the overall energy consumption of IoT devices subject to the FL time constraint. FL time consists of model training computation time and wireless transmission latency. Toward this goal, a joint optimization problem, considering scheduling the IoT devices with the relays, transmit power allocation, and computation frequency allocation, is formulated. Due to the NP-hardness of the joint optimization problem, a global optimal solution is intractable. Therefore, leveraging graph theory, joint near-optimal, and low-complexity suboptimal solutions are proposed. Efficiency of our proposed solutions over several benchmark schemes is verified via extensive simulations. Simulation results show that the proposed near-optimal scheme achieves 6, 4, and 2 times lower energy consumption, respectively, compared to the considered fixed, computation adaptation, and power adaptation schemes. Such an appealing energy efficiency comes at the cost of slightly increased FL time compared to the fixed and computation only adaptation schemes.**

***Index Terms*—Energy consumption, federated learning (FL), relay-assisted Internet of Things (IoT) networks.**

I. Introduction

INTERNET of Things (IoT) is an integral element of the envisioned network society, which aims to provide networked connectivity to the billions of devices. In particular, the next generation IoT leads to the unprecedented growth of computation-demanding applications, such as augmented reality and intelligent transportation, which require significant processing capability at the network edge devices. Cloud computing provides an efficient computation platform for executing the aforementioned applications. However, cloud computing requires efficient offloading of computation-intensive tasks from the energy-constrained mobile devices to the cloud server (CS) of enormous computation capability [1]. Note that the contemporary wireless technologies for IoT, namely, Bluetooth, ZigBee, and low power wide area (LPWA), rely on the unlicensed frequency bands, and thus cannot provide the Quality-of-Service (QoS) guaranteed offloading services to the IoT devices. To this end, thanks to the large bandwidth resources and ubiquitous backhaul connectivity, the existing cellular network infrastructure provides a promising solution to connect the IoT devices with the CS [2]. Nevertheless, it is still challenging for the battery-powered devices/sensors of the network edge to directly communicate with the distant (i.e., cell center) CS while using the dedicated cellular frequency bands. The limited connectivity of the IoT devices can be improved via connecting the cell-edge mobile devices with the network relay(s) instead of the CS directly. Such a relayed transmission not only reduces the energy cost and complexity of the IoT device transmitters, but also improves the reliability and data rate [3]. Essentially, the relay-assisted IoT provides an excellent solution to efficiently offload computation-intensive tasks from the cell-edge IoT devices to the CS.

Recently, the data driven decision making becomes an integral part of IoT networks, thanks to the availability of the enormous data and advancement of the devices' computing power. In particular, machine learning (ML) algorithms are extensively used to predict traffic congestion, user behavior, and QoS of users by analyzing large-scale data collected from the IoT devices. In addition, deep learning techniques are also extensively applied in different systems for analyzing large-scale data collected from the IoT devices. For instance, Mao *et al.* [4] developed deep-learning-based offloading policy optimization strategy to address the dynamics of energy harvesting performance in the satellite-unmanned aerial vehicles (UAV) IoT system. In conventional ML, the collected data from IoT devices (e.g., images, videos, and recorded audios) are offloaded to and processed in the CSs, where the learning models are trained. Such a centralized ML approach is confronted by the huge traffic burden in the wireless links between IoT devices and core network. In addition, the privacy of users'

Manuscript received 25 March 2022; revised 1 May 2022 and 7 July 2022; accepted 25 July 2022. Date of publication 28 July 2022; date of current version 7 December 2022. This work was supported by the Discovery Grant of Natural Science and Engineering Research Council (NSERC), Canada. *(Corresponding author: Mohammed S. Al-Abiad.)*

Mohammed S. Al-Abiad and Md. Jahangir Hossain are with the School of Engineering, The University of British Columbia, Kelowna, BC V1V 1V7, Canada (e-mail: m.saif@alumni.ubc.ca; jahangir.hossain@ubc.ca).

Md. Zoheb Hassan is with the Wireless@Virginia Tech, Bradley Department of ECE, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: mdzoheb@vt.edu).

Digital Object Identifier 10.1109/JIOT.2022.3194546

sensitive data is impeded. Therefore, the conventional centralized learning method is inefficient for next generation IoT networks [5]. Federated learning (FL) is an efficient distributed learning mechanism that allows multiple network edge devices to collaboratively learn a shared model [6]. In FL, the network edge devices train individual learning models using the local data. In contrast to the centralized learning mechanism, the devices only share updated model parameters with the CS. Subsequently, the CS calculates the global model parameter by aggregating the local model parameters of the edge devices. The local and global parameters are updated iteratively until convergence. By distributing the learning tasks between the network edge and the CSs, FL not only reduces the huge traffic burden over a wireless channel, but also protects privacy of IoT data [7]. Recently, Yu *et al.* [8] integrated deep learning and FL to minimize the total offloading delay and network resource while training the models in a distributed manner to protect the edge devices' data privacy. In particular, the work in [8] jointly optimized computation offloading, resource allocation, and service caching placement in multiaccess edge computing system. As such, the total offloading delay and network resource usage and edge devices' data privacy is protected.

FL techniques have exhibited widespread successes in ultra-reliable and latency-sensitive applications, such as virtual reality, traffic prediction, object recognition, etc. [6]–[8]. As an example of latency-sensitive applications for FL is highway traffic prediction. Imagine a highway has different roadside units with camera. They are taking high-quality pictures continuously. The goal is to analyze those pictures rapidly to predict traffic characteristics. To this end, we need an ML model, which will take pictures as input and give traffic prediction as output. However, it is not efficient to send all the pictures to the CS as it will take lots of bandwidth. Hence, FL is an efficient way to build the ML model. Based on the predicted traffic, a controller will automatically adjust the traffic light as such vehicles can move smoothly. To facilitate a fast control in busy highway, the FL must converge rapidly. Therefore, the design of resource and communication-efficient FL systems should efficiently consider low computation and wireless communication delay to meet the acceptable delay requirements of the aforementioned latency-sensitive applications.

There is a tradeoff between the FL time (i.e., computation and wireless communication latencies) and IoT device energy consumption. Reducing the FL time requires more energy to reduce the IoT device model computation time and wireless data transmission latency. The IoT device model computation time is affected by the allocated computing resource, i.e., CPU clock frequency. For instance, increasing computation frequency at the IoT devices can reduce the latency at the cost of the increased energy consumption and visa versa. In addition, the wireless data transmission latency is determined by the wireless transmission rate, which is related to wireless transmission power. In order to reduce the wireless communication latency, we can increase the transmission power, however, this will consume more energy. Therefore, our work jointly investigates both the computation frequency allocation adjustment and wireless transmission power control to minimize the IoT device energy consumption while satisfying the FL time requirements.

It is noteworthy that the performance of FL, including accuracy and convergence jointly depend on selection of the collaborating devices, spectrum resource allocation, power allocation, and computation capability of the collaborating devices [10]. On the one hand, the channel impairments, such as fading and interference can affect the FL accuracy. On the other hand, a set of selected devices can be out-of-coverage due to the large distance, which can affect the convergence of FL. However, the relay-assisted transmission can mitigate the impairments, such as fading and coverage hole. Essentially, relays can assist the out-of-coverage devices to forward the model parameters reliably to the CS, and thereby helps in improving the accuracy and convergence of FL. Moreover, relays can participate in local learning as well [7]. Consequently, relay-assisted IoT provides a convenient platform to implement FL for the network edge devices in IoT networks. In this work, we propose a resource allocation scheme to minimize the overall energy consumption of a delay-constrained FL in relay-assisted IoT networks.

In order to improve the number of connected IoT devices with the relays using limited radio resource blocks (RRBs), nonorthogonal multiple access (NOMA) [9] is employed. Therefore, NOMA is a key solution to address the challenging problem of supporting a large number of IoT devices with the limited number of RRBs in beyond-5G era systems. To this end, developing an innovative resource allocation framework is imperative for harnessing the aforementioned benefits of NOMA-based relay-assisted IoT networks. In the envisioned system, IoT devices of the network edge are connected with the CS via multiple relay nodes. A cluster of IoT devices first upload the locally trained model parameters to a suitable relay node using the NOMA scheme, and subsequently, the relay nodes forward the received model parameters to the CS for global aggregation.

A. Related Works and Motivations

Related Works on Communication-Efficient FL: The performance of a decentralized ML depends on the optimization of wireless links between the network edge devices and parameter server (CS or fog computing nodes). Hence, it is imperative to optimally design the learning-centric resource allocation schemes [12]. In the recent literature, the design of communication-efficient FL was extensively studied. Leveraging the grouping of network edge devices and a decentralized group alternating direction method of multipliers, a jointly communication efficient and fast converging FL algorithm was proposed in [13]. To enhance the accuracy and convergence of FL, it is imperative to enhance the number of collaborating edge devices while using the available spectrum resources efficiently. To this end, a collaborative FL framework was proposed that allows resource-constrained IoT devices to upload model parameters to the nearby devices instead of the distance CS [14]. Moreover, a joint scheduling of network edge devices and RRBs was studied to minimize the FL loss function via applying the Lyapunov optimization framework [15]. In a heterogeneous cellular network, a hierarchical FL framework can effectively enhance the number of devices participating in local learning [16]. In such a hierarchical FL framework, at each round, the network edge devices upload their model parameters only to the nearest F-APs (therein called small base stations), and F-APs periodically upload the average local model parameters to the CS (therein called macro base station) for a global aggregation. Thus, a large number of devices can participate in local learning. Besides, interference among the network edge devices can induce error in FL and increase the convergence time. Accordingly, interference-aware radio resource allocation is also imperative for communication-efficient FL framework. A joint optimization of user selection, RRB allocation, and transmit power allocation was presented to minimize the loss function in the FL training process. Chen *et al.* [17] proposed transmit power allocation of the IoT devices to enhance information freshness in the FL system. Considering the presence of eavesdroppers in an Internet of drones network, Yao and Ansari [18] proposed a secured and delay-constrained FL scheme through transmit power allocations. As compared to the above works, we propose resource allocation and communication-efficient mechanisms for facilitating FL for network edge devices in relay-assisted IoT networks, where relays are used to collect devices' local parameters and forward them to the CS for global aggregation. We show that our framework can effectively optimize resource scheduling, transmit power allocation, and computation frequency allocation and result in communication and energy-efficient relay-assisted IoT network.

Related Works on Energy-Efficient FL: Since the mobile devices are battery-driven, for a sustainable operation of an FL framework, it is imperative to reduce energy consumption of the edge devices. In particular, an energy-efficient or green FL should consider minimizing communication and computation energy. In [19], energy-efficient radio resource allocation was proposed for delay-constrained FL. However, Zeng *et al.* [19] only minimized the communication energy and ignored the computation energy. Wang *et al.* [20] proposed an adaptive FL framework, where the devices can send quantized or compressed model parameters and thus, save energy. However, the radio resource optimization was not presented in [20]. In [21], radio resource allocation was developed to minimize both communication and computation energy in an FL system subject to delay constraints. Yao and Ansari [22] proposed joint transmit power and computation frequency allocation to reduce overall energy consumption of FL in a fog-aided IoT network. Recently, a joint optimization framework considering user scheduling, transmit power allocation, and user's computation frequency allocation was proposed to simultaneously minimize the total energy consumption and maximize the number of the scheduled users [23]. Nevertheless, the studies conducted in [21]–[23] considered orthogonal multiple access (OMA) to connect edge devices with the base station, which can limit the number of collaborating devices. The energy limitation of the collaborating edge devices can also be improved by applying energy-harvesting techniques [24]. Moreover, the work in [25] considered a game theory framework to motivate the network edge devices to participate in local learning while reducing its energy consumption. A survey of different ML and deep learning techniques in green communications can be found in [26]. However, an energy-efficient framework, while jointly considering communication and computation energy, for facilitating FL for network edge devices in relay-assisted IoT networks was not investigated in the existing literature.

Related Works on FL in Relay-Assisted Networks: The problem of FL in relay-assisted network was investigated in [27]–[29]. In particular, leveraging the notion of over-the-air-computing, Lin *et al.* [27] optimized FL accuracy in relay-assisted network. In [28], considering the model owner (i.e., the CS) and mobile devices as leader and followers, respectively, a Stackelberg game was proposed for FL in mobile relay-assisted networks. Qu *et al.* [29] proposed the FL scheme based on two-way relaying. However, the aforementioned works have certain limitations. For instance, the study in [27] optimized FL performance without considering the energy consumption of the devices and latency constraints of the FL. Recall, owing to the limited battery at the edge devices, it is crucial to optimize both communication and computation energy consumption of the edge devices. Although the issue of energy consumption at the devices was considered Feng *et al.* [28] ignored latency introduced by both offloading and local computing in FL. It is recalled that reducing energy consumption may lead to increased latency, which is prohibitive for delay-constrained FL applications. On the other hand, Qu *et al.* [29] ignored the impact of wireless channel fading during decoding at the relay/edge devices and considered OMA to connect multiple devices with the relays. Essentially, the relay-assisted FL algorithm of [29] is not efficient for large-scale IoT networks with time-varying channels. One potential solution is to increase the number of connected devices to the relay using the uplink NOMA scheme, where multiple devices are scheduled in the same RRB. Then, the relay sequentially decodes the devices' data by applying the successive interference cancelation (SIC) scheme [9]. Nevertheless, due to the additional co-channel interference, the optimization of decoding order and power allocation is crucial in NOMA-enabled systems. Bouzinis *et al.* [30] proposed resource allocation to minimize the overall latency of the NOMA-enabled FL scheme. However, due to the consideration of only single hop communication, the proposed resource allocation in [30] is not effective for the edge devices. Therefore, although a relay-assisted network is promising for FL, efficient resource allocations are imperative to reduce the energy consumption of the cell edge devices and improve the number of connected devices with the relay(s).

Motivations: In contrast to the existing works [19]–[22], [24], [25], [27]–[30], our motivation is to develop resource allocation mechanisms to facilitate energy-efficient FL for network edge devices in relay-assisted IoT networks. To the best of our knowledge, an efficient integration of relay-assisted IoT, and NOMA to reduce the energy consumption of the FL scheme is not investigated in the existing literature. Specifically, the envisioned system exhibits the following two challenges. First, the computation and communication latencies of each global iteration in FL crucially depends on the selection of the appropriate devices. This is becasue the CS needs to collect the local model parameters from all the selected devices before model aggregation, and consequently, the devices with poor computation or communication channel, termed as stragglers, can significantly increase the latency. Besides, NOMA can introduce interference among devices. Moreover, since each relay can only support a finite number of edge devices, scheduling appropriate devices to the relays is crucial. Therefore, to effectively reduce the latency of each global iteration, it is imperative to jointly optimize the scheduling of the IoT devices to the available relays and RRBs. Second, there is an inherent tradeoff between FL time (consisting of computation and communication latencies) and energy consumption, i.e., both computation and communication energy are increased to reduce the FL time. Accordingly, a joint optimization of the degrees-of-freedom, namely, power allocation, IoT device scheduling to the relays/RRBs, and computation frequency allocation, is required to satisfy a given FL time constraint and reduce energy consumption. However, due to the interdependence of the resource allocation variables, the required joint optimization exhibits extremely high computational complexity, especially for large-scale networks. Such a fact motivates us to develop a computationally efficient resource allocation scheme to facilitate energy-efficient and delay-constrained FL in NOMA-enabled and relay-assisted IoT networks.

B. Contributions

We investigate resource allocation for energy-efficient and delay-constrained FL in the NOMA-enabled and relay-assisted IoT network. Specifically, we propose a joint optimization of scheduling IoT devices with the relays and RRBs, transmit power allocation, and computation frequency allocation at the IoT devices. To this end, we develop innovative graph-theoretical algorithms leading to computationally efficient solution(s). The main contributions of our work are as follows.

1) For NOMA-enabled and relay-assisted IoT network, we develop a framework to facilitate latency-constrained FL for the energy-limited network edge devices. In the envisioned system, the local models are trained at the IoT devices and the relays upload the collected local model parameters to the CS for aggregation. An optimization problem is formulated to minimize both computation and communication energy consumption of the IoT devices subject to the constraints on FL time, scheduling among the IoT devices, relays, and RRBs, transmit power allocation, and computation frequency allocation. Such a joint optimization problem is NP-hard and thus, its global optimal solution is computationally intractable in the practical systems. To solve the joint optimization problem in a tractable manner, we decompose the joint optimization problem into two subproblems namely, *resource scheduling and power allocation* subproblem and *computation frequency allocation* subproblem. By iteratively solving these subproblems, two efficient solutions to the presented joint optimization problem are obtained.
2) In the proposed first solution, we leverage graph theory to near-optimally solve the resource scheduling and power allocation subproblem. In particular, we design a joint FL graph that solves the IoT device-relay-RRB scheduling and transmit power allocation jointly. Based on the developed resource scheduling and transmit power allocation, we determine a closed-form solution to the computation frequency allocation subproblem. The aforementioned two steps are executed alternately, and the resultant solution is referred to a *joint approach*. The proposed joint approach requires generating all possible NOMA clusters, and consequently, it exhibits a high computational complexity.
3) To reduce the computational complexity of the joint approach, we propose a second solution, where we exploit a graph-pruning technique to solve the resource scheduling and power allocation subproblem. Specifically, we first devise a reduced NOMA graph to generate feasible NOMA clusters. Thereafter, by applying a greedy maximum-weight-independent-set (MWIS) algorithm, we obtain an efficient solution to the reduced NOMA graph. By alternately solving the reduced NOMA graph and the computation frequency allocation subproblem, a low-complexity suboptimal solution approach is devised.

The remainder of this article is organized as follows. The system model is described in Section II. The optimization problem and its transformation are provided in Section III. In Sections IV and V, we develop a joint approach and a low-complexity graph pruning solution to find the optimized IoT device scheduling, power allocation, and frequency computation scheduling decisions, respectively. Simulation results are presented in Section VI, and in Section VII, we conclude this article.

II. System Overview

A. System Model

We consider a relay-assisted IoT system, illustrated in Fig. 1, that consists of a CS, K relays, and N IoT devices. The sets of IoT devices and relays are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$, respectively. The N IoT devices are connected to the relays which are connected to the CS using fronthaul links. We consider that each relay has a limited coverage range that represents the service area of the kth relay within a circle of radius $\mathtt{R}$. The set of IoT devices in the kth relay's coverage range is defined by $\mathcal{N}_k = \{n \in \mathcal{N} | d_{k,n} \leq \mathtt{R}\}$, where $d_{k,n}$ is the distance between the kth relay and the nth IoT device. Our envisioned system considers that the N IoT devices are connected to the relays to offload their local parameters to the CS. Thus, we consider the relay assignment matrix $\mathbf{A}$, where the binary optimization variable $a_{k,n} = 1$ if the nth IoT device is assigned to the kth relay, and $a_{k,n} = 0$ otherwise. This assignment matrix represents the IoT device-relay association.

Let $\mathcal{D}_n$ denote the local data set of the nth IoT device, which is a set of data samples $\{x_i, y_i\}$, where x_i is sample i's input (e.g., image pixels) and y_i is sample i's output (e.g.,

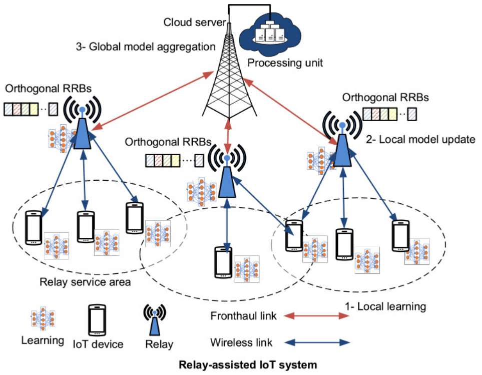

Fig. 1. Illustration of relay-assisted IoT system.

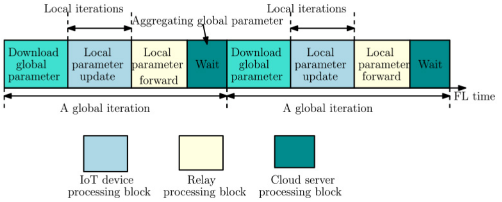

Fig. 2. Illustration of FL process.

label of the image). Similar to [20] and [22], the local loss function on data set of the nth IoT device can be calculated as $L_n(\omega) = (1/D_n)\sum_{i\in\mathcal{D}_n} l_i(\omega)\forall n \in \mathcal{N}$, where $D_n = |\mathcal{D}_n|$ is the number of collected data samples by the nth IoT device and $l_i(\omega)$ is the loss function that measures the local training model error of the ith data sample. Then, the nth IoT device finds the optimum ω_n^* that minimizes $L_n(\omega)$ and uploads it to the scheduled relays for aggregation by the CS. The IoT devices independently train local ML models based on their aggregated local data (e.g., images, videos, and recorded audios). As shown in Fig. 2, the specific process of FL in the t-th iteration can be summarized as: 1) each IoT device downloads the global model parameters $\omega_n(t-1)$ from the CS through the nearest relay; 2) each IoT device updates the local model by its local training data and sends the updated local model parameter $\omega_n(t)$ back to the relays; and 3) the CS aggregates the information from the relays and calculates the new global model parameters. Similar to [18]–[22], this work considers uniform importance for each IoT device, which is meaningful when all the IoT devices have a good quality of data set. Therefore, we consider the wireless channel quality of the IoT device, which will impact the uploaded models from the IoT devices to the relays, i.e., latency and energy consumption. The scenario of considering priority of selecting edge devices which contribute more to the utility improvement in resource allocation can be applied in our work as follows. At the beginning, the CS selects a uniform weight factor, $w_n = 1$, for all IoT devices (same as our current problem setting). Then, at each global iteration, the CS will collect the quantity $||\mathbf{e_n}||$ from all the IoT devices, where $\mathbf{e_n}$ represents the contribution of the nth IoT device to the global model. Then, we update the weight factors and allocate RRBs/selected IoT devices based on the updated weight factors, i.e., the weight factor of the nth IoT device is updated as $w_n = |e_n|/\sum_{n=1}^{N}|e_n|$. As a result, after a certain number of global iterations, only the IoT devices with good channel conditions and good quality of data sets will be selected in updating the global model.

Each IoT device uploads the local information to the scheduled relay via a wireless link. Similar to the resource setting in [31] and [32], we consider that each relay has Z orthogonal RRBs that are denoted by the set $\mathcal{Z} = \{1, 2, \ldots, Z\}$, where IoT devices can transmit their local information to the relays. These RRBs can be used practically as a generic term to denote time/frequency resource block of each relay, e.g., a group of orthogonal subcarriers. Let $\mathbf{S} = \{s_{k,z}^n\}$ be the RRB allocation matrix, where element $s_{k,z}^n = 1$ represents that the nth IoT device is allocated to the kth relay on the zth RRB, and $s_{k,z}^n = 0$ otherwise. In this work, we consider that each IoT device is scheduled to only one RRB [31], [32], and each RRB is scheduled to a set of two IoT devices using NOMA. Such a consideration, on the one hand, reduces the required complexity of scheduling, and on the other hand, simplifies the SIC operations at the relays. Let p_n denote the transmission power of the nth IoT device and let $\mathbf{p}$ be a $1 \times N$ matrix containing the power levels of all IoT devices, i.e., $\mathbf{p} = [p_n]$. Hence, the instantaneous signal-to-interference-plus-noise (SINR) for the link between the nth IoT device and the zth RRB in the kth relay is given by

$$\gamma_{k,z}^n = \frac{s_{k,z}^n p_n \left|G_{k,z}^n\right|^2}{\sum_{\substack{j\in\mathcal{N}_k\setminus n\\ G_{k,z}^j<G_{k,z}^n}} s_{z,k}^j p_j \left|G_{k,z}^j\right|^2 + \sigma^2} \quad \forall (j,n) \in \mathcal{N}_k \quad (1)$$

where σ^2 denotes the additive white Gaussian noise variance and $G_{k,z}^n$ denotes the channel gain for the link between the nth IoT device and the zth RRB in the kth relay. Then, the transmit rate of the nth IoT device to the kth relay over the zth RRB can be given by $R_{k,z}^n = W\log_2(1+\gamma_{k,z}^n)$, where W is the bandwidth of the zth RRB. Consequently, the transmit rate of the nth IoT device is $R_n = \sum_{k\in\mathcal{K}}\sum_{z\in\mathcal{Z}} a_{k,n}s_{k,z}^n W\log_2(1+\gamma_{k,z}^n)$. Each relay uploads its collected local parameters to the CS through a fronthaul link of capacity R_{fh}.

In this work, we propose to solve a complex optimization problem that jointly considers scheduling IoT devices with the relays and RRBs, transmit power allocation, and computation frequency allocation at the IoT devices. For simplicity, we consider perfect channel state information (CSI) between the IoT devices and relays. In practical scenarios, the IoT devices periodically broadcast the pilots and the relays estimate the CSI parameters (i.e., $G_{k,z}^n\ \forall n \in \mathcal{N}, z \in \mathcal{Z}, k \in \mathcal{K}$) from these pilots. After their estimation, these CSI parameters are transmitted to the CS to incorporate them in its decisions on the rates for each IoT device-relay transmission. However, the estimation of these parameters can become outdated during the time of the transmission. Once this outdated CSI is employed, IoT devices' capacity may be miscomputed, thus leading to certain performance degradation [33]. In this context, ML techniques can be used to proactively estimate CSI without using conventional pilot-based estimation [34]. However, such an ML-based method is out of the scope of this work.

Our work is focused on developing resource scheduling, frequency computation, and power allocation mechanisms to facilitate energy-efficient FL for network edge devices in relay-assisted IoT networks under the adopted learning model in [18]. A key challenge of implementing FL for 6G wireless communications is that a large amount of training data is required [35]. However, the off-the-shelf IoT devices usually collect a lot of data, thanks to their superior sensing capability, and FL leverages the local data set collected by the IoT devices. In this context, an interesting problem is to analyze the tradeoff between the local data sizes and learning accuracy of FL. Nevertheless, with the adopted FL model, the joint optimization problem considered in this work is already complex. Essentially, addressing the effectiveness of FL with different training data size is beyond the scope of this work. Therefore, this work abstracts the learning model in [18] based on the training data size in [47] and focuses on facilitating energy-efficient FL for network edge devices in relay-assisted IoT networks.

B. Federated Learning Time

In each iteration, the FL time consists of the computation time for local model training and the transmission time for uploading local model parameters to the relays as well as to the CS. In the local training process, the nth IoT device trains the local model and updates its local parameter until a local accuracy ϵ_l is achieved [20]. Let C_n denote the number of CPU cycles to process one data sample of the nth IoT device, and accordingly, the number of CPU cycles required for one local iteration over all data samples is C_nD_n. Therefore, the computation time for one local iteration in the nth IoT device can be calculated as $((C_nD_n)/f_n)$, where f_n is the computational frequency of the CPU in the nth IoT device (in cycles per second) [36]. Let $\mathbf{f}_{\mathrm{N}}$ be a $1 \times N$ matrix containing the computation frequency allocations of all IoT devices, i.e., $\mathbf{f}_N = [f_n]$. The number of local iterations to reach the local accuracy ϵ_l in the nth IoT device is $T_l = (2/[(2-\delta\beta)\delta\vartheta])\ln(1/\epsilon_l)$, where δ, ϑ, and β are constant parameters [18]. Then, the computation time of the nth IoT device is expressed as follows:

$$T_n^c = T_l \frac{C_nD_n}{f_n}. \tag{2}$$

After performing the local learning at the nth IoT device, suppose that the data size d_n of each resulting local parameter ω_n is fixed over the learning process [37]. Hence, the transmission time of the nth IoT device for uploading its parameters to the kth relay on the zth RRB is $T_n^w = (d_n/R_{k,z}^n)$.

Note that the global model parameters can only be updated by the CS after all local model parameters are received from the relays. Consequently, the FL time τ in each global iteration is dominated by the longest duration time of receiving the parameters among all IoT devices and the longest duration time of forwarding the parameters from the relays to the CS. Moreover, the transmission duration of the kth relay to upload its collected parameters to the CS is $T_k^w = ((\sum_{n\in\mathcal{N}_k} d_n)/R_{fh})$. Hence, the learning time τ of one global iteration can be calculated as follows:

$$\begin{aligned}\tau &= \max_{n\in\mathcal{N}}\{T_n^c + T_n^w\} + \max_{k\in\mathcal{K}}\{T_k^w\}\\ &= \max_{n\in\mathcal{N}}\left\{T_l\frac{C_nD_n}{f_n} + \frac{d_n}{R_{k,z}^n}\right\} + \max_{k\in\mathcal{K}}\left\{\frac{\sum_{n\in\mathcal{N}_k} d_n}{R_{fh}}\right\}.\end{aligned} \tag{3}$$

Inspired by [23], for each global iteration, the learning time τ should be no more than the maximum FL time T_q. Such a constrain is formally expressed as follows:

$$\max_{n\in\mathcal{N}}\left\{T_l\frac{C_nD_n}{f_n} + \frac{d_n}{R_{k,z}^n}\right\} + \max_{k\in\mathcal{K}}\left\{\frac{\sum_{n\in\mathcal{N}_k} d_n}{R_{fh}}\right\} \le T_q \tag{4}$$

and can be written as follows:

$$T_l\frac{C_nD_n}{f_n} + \frac{d_n}{R_{k,z}^n} + \frac{\sum_{n\in\mathcal{N}_k} d_n}{R_{fh}} \le T_q \;\forall n\in\mathcal{N}\;\forall k\in\mathcal{K}. \tag{5}$$

C. Energy Consumption Model

The IoT device's energy is consumed for both local model training and parameter transmission over wireless links that is explained as follows.

1) *Local Computation:* We adopt the widely used energy consumption model which considers that the energy consumption of the nth IoT device to process a single CPU cycle is αf_n^2, where α is a constant related to the switched capacitance [38], [39]. Hence, the energy consumption of the nth IoT device for local computation is $E_n^c = T_lC_nD_n\alpha f_n^2$ [18].
2) *Parameter Transmission:* The energy consumption to upload local model parameters to the relays over wireless links can be denoted by E_n^w and calculated as $p_nT_n^w$. Since the local parameters are forwarded from the relays to the CS over high transmission links, the energy consumption is negligible. Hence, we discard the relays' energy consumption. The key notations are summarized in Table I.

By combining all the aforementioned terms of energy consumption, the total energy consumption of the system in the IoT device local learning scenario can be calculated as follows:

$$E = \sum_{n\in\mathcal{N}}\left(E_n^w + E_n^c\right) = \sum_{n\in\mathcal{N}}\left[\frac{p_nd_n}{R_{k,z}^n} + T_lC_nD_n\alpha f_n^2\right]. \tag{6}$$

III. Problem Formulation and Transformation

A. Problem Formulation

In this work, we devise the energy consumption minimization problem that considers a joint management of computation and communication resources and IoT device selection problem for the delay-constrained FL. Specifically, our proposed framework intelligently selects the active IoT devices that perform local learning and assigns active IoT devices to the suitable relays and RRBs. As such, the objective of $\mathcal{P}1$ is to minimize the overall energy consumption. Therefore, we consider the following constraints.

1) *Constraint C1:* IoT device-RRB/relay assignment constraint where each IoT device is assigned to only one relay and to only one RRB in that relay. This is

TABLE I
SUMMARY OF THE MAIN NOTATIONS AND DEFINITIONS

Variable	**Definition**
$\mathcal{N}_k, \mathcal{K}, \mathcal{Z}$	Sets of N IoT devices, K relays, Z orthogonal RRBs per relay
$\mathcal{N}_k$	Set of IoT devices in the k-th relay's coverage range
$\mathbf{A}, \mathbf{S}$	Relay assignment matrix and RRB allocation matrix
$\mathcal{D}_n$	Local data set of the n-th IoT device
C_n	Number of CPU cycles to process one data sample of the n-th IoT device
f_n, p_n	Computational frequency and power allocations of the n-th IoT device
T_l, ϵ_l	Number of local iterations and local learning accuracy
T_n^c, T_n^w	Computation and transmission times of the n-th IoT device
T_k^w	Transmission time of the k-th relay
D_n, d_n	Number of data samples and local parameter's size of the n-th IoT device
τ	FL time
T_q	FL threshold time
E_n^c, E_n^w	Local computation and transmission energy consumption of the n-th IoT device

becasue we consider an efficient and simple design of relay-assisted IoT system by considering IoT devices-RRBs/relays scheduling policy, similar to [31] and [32].

2) *Constraint C2:* For simplicity of the ensuing analysis, we consider that each RRB can schedule at most two IoT devices using NOMA.
3) *Constraint C3 and C5:* Since we consider energy-constrained IoT devices with limited computation capability, we emphasize that each IoT device has a maximum local computation resource allocation of $f^{\max}$ and maximum transmit power level of $p^{\max}$. In the simulations, the values of $f^{\max}$ and $p^{\max}$ are selected based on [18] and [22], respectively.
4) *Constraint C4:* C4 represents the QoS requirement constraint on the FL time. This constraint indicates the time required for performing one global iteration in sensitive-latency applications, which is widely considered to be 1 s [18]–[22].

The overall energy consumption minimization optimization problem is formulated in $\mathcal{P}_1$ as follows:

$$\mathcal{P}_1\colon \min_{\mathbf{A},\mathbf{S},\mathbf{f}_N,\mathbf{p}} \sum_{n\in\mathcal{N}} \left[\frac{p_n d_n}{R_{k,z}^n} + T_l C_n D_n \alpha f_n^2\right]$$
$$\text{s.t.} \begin{cases} \text{C1:} & \sum_{k\in\mathcal{K}} a_{k,n} = 1 \;\&\; \sum_{z\in\mathcal{Z}} s_{k,z}^n = 1 \;\forall n\in\mathcal{N} \\ \text{C2:} & \sum_{n\in\mathcal{N}} s_{k,z}^n \le 2 \;\forall k\in\mathcal{K}, z\in\mathcal{Z} \\ \text{C3:} & f_n^{\min} \le f_n \le f_n^{\max} \;\forall n\in\mathcal{N} \\ \text{C4:} & \tau \le T_q, \\ \text{C5:} & 0 \le p_n \le p_{\max} \;\forall n\in\mathcal{N} \\ \text{C6:} & a_{i,j}\in\{0,1\}, s_{i,j}^k \in\{0,1\} \end{cases}$$

B. Problem Transformation and Proposed Solution Approach

Problem $\mathcal{P}_1$ is a nonconvex optimization problem. In addition, owing to the coupling of the optimization variables $a_{k,n}$, $s_{k,z}^k$, f_n, and p_n, it is computationally intractable to optimally solve problem $\mathcal{P}_1$. To overcome the aforementioned computational intractability of $\mathcal{P}_1$, an iterative optimization is devised. More specifically, we decompose $\mathcal{P}_1$ into the following two subproblems, namely, 1) IoT device scheduling and transmit power allocation subproblem for a given IoT device's computation frequency allocation and 2) IoT device's computation frequency allocation subproblem for the determined transmit power and IoT device scheduling. Particularly, considering that IoT device's computation frequency allocation, $f_n^* \;\forall n \in \mathcal{N}$, is given, the IoT device scheduling and power allocation are obtained from the following optimization problem:

$$\mathcal{P}_2\colon \min_{\mathbf{A},\mathbf{S},\mathbf{p}} \sum_{n\in\mathcal{N}} \frac{p_n d_n}{W\log_2\left(1+\gamma_{k,z}^n\right)}$$
$$\text{s.t.} \begin{cases} \text{C1, C2, C5} \\ \text{C4:} \quad T_l \frac{C_n D_n}{f_n^*} + \frac{d_n}{R_{k,z}^n} \le T_{q,k} \;\forall n\in\mathcal{N} \end{cases}$$

where $T_{q,k} = T_q - ((\sum_{n\in\mathcal{N}_k} d_n)/R_{fh})$. For the given transmit power allocation and scheduling among the IoT devices, RRBs, and relays, the IoT device's computation frequency allocation is obtained by solving the following optimization problem:

$$\mathcal{P}_3\colon \min_{\mathbf{f}_N} \sum_{n\in\mathcal{N}} T_l C_n D_n \alpha f_n^2$$
$$\text{s.t.} \begin{cases} \text{C3:} \quad f_n^{\min} \le f_n \le f_n^{\max} \;\forall n\in\mathcal{N} \\ \text{C4:} \quad T_l \frac{C_n D_n}{f_n} + \frac{d_n}{R_{k,z}^{*n}} \le T_{q,k} \;\forall n\in\mathcal{N}. \end{cases}$$

From C4, the lower bound of IoT device's computation frequency can be calculated as $f_n \ge ([T_l C_n D_n]/[T_{q,k} - (d_n/R_{k,z}^{*n})])$. For simplicity, we denote $\hat{f}_n = ([T_l C_n D_n]/[T_{q,k} - (d_n/R_{k,z}^{*n})])$. Then, f_n satisfies $f_n \ge \max\{f_n^{\min}, \hat{f}_n\}$, and accordingly, C3 and C4 can be combined as $\max\{f_n^{\min}, \hat{f}_n\} \le f_n \le f_n^{\max}$. Therefore, $\mathcal{P}_3$ can be expressed as follows:

$$\mathcal{P}_4\colon \min_{f_n} \sum_{n\in\mathcal{N}} T_l C_n D_n \alpha f_n^2$$
$$\text{s.t.} \;\max\left\{f_n^{\min}, \hat{f}_n\right\} \le f_n \le f_n^{\max} \;\forall n\in\mathcal{N}. \tag{7a}$$

Lemma 1: The closed-form solution of subproblem $\mathcal{P}_4$ is obtained as follows:

$$f_n = \begin{cases} f_n^{\min}, & \text{if } \hat{f}_n \le f_n^{\min} \\ \hat{f}_n, & \text{if } f_n^{\min} < \hat{f}_n < f_n^{\max} \\ f_n^{\max}, & \text{if } \hat{f}_n \ge f_n^{\max}. \end{cases} \tag{8}$$

Proof: The objective function in $\mathcal{P}_4$ is monotonically increasing with respect to f_n when $f_n \geq 0$. To minimize the objective function, f_n should be in the feasible set of $\{\max\{f_n^{\min}, \hat{f}_n\}, f_n^{\max}\}$. Hence, the closed-form solution is $f_n = \min\{\max\{f_n^{\min}, \hat{f}_n\}, f_n^{\max}\}$. Then, we repetitively perform the following two closed-form procedures.

1) If $([T_l C_n D_n]/[(T_{q,k}/T_g) - (d_n/R_{k,z}^{*n})]) < f_n^{\max}$, the local computation of IoT device n is feasible. Thus, we set $f_n = \max\{f_n^{\min}, ([T_l C_n D_n]/[(T_{q,k}/T_g) - (d_n/R_{k,z}^{*n})])\}$.
2) If $([T_l C_n D_n]/[(T_{q,k}/T_g) - (d_n/R_{k,z}^{*n})]) = f_n^{\max}$, the local computation of the nth IoT device is feasible. Thus, we set $f_n = f_n^{\max}$. ■

An efficient solution to $\mathcal{P}_1$ can be obtained by solving $\mathcal{P}_2$ as in Section IV-A and $\mathcal{P}_4$ using Lemma 1 alternatively. We observe that for the given solution to $\mathcal{P}_2$, a closed-form solution to $\mathcal{P}_4$ can be readily obtained. Essentially, an efficient solution to $\mathcal{P}_1$ is readily obtained as long as $\mathcal{P}_2$ is efficiently solved. However, $\mathcal{P}_2$ is a mixed-integer nonlinear programming problem. Although exhaustive search and branch-and-bound approaches can obtain near-optimal solution to $\mathcal{P}_2$, such approaches are not suitable for the practical systems due to the significantly increased computational complexity. In the next two sections, we present two efficient methods to solve $\mathcal{P}_2$ by leveraging graph theory.

IV. Joint Energy Aware and IoT Device Selection Graph-Based Approach

In this section, we design a joint energy aware and IoT device selection graph-based (JEADS-G) algorithm to near-optimally solve $\mathcal{P}_1$. In what follows, we first present a graph-theory method to solve $\mathcal{P}_1$. Subsequently, we present the overall JEADS-G algorithm, and analyze its complexity.

A. Near-Optimal Graph Theory-Based Solution to $\mathcal{P}_2$

1) Joint FL Graph Design: Let $\mathcal{A}$ denote the set of all possible combinations between IoT devices, relays, and RRBs, i.e., $\mathcal{A} = \mathcal{N} \times \mathcal{Z} \times \mathcal{K}$, and a is an NOMA association which is an element in $\mathcal{A}$, i.e., $a \in \mathcal{A} = \{n_1^a, n_2^a, z^a, k^a\}$. For convenience, n^a represents the nth IoT device in association a. The weighted undirected J-FL graph is denoted by $\mathcal{G}_{\text{J-FL}}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ where $\mathcal{V}$ stands for the set of all the vertices, $\mathcal{E}$ is the set of all the edges, and $\mathcal{W}$ denotes the set of vertex weights. The designed joint FL graph considers all the conflict transmissions between IoT devices across all RRBs in all relays. A vertex $v = \{n_1^v, n_2^v, z^v, k^v\} \in \mathcal{V}$ in this graph is generated for each association in $\mathcal{A}$, i.e., $v = a$ and $|\mathcal{V}| = |\mathcal{A}|$. In the joint FL graph, two distinct vertices v_i representing a and v_j representing a' are adjacent by a scheduling conflict edge if one of the following cases occurs.

1) *CC1:* The same IoT devices (any IoT device or both IoT devices) are associated with both vertices v_i and v_j.
2) *CC2:* The same RRB in the same relay is associated with both vertices v_i and v_j.

Mathematically, two distinct vertices v_i representing a and v_j representing a' are connecting by a conflict edge if and only if $a \cap a' \neq \emptyset$.

To select the IoT device-RRB-relay scheduling that provides a local minimum energy consumption, we design a proper weight $w(v)$ to each vertex $v \in \mathcal{G}_{\text{J-FL}}$. For notation simplicity, we define the utility of IoT devices n and j as $X_n = T_l C_n D_n \alpha f_n^2 + ([p_n d_n]/[R_{k,z}^n])$, $X_j = T_l C_j D_j \alpha f_j^2 + ([p_j d_j]/[R_{k,z}^j])$, respectively. Therefore, the weight of vertex v that reflects the minimum energy consumption of the nth IoT device and the jth IoT device can be given by

$$w(v) = X_{n^v} + X_{j^v} \tag{9}$$

where X_{n^v} and X_{j^v} are the utility of associated IoT devices n and j to vertex v, respectively. The weight of vertex v in (9) is determined by the transmit powers $\{p_n^*, p_j^*\}$, computation frequency allocation $\{f_n^*, f_j^*\}$, RRB z^v, and relay k^v allocated to them.

In what follows, we transform problem $\mathcal{P}_2$ into an MWIS problem. To this end, we present the following remarks about the MWIS.

Remark 1: Any independent set (IS) Γ in graph $\mathcal{G}_{\text{J-FL}}$ must satisfy: 1) $\Gamma \subseteq \mathcal{G}_{\text{J-FL}}$ and 2) $\forall v, v' \in \Gamma$, we have $(v, v') \notin \mathcal{E}$.

Remark 2: A minimal IS in an undirected graph cannot be expanded to add one more vertex without affecting the pairwise nonadjacent vertices.

Remark 3: The IS Γ is referred to as an MWIS of $\mathcal{G}_{\text{J-FL}}$ if it satisfies: 1) Γ is an IS in graph $\mathcal{G}_{\text{J-FL}}$ and 2) the sum weights of the vertices in Γ offers the minimum among all ISs of $\mathcal{G}_{\text{J-FL}}$. Therefore, the MWIS will be denoted as Γ^*.

Using $\mathcal{G}_{\text{J-FL}}$, the optimization problem $\mathcal{P}_2$ for a fixed f_n $\forall n \in \mathcal{N}$ is similar to MWIS problems in several aspects. In MWIS problems, two vertices must be nonadjacent in the graph, and similarly, in problem $\mathcal{P}_2$, two NOMA clusters cannot be allocated with the same RRB or contain at least one IoT device. Moreover, the objective of problem $\mathcal{P}_2$ is to minimize the delay and energy consumption, and similarly, the goal of MWIS is to minimize the weight of all vertices. Consequently, the following theorem characterizes the solution of allocating IoT devices to the RRBs across all relays such that the total energy consumption is minimized.

Theorem 1: The IoT device coordinated scheduling problem $\mathcal{P}_2$ is equivalent to the MWIS problem over the J-FL graph, wherein the weight of a vertex v is given by

$$w(v) = T_l \alpha \Big(C_n D_n f_n^2 + C_j D_j f_j^2 \Big) + \frac{p_n d_n}{R_{k,z}^n} + \frac{p_j d_j}{R_{k,z}^j}. \tag{10}$$

The set of scheduled IoT devices to the zth RRB in the kth relay is obtained by combining the vertices of the MWIS $\mathbf{I}$ in the NOMA-coordinated graph.

Proof: This theorem can be proved by demonstrating the following facts. The first fact establishes the equivalency between $\mathcal{P}_2$ and MWIS problems. Specifically, using $\mathcal{G}$, $\mathcal{P}_2$ is similar to MWIS problems. In MWIS problems, two vertices must be nonadjacent in the graph, and similarly, in problem $\mathcal{P}_2$, two NOMA clusters cannot be allocated with the same RRB or contain at least one IoT device. Afterward, the weight of each vertex is set to be the minimum energy consumption contribution of the corresponding NOMA cluster to the network. Therefore, the MWIS is a feasible solution with the minimum energy consumption, i.e., the MWIS is the feasible solution to $\mathcal{P}_2$. To finalize the proof, we now prove that the weight of the MWIS is the objective function in $\mathcal{P}_2$ to be minimized. Let $\Gamma = \{v_1, v_2, \ldots, v_{|\Gamma|}\}$, $v \in \mathcal{G}$. Let a vertex $v \in \mathcal{V}$ is associated with 2-IoT devices NOMA cluster (n, j). The weight of the MWIS over all the vertices that are representing the corresponding NOMA clusters over all RRBs/relays can be written as

$$\begin{aligned} w(\Gamma) &= \sum_{v\in\Gamma} w(v) \\ &= \sum_{\mathbf{k}\in\mathcal{K}} \sum_{z\in\mathcal{Z}} \left(T_l \alpha \left(C_n D_n f_n^2 + C_j D_j f_j^2 \right) + \frac{p_n d_n}{R_{k,z}^n} + \frac{p_j d_j}{R_{k,z}^j} \right). \end{aligned} \tag{11}$$

Therefore, the problem of minimizing the energy consumption $\mathcal{P}_2$ is equivalent to the MWIS problem among the minimal sets in the J-FL graph. ■

Essentially, any maximal IS in $\mathcal{G}_{\text{J-FL}}$ represents IoT device scheduling and power control satisfies the following criteria.

1) The represented IoT devices by vertices in the MWIS are scheduled to the best RRBs/relays so as to upload their local parameters quickly.
2) The power level of IoT devices in each NOMA cluster is near optimal. This results in a good transmission rate from IoT devices to RRBs, and thus reduces the uplink transmission delay for transmitting the local parameters to the relays.
3) A vertex (representing a cluster of two IoT devices) with a smaller weight will have a higher priority to be selected in participating in the training process to minimize the objective of $\mathcal{P}_2$.

It is worth mentioning that selecting more IoT devices may increase the longest time of uploading parameters, and thereby, violate constraint C4. Hence, each vertex v in the MWIS will be iteratively evaluated to see if the represented IoT devices in that vertex can still meet the delay constraint of a global iteration. If the delay constraint is satisfied, the vertex v is selected.

2) Power Control Optimization: From the designed $\mathcal{G}_{\text{J-FL}}$, we obtain a set of vertices that represent NOMA clusters. Each NOMA cluster includes two IoT devices that simultaneously transmit to a relay over an RRB. For each vertex, we aim to determine transmit power allocations of the IoT devices such that 1) the overall uplink transmission rate is improved by suppressing the interference between the IoT devices and 2) the energy consumption for wireless transmission is reduced. Without loss of generality, we consider a vertex where the nth and the jth IoT devices are clustered, and both IoT devices transmit to the kth relay over the zth RRB. For such a vertex, we formulate the transmit power allocation subproblem as $\mathcal{P}_2 - 1$ as follows:

$$\begin{aligned} \mathcal{P}_2 - 1\colon \max_{\substack{0\le p_n \le p_{\max},\\ 0\le p_j\le p_{\max}}} \; & W\left(\log_2\left(1+\gamma_{k,z}^n\right) + \log_2\left(1+\gamma_{k,z}^j\right)\right) \\ & -V\left(p_n + p_j\right) \\ \text{s.t.} \; & \begin{cases} \text{C8:} \quad W\log_2\left(1+\gamma_{k,z}^n\right) \ge R_{th,n} \\ \text{C9:} \quad W\log_2\left(1+\gamma_{k,z}^j\right) \ge R_{th,j} \end{cases} \end{aligned}$$

In subproblem $\mathcal{P}_2 - 1$, $R_{th,n}$ and $R_{th,j}$ are the required uplink data rates for the nth and jth IoT devices, respectively; and V is a given weight factor. In particular, $R_{th,n} = ([T_g d_n]/[T_{q,k} - T_g T_l((C_n D_n)/f_n^*)])$ and $R_{th,j} = ([T_g d_j]/[T_{q,k} - T_g T_l((C_j D_j)/f_j^*)])$. Essentially, the rate constraints C8 and C9 satisfy the FL delay constraint C4. On the other hand, the weight factor V is selected to strike a suitable balance between capacity and energy consumption of the vertices.

Note that the power allocation depends on the channel gain of the associated IoT devices. To this end, we first define $\Delta_n = (\sigma^2/p_{\max})(2^{R_{th,n}/W} - 1)$ and $\Delta_j = (\sigma^2/p_{\max})(2^{R_{th,j}/W} - 1)$. Thereafter, we consider the following four cases: *Case I:* $|G_{k,z}^n|^2 < \Delta_n$ and $|G_{k,z}^j|^2 < \Delta_j$, *Case II:* $|G_{k,z}^n|^2 \ge \Delta_n$ and $|G_{k,z}^j|^2 < \Delta_j$, *Case III:* $|G_{k,z}^n|^2 < \Delta_n$ and $|G_{k,z}^j|^2 \ge \Delta_j$, and *Case IV:* $|G_{k,z}^n|^2 \ge \Delta_n$ and $|G_{k,z}^j|^2 \ge \Delta_j$. The transmit power allocations, (p_n^*, p_j^*), for each case are given as follows.

Case I: In this case, both IoT devices can not satisfy the rate constraints even using the maximum transmit power. Consequently, both IoT devices suspend their data transmission, and we obtain $p_n^* = 0$ and $p_j^* = 0$.

Case II: In this case, only the nth IoT device can satisfy the required rate constraint, and the transmission of the jth IoT device is suspended. Therefore, we obtain $p_j^* = 0$ and $p_n^* = (\sigma^2/|G_{k,z}^n|^2)(2^{R_{th,j}/W} - 1)$.

Case III: In this case, only the jth IoT device can satisfy the required rate constraint, and the transmission of the nth IoT device is suspended. Therefore, we obtain $p_n^* = 0$ and $p_j^* = (\sigma^2/|G_{k,z}^j|^2)(2^{R_{th,j}/W} - 1)$.

Case IV: In this case, both IoT devices can simultaneously transmit. Without loss of generality, we assume that $|G_{k,z}^j|^2 < |G_{k,z}^n|^2$, i.e., the nth IoT device has a better channel gain compared to the jth IoT device. According to the NOMA principle, the kth relay first decodes the nth IoT device's signal, and subsequently, decodes the jth IoT device's signal after removing the interference from the nth IoT device via applying the SIC technique. We first introduce the following lemma to update the jth IoT device's power allocation

Lemma 2: Assume that the given transmit power allocations for the nth and the jth IoT devices are $\tilde{p}_n$ and $\tilde{p}_j$, respectively. Therefore, the jth IoT device's transmit power allocation to maximize subproblem $\mathcal{P}_2 - 1$ is obtained as follows:

$$p_j = \left[\frac{\frac{\gamma_{k,z}^j}{1+\gamma_{k,z}^j}}{V + \frac{\left(\gamma_{k,z}^n\right)^2}{1+\gamma_{k,z}^n} \frac{|G_{k,z}^j|^2}{\tilde{p}_n |G_{k,z}^n|^2}} \right]_{p_{th}}^{p_{\max}} \tag{12}$$

where $\gamma_{k,z}^n$ and $\gamma_{k,z}^j$ are calculated by plugging $\tilde{p}_n$ and $\tilde{p}_j$ to (1), $p_{th} = (\sigma^2/|G_{k,z}^j|^2)(2^{R_{th,j}/W} - 1)$, and $[\cdot]_{p_{th}}^{p_{\max}}$ denotes projection in the range of $[p_{th}, p_{\max}]$.

Proof: The proof is omitted due to the space limitation. ■

We consider a suboptimal approach to iteratively update the transmit power allocation of both the nth and the jth

IoT devices in the inner and outer loop. Specifically, using a bi-section search method, the outer loop adjusts the power allocation of the nth IoT device such that the rate constraint C8 is satisfied, and the inner loop adjusts the power allocation of the jth IoT device according to Lemma 2. Let us denote the minimum and maximum power level for the nth IoT device as $p_{n,\text{low}}$ and $p_{n,\text{high}}$. The initial transmit power of the nth IoT device is obtained as $p_n = ([p_{n,\text{low}} + p_{n,\text{high}}]/2)$. By plugging the transmit power of the nth IoT device to Lemma 2, the jth IoT device's transmit power is determined. Thereafter, the achievable rate of the nth IoT device is calculated. If $W \log_2(1 + \gamma^n_{k,z}) > R_{th,1}$, $p_{n,\text{high}} \leftarrow p_n$ is applied and if $W \log_2(1 + \gamma^n_{k,z}) < R_{th,1}$, $p_{n,\text{low}} \leftarrow p_n$ is applied. Then, the transmit power of the nth IoT device is updated as $p_n = ([p_{n,\text{low}} + p_{n,\text{high}}]/2)$. The aforementioned procedures are repeated until $|W \log_2(1+\gamma^n_{k,z}) - R_{th,1}|$ approaches a small value. The final values of p_n and p_j provide the required power allocations for case IV.

3) Algorithm Development: Algorithm 1 summarizes the overall steps to solve problem $\mathcal{P}_2$. In particular, for a given allocation of the computation frequency, Algorithm 1 sequentially executes the following two steps to obtain a set of suitable scheduling among the IoT devices, RRBs, and relays (represented by the selected MWIS Γ^*) and the corresponding transmit power allocation.

1) We first design the joint FL graph as follows. We generate all the possible schedules $\mathcal{A}$ of IoT device-NOMA clusters, RRBs, and relays. Afterwards, for each feasible schedule $a \in \mathcal{A}$, a vertex $v \in \mathcal{G}_{\text{J-FL}}$ is generated. We calculate the optimal power levels of each association by solving the optimization problem $\mathcal{P}_2 - 1$. The vertex in $v \in \mathcal{G}_{\text{J-FL}}$ is created by appending the computed power levels and the corresponding rates to that vertex. We repeat the same steps above for all vertices. The J-FL graph is, then, constructed by adding connections according to **CC1** and **CC2**.
2) Subsequently, we iteratively and greedily select the MWIS Γ^* among all the minimal ISs Γ in the J-FL graph, where in each iteration we implement the following procedures. We compute the weight of all generated vertices using (9) and select the minimum weight v^* among all other corresponding vertices. The selected vertex v^* is, then, added to Γ^*, where Γ^* is initially empty. Afterwards, we update the $\mathcal{G}_{\text{J-FL}}$ graph by removing the selected vertices v^* and its connected vertices. This to ensure that the next selected vertex is not in conflict connection with the already selected vertices in Γ^*. We continue the process until no more vertices exist in the J-FL graph $\mathcal{G}_{\text{J-FL}}$. Since each RRB in each relay contributes by a single vertex, the number of vertices in Γ^* is ZK.

B. Overall JEADS-G Algorithm and Computational Complexity

The basic idea of JEADS-G is to iteratively perform the following two steps, namely, 1) obtain a suitable solution to problem $\mathcal{P}_2$ by executing Algorithm 1 while considering a fixed allocation of the computation frequency and 2) update the computation frequency allocation for the obtained solution to problem $\mathcal{P}_2$. The JEADS-G algorithm is summarized in Algorithm 2. We emphasize that Algorithm 1 near-optimally solves $\mathcal{P}_2$, thanks to the generation of all possible NOMA clusters. Since both subproblems of $\mathcal{P}_1$ are near-optimally solve, an iterative procedure presented JEADS-G certainly obtains a converged solution. Due to the decomposed approach of updating resource allocations, the converged solution is not necessarily optimal. However, our simulation results show JEADS-G not only considerably outperforms the benchmark schemes but also provides a reasonable performance gap from the exhaustive search-based optimal solution.

Algorithm 1: Graph Theory-Based Algorithm to Solve $\mathcal{P}_2$

```
Require: $\mathcal{N}$, $\mathcal{K}$, $\mathcal{Z}$, $G^n_{k,z}$, $f_n$, and $p_{\max}$,
  $(n, k, z) \in \mathcal{N} \times \mathcal{K} \times \mathcal{Z}$.
Initialize $\Gamma^* = \emptyset$.
Solve $\mathcal{P}_2$ for fixed $f_n$, $\forall n \in \mathcal{N}$.
Design $\mathcal{G}_{\text{J-FL}}$ according to Section IV-A1.
for each $v \in \mathcal{G}_{\text{J-FL}}$ do
    Solve $\mathcal{P}_2 - 1$ to compute the power allocations
      $\mathbf{p} = \{p^*_{n^v_1}, p^*_{n^v_2}\}$.
    Obtain $v = \{(r^*_{n^v_1}, p^*_{n^v_1}, z^v, k^v), (r^*_{n^v_2}, p^*_{n^v_2}, z^v, k^v)\}$
      according to $\mathbf{p}$.
    Calculate $w(v)$ using (9).
end for
$\mathcal{G}_{\text{J-FL}}(\Gamma^*) \leftarrow \mathcal{G}_{\text{J-FL}}$.
while $\mathcal{G}_{\text{J-FL}}(\Gamma^*) \neq \emptyset$ do
    $v^* = \arg\min_{v \in \mathcal{G}_{\text{J-FL}}(\Gamma)} \{w(v)\}$.
    Set $\Gamma^* \leftarrow \Gamma^* \cup v^*$ and set $\mathcal{G}_{\text{J-FL}}(\Gamma^*) \leftarrow \mathcal{G}_{\text{J-FL}}(v^*)$.
end while
Obtain $\Gamma^*$ .
```

Algorithm 2: Proposed JEADS-G

```
Input: $\mathcal{N}$, $\mathcal{K}$, $\mathcal{Z}$, $C_n D_n$, $T_l$, $W$, $\sigma^2$, $p_{\max}$, $T_q$, $f^{\min}_n$, and
  $f^{\max}_n$.
Output: IoT device scheduling, $p_n$, and $f_n$.
Initialize the number of iteration $t = 1$, $f^{(0)}_n = f^{\min}_n$,
  $f^{(1)}_n = f^{\max}_n$, $\forall n \in \mathcal{N}$.
while $f^{(t)}_n \neq f^{(t-1)}_n$ and $t < T_{\max}$ do
    Solve $\mathcal{P}_2$ as in Algorithm 1.
    Calculate the solution $f^{(t)}_n$ of the problem $\mathcal{P}_4$
      according to Lemma 1.
    $t = t + 1$.
end while
Return IoT device scheduling, $p^{(t)}_n$, and $f^{(t)}_n$.
```

The computational complexity of JEADS-G is dominated by the complexity of constructing the J-FL graph. To construct the J-FL graph, we need to generate all NOMA clusters representing all vertices in the J-FL graph, which needs a computational complexity of $\mathcal{O}\binom{N}{2}$. Then, connecting all these vertices requires a computational complexity of $\mathcal{O}\binom{N}{2}^2$.

Therefore, the overall computational complexity is $\mathcal{O}\binom{N}{2}^2$. Such high complexity is due to generating all the possible NOMA clusters which increases significantly as the number of devices and relays in the network increases. For the high implementation complexity of the joint approach in large-scale networks, in the next section, we develop an efficient and alternative graph pruning solution that has relatively low implementation complexity.

V. Graph Pruning Approach-Based Low-Complexity Solution

The required computational complexity of the proposed JEADS-G algorithm is considerably increased for a large-scale network. To reduce the computational complexity, we propose a sequential pruning graph (SPG) approach. The key idea of the SGP approach is to solve $\mathcal{P}_2$ using a reduced NOMA graph. In such a reduced NOMA graph, only the NOMA clusters significantly contributing to reduce the energy consumption are generated. Essentially, we do not need to generate all the possible NOMA clusters in the network which significantly reduces its size. In what follows, we first present our proposed low-complexity solution approach to solve $\mathcal{P}_2$. Next, we summarize the SPG algorithm to efficiently solve problem $\mathcal{P}_1$.

A. Low-Complexity Solution to Subproblem $\mathcal{P}_2$

In this section, we develop a low-complexity approach to solve $\mathcal{P}_2$. Specifically, we first design a reduced graph for all IoT device-RRB-relay feasible schedules, and subsequently, we efficiently allocate the transmit power of the IoT devices in each schedule. In what follows, the aforementioned two stages are explained.

Stage I (IoT Device Feasible Scheduling): This stage consists of the graph design and greedy MWIS method.

1) Graph Design: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ represent an undirected graph. The graph $\mathcal{G}$ is constructed by generating a vertex v for each 2-IoT devices, RRB, and relay in the network as follows. We start from RRB $z = 1$, and assume that IoT device $n = 1$ is allocated to it. Then, we find the available NOMA clusters according to the possible two scenarios.

1) If IoT device $n = 1$ is not in the service area of the kth relay, we check IoT device $n = 2$ for possible association to RRB z and relay k, and then continue finding the second IoT device. Then, we check the feasibility of constraint C4.
2) If IoT device $n = 1$ is in the service area of the kth relay, then we find the second IoT device $j = n + 1$ (currently, $j = 2$), for the $(n = 1, z = 1)$ pair. Afterwords, we find p_n^*, p_j^* and calculate the rates, and then, we generate a vertex $v = \{(r_n^*, z, k), (r_j^*, z, k)\}$ that represents a feasible NOMA cluster. Given r_n^*, r_j^*, we then compute the weight of that vertex $w(v) = X_n + X_j$ and update $\mathcal{G}$. If adding $j = 2$ is infeasible, we let $j = j + 1 = 3$, and we verify the feasibility and repeat the aforementioned step.

2) MWIS Search Method: In this step, the SGP algorithm itratively and greedily selects the MWIS Γ^* among all the minimal ISs Γ in the graph $\mathcal{G}$, where in each iteration we implement the following procedures. We compute the weight of all generated vertices using (9). The vertex with the minimum weight v^* is selected among all other corresponding vertices. The selected vertex v^* is, then, added to Γ^* that is initially empty. Afterwards, we update the $\mathcal{G}$ graph by removing the selected vertices v^* and its connected vertices. As such, the next selected vertex is not in conflict connection with the already selected vertices in Γ^*. The process continues until no more vertices exist in $\mathcal{G}$. Since each RRB in each relay contributes by a single vertex, the number of vertices in Γ^* is ZK.

Stage II (Power optimization): Solving the power allocation in each vertex for the resulting IoT device-RRB/relay schedule is omitted in this section becasue it can follow the solution of $\mathcal{P}_2 - 1$ in Section IV-A1.

Based on the aforementioned analysis, a low-complexity two-stage algorithm to solve subproblem $\mathcal{P}_2$ is summarized in Algorithm 2.

B. Overall SPG Algorithm and Computational Complexity

Algorithm 4 summarizes the overall steps of the SPG algorithm to obtain a low-complexity solution to problem $\mathcal{P}_1$. Similar to the JEDAS-G algorithm, the SPG algorithm also iteratively solves subproblems $\mathcal{P}_2$ and $\mathcal{P}_4$. The computational complexity of the SPG algorithm is dominated by step 5. In particular, the computational complexity of Algorithm 3 is dominated by the required complexity of the graph construction stage. In order to generate all the vertices using the low-complexity graph punning method, a total of $\mathcal{O}(NKZ)$ computational complexity is required. The required complexity of connecting the generated vertices, i.e., the required complexity of finding neighborhood of the generated vertices is $\mathcal{O}((NKZ)^2)$. Therefore, the overall computational complexity of the proposed SPG algorithm is obtained as $\mathcal{O}(NKZ + (NKZ)^2) \approx \mathcal{O}(N^2K^2Z^2)$. Essentially, the SPG algorithm requires a significantly reduced computational complexity than the near-optimal JEADS-G algorithm.

VI. Numerical Results

A. Simulation Setting and Schemes Under Consideration

In our simulations, we consider a hexagonal cell of radius 1500 m where relays and CS have fixed locations and IoT devices are distributed uniformly within the cell. The CS is located at the cell center. The channel model for IoT device-relay transmissions follows the standard model, which consists of three components: 1) path-loss of $128.1 + 37.6\log_{10}(\text{dis.[km]})$; 2) log-normal shadowing with 4-dB standard deviation; and 3) the Rayleigh channel fading with zero-mean and unit variance. The noise power, relay's power, and maximum' IoT device power are assumed to be -174 dBm/Hz and $q_k = p_{\max} = 3$ W, respectively, [22]. The weighting factor V is set to 0.3. The total number of global and local FL iterations are calculated as $T_g = (2\beta^2/[(2\vartheta - \beta\eta)\vartheta\eta])\ln(1/\epsilon_g)$, $T_l = (2/[(2 - \delta\beta)\delta\vartheta])\ln(1/\epsilon_l)$, respectively, with $\beta = 4, \eta = 1/3, \delta = 1/4, \vartheta = 2, \epsilon_g = \epsilon_l = 10^{-3}$ [18]. The FL time threshold T_q is 1 sec [18]. The bandwidth of each RRB is 20 MHz. Unless otherwise stated, we set the numbers of relays

Algorithm 3: Low-Complexity Two-Stage Algorithm to Solve $\mathcal{P}_2$

```
Data: $\mathcal{N}$, $\mathcal{K}$, $\mathcal{Z}$, $G^n_{k,z}$, $p_{\max}$, and $f^*_n$,
$(n, k, z) \in \mathcal{N} \times \mathcal{K} \times \mathcal{Z}$.
Stage 1: IoT device feasible scheduling
  • Initialize $\mathcal{G} = \emptyset$.
    for $k = 1 : K$ do
      for $z = 1: Z$ do
        Set $n = 1$
        if the n-th IoT device in $\mathcal{N}_k$ then
          Set $j = n + 1$
          while $j < N$ do
            if the j-th IoT device in $\mathcal{N}_k$ then
              Based on $p_n$ and $p_j$, calculate $r_n$, $r_j$.
              Generate vertex
              $v = \{(r^*_n, z, k), (r^*_j, z, k)\}$ and set
              $\mathcal{G} \longleftarrow \mathcal{G} \cup v$.
            end if
            $j = j + 1$
          end while
        else
          $n = n + 1$
        end if
      end for
    end for
  • For each $v \in \mathcal{V}$, finds its neighborhood $\mathcal{N}_{\mathcal{G}}(v)$ according
    to CC1 and CC2.
  • Calculate the weight of each vertex $w(v)$ as in (9).
  • Let $\Gamma^* = \emptyset$, $l = 0$, $\mathcal{G}_l = \mathcal{G}$.
  • MWIS Search Method:
    while $\mathcal{V}(\mathcal{G}_l) \neq \emptyset$ do
      $v^* = \arg\min_{v \in \mathcal{G}_l(\Gamma)}\{w(v)\}$ and set $\Gamma \leftarrow \Gamma \cup v^*$.
      Let $\mathcal{V}(\mathcal{G}_{l+1}) = \mathcal{V}(\mathcal{G}_l(\Gamma))$.
      $l = l + 1$
    end while
  • Output: The MWIS and its corresponding IoT device
    scheduling.
Stage 2: Power allocation: Allocate transmit power in
the NOMA clusters according to the method described in
Section IV-A1.
```

and RRBs to 9 and 4, respectively. The fronthaul capacity R_{fh} is set to 150 Mbit/s. For each IoT device, the number of data samples D_n is randomly chosen from 800 to 1000. Other parameters are summarized in Table II. To assess the performance of our proposed schemes, we simulate various scenarios with different numbers of IoT devices N, data size $\mathtt{D}_n$, number of RRBs Z, number of data samples D_n, computation frequency allocation f_n, and parameter data size. For the sake of comparison, our proposed JEADS-G and SGP schemes are compared with the following baseline schemes, inspired by the existing works [43]–[45], respectively.[1]

Algorithm 4: Proposed SPG Algorithm

```
Input: $\mathcal{N}$, $\mathcal{K}$, $\mathcal{Z}$, $C_nD_n$, $T_l$, $W$, $\sigma^2$, $p_{\max}$, $T_q$, $f^{\min}_n$, and
$f^{\max}_n$.
Output: IoT device scheduling, $p_n$, and $f_n$.
Initialize the number of iteration $t = 1$, $f^{(0)}_n = f^{\min}_n$,
$f^{(1)}_n = f^{\max}_n$, $\forall n \in \mathcal{N}$.
while $f^{(t)}_n \neq f^{(t-1)}_n$ and $t < T_{\max}$ do
  Obtain a solution $\mathcal{P}_2$ by executing Algorithm 3.
  By plugging the updated solution of $\mathcal{P}_2$ Lemma 1,
  calculate the updated computation frequency
  allocation $f^{(t)}_n$, $\forall n$.
  $t = t + 1$.
end while
Return IoT device scheduling, $p^{(t)}_n$, and $f^{(t)}_n$.
```

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Circle radius of relay's service area $\mathtt{R}$	500 m
learning local parameter size, d_n, d_k	$[5 - 10]$ Kbit [18]
IoT device data size, $\mathtt{D}_n$	$[0.5 - 1]$ Mbit
IoT device processing density, C_n	$[600 - 800]$ [18]
IoT device computation frequency, f_n	$[0.0003 - 1]$ G cycles/s
CPU architecture based parameter, α	10^{-28} [38]

1) *Power Adaptation:* This scheme minimizes the energy consumption by optimizing the power level of IoT devices and fixing the computation frequency allocation to its maximum value.

2) *Computation Adaptation:* This scheme minimizes the energy consumption by optimizing the computation frequency allocation and fixing the power level to its maximum value.

3) *Fixed*: This scheme employs random IoT device scheduling and fixes both the computation frequency allocation and transmission power to their maximum values.

4) *OFDMA Scheme:* In this scheme, we consider that the local computation at the IoT devices are executed similar as that in the proposed schemes. The difference is that each RRB is randomly allocated with a single IoT device that can transmit at the maximum power.

B. Simulation Results and Discussion

We adopt three different performance metrics as follows: 1) the *energy consumption* that represents the objective in $\mathcal{P}_1$; 2) number of selected IoT devices; and 3) the *FL time* as expressed in (3).

1) Energy Consumption Performance: In Fig. 3, we plot the energy consumption versus the number of IoT devices. Our proposed JEADS-G and SPG schemes have the following two attributes. First, they judiciously schedule IoT devices to relays/RRBs, adapt the transmission rate of each IoT device, and optimize the transmission power of each IoT

[1] Note that OFDMA, power adaptation, and computation adaptation are well-investigated techniques to deal with models uploading in IoT systems [22]. Consequently, to showcase the effectiveness of our proposed schemes, we compare the proposed schemes with the aforementioned schemes.

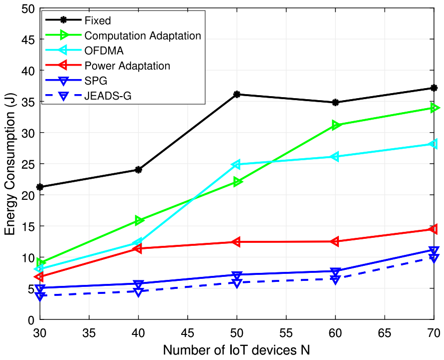

Fig. 3. Energy consumption versus number of IoT devices N for $K = 9$ and $Z = 4$.

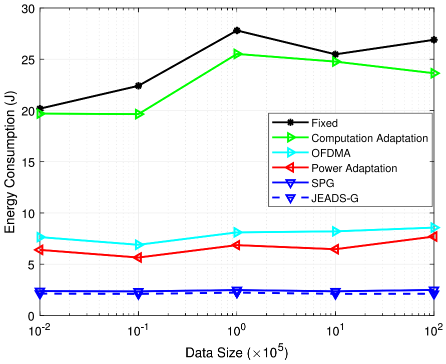

Fig. 4. Energy consumption versus data size for $N = 50$, $K = 9$, and $Z = 4$.

device. Second, JEADS-G and SPG efficiently optimize the computation frequency allocation of IoT devices and relays. Leveraging these two attributes, our proposed JEADS-G and SPG schemes significantly reduce the energy consumption compared to the benchmark schemes, as depicted from both Fig. 3. In particular, the power adaptation scheme selects the maximum computation frequency allocation for each IoT device and each relay. Consequently, the power adaptation scheme results in higher energy consumption for local learning, and it increases the energy consumption of the system in both scenarios. The computation adaptation scheme ignores the power optimization that leads to more interference among the IoT devices and increased offloading transmission time. As a result, the computation adaptation scheme leads to a high energy consumption. Finally, the fixed scheme has the largest energy consumption because it chooses the maximum CPU frequency and transmission power. Accordingly, from an energy consumption perspective, it is inefficient to offload local parameters to relays while ignoring the power allocation and employing random IoT device scheduling to relays/RRBs.

In Fig. 4, we plot the energy consumption versus the data size $\mathtt{D}_n$. When the data size is small (around 1 Kbit), both JEADS-G and SPG work superior in terms of minimizing the energy consumption. When the data size is nearly 10 Mbits, the energy consumption performance of our proposed JEADS-G scheme does not change much and has a performance of 3 J. This is becasue the IoT devices perform local learning on the data and offload the local learning parameters only. Therefore, JEADS-G and SPG do not consume more energy when the data size increases.

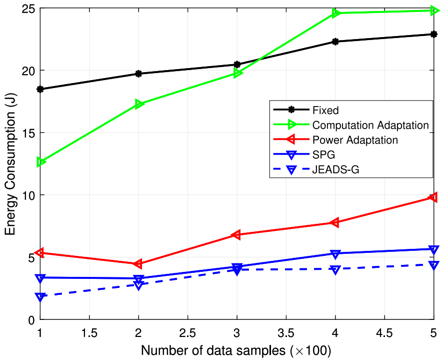

Fig. 5. Energy consumption versus number of data samples for $N = 50$, $K = 9$, and $Z = 4$.

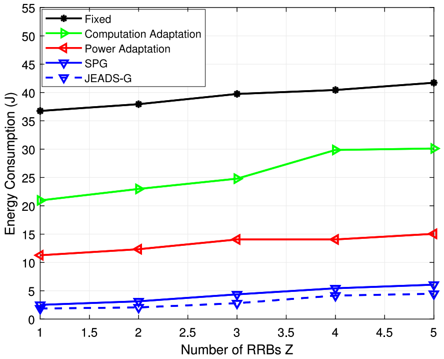

Fig. 6. Energy consumption versus number of RRBs Z for $N = 50$ and $K = 9$.

In Fig. 5, we show the energy consumption versus the number of data samples D_n. The number of data samples affects the CPU-related energy consumption. Similar to our discussions for Fig. 3, the computation adaptation and fixed schemes severely degrades the energy consumption performance. Specifically, the energy consumption of the fixed scheme is increased with the number of data samples. However, the energy consumption of the power adaptation and our proposed JEADS-G and SPG schemes do not significantly change, e.g., see Fig. 5. Using the optimized resource allocations, JEADS-G and SPG incur the least energy consumption for both small and large data samples.

In Fig. 6, we plot the energy consumption versus the number of RRBs Z. As can be seen, the consumed energy of all schemes are increased with the increase in the number of RRBs. This is due to the fact that as the number of RRBs increases, more IoT devices are selected, which in turn increases the energy consumption. Specifically, when $Z = 1$, the maximum number of accommodated IoT devices by the relays is $2ZK = 2 \times 1 \times 9 = 18$, thus the consumed energy of all schemes is low. As the number of RRBs is increased, the energy consumption of all the schemes is slowly increased. This can be explained by the fact that when the number of

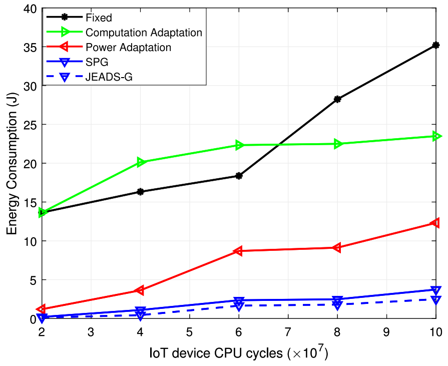

Fig. 7. Energy consumption versus computation frequency for $N = 50$, $K = 9$, and $Z = 4$.

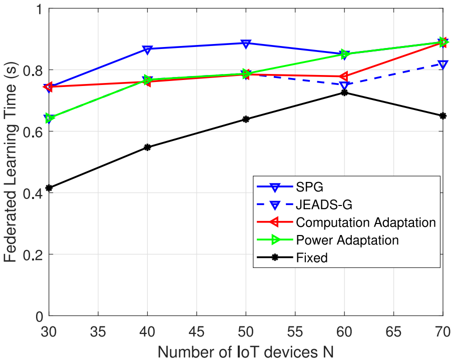

Fig. 8. FL time versus number of IoT devices N for $K = 9$ and $Z = 4$.

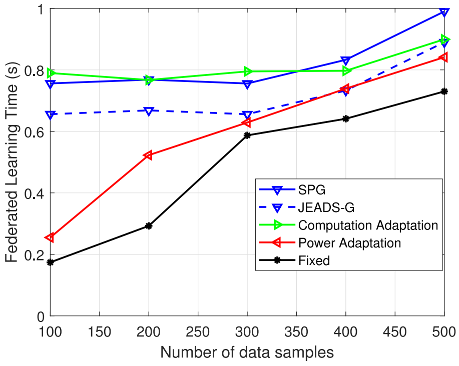

Fig. 9. FL time versus number of data samples D_n for $K = 9$ and $Z = 4$.

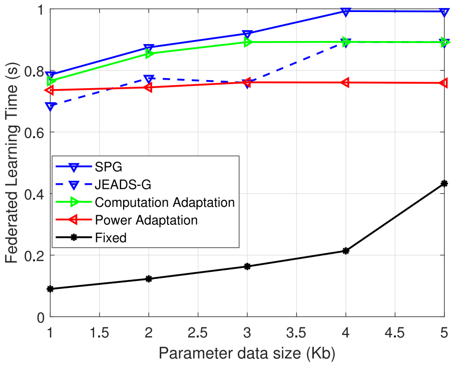

Fig. 10. FL time versus parameter data size d_n for $K = 9$ and $Z = 4$.

RRBs goes beyond 2, no more IoT devices can be accommodated. Thus, the consumed energy of all schemes do not change much. For a fair comparison, we consider that all the schemes serve the same set of IoT devices in the available RRBs of a given relay. The proposed JEADS-G and SPG schemes, however, take benefit from optimizing the transmit power and computation frequency allocation. Essentially, the proposed schemes achieve reduced energy consumption compared to the benchmark schemes.

In Fig. 7, we show the energy consumption versus the number of CPU cycles. The number of CPU cycles ranges from 2×10^7 to 10×10^7. The number of CPU cycles dominates the computation energy consumption. Hence, the energy consumption of both fixed and power adaptation schemes, that fix the computation frequency allocation at the highest value, is considerably increased with the increasing number of computation cycles. On the other hand, the energy consumption of both computation adaptation and JEADS-G and SPG schemes with adjustable CPU frequencies do not change much as shown in Fig. 7(a) and (b). As expected, using both transmit power and computation frequency allocation, JEADS-G and SPG incur the least energy consumption for both small and large numbers of CPU cycles.

2) FL Time Performance That Measures the Time Duration of Model Training Computation and Wireless Transmission in Each Global Round: The presented values of the FL time in Figs. 8–10 are calculated based on (3) that is given in Section II-B. In Figs. 8–10, we plot the FL time versus: number of IoT devices N, number of data samples D_n, and parameter data size d_n, respectively. First, it is clear that the FL time depends on the transmission time and the computation learning time of IoT devices. Since the local learning parameters have small size, the transmission time for offloading such parameters to relays/CS requires smaller portion of the overall FL time compared with the computation training time. Consequently, the FL time is dominated by the computation training time. As can be seen from Figs. 8–10, a fixed scheme, that chooses the maximum CPU frequency, effectively minimizes the FL time at the cost of consuming the most energy as shown in Fig. 3 to Fig. 7. Our proposed JEADS-G and SPG schemes adjust the CPU frequency and power transmissions so that it effectively minimizes the consumed energy within the FL time of 1 s. In Figs. 8 and 10, the FL time of all algorithms does not change much with the number of IoT devices and local parameter size. This is because the FL time is mainly controlled by the longest local training time of one IoT device, which change slowly when the number of IoT device and the local parameter size is increased.

3) Energy Consumption and Number of Selected IoT Devices Performance: In Fig. 11(a) and (b), we plot the consumption power and number of selected IoT devices versus the number of IoT devices N. As explained in Fig. 3, our proposed JEADS-G and SPG schemes offer an improved energy consumption performance as compared to all other schemes. When the number of IoT devices increases, all schemes consume more energy as seen from Fig. 11(a). This is because as the number of

TABLE III
EXECUTION TIMES OF DIFFERENT SCHEMES UNDER CONSIDERATION

Solution	Time(sec)- Small network	Time(sec)- Large network
JEADS-G	0.3524119	21.230017
SPG	0.063895	0.142072
Fixed	0.011926	0.016277
Power Adaptation	0.007987	0.015534
Computation Adaptation	0.013522	0.019362

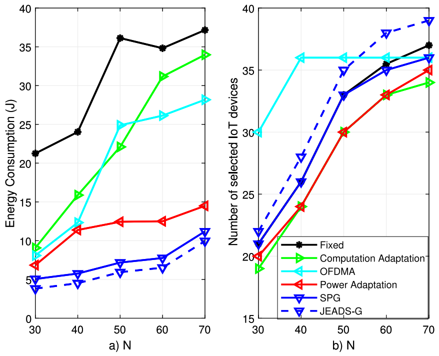

Fig. 11. Energy consumption and number of selected IoT devices versus N for $K = 9$ and $Z = 4$.

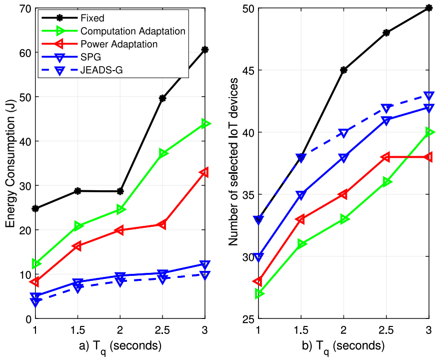

Fig. 12. Energy consumption and number of selected IoT devices versus T_q for $N = 50$, $K = 9$, and $Z = 4$.

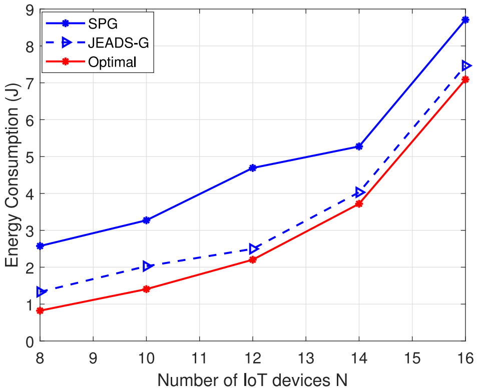

Fig. 13. Energy consumption versus a small number of IoT devices N for $K = 3$ and $Z = 2$.

total IoT devices is increased, more IoT devices are selected for learning as shown in Fig. 11(b). However, Fig. 11(b) depicts that the number of selected IoT devices is slowly increased when $N > 50$. Such an observation is explained by the following argument. Recall, the delay of each FL global iteration is limited by the delay experienced by the straggler device. On the other hand, as the number of devices is increased, the minimum channel gain of the devices is reduced as well. Such a fact eventually increases the delay observed by the straggler device, when the number of available devices is increased. Therefore, to satisfy the delay constraint of each FL global iteration, the number of selected IoT devices is eventually saturated for a large number of available IoT devices.

Further, in Fig. 12(a) and (b), we plot the energy consumption and number of selected IoT devices versus T_q. A larger T_q indicates more IoT devices could potentially be selected to meet a longer deadline and vice versa. As shown in Fig. 12(a) and (b), the fixed scheme always selects the most IoT devices, and accordingly, consumes the most energy. On the other hand, JEASD-G, SPG, and computation and power adaptation schemes select fewer devices than a fixed scheme to minimize the power consumption. Also, JEASD-G and SPG consume the lowest energy but selects more devices than computation and power adaption schemes. As a result, JEASD-G incurs the lowest energy consumption of selected devices, which confirms the fact that JEASD-G can optimize both energy consumption and a number of selected IoT devices under different values of T_q.

4) Optimality and Computational Complexity: In Fig. 13, we plot the performance of our proposed schemes and optimum solution (obtained using an exhaustive graph theoretical search over all possible MWISs) versus the number of devices N for a small network of three relays and two RRBs per relay. It is seen that the proposed JEADS-G scheme achieves near-optimal performance. This is because it considers all NOMA clusters in the network and performs a joint optimization of IoT device scheduling and power allocation, computation allocation, and offloading decisions. Compared to the optimal solution, our proposed SPG scheme has certain degradation since it considers fixed local computations and partial optimization of graph construction.

We further study in Table III the consumed time of MATLAB to execute all schemes in different network setups. We consider two network setups: 1) small setup of five relays, two RRBs per relay, and ten devices and 2) relatively large setup of nine relays, three RRBs per relay, and 20 devices. Table III shows that the proposed JEADS-G scheme requires high computing time than all other solutions in both small and large network setups. This is due to the fact that these schemes jointly generate all NOMA clusters in the network.

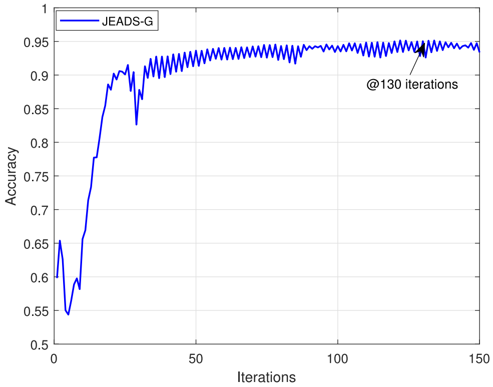

Fig. 14. Accuracy versus number of global iterations for a network of 20 IoT devices for $K = 4$ and $Z = 4$.

Since our work considers a new angle toward developing resource scheduling, frequency computation, and power allocation mechanisms to facilitate energy-efficient FL for network edge devices in relay-assisted IoT networks, we adopt the learning model in [18]. However, here, we conduct simulation results for the learning part under the proposed JEADS-G scheme. Specifically, we consider MNIST [47], a commonly used data set in machine learning. The considered data set contains a set of images for training and a set of images for testing, where each image is one of ten labels. For performing simulations and to consider different degrees of statistical data heterogeneity among the IoT devices, we divide the data sets into the devices' local $\mathcal{D}_n$ of non-i.i.d. data sets, where each local data set contains datapoints from three of the ten labels. Thus, the data sample $D_n = 200$ is selected randomly from the full data set of labels assigned to the nth IoT device. For ML local models at the IoT devices, we consider loss functions from the fully connected neural network (NN) ML classifier. For aggregating the local models at the CS, we used the FedAvg algorithm [18]. As depicted from Fig. 14 that the learning model under the proposed JEADS-G scheme is converged in 130 global iterations upon reaching 95% accuracy, which is similar to the learning model [18]. Such an observation confirms the fact that our proposed resource optimization is built on the realistic FL model.

Based on the aforementioned discussion, we summarize the observations from our presented simulation results as follows. First, although the fixed scheme performs fairly well in terms of reducing the FL time, it exhibits a poor energy consumption performance, which is impractical. Thus, it only serves as a benchmark scheme in this work. Second, optimization of both transmit power and computation frequency significantly impacts the energy consumption of the system. Therefore, optimizing only one factor, while keeping the other factor fixed, can not guarantee the optimal energy efficiency. For this reason, both of our proposed JEADS-G and SPG algorithms considerably outperform the standalone power and computation adaptation schemes. Finally, our proposed JEADS-G and SPG algorithms exhibit a reasonable performance gap from the exhaustive search-based optimal algorithm. Therefore, our proposed algorithm strikes a suitable balance between the optimality and required complexity.

VII. Conclusion

In this article, we investigated the resource allocation strategy to minimize the energy consumption for performing FL in a relay-assisted IoT system subject to FL time constraint. Specifically, we proposed joint optimization of computation frequency allocation, IoT device scheduling, and transmission power control of network edge devices. Leveraging graph theory, we proposed near-optimal joint and low-complexity iterative schemes. The presented numerical results revealed that the proposed schemes substantially reduce the energy consumption compared to the baseline schemes, at the cost of a small increase of the FL learning time. In particular, simulation results showed that the proposed JEADS-G scheme can effectively reduce the energy consumption by around 6, 4, and 2 times lower energy consumption, respectively, compared to: 1) fixed; 2) computation adaptation; and 3) power adaption. Our proposed JEADS-G and SPG schemes have a certain degradation in FL time performance as compared to the benchmark schemes. This small degradation in some numerical results, roughly in the range of 5.42%–15%, comes at the achieved low energy consumption as compared to the benchmark schemes.

References

[1] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, 1st Quart., 2020.

[2] X. Liu and N. Ansari, "Green relay assisted D2D communications with dual batteries in heterogeneous cellular networks for IoT," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1707–1715, Oct. 2017.

[3] Y. M. M. Fouad, R. H. Gohary, and H. Yanikomeroglu, "Chinese remainder theorem-based sequence design for resource block assignment in relay-assisted Internet-of-Things communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3401–3416, May 2018.

[4] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Optimizing computation offloading in satellite-UAV-served 6G IoT: A deep learning approach," *IEEE Netw.*, vol. 35, no. 4, pp. 102–108, Jul./Aug. 2021.

[5] M. Chen *et al.*, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[6] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Syst. Mach. Learn. Conf.*, Stanford, CA, USA, Feb. 2019, pp. 1–15.

[7] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[8] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, Feb. 2021.

[9] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA–MEC offloading," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6207–6211, Jun. 2019.

[10] M. Chen *et al.*, "Distributed learning in wireless networks: Recent progress and future challenges," 2021, *arXiv:2104.02151*.

[11] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 41–47, Dec. 2020.

[12] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, Apr. 2017, pp. 1273–1282.

[13] C. B. Issaid, A. Elgabli, J. Park, M. Bennis, and M. Debbah, "Communication efficient distributed learning with censored, quantized, and generalized group ADMM," 2020, *arXiv:2009.06459*.

[14] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," 2020, *arXiv:2006.02499*.

[15] M. M. Wadu, S. Samarakoon, and M. Bennis, "Joint client scheduling and resource allocation under channel uncertainty in federated learning," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5962–5674, Sep. 2021.

[16] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," 2019, *arXiv:1909.02362*.

[17] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[18] J. Yao and N. Ansari, "Secure federated learning by power control for Internet of drones," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1021–1031, Dec. 2021.

[19] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," 2019, *arXiv:1907.06040*.

[20] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[21] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[22] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided IoT by CPU frequency and wireless power control," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3438–3445, Mar. 2021.

[23] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4385–4395, Mar. 2022.

[24] H. Tran, G. Kaddoum, H. Elgala, C. Abou-Rjeily, and H. Kaushal, "Lightwave power transfer for federated learning-based wireless networks," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1472–1476, Jul. 2020.

[25] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[26] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 210–247, 1st Quart., 2022.

[27] Z. Lin, H. Liu, and Y.-J. A. Zhang, "Relay-assisted cooperative federated learning," *IEEE Trans. Wireless Commun.*, early access, Mar. 8, 2022, doi: 10.1109/TWC.2022.3155596.

[28] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom), IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, Atlanta, GA, USA, Jul. 2019, pp. 815–820.

[29] Z. Qu *et al.*, "Partial synchronization to accelerate federated learning over relay-assisted edge networks," *IEEE Trans. Mobile Comput.*, early access, May 24, 2021, doi: 10.1109/TMC.2021.3083154.

[30] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (WFL) for 6G networks—Part II: The compute-then-transmit NOMA paradigm," 2021, *arXiv:2104.12005*.

[31] M. S. Al-Abiad, M. Z. Hassan, A. Douik, and M. J. Hossain, "Low-complexity power allocation for network-coded user scheduling in Fog-RANs," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1318–1322, Apr. 2021.

[32] M. S. Al-Abiad, A. Douik, S. Sorour, and M. J. Hossain, "Throughput maximization in cloud-radio access networks using cross-layer network coding," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 696–711, Feb. 2022.

[33] M. S. Al-Abiad, A. Douik, and S. Sorour, "Rate aware network codes for cloud radio access networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1898–1910, Aug. 2019.

[34] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 375–388, Jan. 2021.

[35] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," *IEEE Netw.*, vol. 34, no. 6, pp. 272–280, Nov./Dec. 2020.

[36] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput. (HotCloud)*, Berkeley, CA, USA, 2010, p. 4.

[37] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[38] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[39] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process Syst. Signal Image Video Technol.*, vol. 13, no. 2, pp. 203–221, Aug. 1996.

[40] A. Douik, H. Dahrouj, O. Amin, B. Aloquibi, T. Y. Al-Naffouri, and M.-S. Alouini, "Mode selection and power allocation in multi-level cache-enabled networks," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1789–1793, Aug. 2020.

[41] K. Yamaguchi and S. Masuda, "A new exact algorithm for the maximum weight clique problem," in *Proc. 23rd Int. Techn. Conf. Circuits Syst. Comput. Commun. (ITCCSCC)*, Yamaguchi, Japan, 2008, pp. 1–4.

[42] P. R. J. Östergard, "A fast algorithm for the maximum clique problem," *Discrete Appl. Math*, vol. 120, pp. 197–207, Aug. 2002.

[43] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[44] J. Yao and N. Ansari, "QoS-aware power control in Internet of drones for data collection service," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6649–6656, Jul. 2019.

[45] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[46] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4385–4395, Mar. 2022.

[47] L. Yan, C. Corinna, and C. J. Burges. "The MNIST Dataset of Handwritten Digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/ (Accessed: Nov. 1998).

Mohammed S. Al-Abiad (Member, IEEE) received the B.Sc. degree in computer and communications engineering from Taiz University, Taiz, Yemen, in 2010, the M.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2017, and the Ph.D. degree in electrical engineering from The University of British Columbia, Kelowna, BC, Canada, in 2020.

He is currently a Postdoctoral Research Fellow with the School of Engineering, The University of British Columbia. His research interests include cross-layer network coding, optimization and resource allocation in wireless communication networks, machine learning, mobile edge computing, and game theory.

Md. Zoheb Hassan (Student Member, IEEE) received the Ph.D. degree from The University of British Columbia, Vancouver, BC, Canada, in 2019.

He is a Research Assistant Professor with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. Prior to that, he was a Postdoctoral Research Fellow with the École de Technologie Supérieure, Montreal, QC, Canada, and the School of Engineering, The University of British Columbia, Kelowna, BC, Canada. His research interests include resource optimization for wireless communications, signal processing for interference management and detection, and application of machine learning in physical-layer communications.

Dr. Hassan was a recipient of the Natural Science and Engineering Research Council Postdoctoral Fellowship of Canada in 2021. He serves/served as a member of the Technical Program Committee of IEEE IWCMC 2018, IEEE ICC 2019, IEEE ICC 2020, and IEEE Globecom 2021.

Md. Jahangir Hossain (Senior Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from the Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2000, the M.A.Sc. degree from the University of Victoria, Victoria, BC, Canada, in 2003, and the Ph.D. degree from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2007.

He was a Lecturer with BUET. He was a Research Fellow with McGill University, Montreal, QC, Canada, the National Institute of Scientific Research, Quebec, QC, USA, and the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA, Australia. His industrial experience includes a Senior Systems Engineer position with Redline Communications, Markham, ON, Canada, and a Research Intern position with Communication Technology Lab, Intel, Inc., Hillsboro, OR, USA. He is currently a Professor with the School of Engineering, UBC (Okanagan Campus), Kelowna, BC, Canada. His research interests include designing spectrally and power-efficient modulation schemes, applications of machine learning for communications, quality-of-service issues and resource allocation in wireless networks, and optical wireless communications.

Dr. Hossain is serving as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS. He previously served as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and as an Associate Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He regularly serves as a member of the Technical Program Committee of the IEEE International Conference on Communications and the IEEE Global Telecommunications Conference.