

Efficient Federated Meta-Learning over Multi-Access Wireless Networks

Sheng Yue, *Student Member, IEEE*, Ju Ren, *Senior Member, IEEE*, Jiang Xin, *Student Member, IEEE*, Deyu Zhang, *Member, IEEE*, Yaoxue Zhang, *Senior Member, IEEE*, and Weihua Zhuang, *Fellow, IEEE*

Abstract—Federated meta-learning (FML) has emerged as a promising paradigm to cope with the data limitation and heterogeneity challenges in today’s edge learning arena. However, its performance is often limited by slow convergence and corresponding low communication efficiency. In addition, since the available radio spectrum and IoT devices’ energy capacity are usually insufficient, it is crucial to control the resource allocation and energy consumption when deploying FML in practical wireless networks. To overcome the challenges, in this paper, we rigorously analyze the contribution of each device to the global loss reduction in each round and develop an FML algorithm (called NUFM) with a non-uniform device selection scheme to accelerate the convergence. After that, we formulate a resource allocation problem integrating NUFM in multi-access wireless systems to jointly improve the convergence rate and minimize the wall-clock time along with energy cost. By deconstructing the original problem step by step, we devise a joint device selection and resource allocation strategy to solve the problem with theoretical guarantees. Further, we show that the computational complexity of NUFM can be reduced from $O(d^2)$ to $O(d)$ (with the model dimension d) via combining two first-order approximation techniques. Extensive simulation results demonstrate the effectiveness and superiority of the proposed methods in comparison with existing baselines.

Index Terms—Federated meta-learning, multi-access systems, device selection, resource allocation, efficiency.

I. INTRODUCTION

THE integration of Artificial Intelligence (AI) and Internet-of-Things (IoT) has led to a proliferation of studies on edge intelligence, aiming at pushing AI frontiers to the wireless network edge proximal to IoT devices and data sources [1]. It is expected that edge intelligence will reduce time-to-action latency down to milliseconds for IoT applications while minimizing network bandwidth and offering security guarantees [2]. However, a general consensus is that a single IoT device can hardly realize edge intelligence due to its limited computational and storage capabilities. Accordingly, it is natural to rely on collaboration in edge learning, whereby IoT devices work together to accomplish computation-intensive tasks [3].

Sheng Yue, Jiang Xin, Deyu Zhang are with the School of Computer Science and Engineering, Central South University, Changsha, 410083 China. Emails: {sheng.yue, xinjiang, zdy876}@csu.edu.cn.

Ju Ren and Yaoxue Zhang are with the Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing, 100084 China. Emails: {renju,zhangyx}@tsinghua.edu.cn.

Weihua Zhuang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Email: wzhuang@uwaterloo.ca.

Ju Ren is the corresponding author.

Building on a synergy of federated learning [4] and meta-learning [5], federated meta-learning (FML) has been proposed under a common theme of fostering edge-edge collaboration [6]–[8]. In FML, IoT devices join forces to learn an *initial shared model* under the orchestration of a central server such that current or new devices can quickly adapt the learned model to their local datasets via one or a few gradient descent steps. Notably, FML can keep all the benefits of the federated learning paradigm (such as simplicity, data security, and flexibility), while giving a more personalized model for each device to capture the differences among tasks [9]. Therefore, FML has emerged as a promising approach to tackle the heterogeneity challenges in federated learning and to facilitate efficient edge learning [10].

Despite its promising benefits, FML comes with new challenges. On one hand, the number of participated devices can be enormous. The uniform device selection at random, often done in the existing methods, leads to a low convergence speed [9], [11]. Although recent studies [12]–[14] have characterized the convergence of federated learning and proposed non-uniform device selection mechanisms, they cannot be directly applied to FML problems due to the bias and high-order information in the stochastic gradients. On the other hand, the performance of FML in a wireless environment is highly related to its wall-clock time, including the computation time (determined by local data sizes and devices’ CPU types) and communication time (depending on channel gains, interference, and transmission power) [15], [16]. If not properly controlled, a large wall-clock time can cause unexpected training delay and communication inefficiency. In addition, the DNN model training, which involves a large number of data samples and epochs, usually induces high computational cost, especially for sophisticated model structures consisting of millions of parameters [15]. Due to the limited power capacity of IoT devices, energy consumption should also be properly managed to ensure system sustainability and stability [14]. In a nutshell, for the purpose of efficiently deploying FML in today’s wireless systems, the strategy of device selection and resource allocation must be carefully crafted not only to accelerate the learning process, but also to control the wall-clock time of training and energy cost in edge devices. Unfortunately, despite their importance, there are limited studies on these aspects in the current literature.

In this paper, we tackle the above-mentioned challenges in two steps: 1) We develop an algorithm (called NUFM) with a non-uniform device selection scheme to improve the convergence rate of vanilla FML algorithm; 2) based on

NUFM, we propose a resource allocation strategy (called URAL) that jointly optimizes the convergence speed, wall-clock time, and energy consumption in the context of multi-access wireless systems. More specifically, first we rigorously quantify the contribution of each device to the convergence of FML via deriving a tight lower bound on the reduction of one-round global loss. Based on the quantitative results, we present the non-uniform device selection scheme that maximizes the loss reduction per round, followed by the NUFM algorithm. Then, we formulate a resource allocation problem for NUFM over wireless networks, capturing the trade-offs among convergence, wall-clock time, and energy cost. To solve this problem, we exploit its special structure and decompose it into two sub-problems. The first one is to minimize the computation time via controlling devices' CPU-cycle frequencies, which is solved optimally based on the analysis of the effect of device heterogeneity on the objective. The second sub-problem aims at optimizing the resource block allocation and transmission power management. It is a non-convex mixed-integer non-linear programming (MINLP) problem, and to derive a closed-form solution is a non-trivial task. Thus, after deconstructing the problem step by step, we devise an iterative method to solve it and provide a convergence guarantee.

In summary, our main contributions are three-fold.

- We provide a theoretical characterization of contribution of an individual device to the convergence of FML in each round, via establishing a tight lower bound on the one-round reduction of expected global loss. Using this quantitative result, we develop NUFM, a fast-convergent FML algorithm with non-uniform device selection;
- To embed NUFM in the context of multi-access wireless systems, we formulate a resource allocation problem, capturing the trade-offs among the convergence, wall-clock time, and energy consumption. By decomposing the original problem into two sub-problems and deconstructing sub-problems step by step, we propose a joint device selection and resource allocation algorithm (namely URAL) to solve the problem effectively with theoretical performance guarantees;
- To reduce the computational complexity, we further integrate our proposed algorithms with two first-order approximation techniques in [9], by which the complexity of a one-step update in NUFM can be reduced from $O(d^2)$ to $O(d)$. We also show that our theoretical results hold in these cases;
- We provide extensive simulation results on challenging real-world benchmarks (*i.e.*, Fashion-MNIST, CIFAR-10, CIFAR-100, and ImageNet) to demonstrate the efficacy of our methods.

The remainder of this paper is organized as follows. Section II briefly reviews the related work. Section III introduces the FML problem and standard algorithm. We present the non-uniform device selection scheme in Section IV and adapt the scheme for wireless networks in Section V. Finally, Section VI presents the extension to first-order approximation techniques, followed by the simulation results in Section VII and a conclusion drawn in Section VIII.

II. RELATED WORK

Federated learning (FL) [4] has been proposed as a promising technique to facilitate edge-edge collaborative learning [17]. However, due to the heterogeneity in devices, models, and data distributions, a shared global model often fails to capture the individual information of each device, leading to performance degradation in inference or classification [10], [18], [19].

Very recently, based on the advances in meta-learning [5], federated meta-learning (FML) has garnered much attention, which aims to learn a personalized model for each device to cope with the heterogeneity challenges [6]–[9], [11]. Chen *et al.* [6] first introduce an FML method called FedMeta, integrating the model-agnostic meta-learning (MAML) algorithm [5] into the federated learning framework. They show that FML can significantly improve the performance of FedAvg [4]. Jiang *et al.* [7] analyze the connection between FedAvg and MAML, and empirically demonstrate that FML enables better and more stable personalized performance. From a theoretical perspective, Lin *et al.* [8] analyze the convergence properties and computational complexity of FML with strongly convex loss functions and exact gradients. Fallah *et al.* [9] further provide the convergence guarantees in non-convex cases with stochastic gradients. Different from the above gradient descent-based approaches, another recent work [11] develops an ADMM-based FML method and gives its convergence guarantee under non-convex cases. However, due to selecting devices uniformly at random, the existing FML algorithms often suffer from slow convergence and low communication efficiency [9]. Further, deploying FML in practical wireless systems calls for effective resource allocation strategies [15], [20], which is beyond the scope of existing FML literature.

There exists a significant body of works placing interests in convergence improvement and resource allocation for FL [13]–[15], [21]–[36]. Regarding the convergence improvement, Nguyen *et al.* [13] propose a fast-convergent FL algorithm, called FOLB, which achieves a near-optimal lower bound for the overall loss decrease in each round. Note that, while the idea of NUFM is similar to [13], the lower bound for FML is derived from a completely different technical path due to the inherent complexity in the local update. To minimize the convergence time, a probabilistic device selection scheme for FL is designed in [25], which assigns high probabilities to the devices with large effects on the global model. Ren *et al.* [26] investigate a batchsize selection strategy for accelerating the FL training process. Karimireddy *et al.* [30] employ the variance reduction technique to develop a new FL algorithm. Based on momentum methods, Yu *et al.* [27] give an FL algorithm with linear speedup property. Regarding the resource allocation in FL, Dinh *et al.* [15] embed FL in wireless networks, considering the trade-offs between training time and energy consumption, under the assumption that all devices participate in the whole training process. From a long-term perspective, Xu *et al.* [28] empirically investigate the device selection scheme jointly with bandwidth allocation for FL, using the Lyapunov optimization method. Chen *et al.* [14] investigate a device selection problem with “hard” resource

constraints to enable the implementation of FL over wireless networks. Wang *et al.* [29] propose a control algorithm to determine the best trade-off between the local update and global aggregation under a resource budget.

Although extensive research has been carried out on FL, researchers have not treated FML in much detail. In particular, the existing FL acceleration techniques cannot be directly applied to FML due to the high-order information and biased stochastic gradients in the local update phases¹ (see Lemma 2 in Section IV). At the same time, device selection and resource allocation require crafting jointly [6], rather than simply plugging the existing strategies in.

III. PRELIMINARIES AND ASSUMPTIONS

In this section, we introduce federated meta-learning, including the learning problem, standard algorithm, and assumptions for theoretical analysis.

A. Federated Meta-Learning Problem

We consider a set \mathcal{N} of user devices that are all connected to a server. Each device $i \in \mathcal{N}$ has a labeled dataset $\mathcal{D}_i = \{x_i^j, y_i^j\}_{j=1}^{D_i}$ that can be accessed only by itself. Here, the tuple $(x_i^j, y_i^j) \in \mathcal{X} \times \mathcal{Y}$ is a data sample with input x_i^j and label y_i^j , and follows an unknown underlying distribution P_i . Define θ as the model parameter, such as the weights of a Deep Neural Network (DNN) model. For device i , the loss function of a model parameter $\theta \in \mathbb{R}^d$ is defined as $\ell_i(\theta; x, y)$, which measures the error of model θ in predicting the true label y given input x .

Federated meta-learning (FML) looks for a good model initialization (also called meta-model) such that the well-performed models of different devices can be quickly obtained via one or a few gradient descent steps. More specially, FML aims to solve the following problem

$$\min_{\theta \in \mathbb{R}^d} F(\theta) := \frac{1}{n} \sum_{i \in \mathcal{N}} f_i(\theta - \alpha \nabla f_i(\theta)) \quad (1)$$

where f_i represents the expected loss function over the data distribution of device i , *i.e.*, $f_i(\theta) := \mathbb{E}_{(x,y) \sim P_i} [\ell_i(\theta; x, y)]$, $n = |\mathcal{N}|$ is the number of devices, and α is the stepsize. The advantages of this formulation are two-fold: 1) It gives a personalized solution that can capture any heterogeneity between the devices; 2) the meta-model can quickly adapt to new devices via slightly updating it with respect to their own data. Clearly, FML well fits edge learning cases, where edge devices have insufficient computing power and limited data samples.

Next, we review the standard FML algorithm in the literature.

B. Standard Algorithm

Similar to federated learning, vanilla FML algorithm solves (1) in two repeating steps: *local update* and *global aggregation* [9], as detailed below.

- *Local update*: At the beginning of each round k , the server first sends the current global model θ^k to a fraction of devices \mathcal{N}_k chosen uniformly at random with pre-set size n_k . Then, each device $i \in \mathcal{N}_k$ updates the received model based on its meta-function $F_i(\theta) := f_i(\theta - \alpha \nabla f_i(\theta))$ by running τ (≥ 1) steps of stochastic gradient descent locally (also called mini-batch gradient descent), *i.e.*,

$$\theta_i^{k,t+1} = \theta_i^{k,t} - \beta \tilde{\nabla} F_i(\theta_i^{k,t}), \text{ for } 0 \leq t \leq \tau - 1 \quad (2)$$

where θ_i^k denotes the local model of device i in the t -th step of the local update in round k with $\theta_i^{k,0} = \theta^k$, and $\beta > 0$ is the meta-learning rate. In (2), the stochastic gradient $\tilde{\nabla} F_i(\theta)$ is given by

$$\tilde{\nabla} F_i(\theta) := (I - \alpha \tilde{\nabla}^2 f_i(\theta, \mathcal{D}_i'')) \tilde{\nabla} f_i(\theta - \alpha \tilde{\nabla} f_i(\theta, \mathcal{D}_i), \mathcal{D}_i') \quad (3)$$

where \mathcal{D}_i , \mathcal{D}_i' , and \mathcal{D}_i'' are independent batches², and for any batch \mathcal{D} , $\tilde{\nabla} f_i(\theta, \mathcal{D})$ and $\tilde{\nabla}^2 f_i(\theta, \mathcal{D})$ are the unbiased estimates of $\nabla f_i(\theta)$ and $\nabla^2 f_i(\theta)$ respectively, *i.e.*,

$$\tilde{\nabla} f_i(\theta, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \nabla \ell_i(\theta; x, y) \quad (4)$$

$$\tilde{\nabla}^2 f_i(\theta, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \nabla^2 \ell_i(\theta; x, y). \quad (5)$$

- *Global aggregation*: After updating the local model parameter, each selected device sends its local model $\theta_i^k = \theta_i^{k,\tau-1}$ to the server. The server updates the global model by averaging over the received models, *i.e.*,

$$\theta^{k+1} = \frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \theta_i^k. \quad (6)$$

It is easy to see that the main difference between federated learning and FML lies in the local update phase: In federated learning, local update is done using the unbiased gradient estimates while FML uses the biased one consisting of high-order information. Besides, federated learning can be considered as a special case of FML, *i.e.*, FML under $\alpha = 0$.

C. Assumptions

In this subsection, we list the standard assumptions for the analysis of FML algorithms [8], [9], [11].

Assumption 1 (Smoothness). *The expected loss function f_i corresponding to device $i \in \mathcal{N}$ is twice continuously differentiable and L_i -smooth, *i.e.*,*

$$\|\nabla f_i(\theta_1) - \nabla f_i(\theta_2)\| \leq L_i \|\theta_1 - \theta_2\|, \quad \forall \theta_1, \theta_2 \in \mathbb{R}^d. \quad (7)$$

*Besides, its gradient is bounded by a positive constant ζ_i , *i.e.*,* $\|\nabla f_i(\theta)\| \leq \zeta_i$.

Assumption 2 (Lipschitz Hessian). *The Hessian of function f_i is ρ_i -Lipschitz continuous for each $i \in \mathcal{N}$, *i.e.*,*

$$\|\nabla^2 f_i(\theta_1) - \nabla^2 f_i(\theta_2)\| \leq \rho_i \|\theta_1 - \theta_2\|, \quad \forall \theta_1, \theta_2 \in \mathbb{R}^d. \quad (8)$$

¹Different from FML, FL is a first-order method with unbiased stochastic gradients.

²We slightly abuse the notation \mathcal{D}_i as a batch of the local dataset of the i -th device.

TABLE I
KEY NOTATIONS

Notation	Definition	Notation	Definition
θ	Model parameter	$F_i(\cdot)$	Meta-loss function
α, β	Stepsize and meta-learning rate	L_i, ρ_i	Lipschitz continuous parameters
i, j	Indexes of user devices and data samples	σ_G, σ_H	Upper bounds of variances
k	Index of training rounds	γ_G, γ_H	Similarity parameters
t	Index of local update steps	u_i	Contribution to global loss reduction of device i
τ	Number of local update steps	v_i	CPU-cycle frequency of device i
P_i	Underlying data distribution of device i	p_i	Transmission power of device i
\mathcal{D}_i, D_i	Dataset and its size of device i	$z_{i,m}$	Binary variable indicating if device i accesses RB m
\mathcal{N}	Set of devices	η_1, η_2	Weight parameters
n	Number of devices	c_i	CPU cycles for computing one sample by device i
\mathcal{N}_k	Set of participating devices in round k	h_i, I_m	Channel gain of device i and inference in RB m
n_k	Number of participating devices in round k	B, N_0	Bandwidth of each RB and noise power spectral density
x, y	Input and corresponding label	\mathcal{M}, M	Set and number of RBs
$l_i(\theta; x, y)$	Loss of model θ on sample (x, y)	E	Total energy consumption
$f_i(\cdot)$	Expected loss function	T	Total latency

Assumption 3 (Bounded Variance). Given any $\theta \in \mathbb{R}^d$, the following facts hold for stochastic gradient $\nabla l_i(\theta; x, y)$ and Hessians $\nabla^2 l_i(\theta; x, y)$ with $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$\mathbb{E}_{(x,y) \sim P_i} [\|\nabla l_i(\theta; x, y) - \nabla f_i(\theta)\|^2] \leq \sigma_G^2 \quad (9)$$

$$\mathbb{E}_{(x,y) \sim P_i} [\|\nabla^2 l_i(\theta; x, y) - \nabla^2 f_i(\theta)\|^2] \leq \sigma_H^2. \quad (10)$$

Assumption 4 (Similarity). For any $\theta \in \mathbb{R}^d$ and $i, j \in \mathcal{N}$, there exist nonnegative constants $\gamma_G \geq 0$ and $\gamma_H \geq 0$ such that the gradients and Hessians of the expected loss functions $f_i(\theta)$ and $f_j(\theta)$ satisfy the following conditions

$$\|\nabla f_i(\theta) - \nabla f_j(\theta)\| \leq \gamma_G \quad (11)$$

$$\|\nabla^2 f_i(\theta) - \nabla^2 f_j(\theta)\| \leq \gamma_H. \quad (12)$$

Assumption 2 implies the high-order smoothness of $f_i(\theta)$ dealing with the second-order information in the local update step (2). Assumption 4 indicates that the variations of gradients between different devices are bounded by some constants, which captures the similarities between devices' tasks corresponding to non-IID data. It holds for many practical loss functions [37], such as logistic regression and hyperbolic tangent functions. In particular, ψ_i^g and ψ_i^h can be roughly seen as a distance between data distributions P_i and P_j [38].

IV. NON-UNIFORM FEDERATED META-LEARNING

Due to the uniform selection of devices in each round, the convergence rate of the standard FML algorithm is naturally slow. In this section, we present a non-uniform device selection scheme to tackle this challenge.

A. Device Contribution Quantification

We begin with quantifying the contribution of each device to the reduction of one-round global loss using its dataset size and gradient norms. For convenience, define $\zeta := \max_i \zeta_i$, $L := \max_i L_i$, and $\rho := \max_i \rho_i$. We first provide necessary lemmas before giving the main result.

Lemma 1. If Assumptions 1 and 2 hold, local meta-function F_i is smooth with parameter $L_F := (1 + \alpha L)^2 L + \alpha \rho \zeta$.

Sketch of Proof. We expand the expression of $\|\nabla F_i(\theta_1) - \nabla F_i(\theta_2)\|$ into two parts by triangle inequality, followed by bounding each part via Assumptions 1 and 2. The detailed proof is presented in Appendix A of the technical report [39]. \square

Lemma 1 gives the smoothness of the local meta-function F_i and global loss function F .

Lemma 2. Suppose that Assumptions 1-3 are satisfied, and $\mathcal{D}_i, \mathcal{D}'_i$, and \mathcal{D}''_i are independent batches with sizes D_i, D'_i , and D''_i respectively. For any $\theta \in \mathbb{R}^d$, the following holds

$$\|\mathbb{E}[\tilde{\nabla} F_i(\theta) - \nabla F_i(\theta)]\| \leq \frac{\alpha \sigma_G L (1 + \alpha L)}{\sqrt{D_i}} \quad (13)$$

$$\mathbb{E}[\|\tilde{\nabla} F_i(\theta) - \nabla F_i(\theta)\|^2] \leq \sigma_{F_i}^2 \quad (14)$$

where $\sigma_{F_i}^2$ is denoted as

$$\begin{aligned} \sigma_{F_i}^2 := & 6\sigma_G^2 (1 + \alpha L)^2 \left(\frac{1}{D'_i} + \frac{(\alpha L)^2}{D_i} \right) + \frac{3(\alpha \zeta \sigma_H)^2}{D''_i} \\ & + \frac{6(\alpha \sigma_G \sigma_H)^2}{D''_i} \left(\frac{1}{D'_i} + \frac{(\alpha L)^2}{D_i} \right). \end{aligned} \quad (15)$$

Sketch of Proof. We first obtain

$$\begin{aligned} \|\mathbb{E}[\tilde{\nabla} F_i(\theta) - \nabla F_i(\theta)]\| & \leq \|(I - \alpha \nabla^2 f_i(\theta))\mathbb{E}[\delta_2^*] + \mathbb{E}[\delta_1^* \delta_2^*] \\ & \quad + \mathbb{E}[\delta_1^*] \nabla f_i(\theta - \alpha \nabla f_i(\theta))\|, \\ \mathbb{E}[\|\tilde{\nabla} F_i(\theta) - \nabla F_i(\theta)\|^2] & \leq 3 \left(\mathbb{E}[\|\delta_1^*\|^2] \mathbb{E}[\|\delta_2^*\|^2] \right. \\ & \quad \left. + \zeta^2 \mathbb{E}[\|\delta_1^*\|^2] + (1 + \alpha L)^2 \mathbb{E}[\|\delta_2^*\|^2] \right), \end{aligned}$$

where δ_1^* and δ_2^* are given by

$$\begin{aligned} \delta_1^* & = \alpha \left(\nabla^2 f_i(\theta) - \tilde{\nabla}^2 f_i(\theta, \mathcal{D}'_i) \right) \\ \delta_2^* & = \tilde{\nabla} f_i(\theta - \alpha \tilde{\nabla} f_i(\theta, \mathcal{D}_i), \mathcal{D}'_i) - \nabla f_i(\theta - \alpha \nabla f_i(\theta)). \end{aligned}$$

Then, we derive the results via bounding the first and second moments of δ_1^* and δ_2^* . The detailed proof is presented in Appendix B of the technical report [39]. \square

Lemma 2 shows that the stochastic gradient $\tilde{\nabla}F_i(\theta)$ is a biased estimate of $\nabla F_i(\theta)$, revealing the challenges in analyzing FML algorithms.

Lemma 3. *If Assumptions 1, 2, and 4 are satisfied, then for any $\theta \in \mathbb{R}^d$ and $i, j \in \mathcal{N}$, we have*

$$\|\nabla F_i(\theta) - \nabla F_j(\theta)\| \leq (1 + \alpha L)^2 \gamma_G + \alpha \zeta \gamma_H. \quad (16)$$

Sketch of Proof. We divide the bound of $\|\nabla F_i(\theta) - \nabla F_j(\theta)\|$ into two independent terms, followed by bounding the two terms separately. The detailed proof is presented in Appendix C of the technical report [39]. \square

Lemma 3 characterizes the similarities between the local meta-functions, which is critical for analyzing the one-step global loss reduction because it relates the local meta-function and global objective. Based on Lemmas 1–3, we are now ready to give our main result.

Theorem 1. *Suppose that Assumptions 1-4 are satisfied, and \mathcal{D}_i , \mathcal{D}'_i , and \mathcal{D}''_i are independent batches. If the local update and global aggregation follow (2) and (6) respectively, then the following fact holds true for $\tau = 1$*

$$\begin{aligned} \mathbb{E}[F(\theta^k) - F(\theta^{k+1})] &\geq \beta \mathbb{E} \left[\frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \left(\left(1 - \frac{L_F \beta}{2}\right) \|\tilde{\nabla}F_i(\theta^k)\|^2 \right. \right. \\ &\quad \left. \left. - \left(\sqrt{(1 + \alpha L)^2 \gamma_G + \alpha \zeta \gamma_H + \sigma_{F_i}} \right) \right. \right. \\ &\quad \left. \left. \times \sqrt{\mathbb{E} \left[\|\tilde{\nabla}F_i(\theta^k)\|^2 \middle| \mathcal{N}_k \right]} \right) \right] \quad (17) \end{aligned}$$

where the outer expectation of RHS is taken with respect to the selected user set \mathcal{N}_k and data sample sizes, and the inner expectation is only regarding data sample sizes.

Sketch of Proof. Using the smoothness condition of F_i , we express the lower bound of loss reduction by

$$\mathbb{E}[F(\theta^k) - F(\theta^{k+1})] \geq \mathbb{E}[G^k],$$

where G^k is defined as

$$\begin{aligned} G^k &:= \beta \nabla F(\theta^k)^\top \left(\frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \tilde{\nabla}F_i(\theta^k) \right) \\ &\quad - \frac{L_F \beta^2}{2} \left\| \frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \tilde{\nabla}F_i(\theta^k) \right\|^2. \end{aligned}$$

The key step to derive the desired result is providing a tight lower bound for the product of $\nabla F(\theta^k)$ and $\tilde{\nabla}F(\theta^k)$. The detailed proof is presented in Appendix G of the technical report [39]. \square

Theorem 1 provides a lower bound on the one-round reduction of the global objective function F based on the device selection. It implies that different user selection has varying impacts on the objective improvement and quantifies the contribution of each device to the objective improvement,

depending on the variance of local meta-function, task similarities, smoothness, and learning rates. It therefore provides a criterion for selecting users to accelerate the convergence.

From Theorem 1, we have the following Corollary, which simplifies the above result and extends it to multi-step cases.

Corollary 1. *Suppose that Assumptions 1-4 are satisfied, and \mathcal{D}_i , \mathcal{D}'_i , and \mathcal{D}''_i are independent batches with $D_i = D'_i = D''_i$. If the local update and global aggregation follow (2) and (6) respectively, then the following fact holds true with $\beta \in [0, 1/L_F]$*

$$\begin{aligned} \mathbb{E}[F(\theta^k) - F(\theta^{k+1})] &\geq \frac{\beta}{2} \mathbb{E} \left[\frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \sum_{t=0}^{\tau-1} \left(\|\tilde{\nabla}F_i(\theta_i^{k,t})\|^2 \right. \right. \\ &\quad \left. \left. - 2(\lambda_1 + \frac{\lambda_2}{\sqrt{D_i}}) \sqrt{\mathbb{E} \left[\|\tilde{\nabla}F_i(\theta_i^{k,t})\|^2 \middle| \mathcal{N}_k \right]} \right) \right] \quad (18) \end{aligned}$$

where positive constants λ_1 and λ_2 satisfy that

$$\lambda_1 \geq \sqrt{(1 + \alpha L)^2 \gamma_G + \alpha \zeta \gamma_H + \beta \tau \sqrt{35(\gamma_G^2 + 2\sigma_F^2)}} \quad (19)$$

$$\begin{aligned} \lambda_2^2 &\geq 6\sigma_G^2 \left(1 + (\alpha L)^2 \right) \left((\alpha \sigma_H)^2 + (1 + \alpha L)^2 \right) \\ &\quad + 3(\alpha \zeta \sigma_H)^2. \quad (20) \end{aligned}$$

Sketch of Proof. The proof is similar to Theorem 1, with additional tricks in bounding the product of $\nabla F(\theta^k)$ and $\tilde{\nabla}F(\theta^k)$. The detailed proof is presented in Appendix I of the technical report [39]. \square

Corollary 1 implies that the device with a large gradient naturally accelerates the global loss decrease, but a small dataset size degrades the process due to corresponding high variance. Besides, as the device dissimilarities become large, the lower bound (18) weakens.

Motivated by Corollary 1, we study the device selection in the following subsection.

B. Device Selection

To improve the convergence speed, we aim to maximize the lower bound (18) on the one-round objective reduction. Based on Corollary 1, we define the contribution u_i^k device i to the convergence in round k as

$$u_i^k := \sum_{t=0}^{\tau-1} \|\tilde{\nabla}F_i(\theta_i^{k,t})\|^2 - 2(\lambda_1 + \frac{\lambda_2}{\sqrt{D_i}}) \|\tilde{\nabla}F_i(\theta_i^{k,t})\| \quad (21)$$

where we replace the second moment of $\tilde{\nabla}F_i(\theta_i^{k,t})$ by its sample value $\|\tilde{\nabla}F_i(\theta_i^{k,t})\|^2$. Then, the device selection problem in round k can be formulated as

$$\begin{aligned} \max_{\{z_i\}} &\sum_{i \in \mathcal{N}} z_i u_i^k \\ \text{s.t.} &\sum_{i \in \mathcal{N}} z_i = n_k \\ &z_i \in \{0, 1\}, \forall i \in \mathcal{N}. \quad (22) \end{aligned}$$

In (22), z_i is a binary variable: $z_i = 1$ for selecting device i in this round; $z_i = 0$, otherwise. The solution of (22) can be found by the *Select* algorithm introduced in [40, Chapter

9] with the worst-case complexity $O(n)$ (the problem (22) is indeed a *selection problem*). Accordingly, our device selection scheme is presented as follows.

Device selection: Following the local update phase, instead of selecting devices uniformly at random, each device i first computes its contribution scalar u_i^k locally and sends it to the server. After receiving $\{u_i^k\}_{i \in \mathcal{N}}$ from all devices, the server runs the *Select* algorithm and finds the optimal device set denoted by \mathcal{N}_k^* . Notably, although constants λ_1 and λ_2 in (22) consist of unknown parameters such as L , γ_G , and γ_H , they can be either estimated during training as in [29] or directly tuned as in our simulations.

Based on the device selection scheme, we propose the *Non-Uniform Federated Meta-Learning (NUFM)* algorithm (depicted in Algorithm 1). In particular, although NUFM requires an additional communication phase to upload u_i^k to the server, the communication overhead can be negligible because u_i^k is just a scalar.

Algorithm 1: Non-Uniform Federated Meta-Learning (NUFM)

Input: $\alpha, \beta, \lambda_1, \lambda_2$

- 1 Server initializes model θ^0 and sends it to all devices;
- 2 **for** round $k = 0$ to $K - 1$ **do**
- 3 **foreach** device $i \in \mathcal{N}$ **do**
- 4 // Local update
- 4 Initialize $\theta_i^{k,0} \leftarrow \theta^k$ and $u_i^k \leftarrow 0$;
- 5 **for** local step $t = 0$ to $\tau - 1$ **do**
- 6 Compute stochastic gradient $\tilde{\nabla} F_i(\theta_i^{k,t})$ by (3) using batches $\mathcal{D}_i, \mathcal{D}'_i$, and \mathcal{D}''_i ;
- 7 Update local model $\theta_i^{k,t+1}$ by (2);
- 8 Update contribution scalar u_i^k by $u_i^k \leftarrow u_i^k + \|\tilde{\nabla} F_i(\theta_i^{k,t})\|^2 - 2(\lambda_1 + \frac{\lambda_2}{\sqrt{D_i}}) \|\tilde{\nabla} F_i(\theta_i^{k,t})\|$;
- 9 **end**
- 10 Set $\theta_i^k = \theta_i^{k,\tau}$ and send u_i^k to server;
- 11 **end**
- 12 // Device selection
- 12 Once receiving $\{u_i^k\}_{i \in \mathcal{N}}$, server computes optimal device selection \mathcal{N}_k^* by solving (22);
- 13 // Global aggregation
- 13 After receiving local models $\{\theta_i^k\}_{i \in \mathcal{N}_k^*}$, server computes the global model by (6);
- 14 **end**
- 15 **return** θ^K

V. FEDERATED META-LEARNING OVER WIRELESS NETWORKS

In this section, we extend NUFM to the context of multi-access wireless systems, where the bandwidth for uplink transmission and the power of IoT devices are limited. First, we present the system model followed by the problem formulation. Then, we decompose the original problem into two sub-problems and devise solutions for each of them with theoretical performance guarantees.

A. System Model

As illustrated in Fig. 1, we consider a wireless multi-user system, where a set \mathcal{N} of n end devices joint forces to carry out federated meta-learning aided by an edge server. Each round consists of two stages: the computation phase and the communication phase. In the computation phase, each device $i \in \mathcal{N}$ downloads the current global model and computes the local model based on its local dataset; in the communication phase, the selected devices transmit the local models to the edge server via a limited number of wireless channels. After that, the edge server runs the global aggregation and starts the next round. Here we do not consider the downlink communication due to the asymmetric uplink-downlink settings in wireless networks. That is, the transmission power at the server (e.g., a base station) and the downlink communication bandwidth are generally sufficient for global meta-model transmission. Thus, the downlink time is usually neglected, compared to the uplink data transmission time [15]. Since we focus on the device selection and resource allocation problem in each round, we omit subscript k for brevity throughout this section.

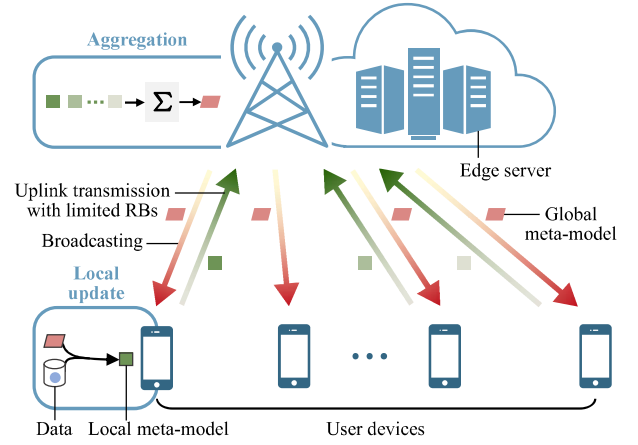


Fig. 1. The architecture of federated meta-learning over a wireless network with multiple user devices and an edge server. Due to limited communication resources, only part of user devices can upload their local models in each training round.

1) *Computation Model:* We denote c_i as the CPU cycles for device i to update the model with one sample, which can be measured offline as a priori knowledge [15]. Assume that the batch size of device i used in local update phase (2) is D_i . Then, the number of CPU cycles required for device i to run a one-step local update is $c_i D_i$. We denote the CPU-cycle frequency of device i as v_i . Thus, the CPU energy consumption of device i in the computation during the local update phase can be expressed by

$$E_i^{cp}(v_i) := \frac{\iota_i}{2} \tau_i c_i D_i v_i^2 \quad (23)$$

where $\iota_i/2$ is the effective capacitance coefficient of the computing chipset of device i [41]. The computational time of device i in a round can be denoted as

$$T_i^{cp}(v_i) := \frac{\tau_i c_i D_i}{v_i}. \quad (24)$$

For simplicity, we set $\tau = 1$ in the following.

2) *Communication Model*: We consider a multi-access protocol for devices, *i.e.*, the orthogonal frequency division multiple access (OFDMA) technique whereby each device can occupy one uplink resource block (RB) in a communication round to upload its local model. There are M RBs in the system, denoted by $\mathcal{M} = \{1, 2, \dots, M\}$. The achievable transmission rate of device i is [25]

$$r_i(\mathbf{z}_i, p_i) := \sum_{m \in \mathcal{M}} z_{i,m} B \log_2 \left(1 + \frac{h_i p_i}{I_m + B N_0} \right) \quad (25)$$

with B being the bandwidth of each RB, h_i the channel gain, N_0 the noise power spectral density, p_i the transmission power of device i , and I_m the interference caused by the devices that are located in other service areas and use the same RB. In (25), $z_{i,m} \in \{0, 1\}$ is a binary variable associated with the m -th RB allocation for device i : $z_{i,m} = 1$ indicates that RB m is allocated to device i , and $z_{i,m} = 0$ otherwise. Each device can only occupy one RB at maximum while each RB can be accessed by at most one device, thereby we have

$$\sum_{m \in \mathcal{M}} z_{i,m} \leq 1 \quad \forall i \in \mathcal{N} \quad (26)$$

$$\sum_{i \in \mathcal{N}} z_{i,m} \leq 1 \quad \forall m \in \mathcal{M}. \quad (27)$$

Due to the fixed dimension of model parameters, we assume that the model sizes of devices are constant throughout the learning process, denoted by S . If device i is selected, the time duration of transmitting the model is given by

$$T_i^{co}(\mathbf{z}_i, p_i) := \frac{S}{r_i(\mathbf{z}_i, p_i)} \quad (28)$$

where $\mathbf{z}_i := \{z_{i,m} \mid m \in \mathcal{M}\}$. Besides, the energy consumption of the transmission is

$$E_i^{co}(\mathbf{z}_i, p_i) := \sum_{m \in \mathcal{M}} z_{i,m} T_i^{co}(\mathbf{z}_i, p_i) p_i. \quad (29)$$

If no RB is allocated to device i in current round, its transmission power and energy consumption is zero.

B. Problem Formulation

For ease of exposition, we define $\mathbf{v} := \{v_i \mid i \in \mathcal{N}\}$, $\mathbf{p} := \{p_i \mid i \in \mathcal{N}\}$, and $\mathbf{z} := \{z_{i,m} \mid i \in \mathcal{N}, m \in \mathcal{M}\}$. Recall the procedure of NUFM. The total energy consumption $E(\mathbf{z}, \mathbf{p}, \mathbf{v})$ and wall-clock time $T(\mathbf{z}, \mathbf{p}, \mathbf{v})$ in a round can be expressed by

$$E(\mathbf{z}, \mathbf{p}, \mathbf{v}) := \sum_{i \in \mathcal{I}} \left(E_i^{cp}(v_i) + E_i^{co}(\mathbf{z}_i, p_i) \right) \quad (30)$$

$$T(\mathbf{z}, \mathbf{p}, \mathbf{v}) := \max_{i \in \mathcal{N}} T_i^{cp}(v_i) + \max_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m} T_i^{co}(\mathbf{z}_i, p_i) \quad (31)$$

where we neglect the communication time for transmitting the scalar u_i . The total contribution to the convergence is

$$U(\mathbf{z}) = \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m} u_i \quad (32)$$

where u_i is given in (21)³.

³One can regularize u_i via adding a large enough constant in (21) to keep it positive.

We consider the following non-convex mixed-integer non-linear programming (MINLP) problem

$$(P) \max_{\mathbf{z}, \mathbf{p}, \mathbf{v}} U(\mathbf{z}) - \eta_1 E(\mathbf{z}, \mathbf{p}, \mathbf{v}) - \eta_2 T(\mathbf{z}, \mathbf{p}, \mathbf{v})$$

$$\text{s.t. } 0 \leq p_i \leq p_i^{\max}, \quad \forall i \in \mathcal{N} \quad (33)$$

$$0 \leq v_i \leq v_i^{\max}, \quad \forall i \in \mathcal{N} \quad (34)$$

$$z_{i,m} \in \{0, 1\}, \quad \forall i \in \mathcal{N}, m \in \mathcal{M} \quad (35)$$

$$\sum_{i \in \mathcal{N}} z_{i,m} \leq 1, \quad \forall m \in \mathcal{M} \quad (26)$$

$$\sum_{m \in \mathcal{M}} z_{i,m} \leq 1, \quad \forall i \in \mathcal{N} \quad (27)$$

where $\eta_1 \geq 0$ and $\eta_2 \geq 0$ are weight parameters to capture the Pareto-optimal trade-offs among convergence, latency, and energy consumption, the values of which depend on specific scenarios. Constraints (33) and (34) give the feasible regions of devices' transmission power levels and CPU-cycle frequencies, respectively. Constraints (26) and (27) restrict that each device can only access one uplink RB while each RB can be allocated to one device at most.

In this formulation, we aim to maximize the convergence speed of FML, while minimizing the energy consumption and wall-clock time in each round. Notably, our solution can adapt to the problem with hard constraints on energy consumption and wall-clock time as in [14] via setting "virtual devices" (see Lemma 4 and Lemma 6).

Next, we provide a joint device selection and resource allocation algorithm to solve this problem.

C. A Joint Device Selection and Resource Allocation Algorithm

Substituting (30), (31), and (32) into problem (P), we can easily decompose the original problem into the following two sub-problems (SP1) and (SP2).

$$(SP1) \min_{\mathbf{v}} g_1(\mathbf{v}) = \eta_1 \sum_{i \in \mathcal{N}} \frac{c_i D_i v_i^2}{2} + \eta_2 \max_{i \in \mathcal{N}} \frac{c_i D_i}{v_i}$$

$$\text{s.t. } 0 \leq v_i \leq v_i^{\max}, \quad \forall i \in \mathcal{N}.$$

$$(SP2) \max_{\mathbf{z}, \mathbf{p}} g_2(\mathbf{z}, \mathbf{p}) = \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m} u_i$$

$$- \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} \eta_1 \frac{S p_i}{B \log_2 \left(1 + \frac{h_i p_i}{I_m + B N_0} \right)}$$

$$- \eta_2 \max_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m} \frac{S}{B \log_2 \left(1 + \frac{h_i p_i}{I_m + B N_0} \right)}$$

$$\text{s.t. } 0 \leq p_i \leq p_i^{\max}, \quad \forall i \in \mathcal{N}$$

$$\sum_{i \in \mathcal{N}} z_{i,m} \leq 1, \quad \forall m \in \mathcal{M}$$

$$\sum_{m \in \mathcal{M}} z_{i,m} \leq 1, \quad \forall i \in \mathcal{N}$$

$$z_{i,m} \in \{0, 1\}, \quad \forall i \in \mathcal{N}, m \in \mathcal{M}.$$

(SP1) aims at controlling the CPU-cycle frequencies for devices to minimize the energy consumption and latency in the computational phase. (SP2) controls the transmission power and RB allocation to maximize the convergence speed while minimizing the transmission cost and communication delay. We provide the solutions to these two sub-problems separately.

1) *Solution to (SP1)*: Denote the optimal CPU-cycle frequencies of (SP1) as $\mathbf{v}^* = \{v_i^*\}_{i \in \mathcal{N}}$. We first give the following lemma to offer insights into this sub-problem.

Lemma 4. *If device j is the straggler of all devices with optimal CPU frequencies \mathbf{v}^* , i.e., $j = \arg \max_{i \in \mathcal{N}} c_i D_i / v_i^*$, the following holds true for any $\eta_1, \eta_2 > 0$*

$$v_i^* = \begin{cases} \min \left\{ \sqrt[3]{\frac{a_2}{2a_1}}, \min_{i \in \mathcal{N}} \frac{c_j D_j v_j^{\max}}{c_i D_i} \right\} & , \text{ if } i = j \\ \frac{c_i D_i v_j^*}{c_j D_j} & , \text{ otherwise.} \end{cases} \quad (36)$$

The positive constants a_1 and a_2 in (36) are defined as

$$a_1 := \eta_1 \sum_{i \in \mathcal{N}} \frac{t_i (c_i D_i)^3}{2(c_j D_j)^2} \quad (37)$$

$$a_2 := \eta_2 c_j D_j. \quad (38)$$

Sketch of Proof. The derivation of the results involves two steps, i.e., expressing v_i^* by v_j^* and deriving v_j^* by solving the corresponding optimization problem. The detailed proof is presented in Appendix D of the technical report [39]. \square

Lemma 4 implies that if the straggler (a device with the lowest computational time) can be determined, then the optimal CPU-cycle frequencies of all devices can be derived as closed-form solutions. Intuitively, due to the contradiction in minimizing the energy consumption and computational time, if the straggler is fixed, then the other devices can use the smallest CPU-cycle frequencies as long as the computational time is shorter than that of the straggler. It leads to the following Theorem.

Theorem 2. *Denote $\mathbf{v}_{\text{straggler};j}^*$ as the optimal solution (i.e., (36)) under the assumption that j is the straggler. Then, the global optimal solution of (SP1) can be obtained by*

$$\mathbf{v}^* = \arg \min_{\mathbf{v} \in \mathcal{V}} g_1(\mathbf{v}) \quad (39)$$

where $\mathcal{V} := \{\mathbf{v}_{\text{straggler};j}^*\}_{j \in \mathcal{N}}$, and g_1 is the objective function in (SP1).

Proof. The result can be directly obtained from Lemma 4. We omit it for brevity. \square

Theorem 2 shows that the optimal solution of (SP1) is the fixed-straggler solution in Lemma 4 corresponding to the minimum objective g_1 . Thus, (SP1) can be solved with computational complexity $O(n)$ by comparing the achievable objective values corresponding to different stragglers.

2) *Solution to (SP2)*: Similar to Section V-C1, we denote the optimal solutions of (SP2) as $\mathbf{z}^* = \{z_{i,m}^* \mid i \in \mathcal{N}, m \in \mathcal{M}\}$ and $\mathbf{p}^* = \{p_i^* \mid i \in \mathcal{N}\}$ respectively, $\mathcal{N}^* := \{i \in \mathcal{N} \mid \sum_{m \in \mathcal{N}} z_{i,m}^* = 1\}$ as the optimal set of selected devices and, for each $i \in \mathcal{N}^*$, RB block allocated to i as m_i^* , i.e., $z_{i,m_i^*}^* = 1$.

It is challenging to derive a closed-form solution for (SP2) because it is a non-convex MINLP problem with non-differentiable “max” operator in the objective. Thus, in the following, we develop an iterative algorithm to solve this problem and show that the algorithm will converge to a local minimum.

We begin with analyzing the properties of the optimal solution in the next lemma.

Lemma 5. *Denote the transmission delay regarding \mathbf{z}^* and \mathbf{p}^* as δ^* , i.e.,*

$$\delta^* := \max_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m}^* \frac{S}{B \log_2 \left(1 + \frac{h_i p_i^*}{I_m + BN_0} \right)}. \quad (40)$$

The following relation holds

$$\delta^* \geq \frac{S}{B \log_2 \left(1 + \frac{h_i p_i^{\max}}{I_m + BN_0} \right)} \quad (41)$$

and p_i^* can be expressed by

$$p_i^* = \begin{cases} \frac{(I_m + BN_0)(2^{\frac{S}{B\delta^*}} - 1)}{h_i} & , \text{ if } i \in \mathcal{N}^* \\ 0 & , \text{ otherwise.} \end{cases} \quad (42)$$

Sketch of Proof. We prove (41) by contradiction and (42) by solving the corresponding transformed problem. The detailed proof is presented in Appendix E of the technical report [39]. \square

Lemma 5 indicates that the optimal transmission power can be derived as a closed-form via (42), given the RB allocation and transmission delay. Lemma 5 also implies that for any RB allocation strategy $\tilde{\mathbf{z}}$ and transmission delay $\tilde{\delta}$ (not necessarily optimal), equation (42) provides the “optimal” transmission power under $\tilde{\mathbf{z}}$ and $\tilde{\delta}$ as long as (41) is satisfied. Based on that, we have the following result.

Theorem 3. *Denote $\mu_{i,m} := (I_m + BN_0)(2^{\frac{S}{B\delta^*}} - 1)/h_i$. Given transmission delay δ^* , the optimal RB allocation strategy can be obtained by*

$$\mathbf{z}^* = \arg \max_{\mathbf{z}} \left\{ \sum_{i,m} z_{i,m} (u_i - e_{i,m}), \text{ s.t. (26) - (35)} \right\} \quad (43)$$

where

$$e_{i,m} := \begin{cases} \eta_1 \delta^* \mu_{i,m} & , \text{ if } \mu_{i,m} \leq p_i^{\max} \\ u_i + 1 & , \text{ otherwise.} \end{cases} \quad (44)$$

Proof. From Lemma 5, Eq. (43) holds if $\mu_{i,m} \leq p_i^{\max}$. On the other hand, when $\mu_{i,m} > p_i^{\max}$, if device i is selected, the transmission delay will be larger than δ^* (see the proof of Lemma 5), which is contradictory to the given condition. Thus, when $\mu_{i,m} > p_i^{\max}$, we set $e_{i,m} = u_i + 1$, ensuring device i not to be selected. \square

Theorem 3 shows the optimal RB allocation strategy can be obtained by solving (43), given transmission delay δ^* . Naturally, problem (43) can be equivalently transformed to a bipartite matching problem. Consider a *Bipartite Graph* \mathcal{G} with source set \mathcal{N} and destination set \mathcal{M} . For each $i \in \mathcal{N}$ and $m \in \mathcal{M}$, denote the weight of the edge from node i to node j as $w_{i \rightarrow j}$: If $u_i - e_{i,m} > 0$, $w_{i \rightarrow j} = e_{i,m} - u_i$; otherwise, $w_{i \rightarrow j} = \infty$. Therefore, maximizing (43) is equivalent to finding a matching in \mathcal{G} with the minimum sum of weights. It means that we can obtain the optimal RB allocation strategy under fixed transmission delay via *Kuhn-Munkres* algorithm with the worst complexity of $O(Mn^2)$ [42].

We proceed to show how to iteratively approximate the optimal δ^* , \mathbf{p}^* , and \mathbf{z}^* .

Lemma 6. Let j denote the communication straggler among all selected devices with respect to RB allocation \mathbf{z}^* and transmission power \mathbf{p}^* , i.e., for any $i \in \mathcal{N}^*$,

$$\begin{aligned} & \sum_{m \in \mathcal{M}} z_{i,m}^* \underbrace{\frac{S}{B \log_2 \left(1 + \frac{h_i p_i^*}{I_m + BN_0} \right)}}_{T_i^{co}(\mathbf{z}_i^*, \mathbf{p}_i^*)} \\ & \leq \sum_{m \in \mathcal{M}} z_{j,m}^* \underbrace{\frac{S}{B \log_2 \left(1 + \frac{h_j p_j^*}{I_m + BN_0} \right)}}_{T_j^{co}(\mathbf{z}_j^*, \mathbf{p}_j^*)}. \end{aligned} \quad (45)$$

Then, the following holds true:

- 1) Define function $f_4(p) := b_1((1+p) \log_2(1+p) \ln 2 - p) - \eta_2$. Then, $f_4(p)$ is monotonically increasing with respect to $p \geq 0$, and has unique zero point $\tilde{p}_j^0 \in (0, b_2]$, where b_1 and b_2 are denoted as follows

$$b_1 := \eta_1 \sum_{i \in \mathcal{N}^*} \frac{I_{m_i}^* + BN_0}{h_i} \quad (46)$$

$$b_2 := 2^{(1 + \sqrt{\max\{\frac{\eta_2}{b_1}, 1\}} - 1) / \ln 2}; \quad (47)$$

- 2) Denote $(SNR)_j := (I_{m_j}^* + BN_0) / h_j$. For $i \in \mathcal{N}^*$, we have

$$p_i^* = \begin{cases} \min \left\{ \tilde{p}_j^0, \min_{i \in \mathcal{N}^*} \frac{h_i p_i^{\max}}{I_{m_i}^* + BN_0} \right\} & , \text{ if } i = j \\ \frac{(SNR)_i}{(SNR)_j} p_j^* & , \text{ otherwise.} \end{cases} \quad (48)$$

Sketch of Proof. We obtain the first result by analyzing the property of f_4 and derive (48) via solving the corresponding optimization problem. The detailed proof is presented in Appendix F of the technical report [39]. \square

Lemma 6 indicates that, given optimal RB allocation strategy \mathbf{z}^* and straggler, the optimal transmission power can be derived by (48), different from Lemma 5 that requires the corresponding transmission delay δ^* . Notably, in (48), we can obtain zero point \tilde{p}_j^0 of f_4 with any required tolerance ϵ by *Bisection method* in $\log_2(\frac{b_2}{\epsilon})$ iterations at most.

Similar to Theorem 2, we can find the optimal transmission power by the following theorem, given the RB allocation.

Theorem 4. Denote $\mathbf{p}_{\text{straggler},j}^*$ as the optimal solution under the assumption that j is the communication straggler, given fixed RB allocation \mathbf{z}^* . The corresponding optimal transmission power is given by

$$\mathbf{p}^* = \arg \max_{\mathbf{p} \in \mathcal{P}} g_2(\mathbf{z}^*, \mathbf{p}) \quad (49)$$

where $\mathcal{P} := \{\mathbf{p}_{\text{straggler},j}^*\}_{j \in \mathcal{N}}$ and g_2 is the objective function defined in (SP2).

Proof. We can easily obtain the result from Lemma 6, thereby omitting it for brevity. \square

Define the communication time corresponding to \mathbf{z} and \mathbf{p} as $T^{co}(\mathbf{z}, \mathbf{p}) := \max_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}} z_{i,m} T_i^{co}(\mathbf{z}_i, p_i)$. Based on

Algorithm 2: Iterative Solution (IVES)

Input: $\eta_1, \eta_2, S, B, \{h_i\}, \{I_m\}$
 1 Initialize $t = 0$ and δ_0 by (50);
 2 **while not done do**
 3 Compute RB allocation strategy \mathbf{z}^t under δ^t using Kuhn-Munkres algorithm based on (43);
 4 Compute transmission power \mathbf{p}^t by (49);
 5 Update $\delta^{t+1} = T^{co}(\mathbf{z}^t, \mathbf{p}^t)$;
 6 **end**
 7 **return** $\mathbf{z}^*, \mathbf{p}^*$

Algorithm 3: User Selection and Resource Allocation (URAL) Algorithm

Input: $\eta_1, \eta_2, S, B, \{h_i\}, \{I_m\}, \{u_i\}, \{c_i\}, \{D_i\}$
 1 Compute \mathbf{v}^* by (39); // Solve (SP1)
 2 Compute \mathbf{z}^* and \mathbf{p}^* by IVES; // Solve (SP2)
 3 **return** $\mathbf{v}^*, \mathbf{z}^*, \mathbf{p}^*$

Theorem 3 and 4, we have the following *Iterative Solution (IVES)* algorithm to solve (SP2).

IVES: We initialize transmission delay δ^0 (based on (42)) as follows

$$\delta^0 = \max_{i \in \mathcal{N}} \frac{S}{B \log_2 \left(1 + \frac{h_i p_i^{\max}}{I_m + BN_0} \right)}. \quad (50)$$

In each iteration t , we first compute an RB allocation strategy \mathbf{z}^t via solving (43) by the *Kuhn-Munkres* algorithm. Then, based on \mathbf{z}^t , we find the corresponding transmission power \mathbf{p}^t by (49) and update the transmission delay by $\delta^{t+1} = T^{co}(\mathbf{z}^t, \mathbf{p}^t)$ before the next iteration. The details of IVES are depicted in Algorithm 2.

Using IVES, we can solve (SP2) in an iterative manner. In the following theorem, we provide the convergence guarantee for IVES.

Theorem 5. If we solve (SP2) by IVES, then $\{g_2(\mathbf{z}^t, \mathbf{p}^t)\}$ monotonically increases and converges to a unique point.

Sketch of Proof. The result is derived via proving $g(\mathbf{z}^t, \mathbf{p}^t) \leq g(\mathbf{z}^{t+1}, \tilde{\mathbf{p}}^{t+1})$ and $g(\mathbf{z}^{t+1}, \tilde{\mathbf{p}}^{t+1}) \leq g(\mathbf{z}^{t+1}, \mathbf{p}^{t+1})$. The detailed proof is presented in Appendix H of the technical report [39]. \square

Although IVES solves (SP2) iteratively, we observe that it can converge extremely fast (often with only two iterations) in the simulation, achieving a low computation complexity.

Combining the solutions of (SP1) and (SP2), we provide the *User selection and Resource Allocation (URAL)* in Algorithm 3 to solve the original problem (P). The URAL can simultaneously optimize the convergence speed, training time, and energy consumption via jointly selecting devices and allocating resources. Further, URAL can be directly integrated into the NUFM paradigm in the device selection phase to facilitate the deployment of FML in wireless networks.

VI. EXTENSION TO FIRST-ORDER APPROXIMATIONS

Due to the computation of Hessian in local update (2), it may cause high computational cost for resource-limited IoT devices. In this section, we address this challenge.

There are two common methods used in the literature to reduce the complexity in computing Hessian [9]:

- 1) replacing the stochastic gradient by

$$\tilde{\nabla} F_i(\theta) \approx \tilde{\nabla} f_i(\theta - \alpha \tilde{\nabla} f_i(\theta, \mathcal{D}_i), \mathcal{D}_i); \quad (51)$$

- 2) replacing the Hessian-gradient product by

$$\begin{aligned} & \tilde{\nabla}^2 f_i(\theta, \mathcal{D}_i'') \tilde{\nabla} f_i(\theta - \alpha \tilde{\nabla} f_i(\theta, \mathcal{D}_i), \mathcal{D}_i') \\ & \approx \frac{\tilde{\nabla} f_i(\theta + \epsilon \tilde{g}_i, \mathcal{D}_i'') - \tilde{\nabla} f_i(\theta - \epsilon \tilde{g}_i, \mathcal{D}_i'')}{2\epsilon} \end{aligned} \quad (52)$$

where $\tilde{g}_i = \tilde{\nabla} f_i(\theta - \alpha \tilde{\nabla} f_i(\theta, \mathcal{D}_i), \mathcal{D}_i')$.

By doing so, the computational complexity of a one-step local update can be reduced from $O(d^2)$ to $O(d)$, while not sacrificing too much learning performance. Next, we show that our results in Theorem 1 hold in the above two cases.

Corollary 2. *Suppose that Assumptions 1-4 are satisfied, and \mathcal{D}_i , \mathcal{D}_i' , and \mathcal{D}_i'' are independent batches. If the local update and global aggregation follow (2) and (6) respectively, we have for $\tau = 1$*

$$\begin{aligned} \mathbb{E}[F(\theta^k) - F(\theta^{k+1})] & \geq \beta \mathbb{E} \left[\frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \left(\left(1 - \frac{L_F \beta}{2}\right) \|\tilde{\nabla} F_i(\theta^k)\|^2 \right. \right. \\ & \quad \left. \left. - \left(\sqrt{(1 + \alpha L)^2 \gamma_G + \alpha \zeta \gamma_H + \tilde{\sigma}_{F_i}} \right) \right. \right. \\ & \quad \left. \left. \times \sqrt{\mathbb{E} \left[\|\tilde{\nabla} F_i(\theta^k)\|^2 \mid \mathcal{N}_k \right]} \right) \right] \end{aligned} \quad (53)$$

where $\tilde{\sigma}_{F_i}$ is defined as follows

$$\tilde{\sigma}_{F_i}^2 = \begin{cases} 4\sigma_G^2 \left(\frac{1}{D_i} + \frac{(\alpha L)^2}{D_i} \right) + 2(\alpha L \zeta)^2 & , \text{ if using (51)} \\ 6\sigma_G^2 \left(\frac{\alpha^2}{\epsilon^2 D_i''} + (1 + 2(\alpha L)^2) \right) & , \text{ if using (52).} \\ \cdot \left(\frac{1}{D_i} + \frac{(\alpha L)^2}{D_i} \right) + 2(\alpha \rho \epsilon)^2 \zeta^4 & \end{cases} \quad (54)$$

Proof. The detailed proof is presented in Appendix I of the technical report [39]. \square

Corollary 2 indicates that NUFM can be directly combined with the first-order approximation techniques to reduce computational cost. Further, similar to Corollary 1, Corollary 2 can be extended to the multi-step cases.

VII. SIMULATION

This section evaluates the performance of our proposed algorithms by comparing with existing baselines in real-world datasets. We first present the experimental setup, including the datasets, models, parameters, baselines, and environment. Then we provide our results from various aspects.

A. Experimental Setup

1) *Datasets and Models:* We evaluate our algorithms on four widely-used benchmarks, namely Fashion-MNIST [43], CIFAR-10 [44], CIFAR-100 [44], and ImageNet [45]. Specifically, the data is distributed among $n = 100$ devices as follows: a) Each device has samples from two random classes; b) the number of samples per class follows a truncated Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 5$ and $\sigma = 5$. We select 50% devices at random for training with the rest for testing. For each device, we divide the local dataset into a support set and a query set. We consider 1-shot 2-class classification tasks, *i.e.*, the support set contains only 1 labeled example for each class. We set the stepsizes as $\alpha = \beta = 0.001$. We use a convolutional neural network (CNN) with max-pooling operations and the Leaky Rectified Linear Unit (Leaky ReLU) activation function, containing three convolutional layers with sizes 32, 64, and 128 respectively, followed by a fully connected layer and a softmax layer. The strides are set as 1 for convolution operation and 2 for the pooling operation.

2) *Baselines:* To compare the performance of NUFM, we first consider two existing algorithms, *i.e.*, FedAvg [4] and Per-FedAvg [9]. Further, to validate the effectiveness of URAL in multi-access wireless networks, we use two baselines, called *Greedy* and *Random*. In each round, the *Greedy* strategy determines the CPU-cycle frequency v_i^g and transmission power p_i^g for device $i \in \mathcal{N}$ by greedily minimizing its individual objective, *i.e.*,

$$v_i^g = \arg \min_{v_i} \left\{ \eta_1 \frac{t_i c_i D_i v_i^2}{2} + \eta_2 \frac{c_i D_i}{v_i}, \text{ s.t. (34)} \right\} \quad (55)$$

$$p_i^g = \arg \min_{p_i} \left\{ \sum_{m \in \mathcal{M}} \frac{z_{i,m}^g (\eta_1 p_i + \eta_2) S}{B \log_2 \left(\frac{1 + h_i p_i}{I_m + B N_0} \right)}, \text{ s.t. (33)} \right\} \quad (56)$$

where $\{z_{i,m}^g\}_{m \in \mathcal{M}}$ is selected at random (*i.e.*, randomly allocating RBs to selected devices). The *Random* strategy decides the CPU-cycle frequencies, transmission powers, and RB allocation for the selected devices uniformly at random from the feasible regions.

3) *Implementation:* We implement the code in TensorFlow Version 1.14 on a server with two Intel[®] Xeon[®] Golden 5120 CPUs and one Nvidia[®] Tesla-V100 32G GPU. The parameters used in the simulation can be found in Table III.

B. Experimental Results

1) *Convergence Speed:* To demonstrate the improvement of NUFM on the convergence speed, we compare the algorithms on different benchmarks with the same initial model and learning rate. We vary the number of participated devices n_k from 20 to 40, and set the numbers of local update steps and total communication rounds as $\tau = 1$ and $K = 50$, respectively. We let $\lambda_1 = \lambda_2 = 1$. As illustrated in Fig. 2 and Table VII-B1, NUFM significantly improve the convergence speed and corresponding test accuracy of the existing FML approach on all datasets⁴. Clearly, it validates the effectiveness

⁴To make the graphs more legible, we draw symbols every two points in Fig. 2.

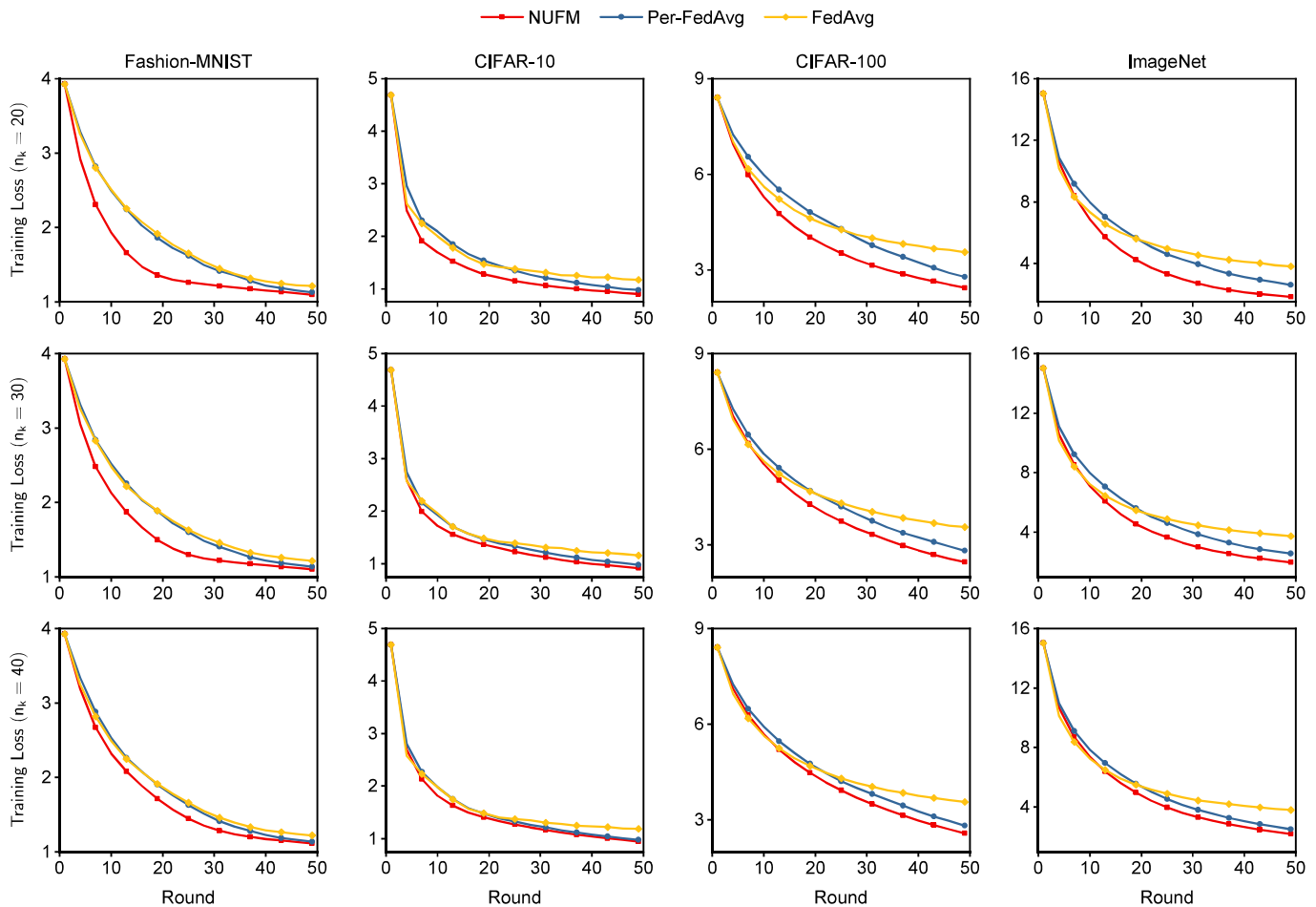


Fig. 2. Comparison of convergence rates under different numbers of participated devices. NUFM significantly accelerates the convergence of the existing FML approach, especially with fewer participated devices. In addition, with more participating devices, the advantages of NUFM weaken, leading to smaller gaps between NUFM and the existing algorithms.

of our proposed device selection scheme that maximizes the lower bound of one-round global loss reduction. Interestingly, Fig. 2 also indicates that NUFM converges more quickly with relatively fewer participated devices. For example, in round 19, the loss achieved by NUFM with $n_k = 20$ decreases by more than 9% and 20% over those with $n_k = 30$ and $n_k = 40$ on Fashion-MNIST, respectively. The underlying rationale is that relatively fewer “good” devices can provide a larger lower bound on the one-round global loss decrease (note that in (17) the lower bound takes the average of the selected devices). More selected devices in each round generally require more communication resources. Thus, the results reveal the potential of NUFM in applications to resource-limited wireless systems.

TABLE II
TEST ACCURACY AFTER FIFTY ROUNDS OF TRAINING.

Algorithm	Fashion-MNIST	CIFAR-10	CIFAR-100	ImageNet
NUFM	68.04%	58.80%	23.95%	34.04%
Per-FedAvg	62.75%	58.22%	21.49%	30.98%
FedAvg	61.04%	54.31%	10.13%	12.14%

2) *Effect of Local Update Steps*: To show the effect of local update steps on the convergence rate of NUFM, we present results with varying numbers of local update steps $\tau = 1, 2, \dots, 10$ in each round. For clarity of illustration, we

compare the loss under different numbers of local steps in the round 19 on Fashion-MNIST and CIFAR-100. Fig. 3 shows that fewer local update steps lead to a larger gap between the baselines and NUFM, which verifies the theoretical result that a small number of local steps can slow the convergence of FedAvg and Per-FedAvg [9, Theorem 4.5]. It also implies that NUFM can improve the computational efficiency of local devices.

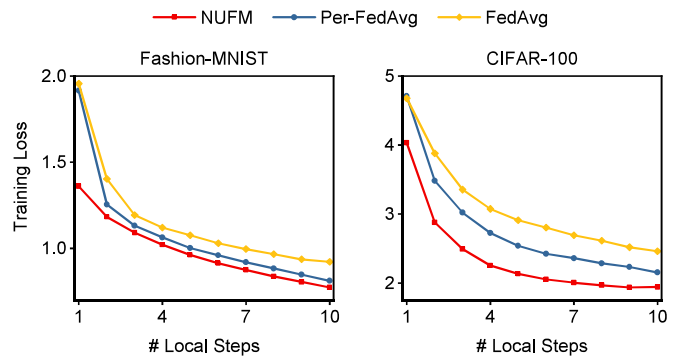


Fig. 3. Effect of local update steps on convergence rates. Fewer local steps lead to larger gaps between NUFM and the existing methods.

3) *Performance of URAL in Wireless Networks*: We evaluate the performance of URAL by comparing with four base-

lines, namely NUFM-*Greedy*, NUFM-*Random*, RU-*Greedy*, and RU-*Random*, as detailed below.

- NUFM-*Greedy*: select devices by NUFM, decide CPU-cycle frequencies, RB allocation, and transmission power by *Greedy* strategy;
- NUFM-*Random*: select devices by NUFM, decide CPU-cycle frequencies, RB allocation, and transmission power by *Random* strategy;
- RU-*Greedy*: select devices uniformly at random, decide CPU-cycle frequencies, RB allocation, and transmission power by *Greedy* strategy;
- RU-*Random*: select devices uniformly at random, decide CPU-cycle frequencies, RB allocation, and transmission power by *Random* strategy.

We simulate a wireless system consisting of $M = 20$ RBs and let the channel gain h_i of device i follow a uniform distribution $U(h^{min}, h^{max})$ with $h^{min} = 0.1$ and $h^{max} = 1$. We set $S = 1$, $B = 1$, and $n_0 = 1$. The interference of RB m is drawn from $I_m \sim U(0, 0.8)$. We set $t_i \sim U(0, 1)$, $c_i \sim U(0, 0.25)$, $p_i^{max} \sim U(0, 1)$, and $v_i^{max} \sim U(0, 2)$ for each $i \in \mathcal{N}$ to simulate the device heterogeneity. In the following experiments, we run the algorithms on FMNIST with local update steps $\tau = 1$ and $\eta_1 = \eta_2 = 1$.

As shown in Fig. 4, URAL can significantly reduce energy consumption and wall-clock time, as compared with the baselines. However, it is counter-intuitive that the *Greedy* strategy is not always better than *Random*. There are two reasons. On one hand, the energy cost and wall-clock time depend on the selection of weight parameters η_1 and η_2 . The results in Fig. 4 imply that, when $\eta_1 = \eta_2 = 1$, *Greedy* pays more attention to the wall-clock time than to energy consumption. Accordingly, *Greedy* achieves much lower average delay than that of *Random*, but sacrificing parts of the energy. On the other hand, the wall-clock time and energy cost require joint control with RB allocation. Although *Greedy* minimizes the individual objectives (55)-(56), improper RB allocation can cause arbitrary performance degradation. Different from *Greedy* and *Random*-based baselines, since URAL aims to maximize the joint objective (P) via co-optimizing the CPU-cycle frequencies, transmission power, and RB allocation strategy, it can alleviate the above-mentioned issues, and achieve a better delay and energy control. At the same time, Fig. 4 indicates that URAL converges as fast as NUFM-*Greedy* and NUFM-*Random* (the corresponding lines almost overlap), which select devices greedily to accelerate convergence (as in NUFM). Thus, URAL can have an excellent convergence rate.

4) *Effect of Resource Blocks*: In Fig. 5, we test the performance of URAL under different numbers of RBs. We vary the number of RBs M from 1 to 50. More RBs enable more devices to be selected in each round, leading to larger energy consumption. As shown in Fig. 5, URAL keeps stable wall-clock time with the increase of RBs. Meanwhile, URAL can control the power in devices to avoid serious waste of energy. It is counter-intuitive that the convergence speed does not always decrease with the increase of RBs, especially for URAL. The reason is indeed the same as that in Section VII-B1. That is, too few selected devices can slow the convergence due to insufficient information provided in each round while a large

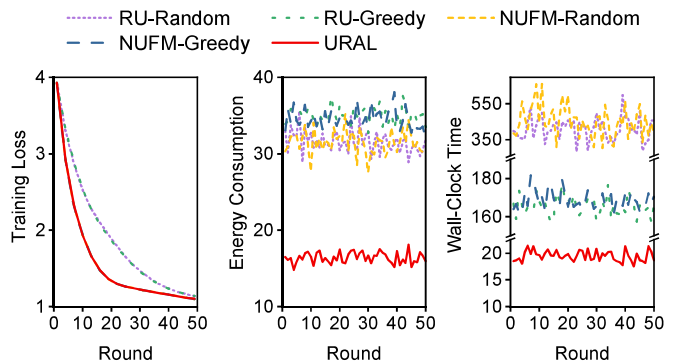


Fig. 4. Comparison of convergence, energy cost, and wall-clock training time. URAL can achieve a great convergence speed and short wall-clock time with low energy consumption.

number of participated devices may weaken the global loss reduction as shown in (1). Therefore, URAL can adapt to the practical systems with constrained wireless resources via achieving fast convergence with only a small set of devices.

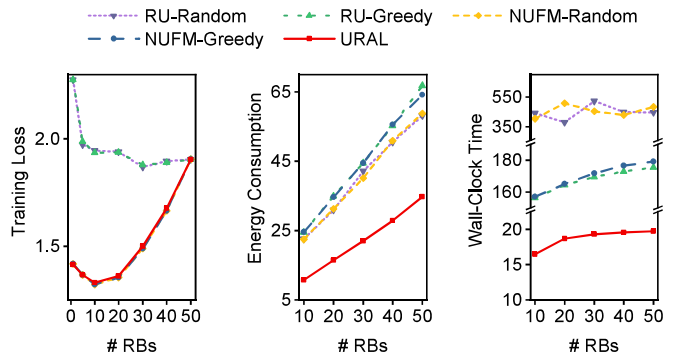


Fig. 5. Comparison of convergence, energy cost, and wall-clock time under different numbers of RBs. URAL can well control the energy cost and wall-clock time with more available RBs. Meanwhile, it can achieve fast convergence with only a small number of RBs.

5) *Effect of Channel Quality*: To investigate the effect of channel conditions on performance, we set the number of RBs $M = 20$ and vary the maximum channel gain h^{max} from 0.25 to 2, and show the corresponding energy consumption and wall-clock training time in Fig. 6. The results indicate that the energy consumption and latency decrease as channel quality improves, because devices can use less power to achieve a relatively large transmission rate.

6) *Effect of Weight Parameters*: We study how weight parameters η_1 and η_2 affect the average energy consumption and wall-clock time of URAL in Fig. 7. We first fix $\eta_2 = 1$ and vary η_1 from 0.5 to 2.5. As expected, the total energy consumption decreases with the increase of η_1 , with the opposite trend for wall-clock time. Then we vary η_2 with $\eta_1 = 1$. Similarly, a larger η_2 leads to less latency and more energy cost. It implies that we can control the levels of wall-clock training time and energy consumption by tuning the weight parameters. In particular, even with a large η_1 or η_2 , the wall-clock time and energy cost can be controlled at low levels. Meanwhile, the convergence rate achieved by URAL is robust to η_1 and η_2 . Thus, URAL can make full use of the resources (including datasets, bandwidth, and power) and

TABLE III
PARAMETERS IN SIMULATION

Parameter	Value
Step size (α) and meta-learning rate (β)	0.001
# edge devices (n)	100
# participating devices (n_k) in Experiments 1 and 2	An integer varying among $\{20, 30, 40\}$
# local updates (τ)	An integer varying among $\{1, 2, \dots, 10\}$
Hyper-parameters (λ_1 and λ_2)	1
Weight parameters (η_1 and η_2)	Real numbers varying among $\{0.5, 1, 1.5, 2.0, 2.5\}$
# RBs	An integer varying among $\{1, 5, 10, 20, 30, 40, 50\}$
CPU cycles of devices (c_i)	Real numbers following $U(0, 0.25)$
Maximum CPU-cycle frequencies of devices (ν_i^{max})	Real numbers following $U(0, 2)$
Maximum transmission powers of devices (p_i^{max})	Real numbers following $U(0, 1)$
Inference in RBs (I_m)	Real numbers following $U(0, 0.8)$
Model size (S), Bandwidth (B), Noise power spectral density (N_0)	1
effective capacitance coefficients of devices (t_i)	Real numbers following $U(0, 1)$
Channel gains of devices	Real numbers following $U(0.1, h^{max})$ where h^{max} varies among $\{0.25, 0.5, 0.75, \dots, 2.0\}$

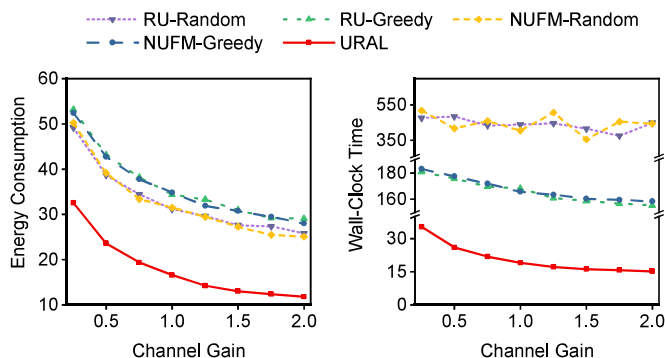


Fig. 6. Effect of channel gains on performance. Worse channel conditions would induce larger transmission power and longer wall-clock time.

achieve great trade-offs among the convergence rate, latency, and energy consumption.

VIII. CONCLUSION

In this paper, we have proposed an FML algorithm, called NUFM, that maximizes the theoretical lower bound of global loss reduction in each round to accelerate the convergence. Aiming at effectively deploying NUFM in wireless networks, we present a device selection and resource allocation strategy (URAL), which jointly controls the CPU-cycle frequencies and RB allocation to optimize the trade-off between energy consumption and wall-clock training time. Moreover, we integrate the proposed algorithms with two first-order approximation techniques to further reduce the computational complexity in IoT devices. Extensive simulation results demonstrate that the proposed methods outperform the baseline algorithms.

Future work will investigate the trade-off between the local update and global aggregation in FML to minimize the convergence time and energy cost from a long-term perspective. In addition, how to characterize the convergence properties and communication complexity of the NUFM algorithm requires further research.

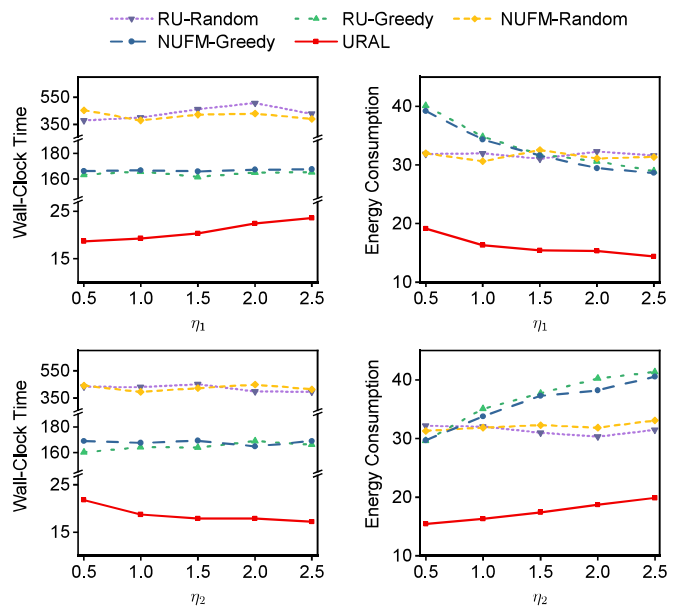


Fig. 7. Effect of weight parameters η_1 and η_2 . A large η_1 achieves lower energy consumption while leading to longer wall-clock time (the average wall-clock time is 18.63 when $\eta_1 = 0.5$; it is 21.53 when $\eta_1 = 2.5$). It is the opposite for η_2 .

ACKNOWLEDGMENT

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62122095, 62072472 and U19A2067, the 111 Project under Grant No. B18059, the National Key R&D Program of China under Grant No. 2019YFA0706403, the Natural Science Foundation of Hunan Province, China under Grant No. 2020JJ2050, and the Young Talents Plan of Hunan Province of China under Grant No. 2019RS2001.

REFERENCES

- [1] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

- [2] G. Plastiras, M. Terzi, C. Kyrkou, and T. Theocharides, "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications," in *Proc. IEEE ASAP*, 2018, pp. 1–7.
- [3] X. Zhang, Y. Wang, S. Lu, L. Liu, W. Shi *et al.*, "Openei: An open framework for edge intelligence," in *Proc. IEEE ICDCS*, 2019, pp. 1840–1851.
- [4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*. PMLR, 2017, pp. 1273–1282.
- [5] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.
- [6] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *ArXiv preprints arXiv: 1802.07876*, 2019.
- [7] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *ArXiv preprints arXiv: 1909.12488*, 2019.
- [8] S. Lin, G. Yang, and J. Zhang, "A collaborative learning framework via federated meta-learning," in *Proc. IEEE ICDCS*, 2020, pp. 289–299.
- [9] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. NIPS*, 2020, pp. 1–12.
- [10] P. Kairouz *et al.*, "Advances and open problems in federated learning," *ArXiv preprints arXiv: 1912.04977*, 2021.
- [11] S. Yue, J. Ren, J. Xin, S. Lin, and J. Zhang, "Inexact-admm based federated meta-learning for fast and continual edge learning," in *Proc. ACM MobiHoc*, 2021, p. 91–100.
- [12] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE ICC*, 2019, pp. 1–7.
- [13] H. T. Nguyen, V. Schwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, 2020.
- [14] 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, 2020.
- [15] C. T. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Networking*, vol. 29, no. 1, pp. 398–409, 2020.
- [16] J. Liu, J. Ren, Y. Zhang, X. Peng, Y. Zhang, and Y. Yang, "Efficient dependent task offloading for multiple applications in mec-cloud system," *IEEE Trans. Mob. Comput.*, 2021.
- [17] Z. M. Fadlullah and N. Kato, "Hcp: Heterogeneous computing platform for federated learning based collaborative content caching towards 6g networks," *IEEE Trans. Emerging Top. Comput.*, 2020.
- [18] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.
- [19] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
- [20] S. Yue, J. Ren, N. Qiao, Y. Zhang, H. Jiang, Y. Zhang, and Y. Yang, "Todg: Distributed task offloading with delay guarantees for edge computing," *IEEE Trans. Parallel Distrib. Syst.*, 2021.
- [21] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, 2020.
- [22] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2019.
- [23] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE ICC Workshops*, 2020, pp. 1–6.
- [24] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, 2020.
- [25] 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, 2020.
- [26] J. Ren, G. Yu, and G. Ding, "Accelerating dnn training in wireless federated edge learning systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 219–232, 2020.
- [27] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," in *Proc. ICML*, 2019, pp. 7184–7193.
- [28] 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, 2020.
- [29] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas in Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.
- [30] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. ICML*, 2020, pp. 5132–5143.
- [31] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE INFOCOM*, 2021.
- [32] W. Luping, W. Wei, and L. Bo, "Cmfl: Mitigating communication overhead for federated learning," in *Proc. IEEE ICDCS*, 2019, pp. 954–964.
- [33] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *Proc. ICLR*, 2019.
- [34] K. C. Sim *et al.*, "Personalization of end-to-end speech recognition on mobile devices for named entities," in *Proc. IEEE ASRU*, 2019, pp. 23–30.
- [35] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *Proc. IEEE ICC*, 2020, pp. 1–6.
- [36] 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, 2020.
- [37] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data," *ArXiv preprints arXiv: 2005.11418*, 2020.
- [38] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *Proc. AISTATS*, 2020, pp. 1082–1092.
- [39] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, and W. Zhuang, "Efficient federated meta-learning over multi-access wireless networks," *arXiv preprint arXiv:2108.06453*, 2021.
- [40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
- [41] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Sig. Proc. Syst. Sig. Image Video Technol.*, vol. 13, no. 2, pp. 203–221, 1996.
- [42] E. W. Weisstein, "Hungarian maximum matching algorithm," <https://mathworld.wolfram.com/>, 2011.
- [43] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *ArXiv preprints arXiv: 1708.07747*, 2017.
- [44] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Technical Report TR-2009, University of Toronto*, 2009.
- [45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. of IEEE CVPR*, 2009, pp. 248–255.