Joint Edge Aggregation and Association for Cost-Efficient Multi-Cell Federated Learning

Tao Wu*, Yuben Qu†, Chunsheng Liu*, Yuqian Jing†, Feiyu Wu†, Haipeng Dai‡, Chao Dong†, Jiannong Cao§
*National University of Defense Technology, China. Email: {wutao20,liuchunsheng17a}@nudt.edu.cn
†Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space,
Nanjing University of Aeronautics and Astronautics, China. Email: {quyuben,jingyuqian,feiyuwu,dch}@nuaa.edu.cn
‡State Key Laboratory for Novel Software Technology, Nanjing University, China. Email: haipengdai@nju.edu.cn
§Department of Computing, The Hong Kong Polytechnic University, Hong Kong. Email: csjcao@comp.polyu.edu.hk

***Abstract*—Federated learning (FL) has been proposed as a promising distributed learning paradigm to realize edge artificial intelligence (AI) without revealing the raw data. Nevertheless, it would incur inevitable costs in terms of training latency and energy consumption, due to periodical communication between user equipments (UEs) and the geographically remote central parameter server. Thus motivated, we study the joint edge aggregation and association problem to minimize the total cost, where the model aggregation over multiple cells just happens at the network edge. After proving its hardness with complex coupled variables, we transform it into a set function optimization problem and prove the objective function is neither submodular nor supermodular, which further complicates the problem. To tackle this difficulty, we first split it into multiple edge association subproblems, where the optimal solution to the computation resource allocation can be efficiently obtained in the closed form. We then construct a substitute function with the supermodularity and provable upper bound. On this basis, we reformulate an equivalent set function minimization problem under a matroid base constraint. We then propose an approximation algorithm to the original problem based on the two-stage search strategy with theoretical performance guarantee. Both extensive simulations and field experiments are conducted to validate the effectiveness of our proposed solution.**

I. Introduction

Recently, edge artificial intelligence (AI), where intelligence is pushed from the network core to the edge via running AI algorithms over edge devices, is seen as a promising key enabler to meet the great vision of ubiquitous intelligence [1]. With the proliferation of various mobile applications on massive Internet of Things (IoTs) devices, *e.g.,* autonomous driving, and health monitoring [2], edge AI also caters to the recent trend of most big data originated from the cloud to the edge. Federated learning (FL), proposed by Google firstly [3], has emerged as a promising distributed machine learning (ML) paradigm to realize the aforementioned edge AI. At its core, FL enables user equipments (UEs) to collaboratively train an ML model without letting out the raw data.

FL has demonstrated its empirical success with theoretical convergence guarantees [4]. Many literature have studied how to improve the learning performance [5]–[11]. Nevertheless, it incurs inevitable costs including training latency and energy consumption. This is because in FL, multiple local training rounds are performed in parallel on those UEs, and then periodical communication happens between them and the central parameter server for model aggregation. Considering the FL in a single cell is constrained by limited network coverage, it cannot support adequate IoT devices and thus leads to inevitable training performance loss. Moreover, since the resources (*e.g.,* communication and computation) are always limited in the wireless network [6], the costs could be unacceptable and a cost-efficient FL design over multi-cell is urgently needed in practice.

Existing researches about FL for edge AI mainly focus on minimizing the training latency [12]–[16], energy consumption [17]–[20], or both of them with trade-offs [21]–[23]. However, most works rely on a central parameter server on the cloud to aggregate the global model, which may incur unsatisfactory latency performance due to the long-range communication delay [24]. Besides, when realizing edge AI in some scenarios such as UAV swarms [25], [26], aggregating FL models on the cloud falls short in real-time nature and cannot cater to the timeliness requirement for emergent task execution. Moreover, aggregating the model on the cloud also faces the drawbacks including privacy concerns, poor scalability, and single point of failure. Although there exist some studies [14]–[17], [19]–[21] exploit the innovative mobile edge computing (MEC) architecture to aggregate the global model at the edge server installed in a base station (BS), they restrict the FL within a single cell which involves a relatively small number of UEs owing to limited coverage of the cell. Therefore, to involve more UEs for better learning performance and reduce the large propagation latency, it is necessary to study optimizing the cost of FL over multiple cells when *the global model is aggregated just at the edge*.

In this work, we study the problem of joint Edge AggRegation and associaTion for tHe cost-efficient multi-cell FL (EARTH). In particular, we take the computation and communication overhead into consideration and investigate how to jointly determine where to aggregate the global model (*i.e.*, *edge aggregation*) and which edge BS should associate

This work is supported by NSFC with No. 62002377, in part by the Hong Kong Scholars Program with No. 2021-101, in part by NSFC with No. 62072303, 62072424, 61872178, 62272223. (*Corresponding author: Yuben Qu.*)

to which UE (*i.e.*, *edge association*) alongside the resource allocation. Our proposed optimization problem yields two main technical challenges. First, edge association has been generally NP-hard. Then combining the resource allocation forms a complex mixed integer nonlinear programming (MINLP) problem with coupled optimization variables. Second, although the objective function could be equal to a set function via some transformation, it is essentially neither submodular nor supermodular, which makes the problem more challenging.

To address the challenging EARTH problem, we first prove its NP-hardness. Then, we split it into multiple edge association subproblems under given edge aggregation decisions, while the optimal solution for computation resource allocation can be obtained in the closed form. Next, we reformulate an equivalent set function optimization problem under a matroid base constraint and analyze the property of the objective function. Finally, we construct a substituted supermodular function with bounded gap and propose an approximation algorithm with theoretical performance guarantee.

Our main contributions are summarized as follows:

- As far as we know, we first study the cost-efficient FL over multiple cells where the global model aggregation happens at the edge. We analyze the problem complexity and identify the root cause of its NP-hardness.
- Via some problem transformation, we reformulate a set function optimization problem under a matroid base constraint where the objective function is neither submodular nor supermodular. Without loss of generality, we propose an innovative approach to decompose the complex objective function, extract the supermodular part and finally construct a substitute function for the non-supermodular part. On this basis, we design a two-stage search-based algorithm with theoretical performance guarantee.
- We conduct both extensive numerical simulations and field experiments to evaluate the performance of the proposed algorithm. The results show that our algorithm can achieve effective and near-optimal performance, while the average differences with the optimal solution in small-scale networks are 0.35% and 0.51%, respectively.

II. Related Works

A large number of studies [12]–[20] have focused on cost optimization in FL. **To reduce the training time**, Song *et al.* [12] studied the joint optimization of computation and communication duration. Vu *et al.* [13] proposed a cell-free massive MIMO scheme to minimize the training time. Xia *et al.* [14] introduced update-importance-based client scheduling schemes using the multi-armed bandit theory. Wei *et al.* [15] proposed the multi-agent multi-armed bandit framework for channel assignment and client selection. **To reduce the energy consumption**, Li *et al.* [17] designed an energy-efficiency oriented compression control scheme. Zeng *et al.* [18] explored an energy-efficient radio resource management in FL. Yang *et al.* [19] and Mo *et al.* [20] both investigated a joint learning and communication resource allocation. At the same time, there existing several researches [21]–[23] devoting to **capture trade-offs** between training latency and energy consumption or minimize the joint cost. All these works introduce the importance weighting indicators of energy and delay in the objective function.

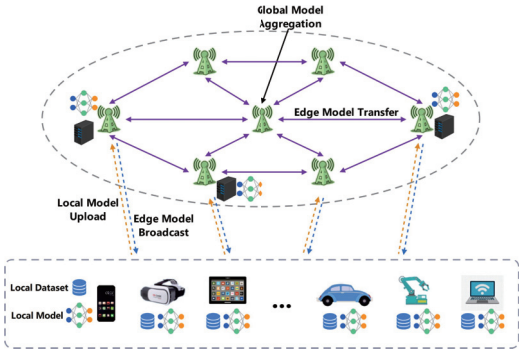

Fig. 1: An overview of federated edge learning over multi-cell networks.

To summarize, the aforementioned works mainly assume the central parameter server is on the cloud, which may suffer the large training latency. Although a few works [14]–[17], [19]–[21] consider the model aggregation at an edge server, they restrict the FL within a single cell and cannot cover a large enough number of UEs. Different from all the above works, we focus on how to minimize the overall cost of FL in terms of training latency and energy cost, while considering both the edge and global model aggregation *happen at the edge*.

III. Model and Problem Formulation

In this work, we consider an edge computing network with a set of UEs $\mathcal{N} := \{1, ..., N\}$ and a set of BSs $\mathcal{M} := \{1, ..., M\}$, where each BS is equipped with an edge server in the system. For simplicity, we interchangeably use BS and edge server in this paper. Due to limited coverage of a single cell, we suppose each BS covers a different set of UEs. These UEs aim to collaboratively train an ML model such as deep neural networks by FL, where each UE $n \in \mathcal{N}$ owns a local training dataset $D_n = \{(X_i, Y_i)\}_{i=1}^{|D_n|}$ with X_i and Y_i denoting the i-th input sample and corresponding labeled output, respectively.

A. Federated Edge Learning Model

The original federated edge learning system carries out the model aggregation on both the remote cloud and the edge [11]. As illustrated in Fig. 1, we focus on all the model aggregation happens within the edge only to achieve faster training, which can suffice the timeliness requirement for emergent task execution such as UAV swarms reconnaissance and early warning [26]. In this FL architecture, the training model goes through model uploading and aggregation in edge servers. An edge server acts as the parameter server to aggregate and broadcast the global model in each global iteration.

Without loss of generality, we assume all the model aggregation is executed by the widely used FedAvg algorithm [3], which can be extended to other FL algorithms. To be specific, for each UE $n \in \mathcal{N}$, it should solve the ML model parameter ω characterizing the output Y_i with loss function $f_n(\omega; X_i, Y_i)$. The loss function with respect to UE n can be defined as

$$F_n(\omega) := \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} f_n(\omega; X_i, Y_i). \quad (1)$$

In the local model update, each UE n should run a number of local rounds denoted as $L(\sigma) = \eta \log(1/\sigma)$ for a large number of iterative algorithm [27] to achieve the local accuracy $\sigma \in (0, 1)$, where η is a constant related to the data size and the ML task. In the t-th local round, each UE n calculates its local update as $\omega_n^t := \omega_n^t - \delta \bigtriangledown F_n(\omega_n^{t-1})$, such that $||\bigtriangledown F_n(\omega_n^t)|| \leq \sigma|| \bigtriangledown F_n(\omega_n^{t-1})||$ holds, where $\delta \in (0, 1)$ is the predefined learning rate [28].

After $L(\sigma)$ local rounds, each UE n will upload its local model ω_n to an associated edge server m. Let $\mathcal{N}_m \subseteq \mathcal{N}$ denote the set of UEs associated to edge server m, and the corresponding edge model is aggregated by averaging the local models as follows.

$$W_m = \frac{\sum_{n \in \mathcal{N}_m} |D_n| \omega_n}{\sum_{n \in \mathcal{N}_m} |D_n|}. \quad (2)$$

Then, edge server m will broadcast W_m to its associated UEs for local model update in the next edge round, until reaching an edge model accuracy σ', which is identical for all edge servers. For a general convex FL task, the number of edge rounds can be obtained as $L'(\sigma', \sigma) = \frac{\eta' \log(1/\sigma')}{1-\sigma}$ [29], where η' is a constant related to the exact learning task. After the edge model aggregation, each edge server will transmit its edge model W_m to one of the edge servers (the selected parameter server) for global model aggregation. The global model is aggregated as follows.

$$W = \frac{\sum_{m \in \mathcal{M}} (\sum_{n \in \mathcal{N}_m} |D_n|) W_m}{\sum_{n \in \mathcal{N}} |D_n|}. \quad (3)$$

B. Latency and Energy Cost Models

To quantify the training overhead, we formulate the latency and energy cost in edge aggregation and association within one global iteration. Let z_m and x_{mn} be the indicator variables for edge aggregation and association, respectively. Variable $z_m \in \{0, 1\}$ denotes whether edge server m is chosen as the parameter server for global model aggregation ($z_m = 1$) or not ($z_m = 0$). $x_{mn} \in \{0, 1\}$ denotes whether UE n is associated to edge server m (*i.e.*, $x_{mn} = 1$ equals $n \in \mathcal{N}_m$) or not ($x_{mn} = 0$ equals $n \notin \mathcal{N}_m$). Considering edge servers generally possess powerful computation capability, the edge model aggregation/ broadcasting latency is very small to be ignored [20].

1) *Latency and energy cost in local model update*: Let α_n denote the number of CPU cycles for UE n to process one sample. The required CPU cycles to run one local round is thus $\alpha_n |D_n|$. We denote the allocated CPU frequency of UE n for computation by f_n. Then, the latency of running $L(\sigma)$ local rounds at UE n is $t_{cmp}^n = L(\sigma) \frac{\alpha_n |D_n|}{f_n}$ and the corresponding energy consumption is $e_{cmp}^n = L(\sigma) \frac{\beta}{2} f_n^2 \alpha_n |D_n|$, where $\frac{\beta}{2}$ is the effective capacitance coefficient [23].

2) *Latency and energy cost in local model uploading*: Following [30], [31], we adopt orthogonal frequency-division multiple access (OFDMA) for uplink channel access. In this case, the communication bandwidth is divided into multiple narrowband sub-channels without interference. For the sake of fairness and simplicity, we assume the bandwidth resource is shared equally among all associated UEs. Then, given the maximal bandwidth B_m^{max} for edge server m, we can achieve the allocated bandwidth to UE n as $B_{mn} = \frac{B_m^{max}}{|\mathcal{N}_m|}$. The achievable uplink data rate for UE n is

$$r_{mn} = \frac{B_m^{max}}{|\mathcal{N}_m|} \log(1 + \frac{h_{mn} p_n}{N_0}), \quad (4)$$

where h_{mn} is the uplink channel gain between UE n and edge server m, p_n is the transmission power of UE n, and N_0 is the background noise. As a result, the latency for UE n transmitting its local model ω_n to edge server m is

$$t_{up\_com}^{mn} = d_n / r_{mn} = \frac{d_n |\mathcal{N}_m|}{B_m^{max} \log(1 + \frac{h_{mn} p_n}{N_0})}, \quad (5)$$

where d_n is the data size of local model ω_n, and the corresponding energy consumption is

$$e_{up\_com}^{mn} = p_n t_{up\_com}^{mn} = \frac{p_n d_n |\mathcal{N}_m|}{B_m^{max} \log(1 + \frac{h_{mn} p_n}{N_0})}. \quad (6)$$

3) *Latency and energy cost in edge/global model transfer*: For the edge/global model transfer between any two edge servers m and m', there exists a propagation latency $t_{pro}^{mm'}$, and $2t_{pro}^{mm'}$ for the round-trip transmission. In practical, the value of $t_{pro}^{mm'}$ is mainly determined by distance between them as well as the number of hops, which is also much smaller than the communication latency to the remote cloud.

As a result, we can conclude that in each round of edge aggregation, the total latency within edge server m is

$$T_{edge}^m = \max_{n \in \mathcal{N}} \{(t_{cmp}^n + t_{up\_com}^{mn}) x_{mn}\}, \quad (7)$$

and the corresponding consumption is

$$E_{edge}^m = \sum_{n \in \mathcal{N}} (e_{cmp}^n + e_{up\_com}^{mn}) x_{mn}. \quad (8)$$

Since it takes $L'(\sigma', \sigma)$ edge rounds to achieve the target edge accuracy σ', we can further conclude that in one global round, the overall latency is given by

$$\begin{aligned} T &= \max_{m \in \mathcal{M}} \{L'(\sigma', \sigma) T_{edge}^m + \sum_{m' \in \mathcal{M}} 2t_{pro}^{mm'} z_{m'}\} \\ &= \max_{m \in \mathcal{M}} \max_{n \in \mathcal{N}} \{L'(\sigma', \sigma)(t_{cmp}^n + t_{up\_com}^{mn}) x_{mn} + \sum_{m' \in \mathcal{M}} 2t_{pro}^{mm'} z_{m'}\}, \end{aligned} \quad (9)$$

and the overall energy consumption is

$$\begin{aligned} E &= \sum_{m \in \mathcal{M}} L'(\sigma', \sigma) E_{edge}^m \\ &= \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} L'(\sigma', \sigma)(e_{cmp}^n + e_{up\_com}^{mn}) x_{mn}. \end{aligned} \quad (10)$$

C. Problem Formulation

Based on the aforementioned models, we consider the system-wide overhead optimization problem in terms of latency and energy cost. Similar to [21], [23], we introduce weighted coefficients $\mu, \nu \in [0,1]$ ($\mu + \nu = 1$) to denote the corresponding importance and strike the balance between latency and energy cost. We then mathematically formulate the problem of joint Edge AggRegation and associaTion for tHe efficient multi-cell federated learning (EARTH) as follows:

$$(\mathbf{P1}): \min_{\mathbf{Z},\mathbf{X},\mathbf{F}} \ \Omega(\mathbf{Z},\mathbf{X},\mathbf{F}) = \mu E + \nu T \tag{11}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{mn} = 1, \forall n \in \mathcal{N}, \tag{11a}$$

$$\sum_{m\in\mathcal{M}} z_m = 1, \forall m \in \mathcal{M}, \tag{11b}$$

$$f_n^{\min} \le f_n \le f_n^{\max}, \forall n \in \mathcal{N}, \tag{11c}$$

$$x_{mn} \in \{0,1\}, z_m \in \{0,1\}, \forall n \in \mathcal{N}, m, m' \in \mathcal{M}. \tag{11d}$$

The above optimization problem involves three variables as $\mathbf{Z}$, $\mathbf{X}$ and $\mathbf{F}$ which are coupled with each other. It is a mixed integer non-linear programming (MINLP) problem, since z_m, x_{mn} are integers and f_n is continuous. Constraint (11a) ensures that each UE should be associated to one of the edge servers. Constraint (11b) states one edge server should be selected as the sole parameter server. Constraints (11c) and (11d) specify the basic range of optimization variables.

IV. Problem Transformation and Analysis

A. Complexity Analysis

In problem $\mathbf{P1}$, there are three types of decision constraints: the $\mathbf{Z}$-constraint, the $\mathbf{X}$-constraint, and the $\mathbf{F}$-constraint.

1) Having $\mathbf{X}$-constraint only: when the edge aggregation decision and the computation frequency are given, *i.e.,* $\mathbf{Z^0} = \{z_m^0\}, \mathbf{F^0} = \{f_n^0\}$, we can denote the objective function in $\mathbf{P1}$ as $\Omega(\mathbf{X})$ and change to $\mathbf{P2}$:

$$(\mathbf{P2}): \min_{\mathbf{X}} \ \Omega(\mathbf{X}) = \mu E(\mathbf{X}) + \nu T(\mathbf{X}) \tag{12}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{mn} = 1, \forall n \in \mathcal{N}, \tag{12a}$$

$$x_{mn} \in \{0,1\}, m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{12b}$$

which is a binary integer programming (BIP) problem with respect to $\mathbf{X}$ only. Then, we have the following theorem.

Theorem 1: The $\mathbf{X}$-constraint alone induces the NP-hardness of EARTH.

Proof: We can prove it by reduction from the *k-median problem* [32], which is omitted due to the space limitation. ■

2) Removing $\mathbf{X}$-constraint: when the edge association decision is given, *i.e.*, $\mathbf{X^0} = \{x_{mn}^0\}$, we have the set of UEs $\mathcal{N}_m$ associated to edger server m. Then, we can loop through all edge servers for parameter server selection, *i.e.*, $z_{m'} = 1$ to compare the objective function value. To simplify the notations, we first introduce the following terms:

$$\begin{aligned}
&\mathrm{A}_n = L'(\sigma',\sigma)L(\sigma)\alpha_n|D_n|,\\
&\mathrm{K}_n = L'(\sigma',\sigma)L(\sigma)\tfrac{\beta}{2}\alpha_n|D_n|,\\
&\Upsilon_{mn} = L'(\sigma',\sigma)e_{up_com}^{mn},\\
&\Psi_{mn} = L'(\sigma',\sigma)t_{up_com}^{mn},\\
&\Phi_m = \textstyle\sum_{m'\in\mathcal{M}} 2t_{pro}^{mm'} z_{m'}.
\end{aligned}$$

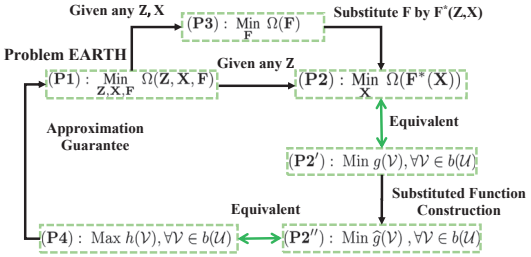

Fig. 2: A flowchart of problem transformation.

Thus, our problem turns into a computation resource allocation problem with respect to $\mathbf{F} := \{f_n\}$ only as follows:

$$\begin{aligned}
(\mathbf{P3}): \min_{\mathbf{F}} \ \Omega(\mathbf{F}) &= \mu E(f_n) + \nu T(f_n)\\
&= \mu \sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}_m} (\mathrm{K}_n f_n^2 + \Upsilon_{mn})\\
&\quad + \nu \max_{m\in\mathcal{M}} \max_{n\in\mathcal{N}_m} \{\frac{\mathrm{A}_n}{f_n} + \Psi_{mn} + \Phi_m\},
\end{aligned} \tag{13}$$

$$\text{s.t.} \quad f_n^{\min} \le f_n \le f_n^{\max}, \forall n \in \mathcal{N}. \tag{13a}$$

Lemma 1: Removing $\mathbf{X}$-constraint makes the problem polynomial-time solvable.

Proof: From the expression, we can find that problem $\mathbf{P3}$ is convex, which can be solved using some convex optimization solvers (*e.g.*, CVX and IPOPT). Thus, the optimal solution $\mathbf{F}^* = \{f_n^*\}$ can be obtained in polynomial time. ■

B. Problem Reformulation

Fig. 2 presents the flowchart of problem transformation. Inspired by Lemma 1, for any $\mathbf{Z}$ and $\mathbf{X}$, we can always obtain the closed form of optimal computation resource allocation $\mathbf{F}^*(\mathbf{Z},\mathbf{X})$ by solving $\mathbf{P3}$ in polynomial time. Moreover, when given the edge aggregation decision $\mathbf{Z}$, we can substituting $\mathbf{F}$ with $\mathbf{F}^*(\mathbf{X})$ and it is equivalent to solve the edge association sub-problem in $\mathbf{P2}$ only regarding $\mathbf{X}$. The objective function is $\Omega(\mathbf{F}^*(\mathbf{X}))$, abbreviated as $\Omega(\mathbf{X})$.

Next, we prove the objective function $\Omega(\mathbf{X})$ can be transformed into a real-valued set function. Let $\mathcal{G} := \{(m,n)|\forall m \in \mathcal{M}, n \in \mathcal{N}\}$, which builds up a one-to-one mapping between edge association variable x_{mn} and element $u = (m,n) \in \mathcal{G}$. $\mathcal{V} \subseteq \mathcal{G}$ present the selected association pairs of edge server and UE, that is, $\mathcal{V} := \{(m,n)|x_{mn} = 1, \forall m \in \mathcal{M}, n \in \mathcal{N}\}$. Then, we can define the new set function $g(\mathcal{V}) := \Omega(\mathbf{X})$, where $x_{mn} = 1$ iff $(m,n) \in \mathcal{V}$. The objective function in $\mathbf{P2}$ becomes $\mathrm{Min}\ g(\mathcal{V})$.

Definition 1: (**Matroid**) [33] A matroid $\mathcal{U}$ is a tuple $(\mathcal{G},\mathcal{I})$, where $\mathcal{G}$ is a finite ground set and $\mathcal{I} \subseteq 2^{\mathcal{G}}$ is a collection of independent sets, such that: (1) $\mathcal{I}$ is nonempty, in particular, $\emptyset \in \mathcal{I}$; (2) $\mathcal{I}$ is downward closed, *i.e.,* if $\mathcal{V}_2 \in \mathcal{I}$ and $\mathcal{V}_1 \subseteq \mathcal{V}_2$, then $\mathcal{V}_1 \in \mathcal{I}$; (3) if $\mathcal{V}_1, \mathcal{V}_2 \in \mathcal{I}$, and $|\mathcal{V}_1| < |\mathcal{V}_2|$, then $\exists u \in \mathcal{V}_2 \backslash \mathcal{V}_1$, such that $\mathcal{V}_1 \cup \{u\} \subseteq \mathcal{I}$. An independent set $\mathcal{V}_1 \in \mathcal{I}$ with the maximum size is defined as a base of the matroid.

According to the constraint in Eq. (12a), each UE n should be associated to only one edge server. That implies, taking n_1 for example, we cannot select the pairs of edge association (m_1,n_1) and (m_2,n_1) simultaneously in set $\mathcal{V}$. Then, we can define $\mathcal{I}$ as a collection of subsets of $\mathcal{G}$, where each subset

consists of some elements of pairs that any UE is associated to only one edge server. We assume there are no less than two UEs and two edge servers, $N \geq 2$ and $M \geq 2$; otherwise, the problem is trivial. Then, we have the following lemma.

Lemma 2: Given $\mathcal{G} := \{(m,n)|\forall m \in \mathcal{M}, n \in \mathcal{N}\}$, the pair $\mathcal{U} := \{\mathcal{G}, \mathcal{I}\}$ is a matroid, where $\mathcal{I}$ is a collection of independent sets, that is, $\mathcal{I} := \{\mathcal{V}|\mathcal{V} \subseteq \mathcal{G}, \forall u_1 = (m_1, n_1), u_2 = (m_2, n_2) \in \mathcal{V}, n_1 \neq n_2\}$. Constraint (12a) in $\mathbf{P2}$ corresponds to a matroid base constraint, *i.e.,* $\mathcal{V} \in b(\mathcal{U})$, where $b(\mathcal{U})$ is the set of bases of $\mathcal{U}$.

Proof: According to the construction of the pair $\mathcal{U} := \{\mathcal{G}, \mathcal{I}\}$, we can prove by contradiction that $\mathcal{U}$ has the same three properties in Definition 1. We omit the specific process due to the space limitation. Therefore, $\mathcal{U}$ is a matroid. Besides, it is easy to obtain that the size of $\mathcal{U}$ is N since there are N UEs. Considering constraint (12a) in $\mathbf{P2}$, $\sum_{m \in \mathcal{M}} x_{mn} = 1$ means that finding an edge association strategy for each UE n is equal to find a set $\mathcal{V} \subseteq \mathcal{G}$, which constitutes a base of $\mathcal{U}$, $\mathcal{V} \in b(\mathcal{U})$. The lemma is thus proved. ■

Therefore, we can transform the edge association subproblem $\mathbf{P2}$ into the set function minimization problem under the matroid base constraint, that is,

$$(\mathbf{P2}'): \ \text{Min } g(\mathcal{V}), \forall \mathcal{V} \in b(\mathcal{U}). \tag{14}$$

C. Modularity Analysis

Definition 2: (**Nonnegativity, Monotonicity, and Supermodularity**) [34]. Given a finite ground set $\mathcal{G}$, a real-valued set function defined as $g: 2^{\mathcal{G}} \to R$, $g(\cdot)$ is called *nonnegative*, *monotone (nondecreasing)*, and *supermodular* if and only if it satisfies following conditions, respectively.

- $g(\emptyset) = 0$ and $g(\mathcal{V}) \geq 0$ for $\forall \mathcal{V} \subseteq \mathcal{G}$ (*nonnegative*);
- $g(\mathcal{V}_1) \leq g(\mathcal{V}_2)$ for $\forall\ \mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{G}$ (*monotone*);
- $g(\mathcal{V}_1 \cup \{u\}) - g(\mathcal{V}_1) \leq g(\mathcal{V}_2 \cup \{u\}) - g(\mathcal{V}_2), \forall \mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{G}, u \in \mathcal{G} \backslash \mathcal{V}_2$ (*submodular*);

Any function $g(\cdot)$ is said to be supermodular if $-g(\cdot)$ is submodular. Supermodularity has an increasing return property while submodularity captures a diminishing return property, which means that the added value of an element to a bigger set is less than that to a smaller set [34]. Thus, in this part, we attempt to analyze the nonnegativity, monotonicity, and modularity (supermodular or submodular) of $g(\mathcal{V})$.

Firstly, $g(\mathcal{V})$ is ***non-negative*** from the function definition. According to the equivalent expression of the objective function in $\mathbf{P2}$ and Eqs. (9), (10), all coefficients including $L'(\sigma', \sigma)$, e^n_{cmp}, p_n, t^n_{cmp}, τ_{mn}, and $2t^{mm'}_{pro}$ in $\Omega(\mathbf{X})$ are nonnegative. Thus, $g(\mathcal{V})$ is non-negative.

Secondly, $g(\mathcal{V})$ is the ***monotone*** since the expansion of any set $\mathcal{V} \subseteq \mathcal{G}$ will relax item $\mathcal{N}_m$ of Eq. (13) in $\mathbf{P3}$ and increase the optimal objective value potentially. For example, when adding one element $u = (m_u, n_u)$ in $\mathcal{V}$, it is equal to associate UE n_u to edge server m_u, which possibly induces more energy cost (the accumulative energy consumption of all edge associations) and increases the system latency (the maximal one induced by edge association) as well. Then, $\forall \mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{G}, g(\mathcal{V}_2) \geq g(\mathcal{V}_1)$, which proves it is monotone.

Lastly, to discuss the ***modularity (supermodular or submodular)*** of the objective function, we need to compare the marginal increment $g(\mathcal{V}_2 \cup \{u\}) - g(\mathcal{V}_2)$ and $g(\mathcal{V}_1 \cup \{u\}) - g(\mathcal{V}_1)$ for any $\mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{G}$ and element $u \in \mathcal{G} \backslash \mathcal{V}_2$, such that $\mathcal{V}_1 \cup \{u\}$ and $\mathcal{V}_2 \cup \{u\}$ are also feasible. Since $g(\mathcal{V})$ has two different components including energy consumption μE and training latency νT, we analyze these two components separately and derive the property as a whole.

To simplify the notation, we introduce the term $\tau_{mn} = \frac{d_n}{B^{max}_m \log(1 + \frac{h_{mn} p_n}{N_0})}$ and rewrite function $\Omega(\mathbf{X})$ as

$$\begin{aligned}
&\Omega(\mathbf{X}) = \mu E(\mathbf{X}) + \nu T(\mathbf{X}) \\
&= \mu \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} L'(\sigma', \sigma)(e^n_{cmp} + p_n \tau_{mn} |\mathcal{N}_m|) x_{mn} \\
&+ \nu \max_{m \in \mathcal{M}} \max_{n \in \mathcal{N}} \{L'(\sigma', \sigma)(t^n_{cmp} + \tau_{mn} |\mathcal{N}_m|) x_{mn} + 2t^{mm'}_{pro}\}.
\end{aligned} \tag{15}$$

Recall the equivalence relation between $g(\mathcal{V})$ and $\Omega(\mathbf{X})$, we can construct another two set functions $g_1(\mathcal{V})$ and $g_2(\mathcal{V})$ corresponding to $\mu E(\mathbf{X})$ and $\nu T(\mathbf{X})$, respectively.

$$\begin{aligned}
g_1(\mathcal{V}) &= \mu E(\mathbf{X})|_{\forall (m,n) \in \mathcal{V}, x_{mn} = 1; \text{otherwise}, x_{mn} = 0} \\
&= \mu \sum_{(m,n) \in \mathcal{V}} L'(\sigma', \sigma)(e^n_{cmp} + p_n \tau_{mn} |\mathcal{N}_m|),
\end{aligned} \tag{16}$$

$$\begin{aligned}
g_2(\mathcal{V}) &= \nu T(\mathbf{X})|_{\forall (m,n) \in \mathcal{V}, x_{mn} = 1; \text{otherwise}, x_{mn} = 0} \\
&= \nu \max_{(m,n) \in \mathcal{V}} \{L'(\sigma', \sigma)(t^n_{cmp} + \tau_{mn} |\mathcal{N}_m|) + 2t^{mm'}_{pro}\}.
\end{aligned} \tag{17}$$

Thus, we can separate function $g(\mathcal{V})$ as $g(\mathcal{V}) = g_1(\mathcal{V}) + g_2(\mathcal{V})$. For $g_1(\mathcal{V})$, we have the following lemma.

Lemma 3: The first component $g_1(\mathcal{V})$ is supermodular.

Proof: For any new element (pair) $u = (m_u, n_u) \in \mathcal{G} \backslash \mathcal{V}_2$, divide set $\mathcal{V}_1$ into two subsets $\mathcal{V}'_1$ and $\mathcal{V}_1 \backslash \mathcal{V}'_1$, where the first component in $\mathcal{V}'_1$ is equal to m_u and that in $\mathcal{V}_1 \backslash \mathcal{V}'_1$ is not, *e.g.,* $\mathcal{V}'_1 = \{(m_u, n_1), (m_u, n_2), ..., (m_u, n_l)\}$ with $l = |\mathcal{V}'_1|$. Note that the difference between $g_1(\mathcal{V}_1 \cup \{u\})$ and $g_1(\mathcal{V}_1)$ is that $g_1(\mathcal{V}_1 \cup \{u\})$ has the extra edge association between m_u and n_u. We can calculate the difference value.

$$\begin{aligned}
&g_1(\mathcal{V}_1 \cup \{u\}) - g_1(\mathcal{V}_1) = g_1(\mathcal{V}'_1 \cup \{u\}) - g_1(\mathcal{V}'_1) \\
&= \mu L'(\sigma', \sigma)((e^{n_1}_{cmp} + e^{n_2}_{cmp} + ... + e^{n_l}_{cmp} + e^{n_u}_{cmp}) \\
&+ (p_{n_1} \tau_{m_u n_1} + p_{n_2} \tau_{m_u n_2} + ... + p_{n_l} \tau_{m_u n_l} + p_{n_u} \tau_{m_u n_u})(l+1)) \\
&- \mu L'(\sigma', \sigma)((e^{n_1}_{cmp} + e^{n_2}_{cmp} + ... + e^{n_l}_{cmp}) \\
&+ l(p_{n_1} \tau_{m_u n_1} + p_{n_2} \tau_{m_u n_2} + ... + p_{n_l} \tau_{m_u n_l} + p_{n_u} \tau_{m_u n_u})) \\
&= \mu L'(\sigma', \sigma)(e^{n_u}_{cmp} + p_{n_1} \tau_{m_u n_1} + p_{n_2} \tau_{m_u n_2} + ... + p_{n_l} \tau_{m_u n_l} \\
&+ p_{n_u} \tau_{m_u n_u}(l+1)).
\end{aligned} \tag{18}$$

In a similar way, we can also divide set $\mathcal{V}_2$ into two subsets $\mathcal{V}'_2$ and $\mathcal{V}_2 \backslash \mathcal{V}'_2$, where the first component in set $\mathcal{V}'_2$ is equal to m_u and that in $\mathcal{V}_2 \backslash \mathcal{V}'_2$ is not. Owing to $\mathcal{V}_1 \subseteq \mathcal{V}_2$, we have $\mathcal{V}'_1 \subseteq \mathcal{V}'_2$ and $\mathcal{V}'_2$ has more elements with first component equaling to m_u, *e.g.,* $\mathcal{V}'_2 = \{(m_u, n_1), (m_u, n_2), ..., (m_u, n_l), ..., (m_u, n_{l'})\}$, where $l' = |\mathcal{V}'_2| \geq l$. Then, referring to the expression in Eq. (18), we can calculate the difference value as follows.

$$\begin{aligned}
&g_1(\mathcal{V}_2 \cup \{u\}) - g_1(\mathcal{V}_2) = g_1(\mathcal{V}'_2 \cup \{u\}) - g_1(\mathcal{V}'_2) \\
&= \mu L'(\sigma', \sigma)(e^{n_u}_{cmp} + p_{n_1} \tau_{m_u n_1} + p_{n_2} \tau_{m_u n_2} + ... + p_{n_l} \tau_{m_u n_l} \\
&+ ... + p_{n_{l'}} \tau_{m_u n_{l'}} + p_{n_u} \tau_{m_u n_u}(l'+1)).
\end{aligned} \tag{19}$$

Therefore, we compare $g_1(\mathcal{V}_2 \cup \{u\}) - g_1(\mathcal{V}_2)$ with $g_1(\mathcal{V}_1 \cup \{u\}) - g_1(\mathcal{V}_1)$ and have

$$\begin{aligned} &g_1(\mathcal{V}_2 \cup \{u\}) - g_1(\mathcal{V}_2) - (g_1(\mathcal{V}_1 \cup \{u\}) - g_1(\mathcal{V}_1)) \\ &= \mu L'(\sigma', \sigma)(p_{n_{l+1}} \tau_{m_u n_{l+1}} + ... + p_{n_{l'}} \tau_{m_u n_{l'}} + p_{n_u} \tau_{m_u n_u}(l' - l)) \\ &\geq 0. \end{aligned} \tag{20}$$

Noticeably, we can achieve the inequality due to the non-negative coefficients including $\mu L'(\sigma', \sigma)$, p_n, τ_{mn} and $l' \geq l$. Thus, we have the result that $g_1(\mathcal{V})$ is supermodular. ■

Lemma 4: The second component $g_2(\mathcal{V})$ is neither *supermodular* nor *submodular*, which essentially determines the property of $g(\mathcal{V})$.

Referring to the property definition, Lemma 4 can be proved by giving two counter examples to show it is not supermodular and submodular, respectively. We omit the proof process due to the space limitation. Therefore, based on the above two lemmas, we can easily obtain the following theorem.

Theorem 2: The reformulated objective function $g(\mathcal{V}), \mathcal{V} \subseteq \mathcal{G}$ is *nonnegative*, *monotone (nondecreasing)*, but neither *supermodular* nor *submodular*.

To the best of our knowledge, for any set function which is neither submodular nor supermodular, there exists no approximation algorithm for optimization problem under the matroid base constraint. One native solution is to enumerate all edge association subsets and compute their value of the objective function which is infeasible in largescare networks.

V. Solution

In this section, we will construct a supermodular function $\widehat{g}(\mathcal{V})$ with the tight upper bound to approximate the original non-supermodular objective function $g(\mathcal{V})$. On this basis, we transform problem $\mathbf{P2'}$ into $\mathbf{P2''}$ and $\mathbf{P4}$, which falls into the scope of maximizing the submodular function. Finally, we design an approximation algorithm based on the two-stage search strategy and prove it is performance-guaranteed.

A. Latency Function Construction with An Upper Bound

We define $\phi = \nu \max\limits_{m \in \mathcal{M}} \max\limits_{n \in \mathcal{N}} \{L'(\sigma', \sigma) t^n_{cmp} + 2t^{mm'}_{pro}\}$ and $\psi = \nu \min\limits_{m \in \mathcal{M}} \min\limits_{n \in \mathcal{N}} \{L'(\sigma', \sigma) t^n_{cmp} + 2t^{mm'}_{pro}\}$ be the maximum and minimum value of the sum of local computation and model transfer latency, respectively. Then, we construct the following latency function

$$\widehat{T}(\mathbf{X}) = \phi/\nu + L'(\sigma', \sigma) \max_{m \in \mathcal{M}} \max_{n \in \mathcal{N}} \{\tau_{mn}\} \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} x_{mn}, \tag{21}$$

which always has the maximum latency components. We have the corresponding set function $\widehat{g}_2(\mathcal{V})$.

$$\begin{aligned} \widehat{g}_2(\mathcal{V}) &= \nu \widehat{T}(\mathbf{X})|_{\forall (m,n) \in \mathcal{V}, x_{mn} = 1; \text{otherwise}, x_{mn} = 0} \\ &= \phi + \nu L'(\sigma', \sigma) \max_{(m,n) \in \mathcal{G}} \{\tau_{mn}\} |\mathcal{V}|. \end{aligned} \tag{22}$$

If we define $\widehat{\Omega}(\mathbf{X}) = \mu E(\mathbf{X}) + \nu \widehat{T}(\mathbf{X})$, the substituted set function can be constructed as

$$\begin{aligned} \widehat{g}(\mathcal{V}) &= g_1(\mathcal{V}) + \widehat{g}_2(\mathcal{V}) \\ &= \mu E(\mathbf{X}) + \nu \widehat{T}(\mathbf{X})|_{\forall (m,n) \in \mathcal{V}, x_{mn} = 1; \text{otherwise}, x_{mn} = 0} \\ &= \widehat{\Omega}(\mathbf{X})|_{\forall (m,n) \in \mathcal{V}, x_{mn} = 1; \text{otherwise}, x_{mn} = 0}. \end{aligned} \tag{23}$$

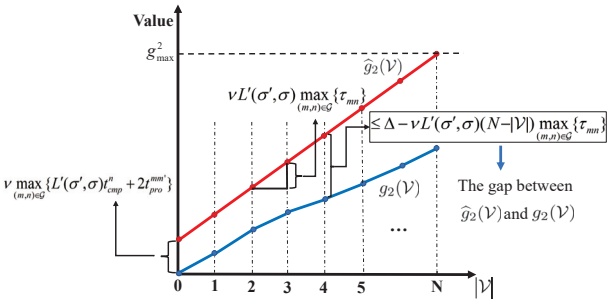

Fig. 3: Gap analysis.

We introduce our method to calculate the value of function $\widehat{g}(\mathcal{V})$: when given any set $\mathcal{V}$, we can easily derive the corresponding $\mathbf{X^0}$ by setting $x_{mn} = 1$ for any $(m, n) \in \mathcal{V}$. Besides, if we have the edge aggregation decision $\mathbf{Z^0}$, *e.g.*, $z_m = 1$, our objective function is equal to $\widehat{\Omega}(\mathbf{F})$ only with respect to variable $\mathbf{F}$. Since we can achieve the value $\Omega(\mathbf{F})$ by solving $\mathbf{P3}$, we can refer to this idea to obtain the value of $\widehat{\Omega}(\mathbf{F})$ as well by solving the following problem $\mathbf{P3'}$.

$$\begin{aligned} (\mathbf{P3'}): \min_{\mathbf{F}} \ & \widehat{\Omega}(\mathbf{F}) = \mu E(f_n) + \nu \widehat{T}(f_n) \\ \text{s.t.} \ & f^{\min}_n \leq f_n \leq f^{\max}_n, \forall n \in \mathcal{N}. \end{aligned} \tag{24}$$

Next, we give the following lemma.

Lemma 5: The set function $\widehat{g}(\mathcal{V})$ is *nonnegative*, *monotone* and *supermodular*. Furthermore, define

$$\Delta = \phi - \psi + \nu L'(\sigma', \sigma)(N \max_{(m,n) \in \mathcal{G}} \{\tau_{mn}\} - \min_{(m,n) \in \mathcal{G}} \{\tau_{mn}\}),$$

the gap between $\widehat{g}(\mathcal{V})$ and $g(\mathcal{V})$ is bounded as follows.

$$g(\mathcal{V}) \leq \widehat{g}(\mathcal{V}) \leq g(\mathcal{V}) + \Delta, \forall \mathcal{V} \subseteq \mathcal{G}. \tag{25}$$

Proof: $\widehat{g}(\mathcal{V})$ is nonnegative according to the definition in Eqs. (16) and (22). Since $g_1(\mathcal{V})$ is monotone and supermodular, we just need prove $\widehat{g}_2(\mathcal{V})$ is monotone and supermodular.

For any adding new element $u = (m_u, n_u)$ to $\mathcal{V}$, which means $x_{m_u n_u} = 1$, it incurs the constant marginal increment $\nu L'(\sigma', \sigma) \max\limits_{(m,n) \in \mathcal{G}} \{\tau_{mn}\}$ referring to the definition in Eq. (22). Naturally, $\widehat{g}_2(\mathcal{V})$ is a linear increasing function which can be regarded as a monotone supermodular function as well.

To find the gap between $\widehat{g}(\mathcal{V})$ and $g(\mathcal{V})$, we present the function curve of both $\widehat{g}_2(\mathcal{V})$ and $g_2(\mathcal{V})$ in Fig. 3. Recall the expression for $g_2(\mathcal{V})$ and $\widehat{g}_2(\mathcal{V})$ in Eqs. (17) and (22), we can easily derive $\widehat{g}_2(\mathcal{V}) \geq g_2(\mathcal{V})$ for any input $\mathcal{V}$. This is because $\widehat{g}_2(\mathcal{V})$ always has the maximal computation and model transfer latency ϕ in the first component, and has the maximal communication latency $\nu L'(\sigma', \sigma) \max\limits_{(m,n) \in \mathcal{G}} \{\tau_{mn}\} |\mathcal{V}|$ for any input $\mathcal{V}$ in the second component. Thus, we can always have $\widehat{g}_2(\mathcal{V}) \geq g_2(\mathcal{V})$.

When $\mathcal{V} = \emptyset$, the difference between $\widehat{g}_2(\mathcal{V})$ and $g_2(\mathcal{V})$ is

$$\begin{aligned} \Delta_{\mathcal{V}} &= \widehat{g}_2(\mathcal{V})_{|\mathcal{V}|=0} - g_2(\mathcal{V})_{|\mathcal{V}|=0} \\ &= \nu \max_{(m,n) \in \mathcal{G}} \{L'(\sigma', \sigma) t^n_{cmp} + 2t^{mm'}_{pro}\} = \phi. \end{aligned} \tag{26}$$

When $\mathcal{V}$ is not an empty set that $|\mathcal{V}| > 0, \mathcal{V} \in b(\mathcal{U})$, we have $g_2(\mathcal{V})_{|\mathcal{V}|=N} \geq \psi + \nu L'(\sigma', \sigma) \min\limits_{(m,n) \in \mathcal{G}} \{\tau_{mn}\}$ and

$$
\begin{aligned}
\Delta_{\mathcal{V}} &= \widehat{g}_2(\mathcal{V}) - g_2(\mathcal{V}) \\
&\leq \phi + \nu L'(\sigma', \sigma)|\mathcal{V}| \max_{(m,n)\in\mathcal{G}} \{\tau_{mn}\} \\
&- (\psi + \nu L'(\sigma', \sigma) \min_{(m,n)\in\mathcal{G}} \{\tau_{mn}\}) \\
&= \phi - \psi + \nu L'(\sigma', \sigma)(|\mathcal{V}| \max_{(m,n)\in\mathcal{G}} \{\tau_{mn}\} - \min_{(m,n)\in\mathcal{G}} \{\tau_{mn}\}) \\
&= \Delta - (N - |\mathcal{V}|) \max_{(m,n)\in\mathcal{G}} \{\tau_{mn}\}. \qquad (27)
\end{aligned}
$$

Obviously, when $|\mathcal{V}| = N$, the difference between $\widehat{g}_2(\mathcal{V})$ and $g_2(\mathcal{V})$ is bounded by Δ. We can achieve $\widehat{g}(\mathcal{V}) \leq g(\mathcal{V}) + \Delta$ for any $\mathcal{V} \in b(\mathcal{U})$ to prove the above lemma. ■

Therefore, by replacing $g(\mathcal{V})$ with $\widehat{g}(\mathcal{V})$, we can formulate the non-negative monotone supermodular minimization problem with a single matroid base constraint, that is,

$$
(\mathbf{P2}'') : \ \text{Min } \widehat{g}(\mathcal{V}) \ , \forall \mathcal{V} \in b(\mathcal{U}). \qquad (28)
$$

B. Local Search Algorithm

According to Lemma 5, $-\widehat{g}(\mathcal{V})$ is non-monotone and submodular. There exist approximation solutions for maximizing a non-negative non-monotone submodular function over bases of a matroid [35]. To apply that solution, we need to transform the objective function into an appropriate non-negative function, *i.e.,* $h(\mathcal{V}) := g_{max} - \widehat{g}(\mathcal{V})$, where g_{max} is the upper bound of $\widehat{g}(\mathcal{V})$ and $h(\mathcal{V}) \geq 0$. Since $g_1(\mathcal{V})$ and $\widehat{g}_2(\mathcal{V})$ achieve the maximum at $\mathcal{V} = \mathcal{G}$, we can define $g^1_{max} := g_1(\mathcal{G}) = \sum_{(m,n)\in\mathcal{G}} L'(\sigma', \sigma)(e^n_{cmp} + p_n \tau_{mn} N)$ and $g^2_{max} := \widehat{g}_2(\mathcal{G}) = \phi + \nu L'(\sigma', \sigma) \max_{(m,n)\in\mathcal{G}} \{\tau_{mn}\} MN$, respectively. Then, we can construct $g_{max} = g^1_{max} + g^2_{max}$ and calculate the value of function $h(\mathcal{V})$. Since minimizing $\widehat{g}(\mathcal{V})$ equals to maximizing $h(\mathcal{V})$, we can transform problem $\mathbf{P2}''$ into $\mathbf{P4}$ as follows.

$$
(\mathbf{P4}) : \ \text{Max } h(\mathcal{V}), \forall \mathcal{V} \in b(\mathcal{U}). \qquad (29)
$$

For the above set function maximization problem, we can refer to the spirit in [35] and design an approximation algorithm in Algorithm 1. The core idea is to loop through all edge servers for edge aggregation selection and compare with the value of $h(\widetilde{\mathcal{V}}_m)$ which can be calculated by executing the two-stage search procedure in Algorithm 2. We then obtain the edge aggregation decision $\mathbf{Z}$, edge association strategy $\mathbf{X}$, and computation resource allocation $\mathbf{F}$ with the maximum $h(\widetilde{\mathcal{V}}_{m'})$.

The two-stage search strategy devotes to decide the output $\mathcal{V}$. In **Stage One** (steps 1-10), we initialize $\mathcal{V}_1$ satisfying the base constraint, *e.g.*, each UE randomly selects an edge server for association in order. We calculate $\widehat{g}(\mathcal{V}_1)$ and obtain the value of $h(\mathcal{V}_1)$. Then, we exploit the local search method only based on the exchange operations to find the updated base $\mathcal{V}_1$, such that the value of $h(\mathcal{V}_2)$ can be increased by a factor of at least $1 + \frac{\varepsilon}{N^4 M^4}$ at each iteration. In **Stage Two** (steps 11-25), we initialize a singleton set $\mathcal{V}_2 = (m_s, n_s) \subseteq \mathcal{G} \backslash \mathcal{V}_1$ with the maximum value $h(\mathcal{V}_2)$, equals the minimum $\widehat{g}(\{(m_s, n_s)\})$. We then run a local search on $\mathcal{G} \backslash \mathcal{V}_1$ using both deletion and exchange operations to obtain an independent set $\mathcal{V}_2 \subseteq \mathcal{G} \backslash \mathcal{V}_1$. After the above process, we compute two disjoint bases of $\mathcal{U}' = \mathcal{U} \backslash \{\mathcal{V}_2\}$, *i.e.*, b_1 and b_2. Thus, we actually obtain three different bases of $\mathcal{U}$, *i.e.*, $\mathcal{V}_1, \mathcal{V}_2 \cup b_1, \mathcal{V}_2 \cup b_2$. We select the best one of the three bases that maximizes the value $h(\mathcal{V})$ and return the result. Last, the algorithm outputs the parameter server decision $\mathbf{Z}$ and $\mathbf{X}$ according to the chosen base.

Algorithm 1: Proposed Algorithm for Problem EARTH

Input: Edge server set $\mathcal{M}$, UE set $\mathcal{N}$, ground set $\mathcal{G} := \{(m,n) | \forall m \in \mathcal{M}, n \in \mathcal{N}\}$, matroid $\mathcal{U} := \{\mathcal{G}, \mathcal{I}\}$, constant $\epsilon > 0$.
Output: $\mathbf{Z}, \mathbf{X}, \mathbf{F}$.

1 Initialize: Let $\mathbf{Z} = \mathbf{0}$, $\mathbf{X} = \mathbf{0}$.
2 **for** $m = 1, 2, ..., M$ **do**
3 $z_m = 1$.
4 Execute the two-stage search procedure with returned $\widetilde{\mathcal{V}}_m$ and $h(\widetilde{\mathcal{V}}_m)$.
5 **end**
6 $m' \leftarrow \arg\max_{m\in\mathcal{M}} h(\widetilde{\mathcal{V}}_m)$.
7 Set $z_{m'} = 1$, $x_{mn} = 1$ and obtain $\mathbf{F}$ by solving $\mathbf{P3}'$ $\forall (m,n) \in \widetilde{\mathcal{V}}_{m'}$.
8 Return $\mathbf{Z}$, $\mathbf{X}$, $\mathbf{F}$.

Algorithm 2: Two-Stage Search for Edge Association

Input: Edge aggregation decision $\mathbf{Z}$, ground set $\mathcal{G} := \{(m,n) | \forall m \in \mathcal{M}, n \in \mathcal{N}\}$, matroid $\mathcal{U} := \{\mathcal{G}, \mathcal{I}\}$, constant $\epsilon > 0$.
Output: Edge association set $\widetilde{\mathcal{V}}$, value of $h(\widetilde{\mathcal{V}})$.

1 **Stage one:**
2 Initialize $\mathcal{V}_1$ with an arbitrary base of matroid $\mathcal{U}$.
3 Obtain the value of $h(\mathcal{V}_1)$, $h(\mathcal{V}_1) = g_{max} - \widehat{g}(\mathcal{V}_1)$.
4 **while** *1* **do**
5 **if** $\exists\ (m,n) \in \mathcal{G} \backslash \mathcal{V}_1$ *and* $(m', n') \in \mathcal{V}_1$ *such that* $\mathcal{V}_1{}' = \mathcal{V}_1 \backslash \{(m', n')\} \cup \{(m,n)\}$ *is a base of* $\mathcal{U}$ *and* $\widehat{h}(\mathcal{V}_1{}') > (1 + \frac{\epsilon}{N^4 M^4}) h(\mathcal{V}_1)$ **then**
6 $\mathcal{V}_1 \leftarrow \mathcal{V}_1{}'$.
7 break
8 **end**
9 **end**
10 **Stage two:**
11 Let $(m_s, n_s) \subseteq \mathcal{G} \backslash \mathcal{V}_1$ be a singleton set with the minimum value $\widehat{g}(\{(m_s, n_s)\})$.
12 Obtain the value of $h(\{(m_s, n_s)\}) = g_{max} - \widehat{g}(\{(m_s, n_s)\})$
13 Initialize $\mathcal{V}_2 = \{(m_s, n_s)\}$.
14 **while** *1* **do**
15 **if** $\exists\ (m,n) \in \mathcal{V}_2$, $\mathcal{V}_2{}' = \mathcal{V}_2 \backslash \{(m,n)\}$, *such that* $h(\mathcal{V}_2{}') \geq (1 + \frac{\epsilon}{N^4 M^4}) h(\mathcal{V}_2)$ **then**
16 $\mathcal{V}_2 \leftarrow \mathcal{V}_2{}'$.
17 **end**
18 **else if** $\exists\ (m,n) \in (\mathcal{G} \backslash \mathcal{V}_1) \backslash \mathcal{V}_2$, $(m', n') \in \mathcal{V}_2 \cup \{\emptyset\}$ *such that* $\mathcal{V}_2{}' = \mathcal{V}_2 \backslash \{(m', n')\} \cup \{(m,n)\} \in \mathcal{I}$ *and* $h(\mathcal{V}_2{}') > (1 + \frac{\epsilon}{N^4 M^4}) h(\mathcal{V}_2)$ **then**
19 $\mathcal{V}_2 \leftarrow \mathcal{V}_2{}'$
20 **end**
21 **else**
22 break
23 **end**
24 **end**
25 Let $\mathcal{U}' = \mathcal{U} \backslash \{\mathcal{V}_2\}$. Compute two disjoint bases of $\mathcal{U}'$, *i.e.*, b_1 and b_2.
26 Choose $\widetilde{\mathcal{V}} = \arg\max_{\mathcal{V} = \mathcal{V}_1, \mathcal{V}_2 \cup b_1, \mathcal{V}_2 \cup b_2} h(\mathcal{V})$ and return.

C. Theoretical Analysis

Theorem 3: Let $\widetilde{\mathbf{Z}}, \widetilde{\mathbf{X}}, \widetilde{\mathbf{F}}$ be the output of Algorithm 1. Let $\mathbf{Z}^*, \mathbf{X}^*, \mathbf{F}^*$ be the optimal solution of $\mathbf{P1}$. Then, we have

$$
\Omega(\widetilde{\mathbf{Z}}, \widetilde{\mathbf{X}}, \widetilde{\mathbf{F}}) \leq \left(\frac{1}{6} - \epsilon\right) \Omega(\mathbf{Z}^*, \mathbf{X}^*, \mathbf{F}^*) + \left(\frac{1}{6} - \epsilon\right) \Delta + \left(\frac{5}{6} + \epsilon\right) g_{max}. \qquad (30)
$$

The time complexity is $O(\frac{1}{\epsilon} N^4 M^5 \log(NM - N) + NM^2)$

Proof: **For the performance guarantee:** The theorem is a corollary of Theorem 5.2 in [35], which proves that there

exists a $(\frac{1}{6}-\epsilon)$-approximation algorithm for maximizing any non-negative submodular function over bases of a matroid. Recall that Algorithm 1 actually designs an approximation algorithm for **P4**, we have $g_{\max}-\widehat{g}(\widetilde{\mathcal{V}}) \geq (\frac{1}{6}-\epsilon)(g_{\max}-\widehat{g}(\mathcal{V}^*))$. Thus, we can derive

$$\begin{aligned} g(\widetilde{\mathcal{V}}) \leq \widehat{g}(\widetilde{\mathcal{V}}) &\leq \left(\frac{1}{6}-\epsilon\right)\widehat{g}(\mathcal{V}^*)+\left(\frac{5}{6}+\epsilon\right)g_{max} \\ &\leq \left(\frac{1}{6}-\epsilon\right)(g(\mathcal{V}^*)+\Delta)+\left(\frac{5}{6}+\epsilon\right)g_{max} \\ &\leq \left(\frac{1}{6}-\epsilon\right)g(\mathcal{V}^*)+\left(\frac{1}{6}-\epsilon\right)\Delta+\left(\frac{5}{6}+\epsilon\right)g_{max}. \end{aligned}$$

Remember the equivalence relation $g(\mathcal{V}^*) \leftrightarrow \Omega(\mathbf{X}^*) \leftrightarrow \Omega(\mathbf{Z}^*, \mathbf{X}^*, \mathbf{F}^*)$, $g(\widetilde{\mathcal{V}}) \leftrightarrow \Omega(\widetilde{\mathbf{X}}) \leftrightarrow \Omega(\widetilde{\mathbf{Z}}, \widetilde{\mathbf{X}}, \widetilde{\mathbf{F}})$ for any given $\mathbf{Z}$ and the optimal $\mathbf{F}^*(\mathbf{X})$, and Algorithm 1 always selects the maximal return $\widetilde{\mathcal{V}}_{m'}$, then we can achieve Eq. (30).

We then analyze the time complexity as follows. Firstly, looping through all edge servers in Algorithm 1 consumes M operations. Secondly, when conducting the two-stage search in Algorithm 2, steps 2-3 construct an arbitrary base $\mathcal{V}_1$ that consumes $O(N)$ operations. Steps 4-10 search a locally optimal base by swap operations that consumes $N(M-1) = O(NM)$ at most. Steps 11-13 initialize $\mathcal{V}_2$ needs at most $N(M-1) = O(NM)$ operations, due to $\mathcal{G}\backslash\mathcal{V}_1 = NM - N$. In steps 14-24, the number of operations is at most $\log_{1+\frac{\varepsilon}{N^4M^4}} \frac{OPT(\mathcal{G}\backslash\mathcal{V}_1)}{OPT(\mathcal{G}\backslash\mathcal{V}_1)/(NM-N)} = O(\frac{1}{\varepsilon}N^4M^4\log(NM-N))$. Finally, step 25 computes two disjoint bases for matroid $\mathcal{U}'$ that needs $O(N)+O(NM) = O(NM)$ operations. Step 26 takes a constant time to find the best one of the three bases. To summarize, the running time is $M(O(N)+O(NM)+O(NM)+O(\frac{1}{\varepsilon}N^4M^4\log(NM-N))+O(NM)) = O(\frac{1}{\varepsilon}N^4M^5\log(NM-N)+NM^2)$, which is polynomial of the problem size. ■

VI. Simulation Evaluation

In this section, we conduct extensive simulations under different settings to verify the performance of the proposed algorithm (labeled as "EARTH" in the figures).

A. Evaluation Setup

We assume all the UEs and edge servers are randomly distributed in the 500 m $\times$500 m area. If no otherwise stated, the number of UEs, edge servers are 50 and 10, respectively. Referring to the setting in [23], the effective computation capacity f_n of each UEs is within $[1, 10]$ GHz. The data transmission power of each UE and edge server are set as $p_n = 0.2$ W and $p_m = 0.3$ W, respectively. The maximum bandwidth of each edge server is the same as $B_m^{max} = B_l^{max} = 2$ MHz. We set the channel gain $h_{mn} = [-50, -10]$ dBm and the noise power $N_0 = -90$ dBm [36]. The parameters setting of FL task is based on the work in [19]. We suppose the collected data size D_n of each UE uniformly distributed in $[5, 10]$ MB and the number of CPU cycles required per bit is in $[40, 100]$ cycles/bit. The uploaded model size is assumed to a constant as 4.5 MB. We also set the typical parameters as $\sigma = \sigma' = 0.1$, $L(\sigma) = L'(\sigma', \sigma) = 5$, $t_{pro}^{mm'} = [5-10]$ s, $\mu = \nu = 0.5$.

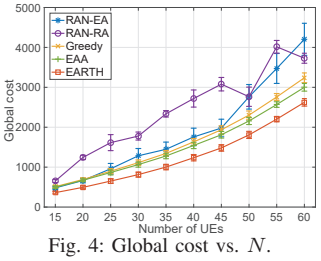

Fig. 4: Global cost vs. N.

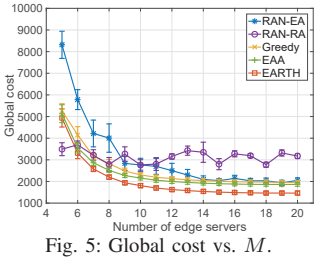

Fig. 5: Global cost vs. M.

B. Baseline Setup

We involve five benchmark algorithms for comparison. **Random Edge Association (RAN-EA)**: each edge server m randomly selects a set of UEs for model aggregation. **Random Resource Allocation (RAN-RA)**: it randomly determined the computation capacity between f_n^{min} and f_n^{max}. **Greedy**: each UE n greedily selects the connected edge server m in an ascending order with the maximum communication bandwidth that can be allocated. **EAA**: it is an algorithm proposed in [23], which iteratively adjusts edge association strategy using the transferring and exchanging method. **Brute-Force**: it utilizes the exhaustive search method to find the optimal solution. Since it cannot be applied in largescare networks, we only show its results under the small network scale in Table I.

C. Evaluation Results

1) *Impact of network size*. As shown in Fig. 4, the global cost has a growth trend when increasing the number of UEs from 15 to 60. This is because more participants increase the accumulated energy cost in local model updating and also lengthen the model uploading time with the diminished allocated bandwidth. Our algorithm performs more cost-efficient and can reduce the global cost by 31.36%, 49.18%, 24.62% and 20.49%, compared with four benchmarks (RAN-EA, RAN-RA, Greedy, and EAA).

Fig. 5 depicts that the global cost gradually decreases when the number of edge servers increases from 5 to 20. The reason is that more edge servers provide UEs more chances with better channel quality for local model uploading and UEs can be allocated more bandwidth to reduce the uploading cost. It is worth noting that when the number of edge servers is 5, the global cost of RAN-RA is lower than EARTH. This is because the random computation frequency might induce a feasible solution closer to the optimal solution than EARTH. Finally, our algorithm reduces the overall cost by 33.04%, 37.06%, 22.36%, and 16.65%, compared with four benchmarks.

2) *Impact of maximum bandwidth B_m^{max}*. Fig. 6 illustrates that the global cost goes down with the increase of bandwidth. This is because the higher bandwidth speeds up the transmission rate to shorten the model uploading time and allow the lower computation frequency. In addition, our algorithm can reduce the overall cost by 36.22%, 32.25%, 22.25% and 15.87%, compared with four benchmarks.

3) *Impact of local data size D_n*. As shown in Fig. 7, we find that the larger size of local data would increase the global cost for all algorithms. We account for this result that the large amount of training data in UEs raises both the model

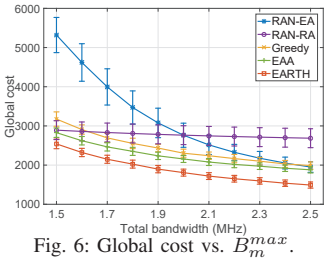

Fig. 6: Global cost vs. B_m^{max}.

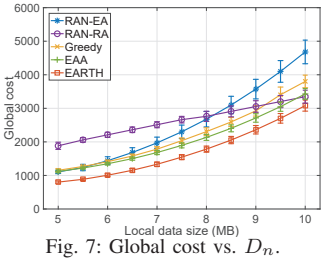

Fig. 7: Global cost vs. D_n.

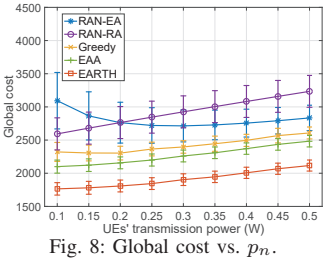

Fig. 8: Global cost vs. p_n.

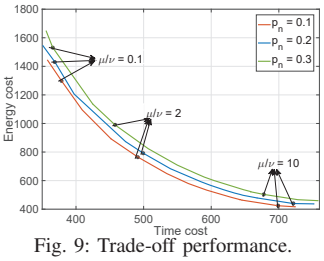

Fig. 9: Trade-off performance.

TABLE I: The near-optimality performance.

Algorithm	Total bandwidth (MHz)										
	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
Brute-Force	217.59	209.62	203.14	197.59	192.73	188.71	185.15	181.97	179.20	176.68	174.38
EARTH	218.55	211.16	204.82	198.22	193.30	188.96	185.15	183.10	180.05	176.68	174.38
Difference	0.44%	0.73%	0.83%	0.32%	0.30%	0.13%	0	0.62%	0.47%	0	0

training latency and cost in local model update process. When the data size increases from 3 MB to 8 MB, our algorithm can reduce the overall cost by 31.99%, 38.18%, 24.36% and 19.01%, compared with four benchmarks.

4) *Impact of UEs' transmission power* p_n. Fig. 8 depicts that there is an growing trend in the global cost when increasing the transmission power from 0.1W to 0.5W. This is because providing the larger transmission power would increase the model uploading cost. Admittedly, the propagation delay can be shortened when improving the transmission power. Nevertheless, its impact on energy consumption weights over the latency saving. In this case, our proposed algorithm still has better performance, reduce the over cost by 31.61%, 34.38%, 21.02% and 15.68% compared with four benchmarks.

5) *Impact of weighted coefficients* μ, ν. In general, there is a conflict between the goals of minimize the energy cost E and the training latency T. We show the trade-off curves in Fig. 9 and find that when μ/ν increases, our curves have a descending trend. This is because a smaller μ/ν means the larger coefficient value of training latency than that of energy cost, which implies the latency important weight plays a leading role in global cost reduction, and vice versa. In addition, our curve is more efficient (means lower energy cost and latency) when the transmission power becomes smaller.

6) *Comparison with the optimal results*. We also compare the performance between our algorithm and the optimal "Brute-Force" algorithm using exhaustive searches for edge association decisions under the small network scale. We set $N = 8$, $M = 3$ and present the comparison result in Table I when varying the maximum communication bandwidth. EARTH performs closely to Brute-Force and even the same when the total bandwidth is 2.1, 2.4 and 2.5 MHz, respectively. The average difference between them is 0.35% of the optimal overhead, which shows the suboptimality.

VII. Field Experiment

To further evaluate our proposed algorithm, we conduct field experiments in Fig. 10. Our testbed consists of several computing modules such as NVDIA Jetson Nano, TX2, where 3 TX2 modules act as edge servers and the other modules serve as 8 UEs, and three routers act as the wireless access point,

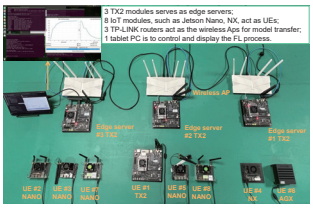

Fig. 10: Experimental scenario.

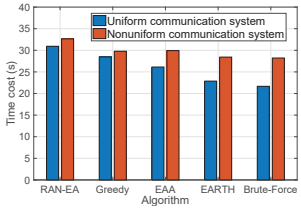

Fig. 11: Total training time in different communication systems.

and a tablet PC. These 8 UEs apply MNIST as the dataset and the learning model is a two-layer DNN.

We consider both the uniform and nonuniform interconnections among edge servers in this small-scale network. For the uniform interconnection, the propagation latency between edge servers are the same and negligible in practice, which is expressed by $t_{pro}^{mm'} = 0$. For the nonuniform interconnection, we can set different timers for TX2 devices to simulate the latency of edge model transfer between edge servers, that is $t_{pro}^{12} = t_{pro}^{21} = 1s, t_{pro}^{13} = t_{pro}^{31} = 5s, t_{pro}^{23} = t_{pro}^{32} = 7s$.

For ease of measurement, we set $\mu = 0$, $\nu = 1$ to compare the system performance in terms of training latency. As shown in Fig. 11, our EARTH algorithm always has a better latency performance when comparing with RAN-EA, Greedy, and EAA. Besides, the training latency in the nonuniform communication system is larger because it spends more time for edge model transmission. Meanwhile, our results are very close to the optimal results and the differences compared to Brute-Force are about 0.26% and 0.76% (0.51% on average) for two different communication systems, respectively.

VIII. Conclusion

To facilitate the implementation of edge AI, we consider the global model aggregation over multiple cells happens at the edge. Then, the joint edge aggregation and association problem is investigated to minimize the total cost. After the complexity analysis, we split into multiple edge association subproblems and transform it into an equivalent set function optimization problem under a matroid base constraint. We then devise a substitute supermodular function with bounded gap and propose an approximation algorithm using the two-stage search strategy. Simulation and field experiments results show that our algorithm has both the superiority and near-optimality.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[2] S. Din and A. Paul, "Smart health monitoring and management system: Toward autonomous wearable sensing for internet of things using big data analytics," *Future Generation Computer Systems*, vol. 91, no. FEB., pp. 611–619, 2018.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. AISTATS*, pp. 1273–1282, 2017.

[4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. MLSys*, 2020.

[5] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE ICC*, 2020, pp. 1–6.

[6] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[7] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th AICMAS*, 2021, pp. 54–66.

[8] 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 Transactions on Networking*, vol. PP, no. 99, pp. 1–12, 2020.

[9] L. Yang, Y. Lu, J. Cao, J. Huang, and M. Zhang, "E-tree learning: A novel decentralized model learning framework for edge ai," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 290–11 304, 2021.

[10] L. Yang, Y. Gan, J. Cao, and Z. Wang, "Optimizing aggregation frequency for hierarchical model training in heterogeneous edge computing," *IEEE Transactions on Mobile Computing*, 2022.

[11] X. Zhou, W. Liang, J. She, Z. Yan, I. Kevin, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5308–5317, 2021.

[12] Y. Song, T. Wang, Y. Wu, L. Qian, and Z. Shi, "Non-orthogonal multiple access assisted federated learning for uav swarms: An approach of latency minimization," in *Proc. IEEE IWCMC*, 2021, pp. 1123–1128.

[13] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive mimo for wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6377–6392, 2020.

[14] W. Xia, W. Wen, K.-K. Wong, T. Q. Quek, J. Zhang, and H. Zhu, "Federated-learning-based client scheduling for low-latency wireless communications," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 32–38, 2021.

[15] K. Wei, J. Li, C. Ma, M. Ding, C. Chen, S. Jin, Z. Han, and H. V. Poor, "Low-latency federated learning over wireless channels with differential privacy," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 290–307, 2021.

[16] D. Chen, C. S. Hong, L. Wang, Y. Zha, Y. Zhang, X. Liu, and Z. Han, "Matching-theory-based low-latency scheme for multitask federated learning in mec networks," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 415–11 426, 2021.

[17] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[18] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE ICC Workshops*, 2020, pp. 1–6.

[19] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[20] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110–124, 2021.

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

[22] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[23] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "Hfel: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6535–6548, 2020.

[24] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[25] Y. Shen, Y. Qu, C. Dong, F. Zhou, and Q. Wu, "Joint training and resource allocation optimization for federated learning in uav swarm," *IEEE Internet of Things Journal*, 2022.

[26] Y. Qu, D. Chao, W. Tao, Z. Yan, D. Haipeng, and F. Wu, "Efficient edge intelligence under clustering for uav swarm networks," in *Proc. IEEE International Conference on Space-Air-Ground Computing (SAGC)*, 2021.

[27] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, 2017.

[28] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 IEEE IJCNN*, 2020, pp. 1–9.

[29] C. Ma, J. Konečnỳ, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[30] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 398–401, 2017.

[31] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2018.

[32] M. Charikar, S. Guha, éva Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *Journal of Computer and System Sciences*, vol. 65, no. 1, pp. 129–149, 2002.

[33] J. G. Oxley, "Matroid theory (oxford graduate texts in mathematics)," *Oxford University Press, Inc.*, 2006.

[34] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-i," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[35] J. Lee, R. S. Mirrokni, R. Nagarajan, and R. Sviridenko, "Maximizing nonmonotone submodular functions under matroid or knapsack constraints," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. p.2053–2078, 2010.

[36] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "Uav-enabled mobile edge computing: Offloading optimization and trajectory design," in *IEEE ICC*, 2018, pp. 1–6.