Min-Max Cost Optimization for Efficient Hierarchical Federated Learning in Wireless Edge Networks

Jie Feng, *Member, IEEE*, Lei Liu, *Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

Abstract—Federated learning is a distributed machine learning technology that can protect users' data privacy, so it has attracted more and more attention in the industry and academia. Nonetheless, most of the existing works focused on the cost optimization of the entire process, while the cost of individual participants cannot be considered. In this article, we explore a min-max cost-optimal problem to guarantee the convergence rate of federated learning in terms of cost in wireless edge networks. In particular, we minimize the cost of the worst-case participant subject to the delay, local CPU-cycle frequency, power allocation, local accuracy, and subcarrier assignment constraints. Considering that the formulated problem is a mixed-integer nonlinear programming problem, we decompose it into several sub-problems to derive its solutions, in which the subcarrier assignment and power allocation are obtained by utilizing the Lagrangian dual decomposition method, the CPU-cycle frequency is obtained by a heuristic algorithm, and the local accuracy is obtained by an iteration algorithm. Simulation results show the convergence of the proposed algorithm and reveal that the proposed scheme can accomplish a tradeoff between the cost and fairness by comparing the proposed scheme with the existing schemes.

Index Terms—CPU-cycle frequency, federated learning, local accuracy, min-max cost, wireless edge networks

1 Introduction

With the unprecedented rapid development of smart devices, they have become an indispensable part of people's daily life, and a massive amount of data is generated each day [1], [2], [3]. The rich data may provide support for machine learning (ML)-based application, such as training the user activity models and predicting the health events models. In the traditional centralized machine learning technology, smart devices (SDs) users upload their data directly to the cloud server for model training. However, since the data is uploaded to the centralized server, the private information of users may be leaked.

Federated learning (FL) is deemed to be an effective technique, which trains an excellent global model at the cloud server [4], [5]. FL is essentially a distributed machine learning, which allows users to train data locally without uploading data directly to the cloud server. In FL, each local user first acquires the current global model from the cloud server, then the shared model is updated with local dataset, and finally, the updated model is transmitted back to the server. By avoiding the employment of centralized training, the user's privacy is effectively protected in the FL technique.

Some works have been done to research the privacy and security of users in federated learning. The authors [6] presented a privacy-preserving and verifiable framework to warrant the confidentially of user's local gradients in federated learning. In [7], the authors explored user-level privacy leakage from the perspective of malicious server attacks, thereby analyzing the privacy leakage problem of federated learning. The authors [8] presented a cryptographic technique that can protect the uploaded information before averaging the parameters of the shared model. In [9], the authors proposed a cost-effective and privacy-enhanced technique for FL to avoid the compromise of adversaries to the shared parameters.

Except for privacy issues, the resource optimization is a challenge for federated learning [10], [11], [12]. Federated learning requires a great many computing and radio resources because users need to train the global model with local data and then sent the updated model back to the cloud server. In [13], the authors proposed a federated averaging algorithm, which can achieve efficient communication by

- *Jie Feng is with the State Key Laboratory of ISN, School of Telecommunication Engineering, Xidian University, Xi'an, Shaanxi 710071, China, and also with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China. E-mail: jiefengcl@163.com.*
- *Lei Liu and Qingqi Pei are with the State Key Laboratory of ISN, School of Telecommunication Engineering, Xidian University, Xi'an, Shannxi 710071, China. E-mail: tianjiaoliulei@163.com, qqpei@mail.xidian.edu.cn.*
- *Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA. E-mail: lik@newpaltz.edu.*

Manuscript received 29 Aug. 2021; revised 27 Oct. 2021; accepted 24 Nov. 2021. Date of publication 30 Nov. 2021; date of current version 23 May 2022.
This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807500, in part by the National Natural Science Foundation of China under Grants 62102297 and 62001357, in part by the Guangdong Basic and Applied Basic Research Foundation under Grants 2020A1515110496 and 2020A1515110079, in part by the China Postdoctoral Science Foundation under Grant 2021M692501, in part by the Fundamental Research Funds for the Central Universities under Grants XJS210105 and XJS210107, and in part by the Open Project of Shaanxi Key Laboratory of Information Communication Network and Security under Grant ICNS202005.
(Corresponding authors: Lei Liu and Qingqi Pei.)
Recommended for acceptance by S.L. Harrell and S.A. Michael.
Digital Object Identifier no. 10.1109/TPDS.2021.3131654

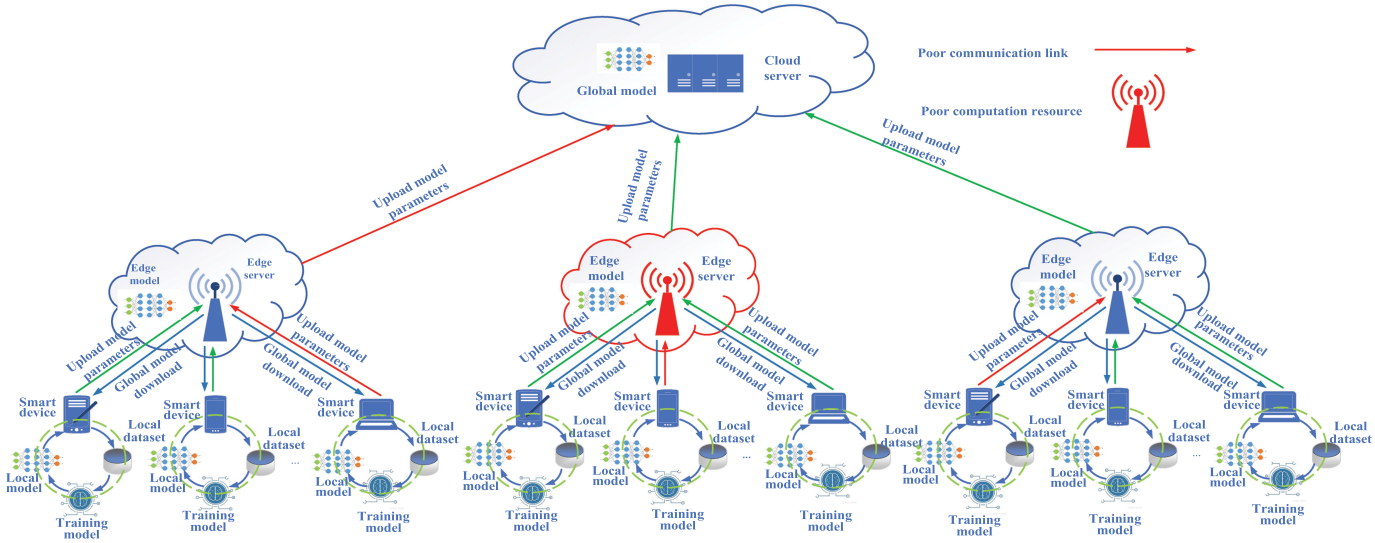


Fig. 1. The system framework.

increasing the frequency of local training or decreasing local training minibatch sizes. The authors in [14] presented a federated learning algorithm that can achieve the tradeoff between the convergence time and the energy consumption of local users. In [15], the authors investigated the application of federated learning in realistic wireless networks and formulated an optimization problem to minimize the loss function of the federated learning.

Although some works have studied the resource optimization of federated learning, there exist new challenges to address. On the one hand, the time of federated learning is determined by two parts: the local computation time and the communication time. Under the premise of the known accuracy level, the learning time is one of the important performance metrics of federated learning. Since the model parameters of all participants can be aggregated only when they are uploaded to the server at the same time, the time for each participant to update the model parameters will affect the convergence rate of federated learning. For this reason, the learning time of individual smart devices needs to be considered. On the other hand, due to the limited energy of SDs, how to achieve optimal allocation of computing resources and radio resources to minimize energy consumption is a major concern.

To tackle these problems, we formulate and investigate the min-max cost problem that specifically minimize the cost of the worst-case participant for federated learning. The contributions of this paper are summarized as follows:

- We propose a hierarchical federated learning optimization framework over wireless edge networks to study the problem of participant cost minimization in the worst case.
- We formulate an optimization problem to accomplish the optimal performance of federated learning by jointly optimizing the local accuracy, subcarrier assignment, transmit power allocation, and computational resource allocation.
- Since the proposed optimization problem is a mixed-integer nonlinear programming problem, it is difficult to obtain its solutions directly. Therefore, we decomposed the original problem into several subproblems to solve. For subcarrier assignment and power allocation, we develop an iteration algorithm to derive the optimal solution. Besides, we design an adaptive harmony search algorithm to obtain the local CPU-cycle frequency and develop an iteration algorithm to get the local accuracy of federated learning.
- The extensive numerical simulation results show the convergence performance of the proposed algorithm and display a tradeoff between the cost and fairness. Meanwhile, compared with the existing algorithms, the performance of the proposed algorithm is better in terms of the cost.

The remainder of this paper is organized as follows. In Section 2, the system model is first minutely described and then the proposed problem is formulated. In Section 3, we design the radio resources allocation algorithm. The design of computation resources allocation and local accuracy algorithms are respectively described in Section 4. In Section 5, we manifest the simulation results and discussions. Ultimately, conclusions are displaced in Section 6.

2 System Model

In this section, we first mention the description of the hierarchical federated learning over wireless edge networks, then discuss the local computation model, edge aggregation model, and cloud aggregation model. Finally, we formulate an optimization problem to minimize the cost of the worst-case user in the system.

2.1 Hierarchical Federated Learning Model in Wireless Edge Networks

As shown in Fig. 1, we consider deploying hierarchical federated learning in wireless edge networks, which is composed of a cloud server and M base stations (BS), indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$. Each BS m is equipped with an edge server and serves N_m smart devices (SM). We let $\mathcal{N}_m = \{1, 2, \ldots, N_m\}$ denote the set of smart devices covered by BS m . Then the total number of smart devices is $N =$

$\sum_{m=1}^{M} N_m$. Assume that each participating smart device $n_m \in \mathcal{N}_m$ has a local dataset $\mathcal{D}_{n_m}$, the size of which is defined as $D_{n_m} = |\mathcal{D}_{n_m}|$.

Then the dataset of an BS m is denoted by $D_m = \sum_{n_m=1}^{N_m} D_{n_m}$ and the global aggregated dataset is defined as $D = \sum_{m=1}^{M} D_m$. In the supervised learning model, the local data set $\mathcal{D}_{n_m}$ of SD n_m consists of a set of data samples. A data sample i is made up of a vector x_i and a scalar y_i, where x_i and y_i are the input and output of the supervised learning model, respectively. The data is generated by the interaction between the user and applications installed on the SD. By using these data in SD, SDs can collaborate to train machine learning models deployed in wireless edge networks.

For a given data sample i, there is a loss function $g_i(w)$, where w is the model parameter in learning problem. Typically, $x_i \in \mathbb{R}^d$ (d denotes the features of input sample vector x_i) and $y_i \in \mathbb{R}$ or $y_i \in \{-1, 1\}$. Then simple examples of the loss function include linear regression, i.e., $g_i(w) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$, support vector machines, i.e., $\max\{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$, and logistic regression, i.e., $-\log(1 + \exp(-y_i x_i^T w)), y_i \in \{-1, 1\}$. In federated learning, the iteration includes three steps: 1) each SD learns the local model based on the training data, 2) edge servers aggregate and update the local model parameters and gradients at every communication round, and 3) the edge servers synchronously uploads their parameters to the cloud server to aggregate the parameters at pre-determined frequency.

In BS m, the local loss function on the dataset $\mathcal{D}_{n_m}$ for SD n_m is

$$G_{n_m}(w) = \frac{1}{D_{n_m}} \sum_{i \in \mathcal{D}_{n_m}} g_i(w). \tag{1}$$

Accordingly, the edge loss function at edge server m is

$$G_m(w) = \frac{1}{D_m} \sum_{n_m=1}^{N_m} D_{n_m} G_{n_m}(w). \tag{2}$$

Similar to [16], we assume that the upper bound of the number of edge iterations is expressed as

$$\mathcal{I}(\epsilon, \theta) = \frac{1}{1-\theta} \mathcal{O}(\log(1/\epsilon)), \tag{3}$$

where ϵ and θ are the edge accuracy and local accuracy, respectively. From (3), we can observe that the number of $\mathcal{I}(\epsilon, \theta)$ will increase sharply when θ is big (i.e., $\theta \to 1$). Hence, it is very important to optimize the local accuracy [16]. Similar to [12], we assume that the edge accuracy ϵ is fixed. For ease of presentation, $\mathcal{O}(\log(1/\epsilon))$ is normalized to 1 [17], and then $\mathcal{I}(\theta) = \frac{1}{1-\theta}$. In the hierarchical FL model, each edge iteration is composed of computation time and uplink and downlink communication time. In the paper, we neglect the downlink time due to the high downlink bandwidth and high BS power compared to uplink communication [12], [18], [19]. However, the computation time relies on the number of local iterations, so the upper bound of the number of local iterations is given by $\log(1/\theta)$.

Similarly, the global loss function on all distributed datasets is defined as

$$G(w) = \frac{1}{D} \sum_{m=1}^{M} D_m G_m(w). \tag{4}$$

Then, the learning problem is formulated as

$$\min_{\mathbf{w} \in \mathbb{R}} G(w). \tag{5}$$

2.2 Local Computation Model

For edge server m, we let C_{n_m} denote the number of CPU cycles that is required by SD n_m to handle a sample data. We assume that all data samples in the system have the same datasize [12]. Then the total number of CPU cycles required by smart device n_m in edge server m when running one local iteration is $C_{n_m} D_{n_m}$. $\boldsymbol{f} = (f_{n_m})$ denotes the CPU-cycles frequency required by smart device n_m to execute one local iteration. By adopting dynamic voltage and frequency scaling (DVFS) technology, smart devices can adaptively change their computing speed to decrease power consumption and shorten computing time [19]. Similar to [20], the power consumption is modeled as $P_{n_m}^{loc} = k_{n_m} f_{n_m}^3$, where k_{n_m} is the effective switched capacitance of the CPU at smart device n_m [21]. Then the computing time required for each local iteration of smart device n_m is given by

$$t_{n_m}^{loc} = \frac{C_{n_m} D_{n_m}}{f_{n_m}}. \tag{6}$$

Accordingly, the total computing time required for smart device n_m to update one local model is expressed as $T_{n_m}^{cmp} = \log(1/\theta) t_{n_m}^{loc}$. The energy consumption for smart device n_m to update one local model is given by

$$E_{n_m}^{loc} = T_{n_m}^{cmp} P_{n_m}^{loc} = \log(1/\theta) k_{n_m} C_{n_m} D_{n_m} f_{n_m}^2. \tag{7}$$

2.3 Edge Computation Model

In hierarchical federated learning, we presume that the system uses the frequency reuses technique. There are K subcarriers in the system, the set of which is denoted as $\mathcal{K} = \{1, 2, \ldots, K\}$, and the bandwidth of each subcarrier is B_0. Let $\boldsymbol{P} = (P_{n_m,k})$ and $\boldsymbol{H} = (h_{n_m,k})$ denote the transmit power and the channel gain from BS m to smart device n on subcarrier k, respectively. The channel-to-interference-plus-noise of smart device n_m on subcarrier k is given by

$$\gamma_{n_m,k} = \frac{P_{n_m,k} h_{n_m,k}}{\sum_{j \in \mathcal{M}, i, j \neq n_m, m} P_{max} g_{i,j} + \sigma^2}, \tag{8}$$

where P_{max} and $g_{i,j}$ are the maximum transmit power that can be allocated on each smart device and the channel gain of smart device i in BS j, respectively. Therefore, the transmit rate of smart device n_m on subcarrier k is expressed as

$$r_{n_m,k} = B_0 \log_2(1 + \gamma_{n_m,k}) = B_0 \log_2(1 + P_{n_m} \bar{h}_{n_m,k}), \tag{9}$$

where

$$\bar{h}_{n_m,k} = \frac{h_{n_m,k}}{\sum_{j \in \mathcal{M}, i, j \neq n_m, m} P_{max} g_{i,j} + \sigma^2} \tag{10}$$

and σ is the noise power for each subcarrier.

Then the total transmit rate and the sum transmit power from smart device n to edge server m are respectively expressed as

$$r_{n_m} = \sum_{k \in \mathcal{K}} x_{n_m,k} r_{n_m,k}, \tag{11}$$

$$p_{n_m}^{com} = \sum_{k \in \mathcal{K}} x_{n_m,k} P_{n_m,k}, \tag{12}$$

where $\boldsymbol{x} = (x_{n_m,k})$ is the subcarrier variable, where $x_{n_m,k} = 1$ means that the subcarrier k is assigned to smart device n_m and $x_{n_m,k} = 0$ otherwise. Let S_{n_m} be the size of the local model parameters (i.e., w_n) updated by smart device n_m. The transmit time of smart device n_m is $T_{n_m}^{com} = S_{n_m}/r_{n_m}$. The energy consumption of smart device n_m can be written as

$$E_{n_m}^{com} = p_{n_m}^{com} T_{n_m}^{com}. \tag{13}$$

Correspondingly, the overall time and the total energy consumption of smart device n_m to complete one edge iteration are respectively given by

$$T_{n_m} = T_{n_m}^{cmp} + T_{n_m}^{com}, \tag{14}$$

$$E_{n_m} = E_{n_m}^{loc} + E_{n_m}^{com}. \tag{15}$$

The computation energy consumption and computation time of edge server m are respective denoted by E_m and T_m.

2.4 Cloud Aggregation Model

Let $\boldsymbol{r} = (r_m)$ and $\boldsymbol{p} = (p_m)$ denote the transmit rate and transmit power from edge server m to cloud server. Denote by U_m the model parameter size of edge server m. Then the transmit time and the energy consumption of cloud server m are respectively given by

$$T_m^{cloud} = \frac{U_m}{r_m}, \tag{16}$$

$$E_m^{cloud} = p_m T_m. \tag{17}$$

Compared with communications, the computing time and energy consumption of cloud server aggregation parameters can be ignored. Hence, we only pay attention to the impact of the difference in communications quality on the hierarchical federated learning.

Then the total cost of individual link (i.e., individual smart devices parameters transmission, individual edge server parameter aggregation, and individual edge server parameters transmission) in one global iteration is given by

$$\begin{aligned} Cost = {} & \lambda^e(\mathcal{I}(\theta)(E_{n_m} + E_m) + E_m^{cloud}) \\ & + \lambda^t(\mathcal{I}(\theta)(T_{n_m} + T_m) + T_m^{cloud}), \end{aligned} \tag{18}$$

where λ^e and λ^t ($\lambda^e, \lambda^t \in [0,1]$) denote the weights of the energy consumption and the latency, respectively.

2.5 Problem Formulation

In the subsection, we formulate an optimization problem of jointly optimizing local accuracy and resource allocation for FL in mobile edge networks. We propose to minimize the maximum cost of all individual links while meeting some constraints. Especially, a joint optimization problem is formulated by considering local accuracy θ, local CPU-cycle frequencies $\boldsymbol{f}$, subcarrier assignment $\boldsymbol{x}$, and transmit power allocation $\boldsymbol{P}$ as follows:

$$\begin{aligned} \min_{\theta,\boldsymbol{x},\boldsymbol{f},\boldsymbol{P}} \max_{n,m} \quad & Cost \\ \text{s.t. } (\text{C1}): \ & 0 \le f_{n_m} \le f_{n_m}^{max}, \ \forall n, m, \\ (\text{C2}): \ & \log(1/\theta)\frac{C_{n_m} D_{n_m}}{f_{n_m}} \le \tau_{max}, \quad \forall n, m, \\ (\text{C3}): \ & \sum_{k \in \mathcal{K}} x_{n_m,k} r_{n,k} \ge R_{n_m}^r, \forall n, m, \\ (\text{C4}): \ & \sum_{n_m=1}^{N_m} x_{n_m,k} \le 1, \forall k, \\ (\text{C5}): \ & x_{n_m,k} \in \{0,1\}, \forall n, k, m, \\ (\text{C6}): \ & 0 \le \sum_{k \in \mathcal{K}} x_{n_m,k} P_{n_m,k} \le P_{n_m}^{max}, \forall n, m, \\ (\text{C7}): \ & 0 \le \theta \le 1, \end{aligned} \tag{19}$$

where $R_{n_m}^r$, $P_{n_m}^{max}$, and τ_{max} are the basic transmission rate, the maximum transmission power of smart device n_m when uploading the updated local model parameters, and the maximum tolerable time, respectively. (C1) is the CPU-cycles frequency constraint of smart device n_m. (C2) indicates that the computing time for smart device n_m to update one local model cannot exceed the maximum tolerable time. (C3) ensures the basic rate requirement of smart devices. (C4) and (C5) state that one smart device can be assigned at most one subcarrier. (C6) is the transmit power constraint of smart devices. (C7) is the feasible region constraint of the local accuracy. Note that the problem (19) is not convex so that it is difficult to solve directly.

All the notations used in the entire paper are listed in Table 1.

3 Design of Radio Resources Allocation Algorithm

3.1 Problem Transformation

In the subsection, we transform the problem (19) by utilizing the fractional programs and the Lagrangian dual decomposition and design an algorithm to obtain the optimal subcarrier assignment and transmit power allocation under the given CPU-cycles frequency $\boldsymbol{f}$ and the local accuracy θ. Then the optimal subcarrier assignment and power allocation can be derived by settling the following problem:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{P}} \max_{n_m} & \frac{\mathcal{I}(\theta) S_{n_m} \left(\lambda^e \sum_{k \in \mathcal{K}} x_{n_m,k} P_{n_m,k} + \lambda^t\right)}{\sum_{k \in \mathcal{K}} x_{n_m,k} r_{n_m,k}} \\ \text{s.t. } (\text{C3}): \ & \sum_{k \in \mathcal{K}} x_{n_m,k} r_{n_m,k} \ge R_{n_m}^r, \forall n, m, \\ (\text{C4}): \ & \sum_{n_m=1}^{N_m} x_{n_m,k} \le 1, \forall k, \\ (\text{C5}): \ & x_{n_m,k} \in \{0,1\}, \forall n, k, m, \\ (\text{C6}): \ & 0 \le \sum_{k \in \mathcal{K}} x_{n_m,k} P_{n_m,k} \le P_{n_m}^{max}, \forall n, m. \end{aligned} \tag{20}$$

TABLE 1
Key Notation Definitions in System Model

Symbol	Definitions	Symbol	Definitions
$\mathcal{N}$	The total number of smart devices	$\mathcal{M}$	The number of edge servers/BS
$\mathcal{N}_m$	The set of all smart devices in BS m	D_{n_m}	The local data set of smart device n_m
θ/ϵ	The local accuracy/edge accuracy	$\mathcal{V}$	The control parameter
$P_{n_m,k}$	The transmit power from smart device n_m on subcarrier	f_{n_m}	The CPU-cycles frequency required by smart device n_m to execute one local iteration
D_{n_m}	The local data set of smart device n_m	$h_{n_m,k}$	The channel gain of smart device n_m on subcarrier k
K	The number of subcarriers	C_{n_m}	The number of CPU cycles of smart device n_m
f_{n_m}	The CPU-cycles frequency required by device n_m to execute one local iteration	$x_{n_m,k}$	The subcarrier k is assigned to smart device n_m
$P_{n_m,k}$	The transmit power from smart device n_m on subcarrier k	$E^{local}_{n_m}/E^{com}_{n_m}$	The energy consumption for smart device n_m in local computation/communications process
$T^{cmp}_{n_m}/T^{com}_{n_m}$	The total computing/transmit time for smart device n_m	σ	The noise power
S_{n_m}	The size of the local model parameters update by smart device n_m	τ_{max}	The tolerable maximum delay
$f^{max}_{n_m}$	The maximum CPU-cycles frequency of device n_m	P_{max}	The maximum transmit power of smart device n_m
$R^r_{n_m}$	The basic transmit rate of smart device n_m	ϱ, ξ, ς	The precision of algorithm
λ^e, λ^t	The weight value of energy consumption and delay	$\mathcal{I}(\theta)$	the number of edge model iterations
B_0	The bandwidth of each subcarrier	$\mathfrak{F}$	The feasible region of constraints
$g_i(\omega)/i$	The loss function/data sample	ω	The parameter of learning model
$G(\omega)$	The global loss function	U_m	The model parameter size of edge server
T^{cloud}_m/E^{cloud}_m	The transmit time and the energy consumption of cloud server	k_{n_m}	The effective switched capacitance of the CPU at smart device n_m
$t^{loc}_{n_m}$	The local computing time	r_m/p_m	The maximum transmit rate and transmit power in cloud model
$r_{n_m,k}$	The transmit rate of devices n in BS m on subcarrier k	$r_{n_m}/P^{com}_{n_m}$	The transmit rate/power of device n in BS m
DW^{max}/DW^{min}	The maximum and minimum value o distance bandwidth	μ^{HMCRm} μ^{HMCRm}	The mean of normal distribution
NI	The number of improvisations	σ^{PARm} σ^{PARs}	The standard deviation of normal distribution

Since $\mathcal{I}(\theta)\log(1/\theta)\left(\lambda^e k_{n_m} C_{n_m} D_{n_m} f^2_{n_m} + \lambda^t C_{n_m} D_{n_m}/f_{n_m}\right) + \lambda^e(\mathcal{I}(\theta)E_m + E^{cloud}_m) + \lambda^t(\mathcal{I}(\theta)T_m + T^{cloud}_m)$ in (19) is a constant, it has no effect on the solution of the subcarrier and power allocation, so it is omitted in (20). We can easily see that the objective function of (20) is non-convex, so (20) is a non-convex optimization problem. However, we can group the problem (20) into a nonlinear fractional programming problem and solve it by using fractional optimization [22]. Let $\mathfrak{F}$ denote the feasible region of $(C3)-(C6)$ in the problem (20). Then, we have

$$\begin{aligned} V^* &= \min_{\{\boldsymbol{x},\boldsymbol{P}\}\in\mathfrak{F}} \max_{n_m} \frac{\mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x_{n_m,k}P_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x_{n_m,k} r_{n_m,k}} \\ &= \max_{n_m} \frac{\mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k}P^*_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}}, \end{aligned} \tag{21}$$

where V^*, as well as $\boldsymbol{x}^*$ and $\boldsymbol{P}^*$ are the optimal value and optimal solution of problem (20), respectively.

To settle the problem (20), we give the following theorem.

Theorem 1. *The optimal solution $\{\boldsymbol{x}^*, \boldsymbol{P}^*\}$ of problem (20) can be achieved if and only if*

$$\begin{aligned} &\min_{\{\boldsymbol{x},\boldsymbol{P}\}\in\mathfrak{F}} \max_{n_m} \left(\mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x_{n_m,k}P_{n_m,k} + \lambda^t\right) - V^* \sum_{k\in\mathcal{K}} x_{n_m,k} r_{n_m,k}\right) \\ &= \max_{n_m} \left(\mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k}P^*_{n_m,k} + \lambda^t\right) - V^* \sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}\right) = 0. \end{aligned} \tag{22}$$

Proof. Similar to [23], we can prove it from sufficiency and necessity. First, the proof of the sufficiency is given as following.

Assume that $\boldsymbol{x}^*, \boldsymbol{P}^*$ is the optimal solution of the problem (20), and $\boldsymbol{x}', \boldsymbol{P}'$ is its feasible solution, then we have

$$\begin{aligned} &\max_{n_m} \mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k}P^*_{n,k} + \lambda^t\right) - V^* \sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n,k} = 0 \\ &\max_{n_m} \mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x'_{n_m,k}P'_{n_m,k} + \lambda^t\right) - V^* \sum_{k\in\mathcal{K}} x'_{n_m,k} r'_{n,k} \geq 0. \end{aligned} \tag{23}$$

Then, from (43), we get

$$\max_{n_m} \frac{\mathcal{I}(\theta)S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k}P^*_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}} = V*, \tag{24}$$

$$\max_{n_m} \frac{\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x'_{n_m,k} P'_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x'_{n_m,k} r'_{n_m,k}} \geq V^*, \tag{25}$$

Therefore, we can obtain that $\boldsymbol{x}', \boldsymbol{P}'$ is also the optimal solution of (14). Then the proof of sufficiency is completed.

Next, let's prove its necessity. For any feasible $\boldsymbol{x}', \boldsymbol{P}'$ in the domain of (14), we have

$$\begin{aligned}
&\max_{n_m} \frac{\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k} P^*_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}} = V^* = 0,\\
&\max_{n_m} \frac{\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x'_{n_m,k} P'_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x'_{n_m,k} r'_{n_m,k}} \geq V^* = 0,
\end{aligned} \tag{26}$$

Hence, since the following equation holds

$$\max_{n}\left(\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k} P^*_{n_m,k} + \lambda^t\right) - V^* \sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}\right) = 0, \tag{27}$$

we have that $\boldsymbol{x}^*, \boldsymbol{P}^*$ is the optimal solution of (14). Then the proof of necessity is completed. □

Observing the above, the problem (20) can be solved by its equivalent problem (22). Nevertheless, since the value of V^* is unknown in advance, we can solve problem (22) by replacing V^* with an update parameter V [23], thereby obtaining the optimal solution of problem (20). Algorithm 1 describes the details of the whole procedure, where the optimization problem in line 2 is given by

$$\begin{aligned}
&\min_{\{\boldsymbol{x},\boldsymbol{P}\}\in\mathfrak{F}} \max_{n_m}\left(\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x_{n_m,k} P_{n_m,k} + \lambda^t\right) - V \sum_{k\in\mathcal{K}} x_{n_m,k} r_{n_m,k}\right)\\
&\text{s.t. } (\text{C3}) - (\text{C6}).
\end{aligned} \tag{28}$$

Algorithm 1. Iterative Radio Resource Allocation Algorithm to Solve (20)

Input: the maximum number of iterations o_{max}, the precision ς, the initial value $V^0 \leftarrow 0$, and $o \leftarrow 0$.
Output: V.

1: **while** $o \leqslant o_{max}$ **do**
2: Solve problem (28) to obtain $\boldsymbol{x}^o$ and $\boldsymbol{P}^o$ under the given V^o.
3: **if** $\left|\max_{n_m}\left(\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^o_{n_m,k} P^o_{n_m,k} + \lambda^t\right) - V^o \sum_{k\in\mathcal{K}} x^o_{n_m,k} r^o_{n_m,k}\right)\right| < \varsigma$ **then**
4: $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^o, \boldsymbol{P}^* \leftarrow \boldsymbol{P}^o$.
5: $V^* \leftarrow \max_{n_m} \frac{\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^*_{n_m,k} P^*_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x^*_{n_m,k} r^*_{n_m,k}}$.
6: break
7: **else**
8: $V^{o+1} \leftarrow \max_{n_m} \frac{\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x^o_{n_m,k} P^o_{n_m,k} + \lambda^t\right)}{\sum_{k\in\mathcal{K}} x^o_{n_m,k} r^o_{n_m,k}}$.
9: **end if**
10: $o \leftarrow o + 1$.
11: **end while**

To solve (28), we introduce a new variable ζ_1 and let $\max_{n_m} Cost = \zeta_1$. Then the original problem (28) is transformed as

$$\begin{aligned}
&\min_{\boldsymbol{x},\boldsymbol{P}} \zeta_1\\
&\text{s.t. } (\text{C3}): \sum_{k\in\mathcal{K}} x_{n_m,k} r_{n_m,k} \geq R^r_{n_m}, \forall n, m,\\
&\quad (\text{C4}): \sum_{n_m=1}^{N_m} x_{n_m,k} \leq 1, \forall k,\\
&\quad (\text{C5}): x_{n_m,k} \in \{0,1\}, \forall n, k, m,\\
&\quad (\text{C6}): 0 \leq \sum_{k\in\mathcal{K}} x_{n_m,k} P_{n_m,k} \leq P^{max}_{n_m}, \forall n, m,\\
&\quad (\text{C8}): \mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} x_{n_m,k} P_{n_m,k} + \lambda^t\right)\\
&\qquad - V \sum_{k\in\mathcal{K}} x_{n_m,k} r_{n_m,k} \leq \zeta_1, \forall n, m.
\end{aligned} \tag{29}$$

From (22), we can observe that $\zeta_1 \geq 0$ when $0 \leq V \leq V^*$ for all feasible $\boldsymbol{x}$ and $\boldsymbol{P}$. Observe (29), we can see that it is a non-convex optimization problem. For this reason, we introduce a new variable $\boldsymbol{\alpha}_{n_m,k} = x_{n_m,k} P_{n_m,k}$ and relax the value of $x_{n_m,k}$ to the interval [0,1]. Particularly, $x_{n_m,k} r_{n_m,k} = 0$ when $x_{n_m,k} = 0$. The problem (29) is rearranged as

$$\begin{aligned}
&\min_{\boldsymbol{x},\boldsymbol{P}} \zeta_1\\
&\text{s.t. } (\text{C3}): \sum_{k\in\mathcal{K}} x_{n_m,k} B_0 \log_2\left(1 + \frac{\boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}}\right) \geq R^r_n, \forall n, m,\\
&(\text{C4}): \sum_{n_m=1}^{N} x_{n_m,k} \leq 1, \forall k,\\
&(\text{C5}): 0 \leq x_{n_m,k} \leq 1, \forall n, k, m,\\
&(\text{C6}): 0 \leq \sum_{k\in\mathcal{K}} \boldsymbol{\alpha}_{n_m,k} P_{n_m,k} \leq P^{max}_{n_m}, \forall n, m,\\
&(\text{C8}): \mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} \boldsymbol{\alpha}_{n_m,k} + \lambda^t\right)\\
&\qquad - V \sum_{k\in\mathcal{K}} x_{n_m,k} \log_2\left(1 + \frac{\boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}}\right) \leq \zeta_1, \forall n, m.
\end{aligned} \tag{30}$$

For optimization problem (30), we have the following theorem to describe its attribute.

Theorem 2. *Problem (30) is a convex problem in $\boldsymbol{x}, \boldsymbol{\alpha}$, and ζ_1.*

Proof. According to the definition of the perspective function, we conclude that if a function $f(x)$ is convex, then its perspective function $g(x,t) = tf(x/t)$ is also convex [24]. According to the conclusion, $x_{n_m,k} \log_2\left(1 + \boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k} / x_{n_m,k}\right)$ is concave. That is because its perspective function $\log_2\left(\boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k}\right)$ is concave. Since the sum of concave functions is still concave, $\sum_{k\in\mathcal{K}} x_{n_m,k} \log_2(1 + \boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k} / x_{n_m,k})$ preserves concavity. Based on the convex optimization theorem [24], the superlevel set of a concave function is convex, so the constraint (C3) is convex. Since $\mathcal{I}(\theta) S_{n_m}\left(\lambda^e \sum_{k\in\mathcal{K}} \boldsymbol{\alpha}_{n_m,k} + \lambda^t\right) - V \sum_{k\in\mathcal{K}} x_{n_m,k} \log_2\left(1 + \boldsymbol{\alpha}_{n_m,k} \bar{h}_{n_m,k} / x_{n_m,k}\right)$ is the sum of non-negative convex function, (C8) is convex. In addition, $(\text{C4}) - (\text{C6})$ are all linear constraints.

Consequently, we can prove that problem (20) is convex optimization problem. □

Noting that (30) satisfies the Slater's condition with zero duality gap, the optimization solutions can be derived via the Lagrangian dual decomposition. The Lagrangian function of (30) is written as (31), shown at the bottom of this page, which $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{n_m}) \succeq 0$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k) \succeq 0$, $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{n_m}) \succeq 0$, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{n_m}) \succeq 0$ are the dual variables associated with constraints (C3), (C4), (C6), and (C8) in the order given.

Accordingly, the dual function is expressed as

$$D(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) = \min_{\boldsymbol{X}, \boldsymbol{\alpha}, \zeta_1 \in (\mathrm{C5})} L(\boldsymbol{X}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}). \tag{32}$$

Observe (32), for a given set of dual variables $\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}$, we can minimize $L(\boldsymbol{X}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu})$ to obtain the resource allocation strategy and ζ_1 by solving the following two subproblems.

3.2 Resource Allocation Strategy

In this subsection, we derive the optimal solutions of $\boldsymbol{x}$ and P by solving the following problem:

$$\begin{aligned}
&\min_{\boldsymbol{x}, \boldsymbol{P}} \Bigg(\sum_{n \in \mathcal{N}} \beta_{n_m} \Bigg(- \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) \Bigg) \\
&+ \sum_{k \in \mathcal{K}} \gamma_k \sum_{n=1}^{N} x_{n_m,k} + \sum_{n_m \in \mathcal{N}_m} \nu_{n_m} \sum_{k \in \mathcal{K}} \alpha_{n_m,k} \\
&+ \sum_{n_m \in \mathcal{N}_m} \mu_{n_m} \Bigg(\mathcal{I}(\theta) S_{n_m} \Bigg(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k} + \lambda^t \Bigg) \\
&- V \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) \Bigg) \Bigg) \\
&\text{s.t. } (\mathrm{C5}): \ 0 \le x_{n_m,k} \le 1, \forall n, k, m,
\end{aligned} \tag{33}$$

◆ Optimal power allocation: For a given variable $\boldsymbol{x}$, since the Karush-Kuhn-Tucker (KKT) conditions exist, we can take the partial derivative of $L(\boldsymbol{x}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu})$ in (33) with respect to $P_{n,k}$ and make it equal to 0. Then the optimal power allocation is given by

$$\begin{aligned}
P^*_{n_m,k} &= \frac{\alpha^*_{n_m,k}}{x_{n_m,k}} \\
&= \left[\frac{B_0(\beta_{n_m} + V\mu_{n_m})}{(\nu_{n_m} + \mu_{n_m} \mathcal{I}(\theta) S_{n_m} \lambda^e) \ln 2} - \frac{1}{\bar{h}_{n_m,k}} \right]^+, \forall n, k, m,
\end{aligned} \tag{34}$$

where $y^+ \triangleq \max\{0, y\}$.

◆ Optimal subcarrier assignment: After the power allocation strategy is known, we obtain the optimal subcarrier assignment policy by calculating the partial derivative of $L(\boldsymbol{x}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu})$ with respect to $x_{n_m,k}$ as follows:

$$\begin{aligned}
&\partial L(\boldsymbol{x}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) / \partial x_{n_m,k} \\
&= - \Bigg((B_0(\beta_{n_m} + \mu_{n_m} V) \Bigg(\log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) \\
&- \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{(x_{n_m,k} + \alpha_{n_m,k} \bar{h}_{n_m,k}) \ln 2} \Bigg) - \gamma_k \Bigg).
\end{aligned} \tag{35}$$

Let $\partial L(\boldsymbol{x}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) / \partial x_{n_m,k} = 0$ and substitute (34) into (35), we further get

$$x^*_{n_m,k} = \begin{cases} 0, & Y_{n_m,k} > \gamma_k, \\ 1, & Y_{n_m,k} < \gamma_k, \end{cases} \tag{36}$$

where

$$\begin{aligned}
Y_{n,k} = &\big(B_0(\beta_{n_m} + \mu_{n_m} V) \\
&\Bigg(\Bigg[\log_2 \Bigg(\bar{h}_{n_m,k} \frac{B_0(\beta_{n_m} + V\mu_{n_m})}{(\nu_{n_m} + \mu_{n_m} \mathcal{I}(\theta) S_{n_m} \lambda^e) \ln 2} \Bigg) \Bigg]^+ \\
&- \frac{1}{\ln 2} \Bigg[1 - \frac{(\nu_{n_m} + \mu_{n_m} \mathcal{I}(\theta) S_{n_m} \lambda^e) \ln 2}{\bar{h}_{n_m,k} B_0(\beta_{n_m} + V\mu_{n_m})} \Bigg]^+ \Bigg) \Bigg).
\end{aligned} \tag{37}$$

Since the channel conditions $\bar{h}_{n_m,k}$ is mutually independent random, $Y_{n,k}$ among different user n cannot be the same for the specified subcarrier k. Therefore, we re-given the subcarrier assignment policy, i.e.,

$$x^*_{n_m,k} = \begin{cases} 1, & \text{if } n_m = \arg\min_{n_m \in \mathcal{N}_m} Y_{n_m,k}, \\ 0, & \text{if } n_m \neq \arg\min_{n_m \in \mathcal{N}_m} Y_{n_m,k}. \end{cases} \tag{38}$$

3.3 Variable ζ_1 Selection

From (30), we can observe that only under a given ζ_1 can we get the solutions of $\boldsymbol{x}$ and $\boldsymbol{P}$. Therefore, we need to further discuss the solution of ζ_1 in the subsection as follows:

$$\begin{aligned}
&\min_{\zeta_1} \Bigg(1 - \sum_{n_m=1}^{N_m} \mu_{n_m} \Bigg) \zeta_1 \\
&\text{s.t. } (\mathrm{C8}): \ \mathcal{I}(\theta) S_{n_m} \Bigg(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k} + \lambda^t \Bigg) \\
&- V \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) \le \zeta_1 \le 0, \forall n_m.
\end{aligned} \tag{39}$$

$$\begin{aligned}
L(\boldsymbol{X}, \boldsymbol{\alpha}, \zeta_1, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) = \zeta_1 &+ \sum_{n_m \in \mathcal{N}_m} \beta_{n_m} \Bigg(R^r_{n_m} - \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) \Bigg) + \sum_{k \in \mathcal{K}} \gamma_k \Bigg(\sum_{n_m=1}^{N_m} x_{n_m,k} - 1 \Bigg) \\
&+ \sum_{n_m \in \mathcal{N}_m} \nu_{n_m} \Bigg(\sum_{k \in \mathcal{K}} \alpha_{n_m,k} - P^{max}_{n_m} \Bigg) + \sum_{n_m \in \mathcal{N}_m} \mu_{n_m} \Bigg(\mathcal{I}(\theta) S_{n_m} \Bigg(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k} + \lambda^t \Bigg) \\
&- V \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \Bigg(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \Bigg) - \zeta_1 \Bigg)
\end{aligned} \tag{31}$$

From (39), we can observe that the optimal solution of ζ_1, i.e., ζ_1^*, can be given by

$$\zeta_1^* = \begin{cases} 0, & \text{if } \sum_{n_m=1}^{N_m} \mu_{n_m} < 1, \\ H_{n_m,k}^*, & \text{if } \sum_{n_m=1}^{N_m} \mu_{n_m} \geq 1, \end{cases} \tag{40}$$

where

$$H_{n_m,k}^* = \max_{n_m} \left(\mathcal{I}(\theta) S_{n_m} \left(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k}^* + \lambda^t \right) - V \sum_{k \in \mathcal{K}} x_{n_m,k}^* B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k}^* \bar{h}_{n_m,k}}{x_{n_m,k}^*} \right) \right). \tag{41}$$

From (34) and (38), we can find that only by knowing the dual variable $\boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{\mu}$ in advance can we obtain the optimal resource allocation strategy. To obtain the dual variable $\boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{\mu}$, we give the dual problem of (30) as follows:

$$\begin{aligned} &\max D(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) \\ &\text{s.t. } \boldsymbol{\beta} \succeq 0, \boldsymbol{\gamma} \succeq 0, \boldsymbol{\nu} \succeq 0, \boldsymbol{\mu} \succeq 0. \end{aligned} \tag{42}$$

Observe that, from (31) and (32), (42) is always convex, because the objective function and the constraint of (42) is linear. Consequently, we can address (42) by employing the subgradient projection method. We first give the following theorem to obtain a subgradient of $D(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu})$.

Theorem 3. *For the dual problem (42) defined by the original problem (30), the update of the dual variable is given by*

$$\beta_{n_m}(l+1) = [\beta_{n_m}(l) + o(l) \Delta \beta_{n_m}(l)]^+, \tag{43}$$

$$\nu_{n_m}(l+1) = [\nu_{n_m}(l) + q(l) \Delta \nu_{n_m}(l)]^+, \tag{44}$$

$$\mu_{n_m}(l+1) = [\mu_{n_m}(l) + g(l) \Delta \mu_{n_m}(l)]^+, \tag{45}$$

where

$$\Delta \beta_{n_m} = R_{n_m}^r - \sum_{k \in \mathcal{K}} x_{n_m,k}^* B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k}^* \bar{h}_{n_m,k}}{x_{n_m,k}^*} \right), \tag{46}$$

$$\Delta \nu_{n_m} = \sum_{k \in \mathcal{K}} \alpha_{n_m,k}^* - P_{n_m}^{max}, \tag{47}$$

$$\Delta \mu_{n_m} = \mathcal{I}(\theta) S_{n_m} \left(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k}^* + \lambda^t \right) - V \sum_{k \in \mathcal{K}} x_{n_m,k}^* B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k}^* \bar{h}_{n_m,k}}{x_{n_m,k}^*} \right) - \zeta_1^*, \tag{48}$$

and l is the index of iteration and $o(l)$, $q(l)$, and $g(l)$ are very small positive step size. We set $o(l) = q(l) = 0.1/l$, $g(l) = 0.01/l$ [24], [25]. $\boldsymbol{\gamma}$ is set a constant in this paper.

Proof. According to the definition of $D(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu})$ in (21), we can get

$$\begin{aligned} D(\boldsymbol{\beta}', \boldsymbol{\gamma}, \boldsymbol{\nu}', \boldsymbol{\mu}') \geq \zeta_1 &+ \sum_{n_m \in \mathcal{N}_m} \beta'_{n_m} \left(R_{n_m}^r - \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \right) \right) \\ &+ \sum_{k \in \mathcal{K}} \gamma_k \left(\sum_{n_m=1}^{N_m} x_{n_m,k} - 1 \right) + \sum_{n_m \in \mathcal{N}_m} \nu'_{n_m} \left(\sum_{k \in \mathcal{K}} \alpha_{n_m,k} - P_{n_m}^{max} \right) \\ &+ \sum_{n_m \in \mathcal{N}_m} \mu'_{n_m} \left(\mathcal{I}(\theta) S_{n_m} \left(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k} + \lambda^t \right) \right. \\ &\left. - V \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \right) - \zeta_1 \right), \end{aligned} \tag{49}$$

where the reason for the inequality is the fact that $\boldsymbol{x}^*$ and $\boldsymbol{P}^*$ are the optimal solutions corresponding to $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\nu}$, and $\boldsymbol{\mu}$.

Then, we rearrange (49) as

$$\begin{aligned} D(\boldsymbol{\beta}', \boldsymbol{\gamma}, \boldsymbol{\nu}', \boldsymbol{\mu}') \geq D(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\mu}) &+ \sum_{n_m \in \mathcal{N}_m} (\beta'_{n_m} - \beta_{n_m}) \left(R_n^r - \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 \left(1 + \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \right) \right) \\ &+ \sum_{n_m \in \mathcal{N}_m} (\nu'_{n_m} - \nu_{n_m}) \left(\sum_{k \in \mathcal{K}} \alpha_{n_m,k} - P_{n_m}^{max} \right) \\ &+ \sum_{n_m \in \mathcal{N}_m} (\mu'_{n_m} - \mu_{n_m}) \left(\mathcal{I}(\theta) S_{n_m} \left(\lambda^e \sum_{k \in \mathcal{K}} \alpha_{n_m,k} + \lambda^t \right) \right. \\ &\left. - V \sum_{k \in \mathcal{K}} x_{n_m,k} B_0 \log_2 (1+) \frac{\alpha_{n_m,k} \bar{h}_{n_m,k}}{x_{n_m,k}} \right) - \zeta_1. \end{aligned} \tag{50}$$

Note that i is a subgradient of a convex function $g(\cdot)$ if $g(x) \geq g(y) + i^T(x-y)$ hold for all x and y in the domain. Hence, Theorem 3 holds. □

To solve (30), we develop an algorithm to obtain the optimal solution, as shown in Algorithm 2.

Algorithm 2. Power Allocation and Subcarrier Assignment Algorithm

Input: Dual variable $\boldsymbol{\beta}(0)$, $\boldsymbol{\nu}(0)$, $\boldsymbol{\mu}(0)$, the maximum number of iteration l_{max}, $l \leftarrow 0$, and the precision ϱ.
Output: Subcarrier assignment $\boldsymbol{x}$ and transmit power $\boldsymbol{P}$.

```
while l ⩽ l_max do
    Obtain power allocation P_{n_m,k} according to (34).
    Assign subcarrier x_{n_m,k} according to (38).
    Obtain β(l+1), ν(l+1), and μ(l+1) based on (43), (44), and (45), respectively.
    Compute i(l) ← β(l+1) − β(l), j(l) ← ν(l+1) − ν(l), and z(l) ← μ(l+1) − μ(l), respectively.
    if ||i(l)||_2 < ϱ, ||j(l)||_2 < ϱ and ||z(l)||_2 < ϱ then
        x*_{n_m,k} ← x_{n_m,k}(l), P*_{n_m,k} ← P_{n_m,k}(l).
        break.
    else
        l ← l + 1.
    end if
end while
```

4 Design of Computation Resource Allocation and Local Accuracy Algorithm

4.1 Computation Resource Allocation Algorithm

In this subsection, we develop a heuristic search algorithm to obtain the optimal computing resource allocation policy under given radio resource allocation strategy $\{\boldsymbol{x}, \boldsymbol{P}\}$ and the local accuracy θ. Then the optimization problem of computing resource allocation is given by

$$\begin{aligned}&\min_{\boldsymbol{f}} \max_{n_m} \mathcal{I}(\theta)\log(1/\theta)\left(\lambda^e k_{n_m} C_{n_m} D_{n_m} f_{n_m}^2 + \lambda^t \frac{C_{n_m} D_{n_m}}{f_{n_m}}\right) + B_{n_m}\\ &\text{s.t. (C1)}: \ 0 \le f_{n_m} \le f_{n_m}^{max}, \ \forall n, m,\\ &\quad\text{(C2)}: \ \log(1/\theta)\frac{C_{n_m} D_{n_m}}{f_{n_m}} \le \tau_{max}, \ \forall n, m,\end{aligned} \tag{51}$$

where

$$\begin{aligned}B_{n_m} = &\mathcal{I}(\theta) S_{n_m}(\lambda_n^e P_n + \lambda_n^t)/r_n + \lambda^e(\mathcal{I}(\theta)E_m + E_m^{cloud})\\ &+ \lambda^t(\mathcal{I}(\theta)T_m + T_m^{cloud}),\end{aligned} \tag{52}$$

and it is a constant.

Since the problem (51) is a continuous optimization problem, we propose an adaptive harmony search (AHS) algorithm [26] to obtain the optimal solution of the problem. The problem (51) includes the following constraints.

- Boundary constraint: $0 \le f_{n_m} \le f_{n_m}^{max}, \ \forall n_m$.
- Performance constraint: $\log\left(\frac{1}{\theta}\right)\frac{C_{n_m} D_{n_m}}{f_{n_m}} \le \tau_{max}, \forall n_m$.

By employing a penalty function, the problem is rearranged as

$$\begin{aligned}&\min_{\boldsymbol{f}} \Phi(\boldsymbol{f})\\ &\text{s.t. } 0 \le f_{n_m} \le f_{n_m}^{max}, \ \forall n,\end{aligned} \tag{53}$$

where

$$\begin{aligned}\Phi(\boldsymbol{f}) = &B_{n_m} + \frac{1}{\vartheta}\sum_{n_m=1}^{N_m} \max\left(0, \log\left(\frac{1}{\theta}\right)\frac{C_{n_m} D_{n_m}}{f_{n_m}} - \tau_{max}\right)\\ &+ \max_{n_m} \mathcal{I}(\theta)\log\left(\frac{1}{\theta}\right)\left(\lambda^e k_{n_m} C_{n_m} D_{n_m} f_{n_m}^2 + \lambda^t \frac{C_{n_m} D_{n_m}}{f_{n_m}}\right),\end{aligned} \tag{54}$$

and $1/\vartheta \longrightarrow 0$ [27]. Observe that, from the problem (53), it and the problem (51) have the same optimal solution.

The proposed AHS algorithm mainly includes the following basic parameters.

- The number of improvisations (NI): it presents the maximum iteration number of the algorithm;
- Pitch adjusting rate (PAR): the parameter is the pitch adjustment rate selected from the harmony vector;
- Distance bandwidth (DW): the parameter is the adjustment range of the continuous variable;
- Harmony memory consideration rate (HMCR): it indicates the probability of selecting one harmony vector (HV) from the harmony memory;
- Harmony memory size (HMS): the harmony memory is regarded as a $HMS \times N$ matrix, in which each row represents the solution of the problem (53).

We take several dynamic parameters to balance the exploration and exploitation of search process. Particularly, HMCR and PAR are adaptively generated, where the value of HMCR (PAR) is normal distributed with mean μ^{HMCRm} (μ^{PARm}) and standard deviation σ^{HMCRs} (σ^{PARs}). Besides, the value of DW is set as a dynamically decreasing function as the number of iteration increase as follows:

$$DW(I) = \begin{cases} DW^{max} - 2I\frac{DW^{max} - DW^{min}}{NI}, & \text{if } I < NI/2,\\ DW^{min}, & \text{if } I \ge NI/2,\end{cases} \tag{55}$$

where DW^{max} and DW^{min} indicate the maximum and minimum value of distance bandwidth, respectively.

Let $F^d = (f^d)$ denote the dth row of the HM, where $d \in \{1, 2, \ldots, HMS\}$. FU and FL are respectively the upper and lower bounds of optimal variable f. i.e., $FU = f_{n_m}^{max}, \ \forall n$ and $FL = 0$. The detail of the proposed algorithm is shown in Algorithm 3. In algorithm, r^1, r^2, and r^3 are the uniform random number generated in the interval [0,1], respectively.

Algorithm 3. Adaptive Harmony Search (AHS) Algorithm

Input: Parameters DW^{max}, DW^{min}, NI, W, μ^{HMCR}, μ^{PARm}, σ^{HMCRs}, σ^{PARs}, $\Phi(\boldsymbol{f})$, the initial value of HM, the maximum iteration number I and iteration index $w \leftarrow 1$.
Output: the optimal computing resource allocation $\boldsymbol{f}$.

for I from 1 to NI **do**
  Generate HMCR and PAR based on μ^{HMCR}, μ^{PARm}, σ^{HMCRs}, and σ^{PARs}.
  Compute the distance bandwidth $DW(I)$ according to (55).
  for j from 1 to N **do**
    if $r^1 < HMCR$ **then**
      $F^{new}(j) \leftarrow F^i(j) \pm r^2 \times DW(I)$, where $i \in \{1, 2, \ldots, HMS\}$.
      if $r^3 < PAR$ **then**
        $F^{new}(j) \leftarrow F^B(j)$, where F^B is the best HV.
      end if
    else
      $F^{new}(j) \leftarrow FL(j) + r^2 \times (FU(j) - FL(j))$.
    end if
  end for
  if $\Phi(F^{new}) < \Phi(F^D)$ **then**
    Update the HM as $F^D \leftarrow F^{new}$ and preserve the HMCR and PAR values, where F^D is worst harmony vector.
  end if
  if $w = W$ **then**
    Recompute the values of μ^{HMCR} and μ^{PARm} based on the saved HMCR and PAR values. Reset $w \leftarrow 1$.
  else
    $w \leftarrow w + 1$.
  end if
end for

4.2 Local Accuracy Optimization Algorithm

After we obtain $\boldsymbol{x}$, $\boldsymbol{P}$, and $\boldsymbol{f}$, the local accuracy optimization problem becomes

$$\min_{\theta} \max_{n_m} \frac{\log\left(\frac{1}{\theta}\right) F_{n_m} + \tilde{E}_{n_m}}{1-\theta}$$
$$\text{s.t. } (\text{C2}): \ \log\left(\frac{1}{\theta}\right) \frac{C_{n_m} D_{n_m}}{f_{n_m}} \leq \tau_{max}, \ \forall n,$$
$$(\text{C7}): \ 0 \leq \theta \leq 1, \tag{56}$$

where

$$F_{n_m} = \lambda^e k_{n_m} C_{n_m} D_{n_m} f_{n_m}^2 + \lambda^t \frac{C_{n_m} D_{n_m}}{f_{n_m}}, \tag{57}$$

and $\tilde{E}_{n_m} = S_{n_m}(\lambda^e P_{n_m} + \lambda^t)/r_{n_m}$ are constants. $Z = \lambda^e E_m^{cloud} + \lambda^t T_m^{cloud}$ is a constant that does not affect the solution of the above problem, so we ignore it in (56).

To obtain the local accuracy θ, we have

$$\eta = \min_{\theta} \max_{n_m} \frac{\log\left(\frac{1}{\theta}\right) F_{n_m} + \tilde{E}_{n_m}}{1-\theta}. \tag{58}$$

Then, the optimization problem (56) is recast as

$$\min_{\theta} \max_{n_m} \log(1/\theta) F_{n_m} + \tilde{E}_{n_m} - \eta(1-\theta)$$
$$\text{s.t. } (\text{C2}), (\text{C7}). \tag{59}$$

To solve the problem (59), we propose an iterative algorithm to obtain the optimal local accuracy. The detail of the proposed algorithm is shown in Algorithm 4.

Algorithm 4. Iteration Local Accuracy Optimization Algorithm

Input: The maximum iteration number V_{max}, the precision ξ, the minimum value $\eta \leftarrow 0$, and $v \leftarrow 0$.
Output: η^*, θ^*.

```
while v ≤ V_max do
    Solve (60) for a given η^v to obtain the local accuracy.
    if |max_{n_m}(Ẽ_{n_m} − F_{n_m} log(θ^v) − η^v(1 − θ^v))| ≤ ξ then
        θ* ← θ^v.
        break
    else
        η^{v+1} ← η^v − (max_{n_m}(Ẽ_{n_m} − F_{n_m} log(θ^v) − η^v(1 − θ^v))) / (θ^v − 1).
        v ← v + 1.
    end if
end while
```

For a given η, we introduce a new variable ζ_2 to rearranged the originally optimization problem (59) as follows:

$$\min_{\theta} \zeta_2$$
$$\text{s.t. } (\text{C2}), (\text{C7}),$$
$$(\text{C8}): \log\left(\frac{1}{\theta}\right) F_{n_m} + \tilde{E}_{n_m} - \eta(1-\theta) \leq \zeta_2, \ \forall n, m. \tag{60}$$

From (60), we observe that it is a convex optimization problem and the solution can be easily obtain.

5 Simulation Results

In this section, a larger number of simulation results are displayed to evaluate the performance of the proposed scheme.

TABLE 2
Summary of The Simulation Parameters

Parameters	Values
Total bandwidth, B	1 MHz
Maximum computing capacity, $f_{n_m}^{max}$	2 GHz
Processing density, C_{n_m}	273.5 cycle/bit
Noise power density, N_0	-174 dBm/Hz
Training size, D_{n_m}	40 KB
Tolerable maximum delay, τ_{max}	0.5 s
Required transmit rate, $R_{n_m}^r$	$2 * 10^4$ bit/s
Maximum transmit power, P_{max}	2 W
$k_{ser,m}$	10^{-27}
Harmony memory size , HMS	5
$\mu^{HMCRm}(\mu^{PARm})$	0.95(0.3)
$\sigma^{HMCRm}(\sigma^{PARm})$	0.01(0.05)
$DW^{max}(DW^{min})$	$(FU - FL)/20(0.0005)$
The number improvisations NI	6000

5.1 Simulation Parameters

For our simulations, we consider a cellular network that consists of 8 smart devices and 3 BSs scattering over a 1×1 km^2. The other simulation parameters are listed in Table 2. The channel is modelled as a frequency-selective channel that is composed of twelve independent Rayleigh multipaths. We adopt the Clarkes flat fading model as the component of the twelve multipaths. The relative power of the component of the twelve multipaths are set $[-2.5, -4, -3.2, 0, -5.2, -7.5, -5.5, -2.8, -10, -8.7, -12, -11]$ dB. We consider the two baseline schemes to verify the performance of the proposed schemes. The first scheme is that focuses on the cost of the entire system, and do not consider the cost of individual smart devices. This scheme is called the network cost optimization scheme (NCS). The second scheme is that focuses on minimizing the training time of the system subject to a total power constraint, which is called the training time minimization scheme (TTS).

5.2 Convergence Performance

Fig. 2 shows the convergence of the outer loop of Algorithm 1, where V is the optimal value of problem (20). We can

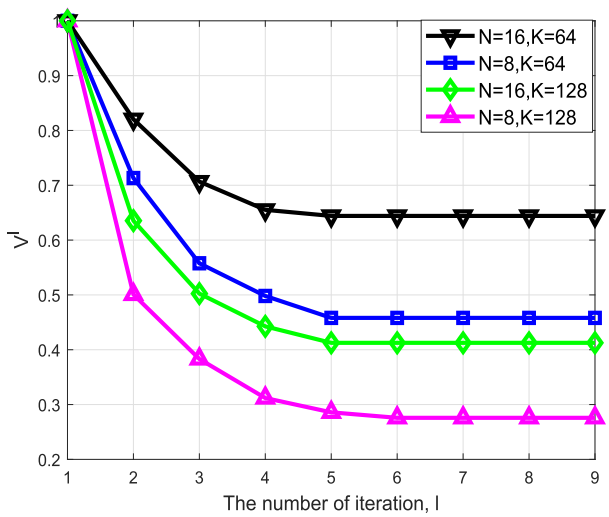


Fig. 2. Convergence of Algorithm 1.

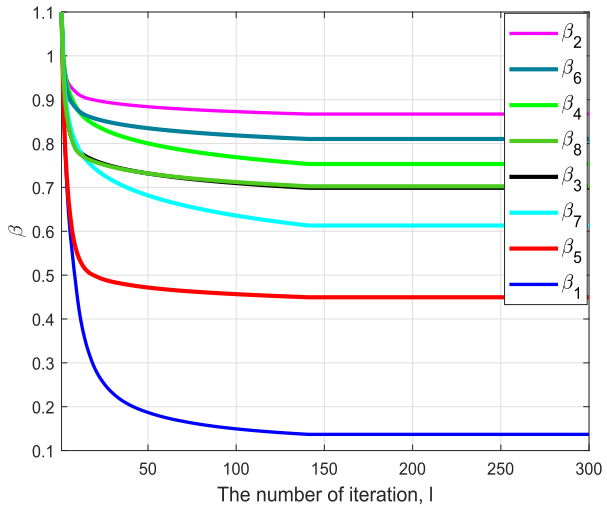


Fig. 3. Convergence of Algorithm 2.

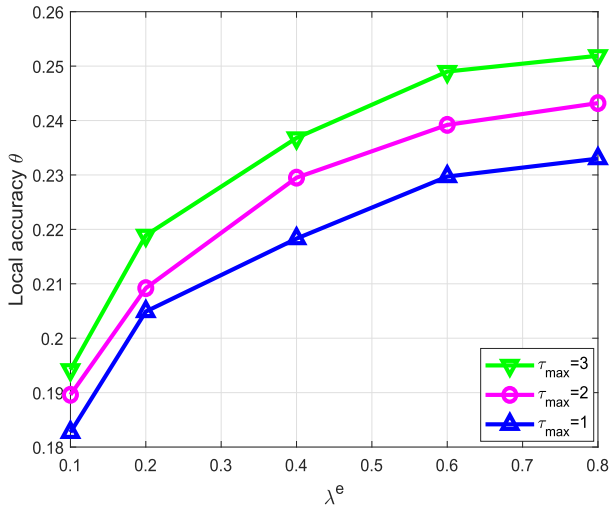


Fig. 5. Local accuracy versus λ^e.

observe that it converges normally in six steps. In order to display the overall convergence rate of Algorithm 1, we further plot the dual variable $\boldsymbol{\beta}$ versus the iteration number to show the convergence of its inner loop, as shown in Fig. 3. From Fig. 3, it is observed that Algorithm 2 has a fast convergence rate. Therefore, Algorithm 1 is cost-effective in terms of computation complexity. Furthermore, we can detect that, from Fig. 2, the cost of the worst-case smart device decreases as the number of subcarrier K increases for a given the number of smart devices N. The reason is that smart devices with good channel conditions are more likely to be assigned subcarriers, and the transmit power allocation is optimized. In Fig. 4, we show the convergence evolution of Algorithm 4, where η is the optimal value of problem (58). It is observe that it has a relatively fast convergence rate, and converges in twenty steps.

5.3 Performance of the Proposed Algorithm

Fig. 5 illustrates how the weight value λ^e affects the local accuracy θ. The local accuracy increases when τ_{max} increases. Further, it is observed that local accuracy increases rapidly with λ^e only when λ^e is below a certain threshold and then keeps growing slowly once λ^e is larger than this threshold. This is because the increase in λ^e will increase the overhead of power consumption (see $Cost = \lambda^e(\mathcal{I}(\theta)(E_{n_m} + E_m) + E_m^{cloud}) + \lambda^t(\mathcal{I}(\theta)\ (T_{n_m} + T_m) + T_m^{cloud})$). Therefore, to balance the overhead between the power consumption and delay, the local accuracy doesn't increase quickly as λ^e increases.

In Figs. 6 and 7, we give performance comparisons among the three schemes, namely the NCS [12], the proposed scheme, and the TTS [10], in terms of the system cost from different perspectives.

Fig. 6 shows the cost of the system and each smart device among the above three schemes. We can observe that the NCS saves the overhead of system at the cost of each smart device efficiency in bad channel conditions and limited computing resource. On the other hand, the proposed scheme can balance the cost of each smart device.

In Fig. 7, we further compare the system performance among the schemes from three aspects: the system cost, the worst cost, and the best cost. It is observed that there is a substantial difference in the cost between the worst and the best cost in the NCS and the TTS, while the cost of each smart device in the proposed scheme is well-balanced with a slightly increased in the system cost. This tradeoff between

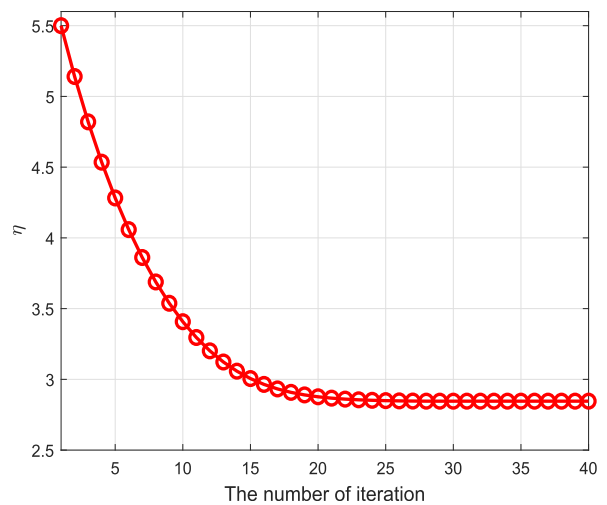


Fig. 4. Convergence of Algorithm 4.

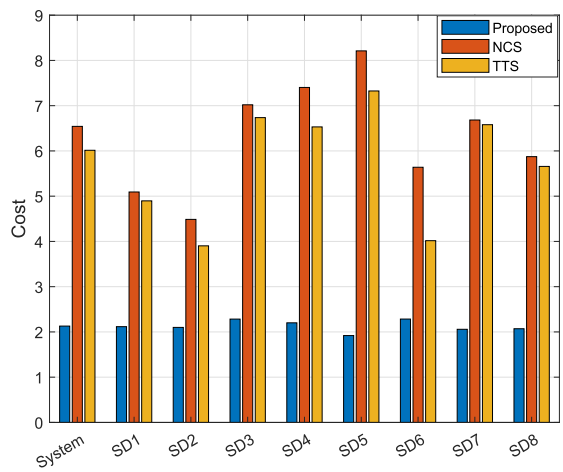


Fig. 6. Comparisons of the cost of the system and each SD.

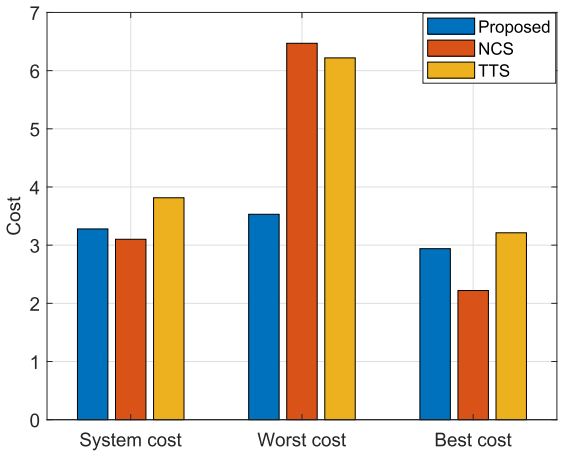


Fig. 7. Comparisons of the system cost, the worst cost, and the best cost.

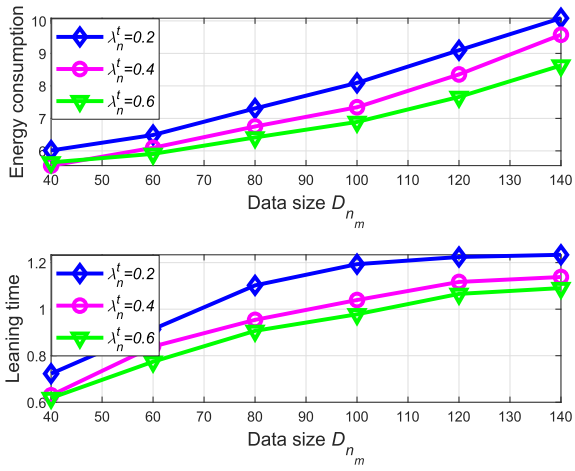


Fig. 9. Power consumption/training time under different data size D_{n_m}.

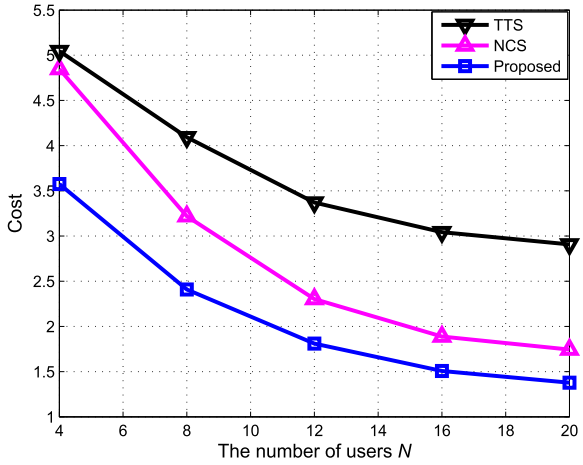


Fig. 8. Cost under different number of smart devices N.

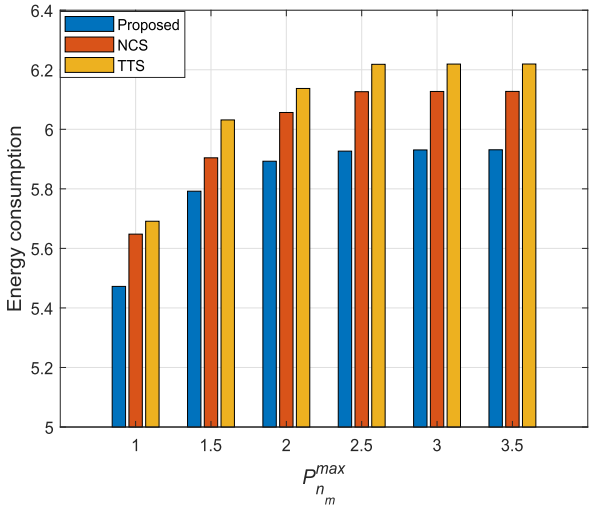


Fig. 10. Power consumption under different maximum transmit power $P_{n_m}^{max}$.

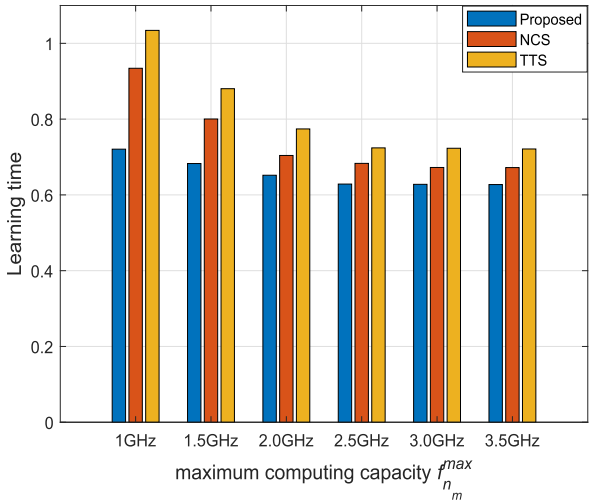


Fig. 11. Training time under different maximum computing capacity of smart devices $f_{n_m}^{max}$.

the system cost and individual smart device fairness is similar to the tradeoff between the throughput and fairness [28].

Fig. 8 shows the impact of the number of smart devices on the cost. Observe that, the cost slowly decreases with the increase in the number of smart devices, then the performance of the proposed scheme is the best, followed by the NCS, and the TTS is the worst. This is because the greater the number of smart devices increases, the more accurate the network trained model, and thus the lower the cost.

In Fig. 9, we investigate the impact the weight value λ^t on the energy consumption and the learning time. Accordingly, we set three weights parameters to display the performance of the proposed scheme, where the three parameters are set to $\lambda^t = 0.2$, $\lambda^t = 0.4$, and $\lambda^t = 0.6$, respectively. Observe that, for a given data size, the energy consumption and the learning time decrease as λ^t increases. Besides, for a given λ^t, it is observed that the values of the power consumption and the learning time increase with the increase in data size D_{n_m}. However, the learning time slowly grows when the data size exceeds a certain value. The reason is that the proposed scheme can achieve the tradeoff between energy consumption and the learning time.

Fig. 10 examines the energy consumption under different maximum transmit power $P_{n_m}^{max}$. It can be observed that the larger the maximum transmit power, the greater the energy consumption. That is because a looser transmit power constraint will cause a big transmit power, then the corresponding energy consumption will be big. However, observe that, the growth of

energy consumption does not increases indefinitely with the increase of $P_{n_m}^{max}$, but keeps unchanged once $P_{n_m}^{max}$ is larger than this threshold. That is because the learning time must be ensured. In addition, the performance of the proposed scheme is the best, followed by the NCS, and the TTS is the worst.

In Fig. 11, we show the impact of the maximum computing capacity $f_{n_m}^{max}$ on the learning time. From Fig. 11, we can observe that the learning time of all schemes decreases with the increase in the maximum computing capacity of smart device. That is because the computing time is monotonically decreasing in the the CPU-cycles frequency (see (6)). Besides, it is observed that the proposed scheme performs better than other schemes.

6 Conclusion

In this paper, we have considered a wireless edge network and investigated the energy consumption and the learning time of hierarchical federated learning by a formulated optimization problem. To meet the performance requirements of the system, we have jointly optimized local accuracy, subcarrier assignment, transmit power allocation, and local CPU-cycle frequency. Furthermore, we have decomposed the original problem into several sub-problem to solve, aiming to reduce the computing complexity of straight settle the formulated problem. Particularly, the subcarrier assignment and the transmit power are obtained by an iteration algorithm, the CPU-cycle frequency is obtained by an adaptive harmony search algorithm, and we have developed an iteration algorithm to obtain the local accuracy. Finally, the simulation results have revealed the performance of the proposed scheme and shown that the proposed method can provide fairness to all smart devices in terms of the cost.

Acknowledgments

We would like to express our appreciation to the reviewers for providing us with valuable comments for improving the manuscript.

References

[1] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021, doi: 10.1109/TWC.2020.3031503.

[2] C. Liu, K. Li, J. Liang, and K. Li, "COOPER-MATCH: Job offloading with a cooperative game for guaranteeing strict deadlines in MEC," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2019.2921713.

[3] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6214–6228, Jul. 2020.

[4] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[5] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2021.3099597.

[6] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, Jul. 2020.

[7] M. Song *et al.*, "Analyzing user-level privacy attack against federated learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.

[8] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1310–1322, Apr. 2020.

[9] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.

[10] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[11] L. U. Khan *et al.*, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[12] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

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

[14] C. Dinh *et al.*, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[15] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "Performance optimization of federated learning over wireless networks," in *Proc. IEEE Global Commun. Conf.*, 2019, pp. 1–6.

[16] C. Ma *et al.*, "Distributed optimization with arbitrary local solvers," *Optim. Methods Softw.*, vol. 32, no. 4, pp. 813–848, 2017.

[17] N. Mhaisen, A. Awad, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 55–66, Jan./Feb. 2022.

[18] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, 2021.

[19] L. Liu *et al.*, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2342–2353, Feb. 2021.

[20] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, Jun. 2020.

[21] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, no. 2/3, pp. 203–221, 1996.

[22] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, 1967.

[23] J. Crouzeix, J. A. Ferland, and S. Schaible, "An algorithm for generalized fractional programs," *J. Optim. Theory Appl.*, vol. 47, no. 1, pp. 35–49, 1985.

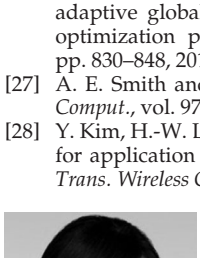
[24] S. Boyd, S. P. Boyd, and L. Vandenberghe, "Convex optimization," Cambridge Univ. Press, 2004.

[25] Y. Li *et al.*, "Energy-efficient subcarrier assignment and power allocation in OFDMA systems with max-min fairness guarantees," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3183–3195, Sep. 2015.

[26] Q. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Appl. Math. Comput.*, vol. 216, no. 3, pp. 830–848, 2010.

[27] A. E. Smith and D. W. Coit, "Penalty functions," *Handbook Evol. Comput.*, vol. 97, no. 1, 1997, Art. no. C5.

[28] Y. Kim, H.-W. Lee, and S. Chong, "Mobile computation offloading for application throughput fairness and energy efficiency," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 3–19, Jan. 2019.



Jie Feng (Member, IEEE) received the PhD degree in communication and information system from Xidian University, China, in 2020. She is currently a lecturer with the Department of Electrical Engineering and Computer Science, Xidian University, Xi'an, China. From 2019 to 2020, she was with Carleton University, Ottawa, ON, Canada, as a visiting PhD student. Her current research interests include mobile-edge computing, blockchain, deep reinforcement learning, device to device communication, resource allocation and convex optimization, and stochastic network optimization.

Lei Liu (Member, IEEE) received the BEng degree in communication engineering from Zhengzhou University, Zhengzhou, China, in 2010, and the MSc and PhD degrees in communication engineering from Xidian University, Xi'an, China, in 2013 and 2019, respectively. From 2013 to 2015, he worked in a technology company. From 2018 to 2019, he was supported by China Scholarship Council to be a visiting PhD student with the University of Oslo, Oslo, Norway. He is currently a lecturer with the Department of Electrical Engineering and Computer Science, Xidian University, China. His research interests include vehicular ad hoc networks, intelligent transportation, mobile-edge computing, and Internet of Things.

Qingqi Pei (Senior Member, IEEE) received the BS, MS and PhD degrees in computer science and cryptography from Xidian University, China, in 1998, 2005 and 2008, respectively. He is currently a professor and member of the State Key Laboratory of Integrated Services Networks, also a professional member of ACM and senior member of Chinese Institute of Electronics and China Computer Federation. His research interests focus on privacy preserving, blockchain and edge computing security.

Keqin Li (Fellow, IEEE) is currently a SUNY distinguished professor of computer science with the State University of New York, New York. He is also a National distinguished professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored more than 800 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds more than 60 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top 10 most influential scientists in distributed computing based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the *ACM Computing Surveys* and the *CCF Transactions on High Performance Computing*. He has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*.