Bandwidth Allocation for Federated Learning with Wireless Providers and Cost Constraints

Jiahao Xue, *Student Member, IEEE,* Zhe Qu, *Member, IEEE,* Jie Xu, *Senior Member, IEEE,* Yao Liu, *Senior Member, IEEE,* Zhuo Lu, *Senior Member, IEEE,*

Abstract—Federated learning (FL) trains a global learning model by using a central server to collaborate with multiple decentralized clients. In a wireless network, the data transmission latency between a client and the FL server is substantially affected by signal quality dynamics and bandwidth allocation. FL clients require synchronized communication at each round to update their models simultaneously, which makes bandwidth allocation methods for conventional wireless tasks infeasible to use. Existing bandwidth allocation studies for FL mainly focused on allocating bandwidth of one bandwidth provider without cost. In this paper, we consider a more practical and challenging problem: how to assign the bandwidth to clients under multiple wireless providers to minimize the FL round length (i.e., the latency that FL finishes one round of model training and updating) with bandwidth capability and cost constraints? We propose a model that maps the problem into a new variant of the knapsack problem, called multi-dimensional max-min multiple knapsacks (MDM^{3}KP). Based on MDM^{3}KP, we create an iterative solution to find the client assignment and bandwidth allocation that minimizes the FL round length. Comprehensive simulation results show that the solution reduces the FL round length by up to 70.8% compared with other benchmarks.

Index Terms—Bandwidth allocation, wireless network, federated learning.

1 Introduction

The development of wireless networks enables more machine learning applications to the mobile computing. As a decentralized approach to machine learning, federated learning (FL) coordinates clients via a centralized server to work together towards training a global machine learning model [1]. However, deploying FL as the application on wireless network confronts dynamic wireless environment and limited wireless resources [2]. It urgently motivates people to develop novel wireless resource management strategies for FL.

Most existing wireless resource allocation algorithm (e.g., [3], [4] for video buffering network) have been designed for conventional wireless network systems. They cannot be used in a network that supports FL. Specifically, at each round of FL, a client involves downloading the global model, computing the new local model, and uploading the local model to the server that computes the new global model for the next round [1], [5]. All clients must be synchronized towards updating the global model and one round ends only when the slowest client finishes its job. This is a unique requirement to design efficient communication mechanisms for FL [6]–[8]. A way to improve the communication efficiency is to improve the uploading and downloading efficiency [9]. In wireless networking, a client with low signal-to-noise ratio (SNR) suffers a low transmission rate [10], and thus spends a longer time on model downloading and uploading. An intuitive way to improve the transmission efficiency is to allocate more bandwidth to the client. However, the total available bandwidth of the wireless bandwidth provider is limited. Though some studies propose to use asynchronous FL to avoid the slowest client, it requires larger computational demand on both clients and central server [11]. Asynchronous FL usually assumes bounded delay, which is impractical [12]. Therefore, we aim to design a wireless communication algorithm for synchronous FL. Given the total limited bandwidth, it is necessary to allocate a bandwidth to each client and avoid let some clients waste time to wait for synchronization.

Although existing studies have taken the bandwidth into consideration, they generally assume that there is one wireless bandwidth provider to allocate the bandwidth to the clients [9], [13], [14]. In many real-world scenarios, bandwidth providers are usually base stations or wireless access points, which can be owned by multiple wireless service providers (e.g., Google Fi, a mobile virtual network operator, can automatically switch between multiple carriers' facilities [15]) and have different bandwidth capabilities (e.g., WiFi 5/6 and 4/5G) and cost constraints (e.g., the monetary cost of data usage). It is important to consider these practical wireless resource conditions to maximize the transmission efficiency. A common performance metric for FL is the round length, which is defined as the total time to finish one round of FL (including the computational time and the communication time in one FL round). In this regard, the communication time due to information exchange of FL critically depends on the bandwidth allocaton in wireless networking, which is our focus in this paper.

- *J. Xue and Z. Lu are with the Department of Electrical Engineering, University of South Florida, Tampa, 33620, USA. E-mail: {jiahao, zhuolu}usf.edu.*
- *Z. Qu is with the School of Computer Science and Engineering, Central South University, Changsha, 410083, CN. E-mail: zhe_qu@csu.edu.cn.*
- *J. Xu is with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, 33146, USA. E-mail: jiexu@miami.edu.*
- *Y. Liu is with the Department of Computer Science and Engineering, University of South Florida, Tampa, 33620, USA. E-mail: yliu@cse.usf.edu.*

Despite substantial efforts [16]–[18] that have been focused on wireless resource allocation across different providers, most of them do not consider the problem of dynamic bandwidth allocation to minimize the FL round length. In particular, these existing methods were designed for conventional tasks over wireless networking and do not consider the synchronized communication model under FL. Though the cost budget can limit bandwidth usage, existing methods do not consider the cost constraints [19], [20]. As a result, a more practical and general problem remains to be fully studied: how to assign the bandwidth to clients under multiple wireless providers to minimize the FL round length under bandwidth capability and cost constraints? The problem consists of two coupled parts: (i) which client is assigned to which provider and (ii) How much bandwidth should be given to a client by a provider.

In this paper, we focus on the wireless resource allocation problem under an FL over wireless network scenario by designing efficient client bandwidth allocation under multiple bandwidth providers with bandwidth and cost constraints. We study how clients should be assigned to bandwidth providers and how much bandwidth should they get in order to minimize the FL round length. We show that the straightforward formulation of the problem leads to a complicated, non-linear, non-convex, combinatorial problem that is NP-hard (in contrast to the known bandwidth allocation problem under one provider that can be solved efficiently by a bisection algorithm in logarithmic time [14]). We propose a model that maps this optimization problem into a new variant of the knapsack problem, called multi-dimensional max-min multiple knapsacks (MDM^{3}KP). Based on MDM^{3}KP, we create an iterative solution to find the client assignment and bandwidth allocation that minimizes the FL round length. We evaluate the proposed MDM^{3}KP solution via comprehensive simulations and demonstrate that under various wireless network settings, the proposed solution is able to reduce the FL round length by up to 67.2% compared with other bandwidth allocation algorithms.

Our main contributions are summarized as follows.

- We formulate the client bandwidth allocation problem under multiple wireless bandwidth providers and take into consideration the bandwidth cost constraints in practical wireless FL scenarios.
- We map the bandwidth allocation problem into a proposed MDM^{3}KP framework and then create an iterative solution to solve the problem. The solution is shown to converge in a limited number of iterations.
- We conduct comprehensive simulations to show the proposed MDM^{3}KP is able to substantially reduce the FL round length and improve the efficiency of FL over wireless networking under various conditions.

The organization of the remainder of this paper is as follows. Section 2 presents the background and the models of FL over wireless networking. Section 3 formulates our research problem and motivation of solution design. Section 4 shows the mapping from the original optimization problem to the MDM^{3}KP framework and describes the proposed solution. Section 5 demonstrates and discusses the simulation results. Finally, we summarize the related work in Section 6 and conclude this paper in Section 7.

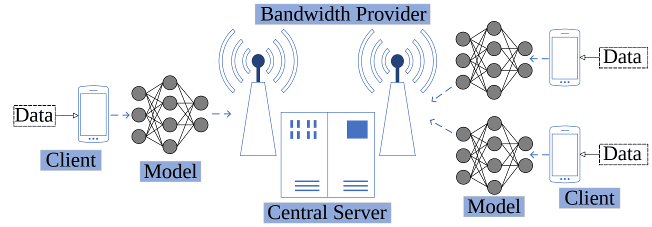


Fig. 1: Multiple wireless bandwidth providers are available for clients in FL.

2 Background and System Models

In this section we briefly discuss how FL works over wireless networking, and introduce each procedure of FL over wireless networking. In a typical model of FL, the clients and the central server jointly train a global model by a distributed approach. FL clients have different contributions toward the global model, which can significantly impact the convergence rate and training accuracy. Therefore, the central server usually adopts a client scheduling algorithm to select a fraction of clients for training [21], [22]. Since different clients have different computing capabilities, local data sizes, and SNRs, client scheduling impacts the computation time and transmission time. This incurs the challenge of synchronization and minimization of the FL round length. In the wireless communication domain, a client finishing the data transmission quickly will help reduce the FL round length, bandwidth allocation thus play a critical role in reducing the FL round length by improving the wireless communication efficiency in FL data exchange.

As shown in Fig. 1, we consider an FL over wireless networking scenario, in which a set of wireless bandwidth providers $\mathcal{I} = \{1, 2, ..., I\}$ is available to FL clients $\mathcal{J} = \{1, 2, ..., J\}$ for their model data exchange with the central server. Note that $\mathcal{J}$ is necessarily the set of all nodes in the network, but is the set determined by an existing client scheduling algorithm [21], [22] to participate the FL process. We assume in this paper that $\mathcal{J}$ are already known and focus on the follow-on bandwidth allocation problem. For each client, we must assign it to a bandwidth provider at the beginning of an FL round. The clients will download the global model, then train and update their local models. The FL round finishes once the server finishes computing the new global model.

The FL round length is the time duration of finishing one FL round, which depends on the slowest client in the wireless network as the fast clients have to wait for the slowest client. Given $\mathcal{J}$ by a client scheduling algorithm that aims to maximize the learning performance, our objective is to minimize the model data communication delay in terms of the FL round length by properly assigning clients to bandwidth providers and allocate them with proper bandwidth. Before we proposed the design for minimization, it is necessary to model all the time duration components that constitute the FL round length.

- Download Transmission (DT). s_j^{DT} is the downloaded model size for the j-th client. Although s_j^{DT} is usually the same for each client in FL, we adopt this notation for the generality of modeling. At each round, each client will be assigned to a provider and obtain bandwidth from it. $\{\kappa_i\}$ is the set of indices of clients that are assigned to i-th provider. $b_{i,j}$ is the assigned bandwidth value of i-th provider to j-th client. We have $b_{i,j} \neq 0$ when $j \in \{\kappa_i\}$ and $b_{i,j} = 0$ when $j \notin \{\kappa_i\}$. In other words, a client can only be assigned to one provider and obtain the bandwidth for communication, and cannot connect to two or more providers at the same time. $R_{i,j}^{DT}$ is the SNR of the download channel from the i-th provider to the j-th client. The DT time based on the Shannon capacity is written as

$$t_{i,j}^{DT} = \frac{s_j^{DT}}{b_{i,j}\log_2(1+R_{i,j}^{DT})}\ ,\ j \in \kappa_i. \tag{1}$$

- Local Computation (LC). This is the local model training time. It is decided by the computing capability, data size, and FL algorithm. The incurred LC time for j-client is denoted as t_j^{LC}.
- Upload Transmission (UT). s_j^{UT} is the local model upload data size and $R_{i,j}^{UT}$ is the SNR of the upload channel from the j-th client to the i-th provider. We write the UT time as

$$t_{i,j}^{UT} = \frac{s_j^{UT}}{b_{i,j}log_2(1+R_{i,j}^{UT})}\ ,\ j \in \kappa_i. \tag{2}$$

- Global Computation (GC). This is the time required by the server to perform model aggregation, which depends on the server's capability. We write the GC time as a constant t^{GC}.

Accordingly, the FL round length is denoted as

$$\max_{i\in\mathcal{I},j\in\kappa_i}\ (t_{i,j}^{DT} + t_j^{LC} + t_{i,j}^{UT} + t^{GC}). \tag{3}$$

3 Bandwidth Allocation under Multiple Providers

In this section, we first state and formulate our research problem, then discuss the direction of creating the iterative solution to our formulation with detailed steps. We summarize major symbols used in our problem formulation in Table 1.

3.1 Problem Statement and Formulation

Based on the modeling in Section 2, our objective is to find the best client-provider assignment $\{\kappa_i\}$ and bandwidth values $\{b_{i,j}\}$ for all clients and bandwidth providers such that we can minimize the FL round length (3). The minimization is round-based, which is not affected by the computation and convergence (e.g, number of iterations) factors over rounds. Since the global computation time t^{GC} is not affected by $\{b_{i,j}\}$ and κ_i in (3), the objective is equivalent to minimizing $\max_{i\in\mathcal{I},j\in\kappa_i}\ (t_{i,j}^{DT} + t_j^{LC} + t_{i,j}^{UT})$, where the model download and upload times $t_{i,j}^{DT}$ and $t_{i,j}^{UT}$ are critical to the value of FL round length.

In real-world scenarios, multiple bandwidth providers are usually available to clients. It is beneficial to assign clients to appropriate providers during wireless resource allocation for various performance gains [18], [23], especially in large-scale wireless networks. In addition, a client can have different SNRs to different providers and the cost of using the bandwidth and data transmission (e.g., the monetary cost of data in practice) can also be different. All these factors will affect the value of $t_{i,j}^{DT}$ and $t_{i,j}^{UT}$. As a consequence, different from existing studies on bandwidth allocation for FL [9], [13], [14], we consider a more practical yet complicated scenario by adding multiple bandwidth providers and the cost constraint to bandwidth allocation for FL over wireless networking. Specifically, we formulate the problem as the following optimization

TABLE 1: Notations.

Variable	Meaning
$\mathcal{I}$	bandwidth provider set
I	total number of providers
$\mathcal{J}$	scheduled client set
J	total number of clients
F	cost constraint of the system
f_i	cost at i-th provider
$B_i^{'}$	available bandwidth at i-th provider
$B_i^{'}$	available bandwidth at i-th provider
$b_{i,j}$	bandwidth allocated to j-th client by i-th provider
B_i	used bandwidth in an iteration
κ_i	clients assigned to i-th provider
$\{b_{i,j}\}$	optimal bandwidth allocation in an iteration
$\{\kappa_i\}$	optimal client assignment in an iteration
$w_{i,j}$	mapped j-th client weight at i-th provider
$p_{i,j}$	mapped j-th client price at i-th provider
t_i^*	optimal FL round length of i-th provider in an iteration

$$\text{Objective: } \min_{b_{i,j},\,\kappa_i}\ \max_{i\in\mathcal{I},\,j\in\kappa_i}\ (t_{i,j}^{DT} + t_j^{LC} + t_{i,j}^{UT}). \tag{4}$$

$$\text{Subjective to: } \kappa_m \cap \kappa_n = \varnothing\ \ \forall m,n \in \mathcal{I} \text{ and } m \neq n, \tag{5}$$

$$b_{i,j} = 0\ \forall\ j \notin \kappa_i, \tag{6}$$

$$\sum_{i\in\mathcal{I}}(f_i \sum_{j\in\kappa_i} b_{i,j}) \leq F, \tag{7}$$

$$\sum_{j\in\kappa_i} b_{i,j} \leq B_i^{'}\ \ \forall i \in \mathcal{I}, \tag{8}$$

where constraint (5) means that any client cannot be assigned to more than one provider at the same time within one round. Constraint (6) shows the j-th client does not obtain any bandwidth from the i-th provider if it is not assigned to the provider. Constraint (7) limits the cost of totally allocated bandwidths within the maximum cost budget F and f_i is a cost factor for the i-th provider. In a real-world scenario, the cost in constraint (7) is usually a monetary cost and associated with the usage of bandwidth. Constraint (8) means that the total allocated bandwidth cannot exceed a provider's maximally available bandwidth $B_i^{'}$.

3.2 Solution Design: Motivation and Overview

In conventional FL wireless network model with one bandwidth provider, the bandwidth allocation can be solved by a bisection algorithm in logarithmic time [14]. Solving (4) for $\{\kappa_i\}$ and $\{b_{i,j}\}$ in our model involves both combinatorial optimization (to obtain $\{\kappa_i\}$) and continuous optimization (to obtain $\{b_{i,j}\}$), which is in fact NP-hard because of the following reason: If (7) is removed, and $b_{i,j},\ \forall i \in \mathcal{I}$ is a constant value, then (4) is a known NP-hard multiple knapsack problem [24], [25]. Adding more constraints and making $b_{i,j}$ a variable will make (4) equivalently NP-hard. Therefore, it is not practical to analytically find out the optimal solution.

Our strategy to solve (4) is to separate it into sub-problems that are easier to solve towards minimizing the FL round length. Firstly, we look at the situation where $\{\kappa_i\}$ is known. Given a fixed, known set of $\{\kappa_i\}$, problem (4) degenerates into a continuous optimization-only sub-problem without any combinatorial optimization. In this sub-problem, suppose that we can design an approach to find $\{b_{i,j}\}$. Then, we define another sub-problem, in which we keep trying to update the values of $\{\kappa_i\}$ (and accordingly finding new values of $\{b_{i,j}\}$) to progressively reduce the FL round length. Thus, our strategy is an iterative approach including two sub-problems: (i) solving (4) given $\{\kappa_i\}$ and (ii) updating new values of $\{\kappa_i\}$. In the following, we describe the proposed approaches to these two sub-problems separately.

3.2.1 *Solving* (4) *given* $\{\kappa_i\}$

In this sub-problem, $\{\kappa_i\}$ is known and (4) is similar to a single-provider allocation problem in [14]. However, (4) still has a budget constraint (7) across multiple providers and we cannot directly solve this sub-problem using multiple rounds of a single-provider allocation algorithm.

Given $\{\kappa_i\}$, problem (4) degenerates into the problem that finds the allocation of $\{b_{i,j}\}$ to minimize the FL round length. However, the allocation of $\{b_{i,j}\}$ in different bandwidth providers is interconnected with each other because of constraint (7). As a result, we aim to decompose the bandwidth allocation problem of multiple providers into multiple independent single-provider problems. To this end, we denote the total bandwidth usage of the i-th provider by $B_i = \sum_{j \in \kappa_i} b_{i,j}$, where B_i is determined by us before we allocate $b_{i,j}$. It is clear that the usage cannot exceed the provider's bandwidth limit, i.e., $B_i \leq B_i'$. Suppose we know the values of $\{B_i\}$, the joint allocation across multiple providers will become multiple single-provider allocation ones. As a recent study [14] constructed a model and designed a bandwidth allocation solution under a single bandwidth provider with a bandwidth budget, we can leverage the results in [14] to solve the bandwidth allocation for the i-th provider, if its budget $\{B_i\}$ is given.

In particular, knowing both $\{\kappa_i\}$ and $\{B_i\}$, we denote by t_i^* the minimum FL round length under the i-th bandwidth provider. It is found in [14] that every client in $\{\kappa_i\}$ should have the same sum of upload time, download time, and local computation time to obtain t_i^*, which is written as

$$t_i^* = t_j^{LC} + \alpha_{i,j}/b_{i,j} \quad \forall j \in \kappa_i, \tag{9}$$

where $\alpha_{i,j} = \frac{s_j^{DT}}{log_2(1+R_{i,j}^{DT})} + \frac{s_j^{UT}}{log_2(1+R_{i,j}^{UT})}$; and the optimal bandwidth allocation is

$$b_{i,j} = \frac{\alpha_{i,j}}{t_i^* - t_j^{LC}} \quad \forall j \in \kappa_i. \tag{10}$$

Given a value of B_i for a single provider, we can use (9) and (10) to solve the bandwidth allocation for the provider. As a result, with $\{\kappa_i\}$ known, the original optimization (4) can be simplified as a problem of allocating total bandwidths $\{B_i\}$ to multiple providers, which is formulated as

$$\text{Objective: } \min_{B_i} \max_{i \in \mathcal{I}} \; t_i^*. \tag{11}$$

$$\text{Subjective to: } B_i < B_i^{'} \; \forall i \in \mathcal{I}, \tag{12}$$

$$\sum_{i \in \mathcal{I}} (f_i B_i) \leq F. \tag{13}$$

There is no analytical solution to the problem (11) as it is a discrete minimax problem [26]–[29]. Heuristic derivative-free optimization has been adopted to solve (11). Common methods, such as genetic algorithms, Nelder-Mead, and particle swarm [30]–[32], are not readily adopted here as they are usually focused on solving a min or max optimization instead of the more complicated minimax problem. As such, we adopt the grid search to find the best $\{B_i\}$ to solve (11). The searching bounds of $\{B_i\}$ can be set empirically and depend on the size of $\{\kappa_i\}$. Intuitively, we should assign a large search range for a provider if it has to support more users. Note that the computational complexity of the grid search is not the bottleneck for the overall solution.

3.2.2 *Iteratively updating* $\{\kappa_i\}$

In the first sub-problem, we can solve the multiple provider allocation if we have a fixed set of $\{\kappa_i\}$. In this second sub-problem, we aim to update $\{\kappa_i\}$ such that we can gradually have new combinations of $\{\kappa_i\}$ that achieve a better FL round length.

Since finding a new set of $\{\kappa_i\}$ belongs to combinatorial optimization, our approach is to map and adapt this sub-problem into a new variant of the knapsack problem [24]. The knapsack problem aims to solve the Max-Min combinatorial optimization problem. It involves multiple knapsacks and its goal is to maximize the knapsack which has minimal cumulative profit. Although it is difficult to directly map the problem in (4) into a standard knapsack problem [24], we can also find some similarity between these two problems. For example, assigning each client to a bandwidth provider is similar to place each item in a knapsack. Furthermore, the capacity of the i-th knapsack is similar to the total bandwidth usage of the i-th provider B_i. This motivates us to design a new type of the knapsack model and enable the mapping. Our proposed model takes $\{B_i\}$ as the input and yields a solution of $\{\kappa_i\}$. We will present our proposed approach to solve the second sub-problem in details in the next section.

3.2.3 *Summary and Initialization*

Fig. 2 shows the overview of our proposed iterative solution to (4) with two steps: 1) In the first sub-problem, under tentative values of $\{\kappa_i\}$, we find the optimal values of $\{B_i\}$; 2) In the second sub-problem, we use the $\{B_i\}$ found in Step 1 to find an updated set of $\{\kappa_i\}$ and the FL round length. Then, we go back to Step 1 and repeat iteratively until we can no longer obtain a better FL round length.

As Fig. 2 shows, the client assignment $\{\kappa_i\}$ is always updated from a previous one in each FL round. We should assign initial values of $\{\kappa_i\}$ at startup and then initialize $\{B_i\}$. We propose two different initialization approaches (Random and SNR-based) to initialize $\{\kappa_i\}$.

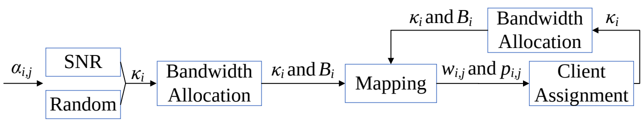


Fig. 2: Overview of the iterative solution

- **Random.** We can always randomly assign clients to bandwidth providers and produce the initial version of $\{\kappa_i\}$. The random initialization can be used as a baseline for performance comparison.
- **SNR-based.** As SNR is one of the most critical factors to be considered for resource allocation in wireless communication. We can initially assign each client to the provider whose signal has the highest SNR such that the client should obtain the best data rate and delay from the provider. To avoid overly imbalanced assignment (e.g., all clients are initially assigned to one provider in rare cases), we set a limit on the number of clients that can be assigned to each provider. When a provider reaches the bandwidth limit, it would be unavailable for initial assignment. Specifically, as j increases from 1 to J, the initialization assigns the j-th client to the i^*-th provider with $i^* = \min_{i\in\mathcal{I}_j} \alpha_{i,j}$, where $\mathcal{I}_j$ is the set of providers that are still available for assignment after the first $j-1$ clients have been assigned.

4 Client Assignment: Formulation and Solution

In this section, we describe how to determine the client assignment $\{\kappa_i\}$ in the second sub-problem towards the iterative solution to (4), which is shown in Fig. 2. We first propose a multi-dimensional max-min multiple knapsacks (MDM^{3}KP) model to formulate the sub-problem of updating client assignment $\{\kappa_i\}$, and then describe our solution to the proposed problem. We do the mapping because MDM^{3}KP can help us to update $\{\kappa_i\}$ more efficiently.

4.1 MDM^{3}KP Formulation

We can find that there is some similarity between this sub-problem and the max-min multiple-knapsack problem (M^{3}KP) as well as multi-dimensional multiple knapsack problem (MDMKP) [24], [25], [33]. In Fig. 3, we illustrate the analogy and mathematical similarity between the three problems. For example, MDMKP consists of multiple knapsacks and the items have different weights and profits when they are placed in different knapsacks, but it is not a max-min problem. M^{3}KP is a max-min problem, but items always have the same weight and profit when placed in different knapsacks. In particular, given items in standard M^{3}KP, every item has a weight and a profit with a capacity constraint that limits the cumulative weight. The objective function aims to find a solution of placing items in knapsacks that can maximize the cumulative profit of the items. The similarity between M^{3}KP and our sub-problem includes (i) both of them belong to the minimax problem involving combinatorial optimization; (ii) assigning clients to bandwidth providers is similar to placing items in knapsacks; and (iii) the capacity constraint in M^{3}KP is similar

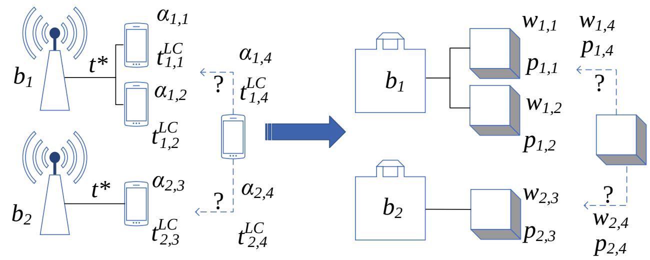


Fig. 3: **Analogy between assignment and knapsack problem**

to constraint (12). Therefore, it motivates us to propose a knapsack-modeling approach to solve the sub-problem.

However, there is no immediate mapping to the knapsack problem without appropriately specifying the weights and profits. Moreover, our modeling (11) leads to multiple weights and profits for each client because of multiple bandwidth providers, which differ from the same weight and profit modeling for each client in each knapsack in M^{3}KP. As a result, we propose a new MDM^{3}KP model to formulate our sub-problem.

$$\text{Objective: } \max \min_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} p_{i,j} x_{i,j}. \tag{14}$$

$$\text{Subjective to: } \sum_{j\in\mathcal{J}} w_{i,j} x_{i,j} \le B_i \qquad \forall i \in \mathcal{I}, \tag{15}$$

$$\sum_{i\in\mathcal{I}} x_{i,j} = 1 \qquad \forall j \in \mathcal{J}, \tag{16}$$

$$x_{i,j} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}, \tag{17}$$

where (15) denotes the cumulative weight cannot exceed the capacity of the i-th knapsack B_i; and $x_{i,j} = 1$ if the j-th client connects to the i-th provider, and 0 otherwise. Now, the key to enable the mapping is to find an approach to obtain weights $w_{i,j}$ and profits $p_{i,j}$ for the j-th client to the i-th bandwidth provider.

1) *Choosing weight $w_{i,j}$:* In MDM^{3}KP, every knapsack can only store a limited number of items because the capacity bounds the cumulative weight from above. In this sub-problem, every provider should also communicate with a limited number of clients because $\{B_i\}$ limits the total bandwidth usage. The larger $b_{i,j}$ a client has, the more weight it should have. This similarity inspires us to use $b_{i,j}$ to decide $w_{i,j}$, i.e., $w_{i,j}$ of the j-th client to the i-th provider is calculated by $w_{i,j} = b_{i,j}$. Note that as shown in Fig. 2, in each iteration, the solution to the first sub-problem produces the values of $b_{i,j}$. As a result, we can use this value as the weight when we solve the second sub-problem.

In this way, we obtain the weight $w_{i,j}$ of the j-th client when the client is assigned to the i-th provider. We also need to calculate the weight $w_{k,j}$ for the k-th provider ($k \in \mathcal{I}$ that the j-th client is not assigned to and $k \neq i$), where the solution to the first sub-problem does not provide. To this end, we consider such a question: in which situation may the j-th client be assigned to $\{\kappa_k\}$ instead of $\{\kappa_i\}$?

We propose to find the result by SNR. Given $\{\kappa_k\}$ and $\{B_k\}$, we can find the client in $\{\kappa_k\}$ which has the closest SNR to the j-th client's SNR, i.e., finding the y-th client such that $y = \min_y |\alpha_{k,j} - \alpha_{k,y}|$. Since they have the closest SNRs, the j-th client is likely to use the closest bandwidth if we replace the y-th client in $\{\kappa_k\}$ with the j-th client. This replacement is likely to have the minimum change to the FL round length. Then, we can compute the bandwidth allocation $b_{k,j}$ using the replacement set $\{\kappa_k\} \cup \{j\} - \{\mathrm{y}\}$, and set the weight as $w_{k,j} = b_{k,j}$. By using this way, we can obtain the weights of the j-th client to all providers.

2) *Choosing profit $p_{i,j}$:* With the optimization structure similarity between our sub-problem and MDM^{3}KP, the objective of our sub-problem is to minimize the FL round length in contrast to the objective of MDM^{3}KP to maximize the profit. This inspires us to use an inversely proportional model to map the FL round length in our problem into the profit in MDM^{3}KP.

To build this model, we first consider the impact of assigning the j-th client to the i-th provider. It is clear when the FL round length for the i-th provider will increase when the j-th client is added. A very large increase value indicates that it may be not a good choice to assign the j-th client to the i-th provider. It should also indicate in the MDM^{3}KP problem that placing the j-th item in the i-th knapsack is a low profit allocation. Therefore, we define the profit as the reciprocal of the FL found length increase value.

In particular, given $\{\kappa_i\}$ and $\{B_i\}$, we use (9) to calculate t_i^*, then remove the j-th client from $\{\kappa_i\}$, compute the new FL round length $\tilde{t}_i^*$, and finally define the profit as

$$p_{i,j} = \frac{1}{t_i^* - \tilde{t}_i^*}$$

To obtain $p_{k,j}\ \forall k \in \mathcal{I}$ and $k \neq i$, we adopt the way similar to choosing the weights: we first replace the y-th client which has the closest SNR the j-th client's SNR, then calculate the profit as $p_{k,j} = \frac{1}{t_k^* - \tilde{t}_k^*}$.

After we obtain the full list of weights and profits for every client, we introduce our approach to solve MDM^{3}KP in the next subsection.

We note that the mapping uses heuristic approximation to sacrifice the solution optimality (which is generally intractable) for a feasible solution to this NP-hard problem. Obtaining the inapproximability of an approximation algorithm is necessary for evaluating the loss of optimality [34]. However, inapproximability of the max-min multiple knapsacks problem is not well explored in existing studies [24], [25]. Therefore, the evaluation on the optimality of mapping may not be feasible and we use simulations to evaluate the performance of the proposed solution.

4.2 Solution Approach

The proposed MDM^{3}KP problem is a more complicated variant of the known M^{3}KP problem, where it is NP-Hard and no proper solution to directly map [24], [25]. Therefore, we create our solution based on adapting the branch and bound algorithm for M^{3}KP in [24].

4.2.1 Algorithm Design

The branch and bound algorithm is designed based on the tree structure, which is commonly used to solve the combinatorial optimization problem, such as the multiple knapsack problem [35]. We illustrate the client assignment using a tree structure: 1) layer of the tree: the j-th layer of the tree shows the possible assignment of the j-th client. The client is assigned to one of the providers; 2) node of the tree: a node on the j-th layer of the tree represents a choice of a bandwidth provider out of the total I providers for the j-th client. Each node has $p_{i,j}$ and $w_{i,j}$. As a result, a path from the root to a leaf in the tree forms a candidate group of client selection $\{\kappa_i\}$. Fig. 4 shows an example with $I = 2$ providers and $J = 2$ clients and a selection path: starting from the root, client 1 is first assigned to provider 2, and then client 2 is assigned to provider 1.

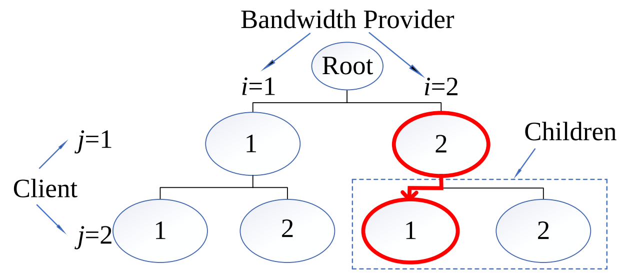


Fig. 4: **Tree Structure.** Every circle is a node. Two nodes in dotted rectangle are the children of upper node. The red path means the first client is assigned to the second provider, while the 2nd client is assigned to the first provider.

The known branch and bound method in [24] cannot be directly used to solve our MDM^{3}KP problem, in which a client has different weight and profit values when the client is assigned to a different bandwidth provider. We adapt the branch and bound algorithm in [24] to MDM^{3}KP and improve its performance.

The number of nodes in the tree increases exponential with increasing the number of I and J. For large values of I and J, it becomes computationally difficult to enumerate all possible paths in the tree to maximize the profit. To design a feasible solution, we first introduce the two following definitions.

Definition 1. (Minimum cumulative profit) Nodes in different layers are connected to form a path. A path is a client assignment $\{\kappa_i\}$ for every bandwidth provider. Given a path, the cumulative profit of i-th bandwidth provider is the sum of $p_{i,j}$ of all i-th nodes on the path. The minimum cumulative profit is the minimum in all providers' cumulative profits on the path.

Fig. 5 shows an example for $\{\kappa_i\}$ selections with their cumulative profits: the first client with profit $p_{1,1}$ and the second one with $p_{1,2}$ are assigned to the first provider that has a cumulative profit of $p_{1,1} + p_{1,2}$; and the third client with $p_{2,3}$ is assigned to the second provider that has a cumulative profit of $p_{2,3}$. Then, the minimum cumulative profit is $\min\{p_{1,1} + p_{1,2}, p_{2,3}\}$.

Definition 2. (Conditional upper bound) A conditional upper bound of the node is an upper bound of the minimum cumulative profit of the node and all its descendants.

Based on the two definitions, we use the two tree-pruning strategies in branch and bound to reduce the set of possible path candidates.

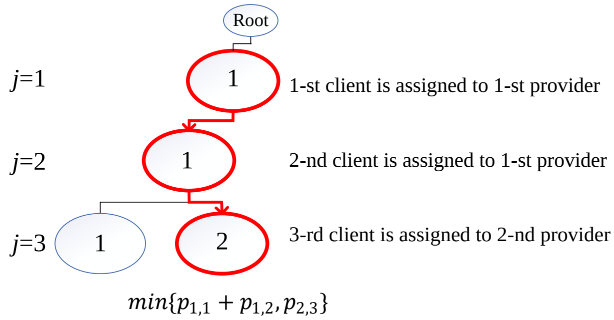


Fig. 5: **Concept of minimum cumulative profit.**

Pruning Strategy 1: Remove all nodes that violate constraint (15).

Pruning Strategy 2: Remove a node and all its descendants if the conditional upper bound of the node is lower than the lower bound of $\max\min_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} p_{i,j}x_{i,j}$ in (14),

- Finding a lower bound: We can leverage the method in [24] to compute the lower bound of $\max\min_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} p_{i,j}x_{i,j}$ in (14). In particular, we search for a candidate of $\{\kappa_i\}$ that makes $\max\min_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} p_{i,j}x_{i,j}$ as large as possible such that we have a tight bound. We adopt the local search as a method to find such a candidate of $\{\kappa_i\}$.
- Finding a conditional upper bound: Choose a node as well as its path (i.e., the path from the root to the node in the tree). According to the solution to the M^{3}KP problem [24], we denote (i) the cumulative profit at the node for the i-th knapsack to be P_i, $\forall i \in \mathcal{I}$, (ii) the upper bound of the possible increase of the cumulative profit as $\phi_i, \forall i \in \mathcal{I}$, and (iii) the upper bound of the possible increase of the cumulative profit of the surrogate relaxation as ϕ_0 [24], [35]. Surrogate relaxation unites multiple knapsacks and their capacities into a single knapsack thus a single knapsack problem is obtained. The problem of surrogate relaxation (18) is a max-min linear programming problem with linear coupled constraints. Specifically, the conditional upper bound of a node is formulated as

$$\text{Objective: } \max_{\Delta P_i} \min_{\forall i\in\mathcal{I}} (P_i + \Delta P_i). \quad (18)$$

$$\text{Subject to: } \sum_{\forall i\in\mathcal{I}} \Delta P_i \le \phi_0, \quad (19)$$

$$0 \le \Delta P_i \le \phi_i,\ \forall i \in \mathcal{I}, \quad (20)$$

where we need to find ΔP_i such that the upper bound $\min_{\forall i\in\mathcal{I}}(P_i + \Delta P_i)$ is as large as possible. Based on P_i, ϕ_i and ϕ_0, [24] computes a conditional upper bound for a node. However, the solution is only limited to a two-dimensional space, which cannot be used in our MDM^{3}KP scenario with more than two providers. To solve (18) in a higher dimensional space, a lexicographic minimax approach [36] is adopted here.

By using the MDM^{3}KP based solution with the two pruning strategies, we can effectively solve the second subproblem. As a result, we show the overall iterative solution of bandwidth allocation and client assignment under multiple bandwidth providers in Algorithm 1.

Algorithm 1 The Iterative Solution.

Input: Weight $w_{i,j}$; Profit $p_{i,j}$, $i \in \mathcal{I}$ and , $j \in \mathcal{J}$; Capacity B_i, $i \in \mathcal{I}$; ϵ
Output: $\mathbf{X}$; FL round length
$w_{i,j} = w_{i,j} \times \epsilon$
Initialize $\mathbf{X}$ as $I \times J$ zero matrix. It is the matrix of $x_{i,j}$
Initialize empty set $\mathcal{D}$
$\mathcal{D} \leftarrow \mathcal{D} \cup \mathbf{X}$
for $j = 1$ to J **do**
  for $\mathbf{X} \in \mathcal{D}$ **do**
    for $i \in \mathcal{I}$ **do**
      $x_{i,j}$ of $\mathbf{X} \leftarrow 1$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \mathbf{X}$
      $x_{i,j}$ of $\mathbf{X} \leftarrow 0$
  for $\mathbf{X} \in \mathcal{D}$ **do**
    If $\mathbf{X}$ violates any one of the 2 strategies
    $\mathcal{D} \leftarrow \mathcal{D} - \mathbf{X}$
for $\mathbf{X} \in \mathcal{D}$ **do**
  Calculate FL round length of $\mathbf{X}$ by (9)
Return: $\mathbf{X}$ with minimal FL round length and its FL round length

Note that we associate an empirical parameter ϵ with the weight $w_{i,j}$ in Algorithm 1. As $w_{i,j} = b_{i,j}$ for given κ_i in our mapping, the sum of $w_{i,j}$ in κ_i is exactly equal to the capacity B_i, which may limit the search directions in the tree due to the constraint $\sum_{j\in\kappa_i} \le B_i$. We choose $\epsilon \in [80\%, 95\%]$ as a relaxation parameter to create more search possibilities in the tree.

4.2.2 *Low-Complexity Version of MDM^{3}KP Solution*

The complexity of the MDM^{3}KP solution which is based on branch and bound grows substantially with the number of providers. Although there are usually not too many providers available in an area, it is still worth further reducing the complexity of the MDM^{3}KP solution when there are more providers for clients to choose. To this end, we design a low-complexity version of the solution by dividing the optimization with more than two providers into several subproblems, each of which has a certain amount of providers. Specifically, the division methods are composed of the following steps:

1) Initializing client assignment by the SNR-based method.
2) Given $\{\kappa_i\}$, according to (11), we solve $\{B_i\}$ as the initialization of $\{B_i\}$.
3) Dividing all I providers to $\lceil I/S \rceil$ groups of S providers (the last group has $I - S\lceil I/S \rceil$). For example, when $I = 5$ and $S = 2$, the 5 providers are divided into 3 groups: the first group includes providers 1 and 2; the second group includes providers 3 and 4, and the last group only contains provider 5. In practice, S is usually 2 or 3 as there are usually a limited number of providers available. The budget F is also divided. Specifically, the divided budget of group k is $\sum_{m\in\mathcal{M}_k} B_m f_m$, where $\mathcal{M}_k$ is the set of provider indices in group k.
4) For each group, at each iteration, we solve (4) individually by the MDM^{3}KP approach, and then obtain the FL round length of each group. As a result, the FL round

length of the original problem is the maximal FL round length among all groups.

Theoretically, the complexity of MDM^{3}KP solution can grow exponentially, which is considered to be the worst case. The low-complexity version via dividing can reduce the base of the exponential growth and accordingly reduces the overall complexity when there are a relatively large number of bandwidth providers available in the network.

4.3 Complexity Analysis

In the following, we adopt the commonly used average case approach [37], [38] to measure the time complexity of the branch and bound algorithm for MDM^{3}KP.

The calculation of the time complexity is as follows: 1) The number of MDM^{3}KP iteration in Fig. 2 is set to be L. 2) The complexity of sorting profit/weight ratio is $O(JlogJ)$. 3) After the profit/weight ratio is sorted, MDM^{3}KP begins to expand nodes. Without pruning, the total expanded nodes in the tree in each iteration is $I(I^J-1)/(I-1)$. We use parameter β_1 to show the percentage of pruned nodes thus the total number of expanded nodes in each iteration of MDM^{3}KP is $\beta_1 I(I^J-1)/(I-1)$. 4) Every expanded node is tested by the two strategies. The complexity of pruning strategy 1 is $O(IJ)+O(I)=O(IJ)$ as $O(IJ)$ is needed to compute the cumulative weight of each node and $O(I)$ is the worst case to find the cumulative weight that violates the capacity. 5) To compute the complexity of pruning strategy 2: first, we find the complexity of finding the lower bound. The 2-opt local search method is applied with complexity of $O(J^2)$ [39]; then, each node needs at most $I \times O(J) = O(IJ)$ to find Φ_0 and Φ_i, and solving (18) incurs the complexity of $O(I)$. Overall, the complexity of finding the conditional upper bound is $O(IJ)$ at each node. 6) There are $I^{J-\beta_2}$ remaining nodes as the output of MDM^{3}KP at each iteration. We need to calculate the FL round length of each node and find the minimum. Since the bisection method has complexity $O(log_2(\gamma_2-\gamma_1)/\zeta)$ [40] (γ and ζ are input parameters for bisection method), the complexity to find minimum is $I^{J-\beta_2} \times O(log_2(\gamma_2-\gamma_1)/\zeta)$.

Based on the analysis, the overall complexity of MDM^{3}KP is $O(LI^{J+2-\beta_1}J + LI^{J-\beta_2}log_2(\gamma_2-\gamma_1)/\zeta)$. For the low-complexity version, I is replaced by S in the complexity notation.

5 Simulation Evaluation

In section, we conduct comprehensive simulations to evaluate the effectiveness of our proposed method. We first introduce the setups, then present and discuss the results.

5.1 Performance Benchmark and Parameter Settings

Benchmarks: We consider FL bandwidth allocation algorithms in Table 2 for the performance comparison purpose. We note that these existing algorithms may be designed not limited to bandwidth allocation, but we only use their bandwidth allocation parts for comparisons. Since their algorithms are not designed for multiple bandwidth providers, we assume that a client is assigned with a provider based on the best SNR, which is common in today's wireless networks.

TABLE 2: Benchmark methods for performance comparison.

Name	Essential Methodology
FedCS	Allocating bandwidths to clients based on the uniform distribution [22].
HybridFL	Allocating bandwidths to clients based on the Gaussian distribution [41].
J-CSBA	Allocating bandwidth proportional to individual round length [9].
O-RANFed	Optimal allocation under single provider with minimum bandwidth constraint for clients [42].
CSIBA	Latency constrained optimal allocation for signal provider, plus uniform allocation if there is excess bandwidth [43].
Random	Our proposed MDM^{3}KP solution with random initialization of client assignment.
SNR	MDM^{3}KP with SNR-based initialization.

Evaluation Metric: We use the FL round length as our performance metric and define the length reduction ratio R as the performance improvement of the SNR-based MDM^{3}KP solution over other benchmarks; i.e.,

$$R = (L_* - L_{\text{SNR}})/L_*, \tag{21}$$

where L_{SNR} is defined as the FL round length achieved by the SNR-based MDM^{3}KP solution and L_* denotes the length by other benchmarks.

FL Networking Settings: We adopt the following default basic settings in our simulations and will evaluate the performance of our proposed solution under a range of parameter values. We note that we follow the similar SNR and transmission rate setups in the literature related to wireless resource allocation for FL [14], [44].

- Download Transmission SNR $R_{i,j}^{DT}$ for the j-th client follows a uniform distribution in [5, 25] dB in different rounds. For the j-th client within each round, its SNR variance among different bandwidth providers is uniformly distributed in [80%, 120%].
- From the j-th client towards the i-th bandwidth provider, the Upload Transmission SNR $R_{i,j}^{UL} = \rho R_{i,j}^{DT}$ with ρ uniformly distributed in [80%, 120%].
- Local Computation time t_j^{LC} for the j-th client follows a uniform distribution in [0.03, 0.07] seconds for each round [14].
- We set data size $s_j^{DT} = s_j^{UT}$ in the same round for the j-th client, and every client has the same data size in the same round. The data size follows a uniform distribution of [0.3, 0.5] Mbits in different rounds [14], [42], [45].
- We obtain the simulation results averaging over 200 runs for each scenario. We choose the default parameter settings as follows: $I = 2$, $J = 20$. $F = 13.2$, $f_1 = 1$, $f_2 = 1.2$. $B_1^{'} = 7.4$ MHz, $B_1^{'} = 6.6$ MHz. These default values are chosen such that the data transmission time is comparable to the FL computation time in the network.
- The limit of the SNR-based initialization is set to be $3/(2I)$ as discussed in Section 3.2.3, and $S = 2$ for the evaluation of the low-complexity version in Section 4.2.2.

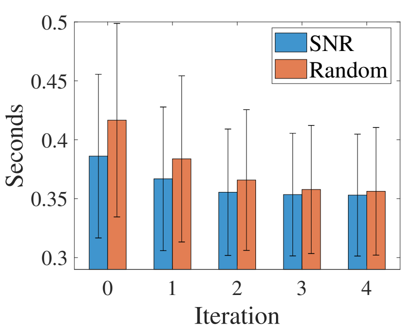


Fig. 6: FL round length of MDM^{3}KP over iterations.

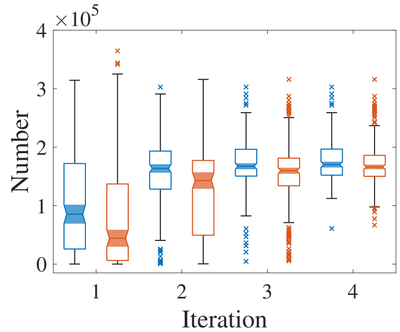


Fig. 7: Number of nodes in constructed tree over iterations.

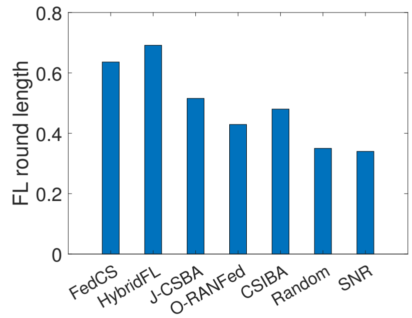
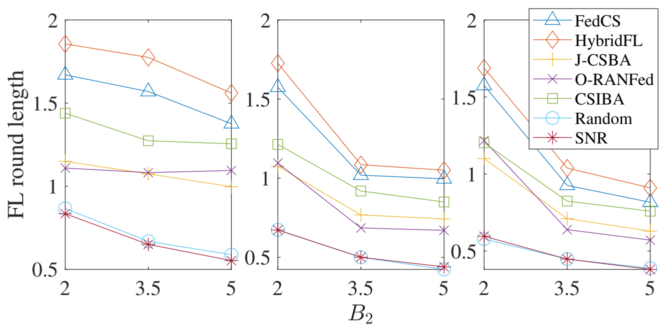
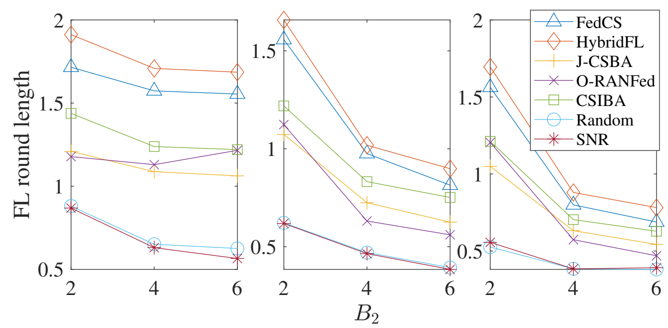


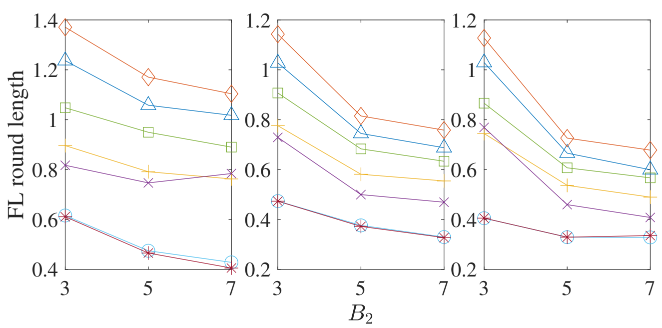
Fig. 8: FL round length comparison between benchmarks.



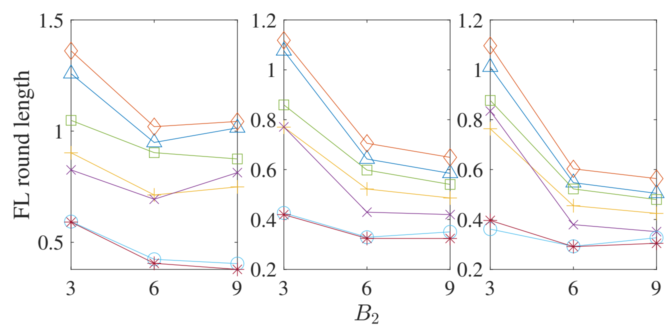
(a) $F = 9$ (From left to right:$B_1 = 2$,3.5,5)



(b) $F = 11$ (From left to right:$B_1 = 2$,4,6)



(c) $F = 13$ (From left to right:$B_1 = 3$,5,7)



(d) $F = 15$ (From left to right:$B_1 = 3$,6,9)

Fig. 9: FL round length of MDM^{3}KP and all benchmarks under various cost budgets.

5.2 Results

In the following, we show the performance of MDM^{3}KP under various settings. In order to show the advantage of the proposed MDM^{3}KP, we organize and present the simulation results as follows. We first show how the internal setups of MDM^{3}KP affects its performance (in terms of the reduction of FL round length and the efficiency) in Section 5.2.1. Then, we show the advantages of MDM^{3}KP over other benchmarks under the default scenario in Section 5.2.2, and under scenarios with different costs and bandwidths in Section 5.2.3. Finally, we add more bandwidth providers in Section 5.2.4 to show that MDM^{3}KP still maintains its performance compared to other benchmarks.

5.2.1 *Performance vs number of iterations of MDM^{3}KP*

In this experiment, we show how the proposed MDM^{3}KP solution performs with different setups (e.g., number of iterations and choice of initialization) used in its algorithm. Fig. 6 shows the performance vs number of iterations of MDM^{3}KP. When the MDM^{3}KP solution starts with either Random or SNR-based initialization proposed in Section 3.2.3, the FL round length (with its standard deviation shown) decreases as the number of iterations increases. In addition, the SNR-based initialization generally leads to a shorter FL length and a faster convergence rate than the Random initialization. For example, the SNR-based initialization decreases from 0.39 seconds to 0.35 seconds, with the ratio 8.6%. The Random initialization decreases from 0.42 seconds to 0.36 seconds, with the ratio 14.6%. The difference between the two types of initialization varies from 0.031 to 0.003 at different iterations.

Fig. 7 box-plots the number of remaining nodes for branch and bound in the tree constructed in our MDM^{3}KP solution for each iteration. We can observe that over the iterations, the mean number of nodes of the tree first increases from around 100000, and then converges at less than 200000. This means that as the algorithm progresses, it finds more possible solution values and keeps checking the values to find a solution that approaches the nearly optimal point. Compared with 2^{20}, we prune 80% to 90% nodes. It is also noted in Fig. 7 that the allocation with SNR-based initialization has more nodes in the tree than Random-Based, indicating that the SNR-based initialization is more likely to have a better solution than Random-based.

5.2.2 Comparison with Different Benchmarks

Next, we show the advantage of MDM^{3}KP over the benchmarks under the default scenario. Fig. 8 compares the FL round length between MDM^{3}KP with SNR-based and Random initializations in comparison with other benchmarks. We can see from the figure that the first 5 benchmarks adopted from others have FL round length of 0.69 seconds to 0.43 seconds; in contrast, MDM^{3}KP with SNR-based initialization achieves the best performance with 0.34 seconds and Random initialization has 0.35 seconds. The results of Fig. 8 show the performance advantage of MDM^{3}KP for bandwidth allocation for FL in the simulated wireless network with default settings. For example, the length reduction ratio of MDM^{3}KP with SNR-based initialization over HybridFL is 50.7% and over O-RANFed is 20.9%.

5.2.3 Performance under Various Cost Budgets

Fig. 8 shows the initial advantage of MDM^{3}KP under our default scenario with a fixed cost budget. We now compare MDM^{3}KP with other benchmark algorithms under various cost budgets. In particular, we measure the FL round length reduction by MDM^{3}KP in comparison with other benchmarks under a range of cost budget values B_1', B_2' and F (with $I = 2$ and $J = 18$) in Fig. 9. From the figure, we can see that overall, MDM^{3}KP with either SNR-based or random initialization has the performance advantage over other benchmarks. The results of SNR-based initialization are generally better than Random initialization. For example, when $F = 9$, $B_1' = 2$ and $B_2' = 5$, the FL round length of HybridFL is 1.56 seconds, and MDM^{3}KP can reduce the length by 64.6% and 62.8% to 0.55 and 0.59 seconds with the SNR-based and Random initializations, respectively. Meanwhile, SNR-based initialization is 6.8% better than Random initialization.

Table 3 shows the FL round length reduction ratios of SNR-based MDM^{3}KP over Random initialization and 5 benchmarks. It is observed from the table that MDM^{3}KP can achieve a length reduction ratio up to 66.5% under a wide range of cost budget values. The reduction ratio can be as low as 13.3% when $F = 15$, $B_1^{'} = B_2^{'} = 9$. Generally the reduction is higher than 25%. Table 3 also shows that the SNR-based initialization is generally better than the Random initialization.

The results of Fig. 9 and Table 3 are obtained with the default values of cost factors $f_1 = 1.0$ and $f_2 = 1.2$. We also evaluate the impacts of cost factors on the FL round length of MDM^{3}KP with SNR-based initialization in Fig. 10. In particular, we fix $I = 2$, $J = 18$, $B_1^{'} = 3.5$, $B_2^{'} = 5$, and $F = 9$ and vary the values of f_1 and f_2 from 0.7 to 1.3. We observe that the FL round length is reduced effectively by MDM^{3}KP for all pairs of f_1 and f_2. For example, when $f_1 = f_2 = 0.7$, the FL round length of J-CSBA is 0.67 seconds and SNR-based MDM^{3}KP is 0.43 seconds. When $f_1 = f_2 = 1.3$, the FL round length of O-RANFed is 0.65 seconds and SNR-based MDM^{3}KP is 0.45 seconds. Generally, the reduction ratio varies between 59.3% and 28.7%. As a result, Fig. 10 demonstrates the uniform benefits achieved by MDM^{3}KP under a range of cost factor values.

5.2.4 Performance with More Providers

Now we evaluate the advantage of using MDM^{3}KP over existing benchmarks in the scenarios of three and four wireless service providers. We adopt the low-complexity version of the MDM^{3}KP solution in Section 4.2.2 to reduce the computational complexity of MDM^{3}KP.

Fig. 11 compares the FL round lengths of MDM^{3}KP and 5 benchmarks under a three-provider scenario with parameters $F = 13.8$, $f_1 = 1$, $f_2 = 1.1$, $f_3 = 1.2$, $B_1^{'} = 3.4$, $B_2^{'} = 5.2$, $B_3^{'} = 4.5$. Generally, we can observe from the figure that MDM^{3}KP is able to substantially reduce the FL round lengths, leading to length reduction ratios ranging from 35.1% to 57.7%. For example, when there are $J = 32$ clients, the length of HybridFL is 1.22 seconds and MDM^{3}KP with SNR-based initialization is 0.52 seconds (i.e., a reduction of 57.4%). SNR-based initialization is generally better than the Random initialization.

Fig. 12 shows the FL round lengths of MDM^{3}KP and other benchmarks under four providers with parameters $F = 18.1$, $f_1 = 1$, $f_2 = 1.19$, $f_3 = 1.09$, $f_4 = 0.9$, $B_1^{'} = 2.1$, $B_2^{'} = 5.89$, $B_3^{'} = 6.38$, $B_4^{'} = 2.39$. When the low-complexity version is applied, we can see similar results: MDM^{3}KP outperforms other benchmarks and achieves the length reduction ratios of 50.2% to 70.8%. Both SNR and random initialization can converge to nearly the same FL round length. Fig. 11 and Fig. 12 show that MDM^{3}KP can always have stable performance when the number of bandwidth providers is increasing.

In addition, when we change from $S = 2$ to $S = 1$ in the low-complexity version, we observe 40% to 43% performance degradation, which means reducing the overall problem into multiple single-provider allocation problems is too coarse and leads to quite a sub-optimal solution. Thus, we need to set at least $S = 2$ for low-complexity MDM^{3}KP.

In summary, our simulation shows that the proposed MDM^{3}KP solution is more efficient to reduce the FL round length than all benchmark methods that were designed without taking into account multiple bandwidth providers and cost constraints. The results show that MDM^{3}KP achieves 13.3% to 70.8% reduction under a wide range of scenarios with different bandwidth and cost constraints as well as numbers of service providers.

6 Related Work

In this section, we summarize studies that are related to the wireless resource management methods for FL. It has been shown that simply removing some slow clients can lead to a biased scheduling that may impact the quality of the FL model [22], [46]. Thus, allocating more wireless resources for slow clients is a more feasible solution in wireless networking. Inefficient allocation can lead to a bottleneck on the wireless communication side of FL [14], [19].

TABLE 3: The FL round length reduction ratios of SNR-based MDM^{3}KP over Random initialization and 5 benchmarks.

	F = 9									F = 11								
B_1' (MHz):	2			3.5			5			2			4			6		
B_2' (MHz):	2	3.5	5	2	3.5	5	2	3.5	5	2	4	6	2	4	6	2	4	6
Random (%):	3.6	2.7	6.8	0.2	0.1	-3.8	-3.0	0.1	2.0	1.3	3.1	9.6	0.5	1.1	2.3	-5.7	-0.1	-2.8
FedCS (%):	50.0	58.6	59.8	57.3	50.9	55.8	62.1	51.7	53.3	49.4	60.0	63.6	60.3	52.4	52.8	64.4	52.0	43.1
HybridFL (%):	55.0	63.4	64.6	61.0	53.9	58.1	64.7	56.9	58.1	54.6	63.1	66.5	62.7	54.3	57.2	67.2	56.3	49.9
J-CSBA (%):	27.4	39.5	44.6	37.5	34.9	40.8	45.8	37.0	39.4	28.2	42.0	46.8	42.4	35.9	38.6	47.0	39.2	27.9
O-RANFed (%):	24.8	39.9	49.5	38.6	27.0	34.4	51.0	30.0	33.3	26.3	44.2	53.5	44.9	26.4	31.5	53.8	33.0	16.6
CSIBA (%):	42.0	48.9	56.0	44.6	45.5	48.2	50.4	45.7	49.8	39.6	49.1	53.7	49.2	44.2	48.9	54.1	45.4	37.7
	F = 13									F = 15								
B_1' (MHz):	3			5			7			3			6			9		
B_2' (MHz):	3	5	7	3	5	7	3	5	7	3	6	9	3	6	9	3	6	9
Random (%):	0.8	1.9	5.3	0.1	1.1	0.7	-0.1	0.0	-2.2	0.4	4.3	6.4	1.6	1.4	7.6	-9.6	0.4	6.8
FedCS (%):	50.6	56.0	60.2	53.9	50.0	52.4	60.6	50.6	44.0	53.0	57.3	62.7	61.0	49.5	44.6	60.8	46.8	39.7
HybridFL (%):	55.4	60.2	63.3	58.6	54.3	56.8	64.0	54.7	50.5	56.6	60.2	63.8	62.5	54.1	50.1	63.9	51.7	46.0
J-CSBA (%):	31.8	41.2	46.8	39.0	36.0	41.0	45.6	38.7	31.5	34.5	43.1	49.5	45.5	37.9	33.4	48.1	36.0	28.2
O-RANFed (%):	25.2	37.7	48.3	35.1	25.5	30.2	47.3	28.4	17.8	28.4	41.5	53.5	45.6	24.5	22.8	52.5	23.2	13.3
CSIBA (%):	41.7	51.0	54.5	47.8	45.5	48.3	53.2	45.8	40.8	43.6	55.1	56.8	51.2	45.8	40.1	54.8	44.3	36.5

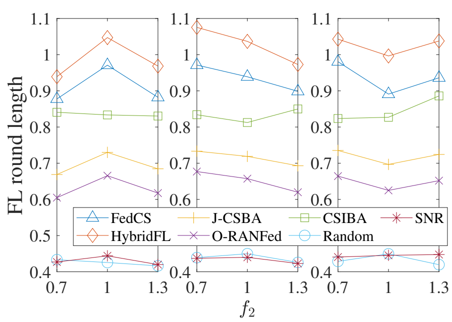


Fig. 10: FL round length of MDM^{3}KP and other benchmarks under various f_1 and f_2. (From left to right: $f_1 = 0.7$, 1, 1.3.)

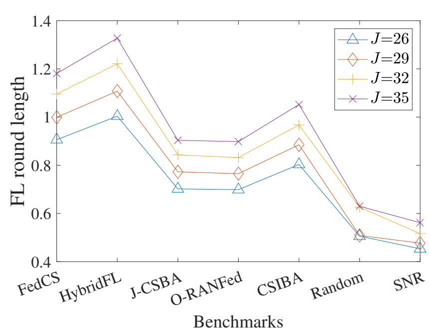


Fig. 11: FL round length of MDM^{3}KP and benchmarks under three providers.

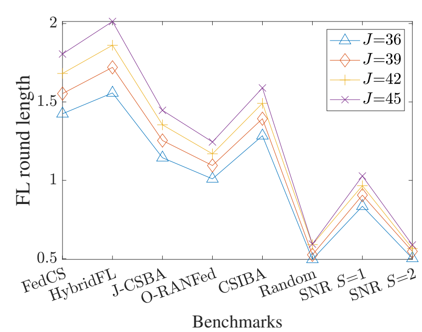


Fig. 12: FL round length of MDM^{3}KP and benchmarks under four providers.

Our algorithm happens after client scheduling procedure. It is complementary to the client scheduling and focuses on how to efficiently assign bandwidth and its provider to each client. Bandwidth allocation and client assignment play two important roles in wireless communication, which is a bottleneck on the communication efficiency in FL [14].

Bandwidth allocation: Bandwidth allocation approaches have been proposed to improve the service efficiency over wireless networking [3], [4], [9], [13], [14], [20], [44], [47]–[52]. Although bandwidth allocation algorithms [3], [4], [52] have been proposed for conventional wireless applications, they cannot be readily adapted to FL because they do not have synchronized communication model under FL. To improve the FL communication efficiency, a typical bandwidth allocation solution is to optimize the bandwidth usage among all clients with a focus on the client with the worse SNR that can lead to the worst data transmission time [9], [13], [14], [20], [44], [47]–[51]. For example, the work in [14] optimized the bandwidth allocation of multiple FL services when it is necessary to consider the fairness of different services. A Markov decision process was considered in [9] for joint client scheduling and bandwidth allocation. The study in [47] considered the trade-off between the FL accuracy and the wireless latency when allocating the bandwidth. The work in [13] increased the FL learning rate by optimizing the bandwidth of the wireless network and client scheduling policy. Most of them focused on single bandwidth provider and did not fully consider the cost of bandwidth offered by the provider. In this work, we consider a more practical and complicated scenario, where FL clients are able to select different providers according to their costs and design the MDM^{3}KP solution to minimize the FL round length under this scenario.

Client assignment: Existing studies also considered the client assignment problem in wireless networks to improve the quality of network service [16]–[18], [23], [53], [54]. For example, the work of [23] proposed a deep learning framework to minimize the energy consumption of cloud computing queue by client assignment, which is subject to the delay constraint. However, the pretrained deep learning framework can only obtain client assignment for the delay tolerant sequential packets queue, not the communication of FL. The work of [18] adopted a machine learning approach to minimize the energy and time consumption in FL for balloon networks by adjusting the client assignment in order to meet the client needs. Most studies did not deal with the bandwidth allocation or cost problems. Our work can be considered as a systematic study of assigning bandwidth from each provider to every client in a wireless network to minimize the FL round length with practical cost constraints.

7 Conclusion

This paper studies the joint problem of the client assignment and the bandwidth allocation, which aims to minimize the round length for FL over wireless networking. We propose an approach that maps the challenging problem into a new MDM ^{3}KP model and create an iterative algorithm to solve the problem. Our simulation results show that the proposed MDM ^{3}KP solution can converge fast with realistic parameters and achieve a much shorter FL round length compared with other bandwidth allocation benchmarks.

Acknowledgement

This work at University of South Florida and University of Miami was supported in part by NSF 2319781 and NSF 2319780, respectively.

References

[1] 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 Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[2] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, 2021.

[3] M. M. Nasralla, N. Khan, and M. G. Martini, "Content-aware downlink scheduling for LTE wireless systems: A survey and performance comparison of key approaches," *Computer Communications*, vol. 130, pp. 78–100, 2018.

[4] S. Colonnese, F. Cuomo, L. Chiaraviglio, V. Salvatore, T. Melodia, and I. Rubin, "Clever: A cooperative and cross-layer approach to video streaming in HetNets," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1497–1510, 2017.

[5] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[6] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[8] W. Luping, W. Wei, and L. Bo, "CMFL: Mitigating communication overhead for federated learning," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*. IEEE, 2019, pp. 954–964.

[9] H. Ko, J. Lee, S. Seo, S. Pack, and V. C. Leung, "Joint client selection and bandwidth allocation algorithm for federated learning," *IEEE Transactions on Mobile Computing*, 2021.

[10] E. Desurvire, *Classical and quantum information theory: an introduction for the telecom scientist*. Cambridge university press, 2009.

[11] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *arXiv preprint arXiv:2109.04269*, 2021.

[12] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (WFL) for 6G networks-part i: Research challenges and future trends," *IEEE Communications Letters*, vol. 26, no. 1, pp. 3–7, 2021.

[13] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2020.

[14] J. Xu, H. Wang, and L. Chen, "Bandwidth allocation for multiple federated learning services in wireless edge networks," *IEEE Transactions on Wireless Communications*, 2021.

[15] Y. Li, J. Zheng, Z. Li, Y. Liu, F. Qian, S. Bai, Y. Liu, and X. Xin, "Understanding the ecosystem and addressing the fundamental concerns of commercial mvno," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1364–1377, 2020.

[16] C. Makaya and S. Aissa, "Joint scheduling and base station assignment for CDMA packet data networks," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall*, vol. 3. IEEE, 2003, pp. 1693–1697.

[17] K. Navaie and H. Yanikomeroglu, "Downlink joint base-station assignment and packet scheduling algorithm for cellular CDMA/TDMA networks," in *2006 IEEE International Conference on Communications*, vol. 9. IEEE, 2006, pp. 4339–4344.

[18] S. Wang, M. Chen, C. Yin, W. Saad, C. S. Hong, S. Cui, and H. V. Poor, "Federated learning for task and resource allocation in wireless high-altitude balloon networks," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 460–17 475, 2021.

[19] 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.

[20] J. Kuang, M. Yang, H. Zhu, and H. Qian, "Client selection with bandwidth allocation in federated learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 01–06.

[21] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, and Y. Liu, "Context-aware online client selection for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, 2022.

[22] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[23] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[24] T. Yamada, "Max-min optimization of the multiple knapsack problem: an implicit enumeration approach," pp. 351–362, 2002.

[25] F. Al-Maliky, M. Hifi, and H. Mhalla, "An exact solution search for the max-min multiple knapsack problem," in *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2014, pp. 117–122.

[26] S. Zuhe, H. Z. Yu, and M. Wolfe, "An interval maximum entropy method for a discrete minimax problem," *Applied mathematics and computation*, vol. 87, no. 1, pp. 49–68, 1997.

[27] M. A. Wolfe, "On discrete minimax problems in R using interval arithmetic," *Reliable Computing*, vol. 5, no. 4, pp. 371–383, 1999.

[28] R. Rangarajan, "On the resolution of certain discrete univariate max–min problems," *Computational Optimization and Applications*, vol. 68, no. 1, pp. 163–192, 2017.

[29] A. Nazemi and M. Mortezaee, "A novel collaborate neural dynamic system model for solving a class of min–max optimization problems with an application in portfolio management," *The Computer Journal*, vol. 62, no. 7, pp. 1061–1085, 2019.

[30] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-a literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 380–384.

[31] F. Gao and L. Han, "Implementing the nelder-mead simplex algorithm with adaptive parameters," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, 2012.

[32] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," *Swarm and Evolutionary Computation*, vol. 63, p. 100868, 2021.

[33] J. Wang, T. Liu, K. Liu, B. Kim, J. Xie, and Z. Han, "Computation offloading over fog and cloud using multi-dimensional multiple knapsack problem," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[34] D. S. Hochba, "Approximation algorithms for NP-hard problems," *ACM Sigact News*, vol. 28, no. 2, pp. 40–52, 1997.

[35] S. Martello and P. Toth, "A bound and bound algorithm for the zero-one multiple knapsack problem," *Discrete Applied Mathematics*, vol. 3, no. 4, pp. 275–288, 1981.

[36] H. Luss, "On equitable resource allocation problems: A lexicographic minimax approach," *Operations Research*, vol. 47, no. 3, pp. 361–378, 1999.

[37] B. Krishnamoorthy, “Bounds on the size of branch-and-bound proofs for integer knapsacks,” *Operations Research Letters*, vol. 36, no. 1, pp. 19–25, 2008.

[38] H. He, H. Daume III, and J. M. Eisner, “Learning to search in branch and bound algorithms,” *Advances in neural information processing systems*, vol. 27, 2014.

[39] K. Rocki and R. Suda, “Accelerating 2-opt and 3-opt local search using GPU in the travelling salesman problem,” in *2012 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2012, pp. 489–495.

[40] Y. Ye, “Combining binary search and Newton’ s method to compute real roots for a class of real functions,” *Journal of Complexity*, vol. 10, no. 3, pp. 271–280, 1994.

[41] W. Wu, L. He, W. Lin, and R. Mao, “Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1539–1551, 2020.

[42] A. K. Singh and K. K. Nguyen, “Joint selection of local trainers and resource allocation for federated learning in open RAN intelligent controllers,” in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 1874–1879.

[43] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, “System optimization of federated learning networks with a constrained latency,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 1095–1100, 2021.

[44] Y. Ji, Z. Kou, X. Zhong, H. Li, F. Yang, and S. Zhang, “Client selection and bandwidth allocation for federated learning: An online optimization perspective,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 5075–5080.

[45] 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. 29, no. 1, pp. 398–409, 2020.

[46] 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.

[47] W. Shi, S. Zhou, and Z. Niu, “Device scheduling with fast convergence for wireless federated learning,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[48] J. Xu and H. Wang, “Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2020.

[49] 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.

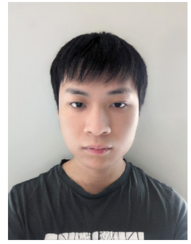
[50] C. Liu, C. Guo, Y. Yang, M. Chen, H. V. Poor, and S. Cui, “Optimization of user selection and bandwidth allocation for federated learning in VLC/RF systems,” in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[51] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, “Federated learning over wireless IoT networks with optimized communication and resources,” *IEEE Internet of Things Journal*, 2022.

[52] H. Kim and Y. Han, “A proportional fair scheduling for multicarrier transmission systems,” *IEEE Communications letters*, vol. 9, no. 3, pp. 210–212, 2005.

[53] S. Moon, H. Kim, and Y. Yi, “Brute: Energy-efficient user association in cellular networks from population game perspective,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 663–675, 2015.

[54] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. Leung, and H. V. Poor, “Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 1936–1947, 2017.



Jiahao Xue received the B.S. degree from Huazhong, Wuhan, China, in 2018, and the M.S. degree from the University of California Santa Cruz, Santa Cruz, CA, USA, in 2021. He is currently a Ph.D. student of Systems and Security in the Department of Electrical Engineering, University of South Florida, Tampa, FL, USA. His primary research interests include network and mobile system security, and machine learning for networks.

Zhe Qu received the BS degree from Xiamen University in 2015, the MS degree from the University of Delaware in 2017 and the Ph.D. degree from University of South Florida in 2022. He is currently a tenure-track Associate Professor with the School of Computer Science and Engineering, Central South University, China. His primary research interests include network and mobile system security, and machine learning for networks.

Jie Xu received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from UCLA in 2015. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Miami. His research interests include mobile edge computing/intelligence, machine learning for networks, and network security. He received the NSF CAREER Award in 2021.

Yao Liu Dr. Yao Liu is an Associate Professor in the Department of Computer Science and Engineering, University of South Florida. She received her Ph.D in Computer Science from North Carolina State University in 2012. Dr. Liu’s research is related to computer and network security, Internet of Things, and machine learning. She has been in the editorial boards of IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE/ACM Transactions on Networking, and ACM Transactions on Privacy and Security. Dr. Liu was an NSF CAREER Award recipient in 2016. She also received the ACM CCS Test-of-Time Award by ACM SIGSAC in 2019.

Zhuo Lu is an Associate Professor in Department of Electrical Engineering, University of South Florida. He is also affiliated with the Florida Center for Cybersecurity and by courtesy with Department of Computer Science and Engineering. Dr. Lu received his Ph.D. degree from North Carolina State University in 2013. Dr. Lu’s research has been mainly focused on both theoretical and system perspectives on communication, network, and security. His recent research is equally focused on machine learning and AI perspectives on networking and security. He received the NSF CISE CRII award in 2016, the Best Paper Award from IEEE GlobalSIP in 2019, and the NSF CAREER award in 2021.