Optimization of Bandwidth Usage in Wireless Federated Learning Systems

by

Pedram Zamani

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Communications

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Machine learning (ML) and the Internet of Things (IoT) have many promises but also raise many concerns and challenges. In particular, when several users collaborate in training an ML model, preserving the privacy of their data is quite challenging. Federated learning (FL), as an ML technique based on distributed computing, has been proposed to address this and other challenges of collaborative ML training. While FL has gained vast popularity in both academia and industry, its deployment in practice, especially in time-sensitive and energy-limited wireless applications, is challenging. Moreover, scarce communication resources such as bandwidth must be efficiently utilized.

In wireless FL systems, bandwidth allocation plays a crucial role in determining the overall performance, including training latency, model accuracy, and energy consumption. The limited availability of bandwidth, coupled with the need for synchronized communication among FL clients, makes bandwidth allocation a complex optimization problem.

Hence, in this work, we explore the problem of minimizing the total bandwidth usage in a wireless FL system under time and energy constraints. We formulate this problem as a non-convex optimization problem that aims to minimize the total bandwidth usage of the system while respecting the mentioned practical constraints. By decomposing the problem into two subproblems, we show that it can be solved efficiently using convex optimization and iterative search techniques. Our proposed algorithm finds the optimal solution while enjoying low complexity, making it suitable for real-world implementations. Through comprehensive simulations, we demonstrate

the efficiency of our approach and analyze different aspects of the problem.

This work contributes to the ongoing research efforts in optimizing FL for wireless networks, addressing the critical challenges of bandwidth allocation along with cost constraints. The insights gained from this study can help in developing more robust and efficient FL systems for a wide range of time-sensitive and energy-constrained wireless applications.

Preface

This thesis is an original work by Pedram Zamani. A version of Chapter 3 of this thesis has been submitted to be published as Zamani, P., Pourtahmasi Roshandeh, K., & Ardakani, M. (2024) ”An Optimal Algorithm on Bandwidth Minimization in Wireless Federated Learning” IEEE Communication Letters.

Acknowledgements

First and foremost, I would like to express my deepest gratitude for my supervisor, Professor Masoud Ardakani, whose constant guidance and unwavering support helped me throughout this journey. It would be impossible to complete this work without his help. Whenever I was facing challenges or setbacks from time to time, he was there to support me with patience and professional help. I appreciate and admire his great character as a supervisor.

Additionally, I want to thank my dear mentor, Koosha Pourtahmasi Roshandeh, who significantly aided my research's progress for this work. He assisted me in familiarizing myself to the research world for the first time, and helped me overcome the burdens along the way. Moreover, his words of motivation, encouragement, and advice have always been a great help to me on a personal level.

I would like also to thank Professor Chintha Tellambura and Professor Ali Khajehoddin, for being on the examination committee. I appreciate their valuable time spent reading and providing comments on my work. Their insightful feedback helped me improve and present the best of this work.

Last but not least, I would like to thank my family and friends for constantly encouraging me and supporting me through all the ups and downs. Their presence in my life is always heartwarming.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Machine learning (ML) has become an integral part of modern technology, driving advancements across various industries by enabling systems to learn from data and make informed decisions. Traditionally, ML involves the centralization of data from multiple sources onto a single server or data center, where a model is trained on this aggregated dataset. This centralized approach has been the backbone of many successful applications, from image recognition and natural language processing to predictive analytics and autonomous systems. By leveraging large datasets, powerful computational resources, and sophisticated algorithms, traditional ML has achieved remarkable accuracy and efficiency in a wide range of tasks.

However, this centralized model of learning is not without its limitations. One significant challenge is the issue of data privacy and security. As more sensitive and personal data is generated by users and devices, the risks associated with centralizing this data—such as breaches, unauthorized access, and compliance with privacy regulations have become more pronounced. Additionally, the sheer volume of data generated at the edge, by devices such as smartphones, sensors, and autonomous vehicles, creates logistical challenges in terms of data transfer, storage, and processing. The need to move large datasets to a central server can result in significant communication overhead, latency, and energy consumption, making centralized ML impractical for certain applications.

To address these challenges, federated learning (FL) has emerged as a decentralized alternative to traditional ML [1–3]. FL enables multiple devices, often referred to as clients, to collaboratively train a shared global model without exchanging their local data. Instead of centralizing the data, FL keeps it localized on edge devices, with each device performing local training on its own dataset. Periodically, these devices send updates, typically in the form of model parameters or gradients, to a central server. The server aggregates these updates to improve the global model, which is then sent back to the devices for further local training. This process iteratively continues until the model converges to the desired level of accuracy.

FL offers several key advantages over traditional ML. By keeping data localized, FL significantly enhances data privacy and security, reducing the risks associated with data breaches and unauthorized access. It also reduces the communication overhead associated with transferring large datasets, as only the model updates are transmitted between devices and the server. This makes FL particularly well-suited for scenarios where data is generated and stored on edge devices, such as in healthcare [4, 5], finance [6], smart cities [7] and many IoT applications [8]. Moreover, FL enables the utilization of the computational resources available on edge devices, distributing the training workload and reducing the reliance on centralized data centers.

The rise of edge computing and the proliferation of devices such as smartphones, IoT sensors, and autonomous vehicles have positioned FL as a key technology for future AI systems. These devices continuously generate vast amounts of data that can be harnessed to improve machine learning models. However, the challenge lies in the effective utilization of resources—both computational and communication—available in these devices. FL promises to leverage the collective intelligence of these devices without overwhelming them or the network with excessive computational or communication demands, which is critical in ensuring the sustainability of such systems.

In a typical FL process, each participating device performs local training on its dataset and periodically sends updates, usually in the form of model parameters or gradients, to a central server. The server then aggregates these updates to improve the global model, which is subsequently sent back to the devices for further local training. This iterative process continues until the model converges to a desired level of accuracy. However, the communication overhead associated with transmitting model updates can be substantial, particularly in wireless environments where bandwidth is limited and costly. This overhead not only strains the network but can also lead to increased latency, which is detrimental to time-sensitive applications of FL.

The integration of FL with wireless environments introduces new complexities and challenges. Wireless networks, by their nature, are characterized by variable bandwidth, latency, and reliability. Unlike wired networks, where resources can be more easily managed and scaled, wireless networks must contend with factors such as signal interference, channel fading, and congestion. These factors can significantly impact the performance and efficiency of FL systems, particularly when dealing with large-scale deployments or environments with fluctuating network conditions.

In wireless FL, the communication between edge devices and the central server typically occurs over shared wireless channels. The bandwidth of these channels is often a limited resource, especially in scenarios involving large numbers of devices, such as smart cities, autonomous vehicle fleets, or widespread IoT deployments. Efficiently managing this bandwidth is critical for ensuring that the FL process remains scalable and responsive, particularly as the number of participating devices increases. Without careful management, the cumulative demand for bandwidth can lead to network congestion, which can severely degrade the performance of the FL process.

Moreover, edge devices in wireless environments are often constrained by limited computational resources and energy supplies. These devices, such as smartphones or IoT sensors, typically operate on battery power and have limited processing capabilities. Therefore, the computational burden of local training and the energy costs associated with communication must be carefully balanced. Excessive computational demands can drain the device's battery, while frequent communication can consume significant amounts of bandwidth and energy, leading to increased latency and potential model degradation. This balancing act is crucial in ensuring that FL can be sustainably deployed in real-world environments.

The primary motivation for this work stems from the need to optimize the use of communication resources, particularly channel bandwidth, in wireless FL environments. Bandwidth is arguably the most valuable and constrained resource in such settings, as it directly influences the scalability, efficiency, and feasibility of deploying FL on a large scale. In scenarios where many devices are participating in the FL process, the cumulative demand for bandwidth can lead to network congestion, increased latency, and higher operational costs. Therefore, efficient bandwidth allocation is crucial for sustaining the performance of FL systems, especially as these systems scale to include more devices and handle more complex models.

The optimization framework we propose has broad applications across various industries where FL is poised to play a transformative role. One of the most immediate applications is in the telecommunications industry. Telecom companies are increasingly exploring FL as a way to leverage data generated by users' devices without violating privacy. However, the efficiency of these deployments hinges on the effective use of bandwidth, a resource that telecom operators must manage meticulously to ensure service quality and cost-effectiveness. Our framework provides a tool for telecom companies to optimize bandwidth allocation in FL systems, allowing them to maximize the number of devices that can participate in the learning process without overwhelming the network or compromising the quality of the service provided.

Another key application is in autonomous systems, such as fleets of self-driving cars. These vehicles generate and process massive amounts of data in real-time, and FL can be used to enable them to learn collaboratively from each other's experiences. In such a scenario, the efficient use of wireless communication channels is critical to ensuring that vehicles can quickly and reliably exchange model updates without sacrificing safety or performance. Our optimization approach ensures that bandwidth is used efficiently, allowing for faster and more reliable communication between vehicles, which is essential for the real-time decision-making required in autonomous systems.

In the healthcare sector, FL offers a way to train models on decentralized medical data, enabling hospitals and other healthcare providers to collaborate on machine learning projects without sharing sensitive patient data. However, healthcare data is often large and complex, making bandwidth a significant concern. Our framework can help healthcare providers implement FL systems that are both privacy-preserving and efficient, ensuring that medical data can be used to its fullest potential without overburdening the network or compromising patient privacy. This is particularly important in healthcare, where timely and accurate model updates can be crucial for patient outcomes.

The concept of smart cities involves the integration of technology to manage urban resources more efficiently. FL can be used in smart cities to enable devices such as traffic cameras, environmental sensors, and public transportation systems to learn from each other and improve their operations. In such environments, bandwidth is a shared resource that must be managed carefully to avoid congestion and ensure that all devices can communicate effectively. Our optimization framework can be applied to manage bandwidth in smart cities, ensuring that FL processes are both effective and sustainable, which is essential for the long-term success of smart city initiatives.

The primary contribution of this work is the development of an optimization framework that minimizes total bandwidth usage in wireless FL environments. This framework is designed to balance the competing demands of communication and computation resources while maintaining model accuracy and adhering to latency and energy constraints. Our approach is novel in its focus on bandwidth optimization as the central objective, setting it apart from existing techniques that primarily address overhead reduction.

This work represents a significant step forward in the optimization of resource-constrained FL systems, offering a novel approach to managing the most critical resource in wireless networks—bandwidth. By addressing the unique challenges posed by wireless FL environments, our framework has the potential to enable more efficient and scalable deployments of FL across a wide range of applications, from telecommunications to healthcare, autonomous systems, and beyond. This approach not only enhances the practicality of FL in real-world settings but also opens up new possibilities for its application in increasingly complex and demanding environments.

1.2 Thesis Objectives

This study stands at the intersection of federated learning systems and wireless communication, two rapidly evolving fields that promise to reshape the landscape of distributed computing. Considering the discussed significance of channel bandwidth as the most valuable communication resource, we aim to use it in the most efficient manner in wireless federated learning systems. The study delves into the problem of optimizing computation resources, accuracy, and bandwidth allocation in order to reach minimal bandwidth usage. This involves formulating an optimization problem that models the wireless federated learning system while considering constraints on energy consumption and time limit, which are critical factors in real-world applications such as mobile edge computing and IoT networks.

The goal is to handle the complexities inherent in this problem by converting non-convex optimization aspects of the problem into manageable convex formulations, which are more tractable and can be solved efficiently using standard optimization techniques. By doing so, the research attempts to achieve optimal solutions that deliver our objective of bandwidth usage minimization. The outcomes of this study could have significant implications for the design of future wireless federated learning systems, potentially enabling more scalable and robust deployments in a variety of environments, from urban centers with high network congestion to remote areas with limited connectivity.

1.3 Thesis Organization

The following chapters from this thesis are organized as follows. First, in Chapter 2, we delve into the background behind this work. We will review the literature and provide an overview of related works in the field. Subsequently, we will provide some base knowledge on the concepts and techniques used in this work.

In Chapter 3, we will first explain the system model to our work. Then, we will formulate a standard optimization problem based on our model. After analyzing the problem, we will propose a solution along with step-by-step proofs. This will be followed by a numerical results section to examine the proposed solution in a comprehensive manner. Finally, we will go through the same process for an extended version of the problem, which we call the Dual Accuracy form.

Chapter 4 will include a conclusion to this thesis, explaining a gist of the methodology, the contributions and possible applications of our work. It is finished by presenting some possible future research directions that can be built on this research to make further advancements in the domain.

Chapter 2

Background

2.1 Literature Review

Since the emergence of federated learning as a new approach to replace the classic, centralized way of machine learning, numerous publications have explored different aspects of wireless federated learning systems. In this section, we will review some of the most influential works, while categorizing them based on various criteria. These criteria include objective functions, wireless setups, and applications of the works published in the field. We also inspect some works that are merely focused on mathematical analysis of FL frameworks.

Based on Optimization Goals

State-of-the-art works have been published that have focused their goal on optimizing wireless FL for efficient use of resources such as reducing the energy consumption or latencies in the FL process. In some cases, objective functions also consider other factors such as packet error rate [9], or data size [10, 11]. The most popular approaches towards these goals are client selection and bandwidth allocation. However, in most works, other optimization parameters exist along with these two.

The authors in [12] have considered a hierarchical FL system. They have defined a cost function that consists of a weighted sum of the energy consumption and the latency. The authors propose different algorithms to find the optimal radio resource allocation, computation resource allocation, and local accuracy in order to minimize the mentioned cost function. In [13], the authors minimize the total energy consumption of the users under a latency constraint, over variables including (but not limited to) bandwidth, local accuracy, and CPU-cycle frequency. Also, [14] proposes maximizing the number of participating clients in each FL global iteration and minimizing the bandwidth allocated to the selected clients. However, the proposed discrete water-filling method for bandwidth allocation distributes the communication resource blocks subject to only a latency constraint.

The study in [15], focused on vehicular edge computing (VEC), addresses two key challenges: selecting suitable vehicles for participating in the FL process and optimizing the allocation of computational and communication resources to minimize training time and energy consumption. By formulating the problem as a joint optimization of vehicle selection, resource allocation, and FL model training, the paper proposes strategies that balance these factors, leading to more effective and scalable VEC systems.

The paper [16] tackles the critical trade-offs between computation and communication latencies and their impact on federated learning time and user equipment (UE) energy consumption. The authors address this by formulating an optimization problem which captures these trade-offs. Although the formulated problem is inherently non-convex, the authors decompose it into three convex sub-problems and derive closed-form solutions for each, leading to a globally optimal solution. These solutions provide key insights into optimizing learning time, accuracy, and energy cost in FL over wireless networks. This work is among the most reputable state-of-the-art works in the field of wireless FL.

The paper [10] focuses on optimizing federated learning in wireless networks by addressing client selection and bandwidth allocation. The optimization problem is framed as a mixed-integer problem aimed at minimizing the defined cost function, which consists of latency and accuracy, while adhering to long-term energy constraints across all training rounds. The approach involves an online optimization strategy to balance these factors effectively.

There are more examples of works that have focused on either minimizing the energy consumption [17–20], training time [21, 22], or a joint cost function of both [23, 24].

Based on Wireless Setups

In the sense of wireless setups, Various works with FL frameworks in simple to more complicated wireless system models, such as reconfigurable intelligent surfaces (RIS), mmWave, HetNets, interference, Non-orthogonal multiple access (NOMA), Multiple Input Multiple Output (MIMO), were investigated in the literature [19, 21, 25–29]. For example, authors in [21] work on a framework to reach minimal training time in FL over a cell-free massive multiple-input multiple-output (MIMO) network, with the total allocated bandwidth to the system as a given constant.

In [25] and [26], authors investigate FL in different wireless environments, i.e. an mm-wave massive MIMO network with hybrid beamforming and an RIS-assisted massive MIMO, and compare the performance, the transmission overhead, and the tolerance to corruptions in channel data in them against centralized learning.

The proposed FL scheme in [27] considers an RIS-assisted mmWave communication system, and works on maximizing the achievable rate of the received signal, with a given total system bandwidth. However, unlike our work, it does not include the energy consumption of participating users and a target accuracy for local users or the global model in its system model.

The work in [28], proposing a wireless FL system model with limited resource blocks (RBs) and multiple users with interference, performs user selection and resource allocation to minimize the total time it takes to complete the FL training process. This work, too, does not include an adjustable local accuracy and considers a constant number of iterations T that is large enough to guarantee the convergence of FL. It also does not model or limit the energy consumption of local users.

Authors in [29] aim to minimize the total energy consumption in hierarchical FL over heterogeneous networks (HetNets), considering a constant bandwidth. This work has no constraint on the completion time of users at each global iteration or the whole FL process. In [19], the total energy consumption of users is minimized considering a given bandwidth allocation in a multi-tier NOMA-enabled HetNet FL environment.

Based on Applications

From an application point of view, numerous applications of FL in different areas have been explored in the literature. In healthcare, FL has been used to improve predictive models for patient outcomes by training on sensitive medical data from multiple hospitals without centralizing the information, thus preserving patient privacy [4, 5, 30].

In finance, FL can enable multiple financial institutions to collaboratively train machine learning models without directly sharing their sensitive data. This application of FL is relevant for scenarios like fraud detection, credit scoring, and financial risk analysis, where institutions can benefit from shared insights while adhering to strict privacy regulations [6, 31].

In autonomous driving, FL facilitates the development of robust vehicle perception systems by leveraging data from a fleet of vehicles to improve object detection and navigation algorithms in a privacy-preserving manner [15, 32].

In agriculture, FL supports precision farming by combining data from various farms to optimize crop management and yield predictions while maintaining the privacy of proprietary farming techniques [33].

In smart homes, FL improves the functionality of devices like thermostats and security cameras by learning from user behavior across multiple homes, thus enhancing personalization while keeping user data decentralized [34, 35].

Additionally, smart keyboard applications use FL to enhance text prediction and auto-correction by training language models on users' typing data while keeping the data on their devices, thus providing more accurate suggestions without compromising privacy [36, 37].

In smart cities, FL is utilized for numerous purposes as well. Federated learning enhances traffic management by optimizing signal control, congestion prediction, and route planning through decentralized model training. It improves public safety with real-time surveillance and anomaly detection while preserving privacy. Environmental monitoring benefits from FL through collaborative air quality prediction and pollution control. For energy management, FL predicts demand and efficiently manages smart grids without sharing sensitive data. [7, 38]. These diverse applications highlight the versatility of FL in solving complex, privacy-sensitive problems across a range of industries [8].

Fundamental Works

Finally, some works are solely dedicated to proposing fundamental algorithms and frameworks for FL or analyzing them regarding convergence rate, accuracy, and communication or computation efficiency.

The original work in [1] is one of these works, introducing federated learning as a decentralized approach that allows training deep networks while keeping data on users' devices to enhance privacy. The authors propose the FederatedAveraging (FedAvg) algorithm, which iteratively averages locally computed updates, significantly reducing communication rounds compared to traditional methods. This approach effectively addresses the challenges of unbalanced and non-IID data distributions, enabling efficient training on decentralized data.

The paper [2] by Konečný et al. introduces federated optimization as a framework for training machine learning models across numerous devices while keeping data localized to preserve privacy. It emphasizes the need for communication efficiency and proposes a new algorithm tailored for sparse convex problems, addressing the unique challenges of distributed learning in scenarios where data is unevenly distributed across many devices. This work lays the groundwork for future research in privacy-preserving distributed learning techniques, particularly in mobile and edge computing contexts.

In [39], authors propose a communication-efficient framework, namely COCOA, that uses local computation in a primal-dual setting to dramatically reduce the amount of necessary communication. They provide a strong convergence rate analysis for this class of algorithms, as well as experiments on real-world distributed datasets with implementations in Spark.

Ma et al., in their widely adopted work "Distributed Optimization with Arbitrary Local Solvers" [40], propose a flexible framework that allows for the use of any local solver on individual machines while still achieving competitive performance in a distributed setting. The authors provide strong primal-dual convergence guarantees for their approach and demonstrate through theoretical analysis and experiments that it can outperform specialized distributed methods by leveraging well-tuned local solvers.

The papers "Semi-Stochastic Coordinate Descent" [41] and "Semi-Stochastic Gradient Descent Methods" [42] introduce novel optimization techniques that combine deterministic and stochastic steps for minimizing strongly convex functions. The Semi-Stochastic Coordinate Descent (S2CD) method alternates between full gradient computations and stochastic coordinate updates, using non-uniform distributions to select both the function and coordinate to update. Similarly, the Semi-Stochastic Gradient Descent (S2GD) method alternates between full gradient evaluations and multiple stochastic gradient steps. Both methods aim to achieve faster convergence rates than traditional stochastic methods, offering a balance between the computational efficiency of stochastic approaches and the stability of deterministic methods. These approaches represent significant advancements in optimization techniques for large-scale machine learning problems, providing improved efficiency and theoretical guarantees without the need for parameter tuning typically required in purely stochastic methods.

2.2 Gap in Research

While existing research has made strides in reducing communication costs in FL, most efforts have focused on techniques that reduce the size of the transmitted data, such as quantization [43–45], and compression [46–48]. These techniques primarily aim to decrease the overhead associated with each communication round by reducing the amount of data that needs to be transmitted. However, they do not address the broader issue of minimizing the total bandwidth usage across all FL users. To the best of our knowledge, the specific problem of minimizing total bandwidth consumption, while also considering computational resources, model accuracy, latency, and energy constraints, has not been thoroughly investigated in the literature of wireless FL.

This work aims to fill this gap by formulating the problem of minimizing total bandwidth usage in a wireless FL setup as a non-convex optimization problem. The proposed approach considers the interplay between communication and computation resources, and accuracy ensuring that the optimization process balances these demands while maintaining the accuracy of the learning model. By decomposing the problem and applying convex optimization techniques, we are able to derive a solution that effectively optimizes bandwidth usage, thereby enhancing the efficiency and scalability of FL in wireless environments.

2.3 Background Information

2.3.1 Optimization

Mathematical optimization [49] is a fundamental area of mathematics and applied sciences that deals with the problem of finding the best solution from a set of feasible solutions. The essence of optimization is to make decisions that maximize or minimize an objective function—often representing cost, profit, efficiency, or some other measure of interest—subject to a set of constraints. These constraints define the conditions that any feasible solution must satisfy.

At its core, an optimization problem is composed of three key elements:

1. **Objective Function:** This is the function that needs to be optimized—either maximized or minimized. For example, in a business context, the objective might be to minimize costs or maximize profits.

2. **Decision Variables:** These are the variables that can be controlled or adjusted to optimize the objective function. For example, these could be the quantities of different products to manufacture or the amount of resources to allocate.

3. **Constraints:** These are the conditions or restrictions placed on the decision variables. They define the feasible region within which the solution must lie. Constraints could be physical limits, resource capacities, or specific requirements that must be met.

Mathematical optimization is a powerful tool used in various fields such as engineering, economics, finance, logistics, and machine learning, to name a few. By formulating problems as optimization problems, one can apply a wide range of mathematical techniques to find optimal or near-optimal solutions.

Categorization of Optimization Problems

Optimization problems can be broadly categorized based on several criteria. Understanding these categories helps in choosing the appropriate methods and tools for solving specific problems. Below is a structured, top-down categorization of optimization problems.

- **Based on the Objective Function**
  - **Linear Optimization:**
    - In linear optimization, the objective function and constraints are linear functions of the decision variables. Linear optimization problems are

relatively well understood and can often be solved efficiently.

    * Example: Linear Programming (LP).

  - **Nonlinear Optimization:**

    * In nonlinear optimization, at least one of the objective functions or constraints is nonlinear. Nonlinear optimization problems can be more challenging to solve due to the complexity of the objective function and constraints.

    * Nonlinear optimization can be further divided into:

      · **Convex Optimization:** Where the objective function is convex, and the feasible region is a convex set. These problems have a single global optimum.

      · **Non-convex Optimization:** Where the objective function or feasible region is non-convex, potentially leading to multiple local optima, making the problem harder to solve.

- **Based on the Nature of the Decision Variables**

  - **Continuous Optimization:**

    * Decision variables can take any value within a given range (usually real numbers).

    * Example: Minimizing a cost function over a continuous range of inputs.

  - **Discrete Optimization:**

    * Decision variables can only take discrete values, often integers.

    * Example: Integer Programming (IP), where the decision variables must be integers.

- **Based on the Structure of the Constraints**

- **Unconstrained Optimization:**
  - No constraints are imposed on the decision variables.
  - Example: Finding the minimum of a function over its entire domain.
- **Constrained Optimization:**
  - The problem includes equality and/or inequality constraints that restrict the feasible region.
  - Example: Portfolio optimization with budget constraints.

Optimization problems can also be categorized based on their deterministic or stochastic nature. In deterministic optimization, all parameters in the problem are known with certainty. On the other hand, stochastic optimization involves uncertainty in the parameters and is often modelled using random variables.

Convex Optimization

Convex optimization [50] is a fundamental class of optimization problems where the objective function is convex and the feasible region is defined by convex constraints. A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if, for any $x_1, x_2 \in \mathbb{R}^n$ and any $\theta \in [0, 1]$, the inequality

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$$

holds. This definition implies that for a convex function, the line segment between any two points on its graph lies above or on the graph itself. Convex sets and functions are particularly of significant interest because they guarantee that any local minimum is also a global minimum, which greatly simplifies the problem-solving process compared to non-convex cases.

Mathematical Conditions for Convexity

Several mathematical conditions can be used to determine if a function is convex:

- **First-Order Condition:** A differentiable function f is convex if and only if its gradient satisfies the condition:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for all } x, y \in \mathbb{R}^n.$$

This condition means that the function lies above its first-order Taylor expansion at every point.

- **Second-Order Condition:** For twice-differentiable functions, convexity can be determined by the Hessian matrix $\nabla^2 f(x)$. The function f is convex if and only if the Hessian is positive semidefinite at every point:

$$\nabla^2 f(x) \succeq 0 \quad \text{for all } x \in \mathbb{R}^n.$$

A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is said to be positive semidefinite if, for all non-zero vectors $z \in \mathbb{R}^n$, the quadratic form $z^T A z \geq 0$. In other words, a matrix A is positive semidefinite if:

$$z^T A z \geq 0 \quad \text{for all } z \in \mathbb{R}^n.$$

The positive semidefiniteness of the Hessian $\nabla^2 f(x)$ indicates that the function curves upwards at every point, which is a key characteristic of convex functions.

- **Convexity of Set-Based Functions:** If a function f is defined over a convex set $S \subseteq \mathbb{R}^n$ and the function is convex on S, then any minimization of f over S is guaranteed to have a globally optimal solution.

Key Properties of Convex Optimization Problems

- **Global Optimality:** The convexity of both the objective function and feasible region ensures that any local minimum is also a global minimum. This property eliminates the possibility of suboptimal solutions trapped in local minima.

- **Efficient Solvability:** Convex optimization problems can often be solved using efficient algorithms with polynomial-time complexity, such as interior-point methods, projected gradient descent, and other techniques specifically designed for convex structures. These algorithms benefit from the problem's well-behaved geometry, ensuring reliable convergence even in high-dimensional spaces.

The Lagrangian and KKT Conditions in Convex Optimization

The Lagrangian is a function that combines the objective function and the constraints of an optimization problem into a single expression. For a given optimization problem:

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

the Lagrangian is defined as:

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

where λ_i and ν_j are the Lagrange multipliers associated with the inequality and equality constraints, respectively.

The Karush-Kuhn-Tucker (KKT) conditions are a set of necessary conditions that a solution must satisfy to be optimal for a constrained optimization problem. When the optimization problem is convex, these conditions are not only necessary but also sufficient for optimality.

KKT Conditions

The KKT conditions for this problem are:

1. **Primal Feasibility:**

$$g_i(x^*) \leq 0, \quad i = 1, \ldots, m$$

$$h_j(x^*) = 0, \quad j = 1, \ldots, p$$

Ensures the optimal solution (x^*) satisfies the constraints.

2. **Dual Feasibility:**

$$\lambda_i \geq 0, \quad i = 1, \ldots, m$$

Where λ_i are the Lagrange multipliers associated with the inequality constraints.

3. **Stationarity:**

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{p} \nu_j \nabla h_j(x^*) = 0$$

The gradient of the Lagrangian must vanish at x^*.

4. **Complementary Slackness:**

$$\lambda_i g_i(x^*) = 0, \quad i = 1, \ldots, m$$

Ensures that either a constraint is active or its corresponding Lagrange multiplier is zero.

Application in This Work

In this work, we apply the KKT conditions to solve a convex optimization problem, focusing on optimizing bandwidth usage in a wireless federated learning system. This involves minimizing a convex objective function subject to constraints such as time and energy limits, which are critical for time-sensitive applications. The KKT conditions help us systematically derive the optimal solution, ensuring that it satisfies all constraints while minimizing the objective function, which in this case is bandwidth usage. This approach is significant because channel bandwidth is often the most expensive communication resource in wireless systems, making efficient use of it crucial.

2.3.2 Log-Distance Path Loss Model

The Log-Distance Path Loss Model is a widely used empirical model in wireless communication to predict the path loss of a signal over a distance [51]. It is particularly useful in environments where the exact propagation characteristics are complex and not easily modeled analytically. This model helps in understanding how signal strength diminishes with distance and is crucial for designing and optimizing wireless communication systems.

Path loss refers to the reduction in power density of an electromagnetic wave as it propagates through space. It is influenced by various factors including distance, frequency, and the environment (e.g., urban, rural, indoor). Path loss is a critical parameter in the design of wireless networks because it impacts the coverage area and the quality of service.

The Log-Distance Path Loss Model generalizes the Free Space Path Loss model by incorporating a path loss exponent, which accounts for the environment's impact on signal attenuation. The basic form of the path loss model can be expressed both in dB (logarithmic scale) and non-dB (linear scale) as follows:

- **dB Form:**
$$PL(d) = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right)$$
- **Linear Scale Form:**
$$L(d) = L(d_0) \left(\frac{d}{d_0} \right)^n$$

where:

- $PL(d)$ (in dB) and $L(d)$ (unitless) represent the path loss at distance d,
- $PL(d_0)$ (in dB) and $L(d_0)$ (unitless) represent the path loss at a reference distance d_0,
- n is the path loss exponent (unitless),
- d is the distance between the transmitter and receiver (in meters, m),
- d_0 is the reference distance (in meters, m).

The reference path loss $PL(d_0)$ can be determined through empirical measurements or calculated using the Free Space Path Loss formula:

$$PL(d_0) = 20 \log_{10} \left(\frac{4\pi d_0 f}{c} \right)$$

where:

- f is the frequency of the signal (in Hertz, Hz),

- c is the speed of light (approximately 3×10^8 m/s).

The path loss exponent n characterizes the rate at which the path loss increases with distance. Its value depends on the specific propagation environment:

- In free space, $n = 2$.
- In urban area cellular radio, n typically ranges from 2.7 to 3.5.
- In shadowed urban cellular radio, n can range from 3 to 5.
- In building line-of-sight environments, n can range from 1.6 to 1.8.
- In obstructed in-building environments, n can range from 4 to 6.
- In obstructed in factories environments, n can range from 2 to 3.

The received power $P_r(d)$ at distance d can be expressed based on the received power at a reference distance d_0 using the following relationship:

$$P_r(d) = P_r(d_0) \cdot \left(\frac{d_0}{d}\right)^n$$

where:

- $P_r(d)$ is the received power at distance d (in watts, W),
- $P_r(d_0)$ is the received power at reference distance d_0 (in watts, W).

The Log-Distance Path Loss Model is used in various applications such as:

- *Cellular Network Planning:* Helps in determining cell coverage areas and planning base station placements.
- *Wireless Sensor Networks:* Assists in estimating communication range and network connectivity.

- *Indoor Positioning Systems:* Used to model signal strength for location estimation.
- *Link Budget Analysis:* Aids in calculating the link budget for ensuring reliable communication links.

The Log-Distance Path Loss Model is a practical tool for predicting path loss in various wireless communication environments. By incorporating the path loss exponent and, optionally, the shadowing effect, it provides a more realistic representation of signal attenuation compared to the Free Space Path Loss model. Understanding this model is essential for the effective design and optimization of wireless communication systems.

2.3.3 Federated Learning

Federated learning is a decentralized approach to machine learning where multiple clients collaboratively train a global model without directly sharing their data. Unlike traditional centralized machine learning, where all data is aggregated and processed in a central server, FL allows model training to occur locally on devices, transmitting only model updates instead of raw data. This decentralized strategy addresses concerns related to data privacy, bandwidth, and scalability.

With the rise of privacy regulations, such as the General Data Protection Regulation (GDPR) and the expansion of edge devices (e.g., smartphones, IoT sensors), the need to train models on distributed, sensitive, and heterogeneous data has increased. FL provides a solution by enabling collaborative learning across these decentralized data sources without the need to centralize the data itself. This leads to significant benefits in terms of privacy preservation, reduced communication costs, and scalability.

Federated Learning Workflow

The FL process involves a series of key steps that allow for training a global model using distributed data while preserving data privacy. The standard workflow is as follows:

1. **Local Training:** Each client n starts by downloading the current global model parameters $w^{(t)}$ from the central server. With these parameters, each client trains its local model using its own dataset $\mathcal{D}_n$. The local loss function for client n is defined as:

$$F_n(\boldsymbol{w}) = \frac{1}{D_n} \sum_{(x_i, y_i) \in \mathcal{D}_n} f_n(\boldsymbol{w}, x_i, y_i),$$

   Here, D_n denotes the number of data samples at client n.

   where $f_n(\boldsymbol{w}, x_i, y_i)$ represents the loss function for client n. This could be any suitable loss function depending on the local solver used. The local training process updates the model parameters to minimize this loss function over the client's local data.

2. **Model Update:** After completing local training, each client n sends its updated model parameters $w_n^{(t+1)}$ to the central server. This transmission includes the new parameters and possibly additional metadata, such as the number of data samples used in training.

3. **Global Aggregation:** The server aggregates the received model updates to form a new global model. The aggregation process is typically done by computing a weighted average of the received model parameters, based on the amount of data each client contributed. The updated global model $w^{(t+1)}$ is computed as:

$$w^{(t+1)} = \frac{\sum_{n=1}^{N} D_n w_n^{(t+1)}}{\sum_{n=1}^{N} D_n}.$$

4. **Iteration:** The process repeats by distributing the updated global model $w^{(t+1)}$ back to the clients for the next round of training. This iterative process continues until the model converges to a satisfactory level of performance. Convergence is measured based on the global model's accuracy or loss, with the goal being to achieve a global model that performs well across all clients.

   The number of global iterations needed to achieve convergence depends on various factors, including the local and global accuracy and efficiency of the local training procedures. For example, if local training involves iterative optimization algorithms, the number of iterations required to reach local accuracy θ can impact the overall number of global iterations needed for convergence.

   The convergence of the global model is typically assessed by monitoring the changes in model accuracy or loss across iterations. The exact number of iterations required is influenced by the performance of both the global aggregation process and the local training algorithms used by the clients.

Key Concepts in Federated Learning

FL introduces several challenges compared to traditional centralized training, particularly in the context of data heterogeneity, communication efficiency, and privacy.

Data Heterogeneity (Non-IID Data)

One of the main challenges in FL is the non-IID (non-Independent and Identically Distributed) nature of the data across clients. Since clients typically generate and store data in specific contexts, their data distributions can be highly skewed. For instance, user behavior on mobile devices may vary greatly across users, leading to significant differences in local data distributions.

Communication Efficiency

Since model updates are exchanged frequently between the server and clients, communication efficiency is a primary concern in FL. Techniques to improve communication efficiency include:

- **Compression:** Techniques such as quantization and sparsification reduce the size of transmitted model updates [43–48].
- **Partial Model Updates:** Transmitting only significant or selected portions of the model.
- **Client Sampling:** In each round, only a subset of clients is selected to participate, reducing the total communication overhead [10].

Privacy and Security

FL inherently preserves privacy by keeping raw data on devices, but additional techniques are often needed to enhance privacy and security further:

- **Differential Privacy:** Adds noise to model updates before sharing to prevent individual data points from being inferred.
- **Secure Aggregation:** Cryptographic techniques like homomorphic encryption or multi-party computation enable the server to aggregate model updates without accessing individual updates.

Core Algorithms in Federated Learning

Several algorithms have been developed to address the unique challenges in FL, focusing on aspects such as non-IID data handling, efficient communication, and stable optimization. Below are some key algorithms:

Federated Averaging (FedAvg)

FedAvg is one of the most widely used algorithms in federated learning. It was also among the very first FL algorithms introduced. It combines local stochastic gradient descent (SGD) with global model averaging. The algorithm begins with the server initializing the global model with parameters $w^{(0)}$. At each training round t, a subset S_t of clients is selected to participate. Each selected client $n \in S_t$ then downloads the global model $w^{(t)}$ and performs local training, updating its model according to the rule $w_n^{(t+1)} = w^{(t)} - \eta \nabla F_n(w^{(t)})$, where η is the learning rate and $F_n(w)$ represents the local loss function for client n. After local training, the server aggregates the updated models from all clients using a weighted average formula: $w^{(t+1)} = \frac{\sum_{n=1}^{N} D_n w_n^{(t+1)}}{\sum_{n=1}^{N} D_n}$, where D_n represents the number of data samples held by client n. This process of client selection, local training, and global aggregation is repeated iteratively until the model converges. While FedAvg is highly effective in scenarios with IID (independent and identically distributed) data, its performance can degrade in non-IID settings due to the variance in local models across clients.

Federated Proximal (FedProx)

FedProx is a variant of FedAvg that introduces a proximal term in the local objective function, which helps stabilize training in the presence of non-IID data. The modified objective for each client is:

$$\min_{w} F_n(w) + \frac{\mu}{2} \|w - w^{(t)}\|^2,$$

where μ is a regularization parameter controlling the deviation from the global model $w^{(t)}$. By restricting the local models from drifting too far, FedProx improves robustness in heterogeneous data environments.

DANE (Distributed Approximate Newton Method)

DANE is a distributed optimization algorithm designed to handle the heterogeneity of data by combining local quadratic approximations with global consensus updates. Unlike FedAvg, which simply averages local updates, DANE aligns local updates more closely with the global objective using a quadratic approximation.

Each client solves the following local subproblem:

$$\min_w \left[L_n(w) + \frac{\rho}{2} \|w - w^{(t)}\|^2 \right],$$

where ρ is a parameter controlling the strength of regularization. DANE's quadratic regularization allows it to be more resilient to non-IID data distributions by better balancing local and global objectives.

COCOA (Communication-Efficient Distributed Dual Coordinate Ascent)

COCOA is a communication-efficient distributed optimization algorithm that focuses on solving convex optimization problems in the dual space. Unlike FedAvg, which works in the primal space, COCOA leverages dual coordinate ascent, leading to better performance and faster convergence in some scenarios.

COCOA optimizes the dual objective:

$$\min_\alpha \sum_{n=1}^{N} g_n(\alpha_n) + \frac{\lambda}{2} \| \sum_{n=1}^{N} \alpha_n \|^2,$$

where α_n represents the dual variables for client n, and $g_n(\alpha_n)$ is the dual objective specific to each client. COCOA allows clients to perform multiple local updates before transmitting compressed dual updates, which reduces communication overhead while maintaining good model performance.

Semi-Stochastic Coordinate Descent (S2CD)

Key ideas:

- **Random coordinate selection:** A subset of coordinates is selected randomly at each iteration.
- **Exact gradient updates:** The selected coordinates are updated using the exact gradient information.
- **Local accuracy:** A parameter controls the level of accuracy required for the updates of the selected coordinates.

Bound on local iterations: Let us denote:

- f as the objective function to be minimized.
- $x \in \mathbb{R}^n$ as the optimization variable.
- S_t as the random subset of coordinates selected at iteration t.
- θ as the desired level of accuracy.

The S2CD algorithm updates the selected coordinates x_{S_t} using the following rule:

$$x_{S_t} := x_{S_t} - \alpha_t \nabla_{S_t} f(x)$$

where α_t is the step size at iteration t and $\nabla_{S_t} f(x)$ is the gradient of f with respect to the coordinates in S_t.

The bound on local iterations for S2CD is typically derived using techniques from optimization theory, such as the Polyak-Ribière conjugate gradient method. The exact bound depends on the properties of the objective function and the desired level of accuracy. However, a common result is that the number of local iterations required to achieve θ-accuracy is proportional to $\log(1/\theta)$.

Semi-Stochastic Gradient Descent (S2GD)

Key ideas:

- **Random coordinate selection:** A subset of coordinates is selected randomly at each iteration.
- **Stochastic gradient updates:** The selected coordinates are updated using stochastic gradient estimates.
- **Local accuracy:** A parameter controls the level of accuracy required for the updates of the selected coordinates.

Bound on local iterations: The bound on local iterations in S2GD is similar to that of S2CD, but it may be slightly looser due to the use of stochastic gradient estimates. The exact bound depends on the variance of the stochastic gradient estimates and the desired level of accuracy.

However, a common result is that the number of local iterations required to achieve θ-accuracy is proportional to $\log(1/\theta)$, with a constant factor that depends on the variance of the stochastic gradient estimates.

Comparison of S2CD and S2GD

- **Convergence rate:** Both S2CD and S2GD can achieve faster convergence rates than full CD or SGD, especially for large-scale problems.

- **Accuracy:** S2CD generally provides higher accuracy than S2GD due to the use of exact gradients.

- **Computational efficiency:** S2GD can be more computationally efficient than S2CD for large-scale problems where computing exact gradients is expensive.

- **Generalization performance:** Both S2CD and S2GD can achieve good generalization performance, especially when combined with techniques like regularization.

Distributed Optimization with Arbitrary Local Solvers

The "Distributed Optimization with Arbitrary Local Solvers" [40] proposes a framework that generalizes many FL algorithms by providing a flexible approach to how local models are updated. Instead of prescribing a specific algorithm like FedAvg or COCOA, this framework allows clients to use any local solver for optimization. The key idea is to decouple the local updates from the global consensus step, enabling more efficient and diverse optimization strategies across clients.

Mathematically, the framework involves clients optimizing their local models according to their preferred method and then sending updates to the server, which aggregates them using an appropriate consensus mechanism. The global update is typically expressed as:

$$w^{(t+1)} = w^{(t)} + \gamma \sum_{n=1}^{N} \frac{D_n}{\sum_{n=1}^{N} D_n} (w_n^{(t+1)} - w^{(t)}),$$

where γ is a step-size parameter controlling the contribution of local updates to the global model.

This framework is especially useful when clients have different computational resources or prefer different optimization techniques, allowing for a more adaptive and scalable FL system.

Application in This Work

As discussed, there have been lots of theoretical state-of-the-art works advancing the field of federated learning and forming fundamental grounds for future studies to build on [16, 39–42]. In this work, we utilize some of the most significant results of these studies, which are hereby explained.

1. The upper bound on the number of local iterations required to solve the local problem for a wide range of iterative algorithms is $O(\log(1/\theta))$, where $0 \leq \theta \leq 1$ represents the desired local accuracy with which the local problem needs to be solved. Here, $\theta = 0$ means optimal solving of the local problem is required and $\theta = 1$ means no progress in solving the local problem. [16, 41, 42].

2. the number of global iterations required for the global model to converge with respect to global accuracy $0 \leq \varepsilon \leq 1$ and local accuracy θ is shown to be [16, 40]:

$$K(\varepsilon, \theta) = \frac{\mathcal{O}(\log(1/\varepsilon))}{1 - \theta}. \tag{2.1}$$

These results hold for a μ-strongly convex loss function, which is a common assumption in the literature. A function $f(x)$ is called μ-strongly convex if, for some $\mu > 0$, it satisfies the following condition for all x, y in its domain:

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2$$

This condition implies that the function is not only convex but has a certain level of curvature controlled by μ, which ensures faster convergence in optimization.

We will follow a similar system model and assumptions in this work and utilize these results in our problem formulation.

2.3.4 Differential Evolution

Differential Evolution (DE) [52] is a population-based optimization algorithm used to solve complex, non-linear, and multi-dimensional problems. As a meta-heuristic, DE operates by evolving a population of candidate solutions over generations to find the optimal solution. The algorithm starts by randomly initializing a population of vectors, each representing a potential solution. At each iteration, three key steps occur: mutation, crossover, and selection. In the mutation step, a donor vector is generated by combining the differences between randomly selected population members. This donor vector is then combined with the current vector in the crossover step to produce a trial vector. The trial vector is compared with the current vector, and if it performs better, it replaces the current vector in the next generation. This process allows the population to progressively converge toward the optimal solution.

The strength of DE lies in its simplicity and its ability to handle complex, noisy, or non-differentiable objective functions. It is robust against getting stuck in local optima, making it particularly powerful for problems where traditional methods struggle. Compared to other meta-heuristic algorithms, such as Genetic Algorithms and Particle Swarm Optimization, DE typically requires fewer parameters to be fine-tuned, which can simplify its implementation. However, DE also excels in maintaining diversity within the population, which enhances exploration while balancing the exploitation of known good solutions. DE is widely used in various applications, such as engineering design (e.g., optimizing structures or systems), control systems, and machine learning for hyperparameter tuning. Its ability to efficiently explore large, complex spaces and find global solutions makes it an attractive choice for a wide range of optimization tasks. We will use DE in this work as a benchmark for comparison to our results.

Chapter 3

Optimal Algorithm on Bandwidth Minimization in Wireless Federated Learning

In this chapter, we will explain the system model and the problems in our work. Then, we provide the details of our solution to these problems. Finally, we will explore the numerical results in various scenarios.

3.1 System Model

We consider a wireless FL system consisting of a BS equipped with an edge server and N learning users. Let $\mathcal{N} = \{1, 2, \cdots, N\}$ denote the set of FL users and $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{D_n}$, denote the local dataset of user n. An illustration of the system is shown in Figure 3.1. For data sample i, let x_i and y_i denote the input data and its corresponding output, respectively. Also, $D_n = |\mathcal{D}_n|$ indicates the number of samples in the n-th local dataset.

3.1.1 Federated Learning Process

Each user n, training a local model, has a local loss function $F_n(\boldsymbol{w})$ over its dataset $\mathcal{D}_n$, which is obtained as:

$$F_n(\boldsymbol{w}) = \frac{1}{D_n} \sum_{(x_i, y_i) \in \mathcal{D}_n} f_n(\boldsymbol{w}, x_i, y_i) . \tag{3.1}$$

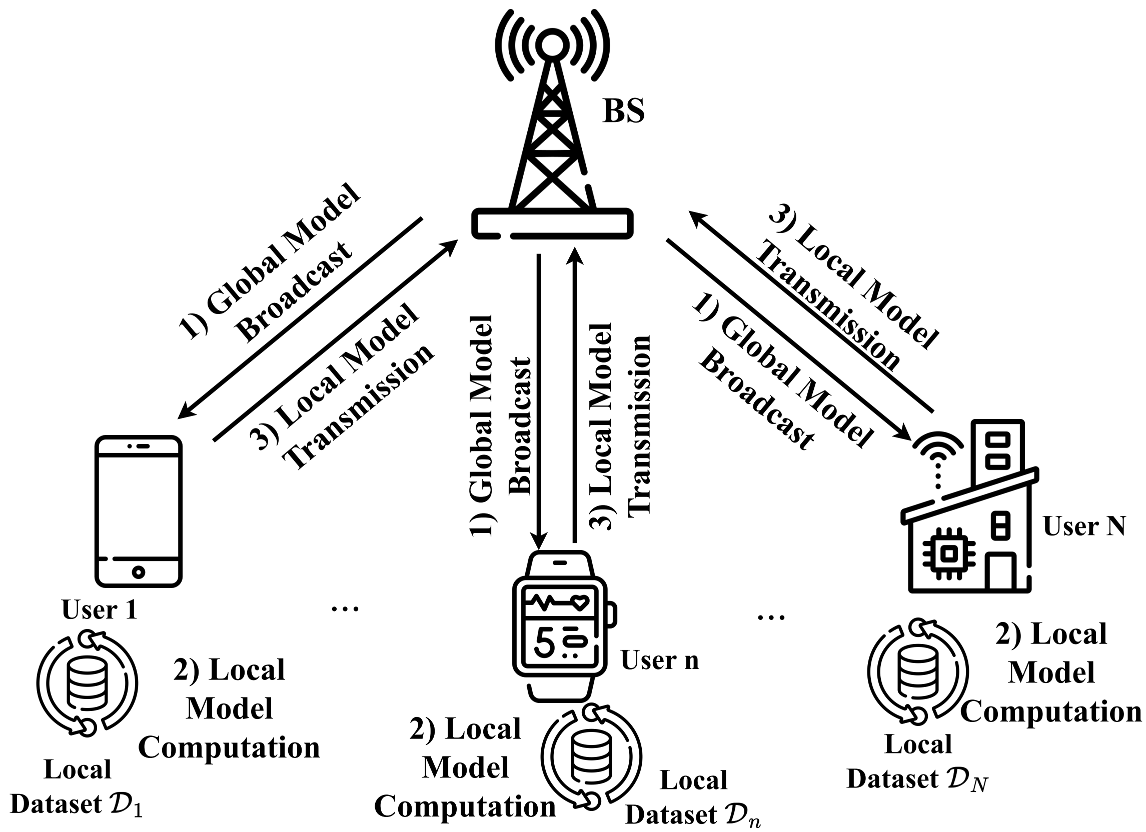


Figure 3.1: System Model

In (3.1), $f_n(\boldsymbol{w}, x_i, y_i)$ is a loss function such as $f_n(\boldsymbol{w}, x_i, y_i) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$ for linear regression, or $f_n(\boldsymbol{w}, x_i, y_i) = \max\{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$ for support vector machine. In an FL system, the objective is to find the optimal training model parameter w^*, which minimizes the loss function

$$F(\boldsymbol{w}) \triangleq \frac{\sum_{n=1}^{N} D_n F_n(\boldsymbol{w})}{\sum_{n=1}^{N} D_n}. \tag{3.2}$$

Each round of the FL process, i.e. a global iteration, includes the following steps:

1. *Global Model Broadcast:* The edge server sends the latest global model parameters to all users. In this paper, we ignore the downlink time due to the high BS power and high downlink bandwidth compared to uplink[12].

2. *Local Model Computation:* Using the received global model parameters, each user trains its model on its local dataset iteratively to achieve a model update

with a local accuracy $0 \leq \theta \leq 1$, i.e., $\|\nabla F_n\left(w_n^{(r)}\right)\| \leq \theta\|\nabla F_n\left(w_n^{(r-1)}\right)\|$ [16]. Here, ∇F_n and $w_n^{(r)}$ denote the gradient of the local loss function and the model parameter of user n at local iteration r, respectively.

3. *Local Model Transmission:* At this stage, users upload their model parameters updates to the BS, and the server aggregates them to update the global model.

These steps are repeated until the loss function converges to a global accuracy $0 \leq \varepsilon \leq 1$, i.e., $\|\nabla F\left(w^{(m)}\right)\| \leq \varepsilon\|\nabla F\left(w^{(m-1)}\right)\|$, where $w^{(m)}$ denotes the global model parameters at round m. To achieve this objective, the number of global iterations required with respect to global accuracy ε and local accuracy θ is shown to be [16]:

$$K(\varepsilon, \theta) = \frac{\mathcal{O}(\log(1/\varepsilon))}{1-\theta}. \tag{3.3}$$

Assuming that the global accuracy is fixed, $\mathcal{O}(\log(1/\varepsilon))$ can be normalized to 1 for ease of presentation to get $K(\theta) = \frac{1}{1-\theta}$. In addition, the number of local iterations also depends on the local accuracy θ. The number of local iterations required to achieve accuracy θ is shown to be $\log(1/\theta)$ [12, 13, 16, 24]. As can be observed, the number of both global and local iterations is affected by the local accuracy. Thus, it is important to involve the local accuracy of an FL system in any efficient design.

3.1.2 Computation Model

For any FL user n, let C_n denote the number of required CPU cycles to process a data sample for training. Assuming all data samples are of the same size, the total number of required CPU cycles for the user to perform one local iteration would be C_nD_n. Therefore, the total time for a single local iteration at device n can be defined as[12, 16]

$$T_n^{cmp} = \frac{C_nD_n}{f_n}, \tag{3.4}$$

where f_n is the CPU-cycle frequency of user n. Furthermore, we can model the energy consumption of one local iteration at device n as [10, 12]

$$E_n^{cmp} = k_nC_nD_nf_n^2, \tag{3.5}$$

where k_n denotes the effective switched capacitance for user n, a quantity that depends on the CPU architecture[13].

3.1.3 Communication Model

In this subsection, we will model the latency and the energy consumption of the FL steps. We will later use these models to consider practical constraints for our problem.

After each user has completed the local model training and has model updates ready to be shared, it transmits the new parameter to the BS. Let S_n denote the size of the updated local model parameter, i.e. w_n, that needs to be sent to the edge server at the end of each global iteration. Then, for uploading local parameters to the server, the total transmission time and the total energy consumed for the transmission can be respectively expressed as follows [12]:

$$T_n^{com} = \frac{S_n}{r_n}, \tag{3.6}$$

$$E_n^{com} = p_n T_n^{com} = p_n \frac{S_n}{r_n}. \tag{3.7}$$

In (3.7), p_n is the transmit power of user n, and r_n denotes the transmit rate of the n-th user and is given by [13]

$$r_n = b_n \log_2 \left(1 + \frac{h_n p_n}{b_n N_0}\right), \quad \forall n \in \mathcal{N}, \tag{3.8}$$

where b_n is the channel bandwidth allocated to user n, h_n is the channel gain between user n and the BS, and N_0 is the power spectral density of the Gaussian noise. Hence, the total training and uploading time for user n in round m equals [12]:

$$T_n^{(m)} = \log(1/\theta) T_n^{cmp} + T_n^{com}. \tag{3.9}$$

Finally, the total energy consumed by each user for round m and for the whole process, respectively, can be written as[13]:

$$E_n^{total} = \frac{1}{1-\theta} E_n^{(m)} = \frac{1}{1-\theta} (\log(1/\theta) E_n^{cmp} + E_n^{com}). \tag{3.10}$$

Many applications of FL are time-sensitive and the participating devices are energy-limited. Given this, it is desirable to limit the user's completion time in each global iteration, consisting of local model computation and local model transmission, as well as its total energy consumption across all global iterations of the FL process. This observation is reflected in our work as constraints in the formulated problem.

3.2 Problem Formulation

As previously discussed, efficient utilization of the channel bandwidth, as an extremely valuable communication resource, is significantly important. Hence, we aim to minimize the total bandwidth usage of the wireless FL system described in Section 3.1. To this end, we define the objective function of our problem to be the sum of the channel bandwidths allocated to all the users. We set our optimization variables to be the local accuracy θ, the set of CPU cycle frequencies $f = \{f_n\}, n \in \mathcal{N}$, and the set of channel bandwidths $b = \{b_n\}, n \in \mathcal{N}$. We also consider constraints on the user's completion time in each global iteration m, and the total energy consumption of each user across all global iterations. These two values are expressed as $T_n^{(m)}$ in (3.9) and E_n^{total} in (3.10), respectively. Thus, to limit the maximum energy consumed by a user, we substitute (3.5) and (3.7) in (3.10). Then, we set the result to be at most γ_E. We also derive the constraint on the user's completion time by substituting (3.4) and (3.6) in (3.9). The corresponding maximum value for $T_n^{(m)}$ is γ_T. We can then formulate the optimization problem as follows:

$$\min_{\theta,b,f} \sum_{n=1}^{N} b_n \tag{3.11a}$$

$$\text{s.t. } 0 \leq f_n \leq f_n^{\max}, \forall n, \tag{3.11b}$$

$$\log(1/\theta)\frac{C_n D_n}{f_n} + \frac{S_n}{r_n} \leq \gamma_T, \forall n \tag{3.11c}$$

$$0 \leq b_n \leq B_{max}, \forall n, \tag{3.11d}$$

$$\frac{1}{1-\theta}(\log(1/\theta)k_n C_n D_n f_n^2 + p_n \frac{S_n}{r_n}) \leq \gamma_E, \forall n \tag{3.11e}$$

$$0 \leq \theta \leq 1. \tag{3.11f}$$

In this problem, constraint (3.11b) determines the feasible range of CPU-cycle frequency of the participating users. For each user n, (3.11c) indicates that the total time for computing and uploading the results cannot exceed γ_T. Constraint (3.11d) limits the bandwidth allocated to each user to B_{max}. In (3.11e), we limit the maximum total energy consumption of each user across the entire global iterations of the FL process to γ_E. The feasible range of the local accuracy is given by (3.11f).

3.3 Solution

The proposed problem in (3.11) is not a convex optimization problem with respect to θ, f, and b. However, we realize that the parameter causing this non-convexity is the local accuracy θ. That means assuming θ is fixed, (3.11) turns into a convex optimization problem with respect to b and f. We take advantage of this fact and decompose the problem into two sub-problems. First, assuming θ is a fixed given value θ_c, we obtain f and b in terms of θ_c. Then, we find the optimal value for θ_c which minimizes the total bandwidth usage. In the first step, we reformulate the non-convex problem in (3.11) into the following optimization problem

$$\text{SUB1:} \min_{b,f} \sum_{n=1}^{N} b_n \tag{3.12a}$$

$$\text{s.t. } 0 \leq f_n \leq f_n^{\max}, \forall n, \tag{3.12b}$$

$$\log(1/\theta_c)\frac{C_n D_n}{f_n} + \frac{S_n}{r_n} \leq \gamma_T, \forall n \tag{3.12c}$$

$$0 \leq b_n \leq B_{max}, \forall n, \tag{3.12d}$$

$$\frac{1}{1-\theta_c}(\log(1/\theta_c)k_n C_n D_n f_n^2 + p_n \frac{S_n}{r_n}) \leq \gamma_E, \forall n, \tag{3.12e}$$

where θ is a given constant $0 \leq \theta_c \leq 1$. We then solve (3.12) to obtain f_{n,θ_c} and b_{n,θ_c} in terms of θ_c. Note that the constraints in SUB1 are the same as the original problem in (3.11), except (3.11f) which is omitted because θ is not a parameter in this case and is assumed a given constant. Here, we assume that substituting the given θ_c in f_{n,θ_c} and b_{n,θ_c}, does not violate constraints (3.12b) to (3.12e). We later include these constraints in our next sub-problem to ensure this assumption is fulfilled.

After solving the first sub-problem in (3.12), we obtain the optimal bandwidths and CPU-cycle frequencies for a given constant local accuracy θ_c. However, the optimal solution to the problem in (3.12) still relies on θ_c. Hence, we then need to find the optimal answer to θ_c, denoted by θ^*, that minimizes the total bandwidth usage. To this end, we formulate a second sub-problem as follows:

$$\text{SUB2:} \min_{\theta_c} \sum_{n=1}^{N} b_{n,\theta_c} \tag{3.13a}$$

$$\text{s.t. } 0 \leq f_{n,\theta_c} \leq f_n^{\max}, \forall n, \tag{3.13b}$$

$$\log(1/\theta_c)\frac{C_n D_n}{f_{n,\theta_c}} + \frac{S_n}{r_{n,\theta_c}} \leq \gamma_T, \forall n \tag{3.13c}$$

$$0 \leq b_{n,\theta_c} \leq B_{max}, \forall n, \tag{3.13d}$$

$$\frac{1}{1-\theta_c}(\log(1/\theta_c)k_n C_n D_n f_{n,\theta_c}^2 + p_n \frac{S_n}{r_{n,\theta_c}}) \leq \gamma_E, \forall n \tag{3.13e}$$

$$0 \leq \theta_c \leq 1, \tag{3.13f}$$

where we aim at obtaining θ^*.

To demonstrate the optimality of the solution (b^*, f^*, θ^*) obtained through this approach, we use proof by contradiction. Assume that (b^*, f^*, θ^*) is not the optimal solution to the original problem. This assumption implies that there exists another set of parameters (b', f', θ') such that the objective function value at (b', f', θ') is less than the value at (b^*, f^*, θ^*). We show the contradiction of this assumption in the cases where $\theta' = \theta^*$ and where $\theta' \neq \theta^*$ in the next two paragraphs, respectively.

Firstly, note that for each fixed value of θ_c, the parameters f_{n,θ_c} and b_{n,θ_c} are optimized to minimize the objective function by solving the sub-problem in (11). Therefore, for the specific value θ', the values $f_{n,\theta'}$ and $b_{n,\theta'}$ are chosen such that they yield the lowest possible objective function value given θ'. Hence, if $\theta' = \theta^*$, b' and f' cannot provide a better objective function value than $b(\theta')$ and $f(\theta')$, i.e. the set of $f_{n,\theta'}$ and $b_{n,\theta'}$ for all N users.

Furthermore, since we perform a comprehensive search over all possible values of θ in Algorithm 1, the value θ^* is selected because it results in the global minimum of the objective function across all θ. Thus, (b^*, f^*, θ^*) represents the combination of parameters that achieves the lowest objective function value over all possible values of θ, including θ'. Hence, if θ' had yielded a lower value for the objective function, it would have been chosen instead of θ^* during our iterative search.

Therefore, our initial assumption that (b^*, f^*, θ^*) is not optimal must be false, as it contradicts the fact that for any given θ, the parameters b and f are optimally chosen, and θ^* is the value that minimizes the objective function globally between all possible values of θ. Thus, the solution (b^*, f^*, θ^*) is indeed the optimal solution to the original problem.

In the next subsections, we provide the solutions to the sub-problems (3.12) and (3.13).

$$H_t = \begin{pmatrix} \frac{2h^2p^2S\log(2)}{b^5N_0^2R^2\log^3(R)} + \frac{h^2p^2S\log(2)}{b^5N_0^2R^2\log^2(R)} - \frac{4hpS\log(2)}{b^4N_0R\log^2(R)} + \frac{2S\log(2)}{b^3\log(R)} & 0 \\ 0 & \frac{2CD\log\left(\frac{1}{\theta_c}\right)}{f^3\log(2)} \end{pmatrix} \tag{3.14a}$$

$$R = 1 + \frac{hp}{bN_0} \tag{3.14b}$$

3.3.1 Proof of Convexity of SUB1

We mainly focus on proving the convexity for constraints (3.12c) and (3.12e), since the convexity of the objective function and other constraints are evident. To this end, we will use the second-order condition of convexity, i.e. function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if $\nabla^2 f(x) \succeq 0$ for all $x \in \mathrm{dom}(f)$. In this statement, $\nabla^2 f(x)$ represents the Hessian matrix of the function $f(x)$, $\succeq 0$ denotes that the Hessian matrix is positive semi-definite, and $\mathrm{dom}(f)$ is the domain of the function $f(x)$.

We will start the proof for (3.12c) by deriving its Hessian matrix H_t. For the sake of simplicity and without loss of generality, we omit the parameters b_m and f_m where $m \neq n$ in the Hessian matrices and the rest of the proof, since they all turn into zero elements. We also avoid repeating the n indices in the equations. The Hessian matrix of $\log(1/\theta_c)\frac{C_nD_n}{f_n} + \frac{S_n}{r_n}$ with respect to b_n and f_n is shown in (3.14).

We know matrix H_t is positive semi-definite if and only if its eigenvalues are non-negative. The eigenvalues of a diagonal matrix are the elements on its main diagonal. For matrix H_t, the eigenvalue equal to $H_{t_{2,2}}$, i.e. the entry in the second row and second column of matrix H_t, obviously cannot be negative in the domain of our problem, since all the present variables are non-negative and $1 \leq \frac{1}{\theta_c}$. With some mathematical manipulation on $H_{t_{1,1}}$ one can simplify it as (3.15). Using the non-negativity of the present variables in the equation, it can be shown that (3.15) is also non-negative. The only part in (3.15) that is not evidently contributing to its non-negativity, is the term inside the parenthesis in the numerator. This term is

$$H_{t_{1,1}} = \frac{S\log(2)\left(2(bN_0+hp)^2\log^2(R) - hp(4bN_0+3hp)\log(R) + 2h^2p^2\right)}{b^3(bN_0+hp)^2\log^3(R)} \tag{3.15}$$

$$H_e = \begin{pmatrix} \frac{\frac{2h^2p^3S\log(2)}{b^5N_0^2R^2\log^3(R)} + \frac{h^2p^3S\log(2)}{b^5N_0^2R^2\log^2(R)} - \frac{4hp^2S\log(2)}{b^4N_0R\log^2(R)} + \frac{2pS\log(2)}{b^3\log(R)}}{1-\theta_c} & 0 \\ 0 & \frac{2CDk\log\left(\frac{1}{\theta_c}\right)}{(1-\theta_c)\log(2)} \end{pmatrix} \tag{3.16}$$

a quadratic of the form $ax^2 + bx + c$, considering $\log\left(1 + \frac{hp}{bN_0}\right)$ as x. Calculating its discriminant $\Delta = -h^2p^2\left(8bN_0hp + 7h^2p^2\right)$, we realize the discriminant to this function is negative in our domain. Hence, since one can see that the equation is positive for arbitrary values of the variables in the domain of the problem, the function is positive throughout the domain. Thus, we can prove the positive semi-definiteness of the H_t matrix and hence the convexity of (3.12c).

Now, we move forward to prove the convexity for constraint (3.12e). Similarly, we want to obtain its hessian matrix, denoted by H_e, and prove that it is positive semi-definite. H_e is derived and shown in (3.16).

As can be seen, (3.16) is also a diagonal matrix with its eigenvalues on its main diagonal. It is evident that $H_{e_{2,2}}$ cannot be negative in the domain of our problem, considering $0 \leq \theta_c \leq 1$ and the non-negativity of other present variables. It can also be noted that $H_{e_{2,2}} = \frac{p}{1-\theta_c}H_{t_{2,2}}$. Thus, as $H_{t_{2,2}}$ is non-negative, $H_{e_{2,2}}$ cannot be negative as well. Hence, the eigenvalues of the hessian matrix for constraint function in (3.12e) are proven to be non-negative, which proves its convexity.

Therefore, the constraints and the objective function are convex in sub-problem (3.12), and hence it is a convex optimization problem.

3.3.2 SUB1 Solution

We can observe that the sub-problem in (3.12) is now a convex problem. Therefore, one can leverage the Karush-Kuhn-Tucker (KKT) conditions analysis to derive the optimal local CPU-cycle frequencies f_{n,θ_c} and bandwidth allocations b_{n,θ_c}[50]. The

$$f_{n,\theta_c} = \frac{\sqrt[3]{27C_n^3D_n^3k_n^2p_n\log^3\left(\frac{1}{\theta_c}\right)+A}}{3\sqrt[3]{2}C_nD_nk_n\log\left(\frac{1}{\theta_c}\right)} - \frac{\sqrt[3]{2}\log^3(2)(\gamma_T p_n + \gamma_E\theta_c - \gamma_E)}{\sqrt[3]{27C_n^3D_n^3k_n^2p_n\log^3\left(\frac{1}{\theta_c}\right)+A}} \tag{3.17a}$$

$$A = \sqrt{729C_n^6D_n^6k_n^4p_n^2\log^6\left(\frac{1}{\theta_c}\right) + 108C_n^3D_n^3k_n^3\log^3\left(\frac{1}{\theta_c}\right)\log^3(2)(\gamma_T p_n + \gamma_E\theta_c - \gamma_E)^3} \tag{3.17b}$$

$$b_{n,\theta_c} = -\frac{f_nh_np_nS_n\log^2(2)}{h_np_n\left(\gamma_T f_n\log(2) - C_nD_n\log\left(\frac{1}{\theta_c}\right)\right)W(z) + f_nN_0S_n\log^2(2)} \tag{3.18a}$$

$$z = -\frac{f_nN_0S_n\log^2(2)\exp\left(-\frac{f_nN_0S_n\log^2(2)}{\gamma_T f_nh_np_n\log(2) - C_nD_nh_np_n\log\left(\frac{1}{\theta_c}\right)}\right)}{\gamma_T f_nh_np_n\log(2) - C_nD_nh_np_n\log\left(\frac{1}{\theta_c}\right)} \tag{3.18b}$$

---

following theorem gives the optimal solution to the sub-problem in (3.12).

Theorem 1 *The optimal solution to the problem (3.12), i.e. the local CPU-cycle frequency and channel bandwidth allocated for n-th user are given in (3.17) and (3.18), respectively, where W is the Lambert W function [53].*

Proof: To begin delving into the KKT conditions, the Lagrangian function of the problem can be expressed as:

$$\begin{aligned} L(f,b,\lambda) &= \sum_n b_n + \sum_n \lambda_{1,n}\left(f_n - {f_n}^{\max}\right) + \sum_n \lambda_{2,n}\left(-f_n\right) \\ &+ \sum_n \lambda_{3,n}\left(\log(1/\theta_c)\frac{C_nD_n}{f_n} + \frac{S_n}{r_n} - \gamma_T\right) \\ &+ \sum_n \lambda_{4,n}\left(b_n - B_{\max}\right) + \sum_n \lambda_{5,n}\left(-b_n\right) \\ &+ \sum_n \lambda_{6,n}\left(\frac{1}{1-\theta_c}\left(\log(1/\theta_c)k_nC_nD_nf_n^2 + p_n\frac{S_n}{r_n}\right) - \gamma_E\right) \end{aligned} \tag{3.19}$$

We utilize KKT conditions to analyze the problem and find the solution. Due to complementary slackness, $\forall n \in \mathcal{N}$, we have

$$\lambda_{1,n}\left({f^*}_n - {f_n}^{\max}\right) = 0 \tag{3.20a}$$

$$\lambda_{2,n}\left(-{f^*}_n\right) = 0 \tag{3.20b}$$

$$\lambda_{3,n}\left(\log(1/\theta_c)\frac{C_n D_n}{f^*_n} + \frac{S_n}{r^*_n} - \gamma_T\right) = 0 \tag{3.20c}$$

$$\lambda_{4,n}\left({b^*}_n - B_{\max}\right) = 0 \tag{3.20d}$$

$$\lambda_{5,n}\left(-{b^*}_n\right) = 0 \tag{3.20e}$$

$$\lambda_{6,n}\left(\frac{1}{1-\theta_c}\left(\log(1/\theta_c)k_n C_n D_n {f^*}^2_n + p_n\frac{S_n}{r^*_n}\right) - \gamma_E\right) = 0. \tag{3.20f}$$

Note that r_n is marked as r^*_n because it includes b^*. Additionally, the gradient of the Lagrangian function should vanish for f^* and b^*, i.e.

$$\begin{aligned}\frac{\partial L(f^*, b^*, \lambda^*)}{\partial f_n} &= \lambda_{1,n}(1) + \lambda_{2,n}(-1)\\ &+ \lambda_{3,n}\left(-\frac{CD\log\left(\frac{1}{\theta_c}\right)}{{f^*}^2\log(2)}\right) + \lambda_{6,n}\left(\frac{2kCDf^*\log\left(\frac{1}{\theta_c}\right)}{(1-\theta_c)\log(2)}\right) = 0\end{aligned} \tag{3.21}$$

$$\begin{aligned}\frac{\partial L(f^*, b^*, \lambda^*)}{\partial b_n} &= 1 + \lambda_{3,n}\left(\frac{hpS\log(2)}{(b^*)^3N_0\left(1+\frac{hp}{b^*N_0}\right)\log^2\left(1+\frac{hp}{b^*N_0}\right)}\right.\\ &\left.-\frac{S\log(2)}{(b^*)^2\log\left(1+\frac{hp}{b^*N_0}\right)}\right) + \lambda_{4,n}(1) + \lambda_{5,n}(-1) +\\ &\frac{\lambda_{6,n}}{1-\theta_c}\left(\frac{hp^2S\log(2)}{(b^*)^3N_0\left(1+\frac{hp}{b^*N_0}\right)\log^2\left(1+\frac{hp}{b^*N_0}\right)} - \frac{pS\log(2)}{(b^*)^2\log\left(1+\frac{hp}{b^*N_0}\right)}\right) = 0.\end{aligned} \tag{3.22}$$

We want to prove that the optimal solution is always derived from the case when $\lambda_{3,n} \neq 0$ and $\lambda_{6,n} \neq 0$, which results in (3.17) and (3.18). Hence, the terms in front of them in (3.20c) and (3.20f) must be zero, i.e. constraints (3.12c) and (3.12e) are active. We will do that using proof by contradiction and we show that all the cases except the mentioned one will lead to contradiction.

Step 1: λ_2 and λ_5 must equal zero

If any $\lambda_{2,n}$ or $\lambda_{5,n}$ is non-zero, it means $f_n = 0$ or $b_n = 0$, and primal constraint (3.12c) will not be fulfilled. Hence, due to the primal feasibility condition of the KKT conditions, $\lambda_{2,n}$ and $\lambda_{5,n}$ have to be zero.

Step 2: contradiction of case $\lambda_{3,n} = 0$ and $\lambda_{6,n} = 0$

As can be noted, assuming $\lambda_{3,n}$ and $\lambda_{6,n}$ are both zero, directly leads to the contradiction $1 + \lambda_{4,n} = 0$ in (3.22). Due to KKT conditions, $0 \leq \lambda_{i,n}$ for any $n \in \mathcal{N}$ and $i \in \{1, 2, 3, 4, 5, 6\}$. Hence, we proceed with proof by contradiction to show that all the cases where one of them is zero and one of them is non-zero also lead to contradiction. Next, we will analyze the 4 possible cases for $\lambda_{1,n}$ and $\lambda_{4,n}$.

Step 3: λ_1 cannot be zero

If we consider $\lambda_{1,n} = 0$, and assume only one of $\lambda_{3,n}$ and $\lambda_{6,n}$ is zero, we face a contradiction since the numerator that remains in (3.21) cannot be zero in the domain of our problem. Therefore, we can conclude that $\lambda_{1,n}$ cannot equal 0.

Step 4: contradiction of case $\lambda_{3,n} = 0$ and $\lambda_{6,n} \neq 0$

Subsequently, we can see that if $\lambda_{3,n} = 0$ and $\lambda_{6,n} \neq 0$, we reach a contradiction in (3.21) where one of $\lambda_{1,n}$ or $\lambda_{6,n}$ has to be negative. Thus, the remaining cases to be examined are when $\lambda_{4,n} = 0$ (namely **case I**) and $\lambda_{4,n} \neq 0$ (namely **case II**), both assuming $\lambda_{1,n} \neq 0$, $\lambda_{3,n} \neq 0$and $\lambda_{6,n} = 0$.

Step 5: contradiction of case $\lambda_{3,n} \neq 0$ and $\lambda_{6,n} = 0$

First, we should note that our solution in (3.18) yields the optimal b_{n,θ_c} with respect to f_{n,θ_c} by solving the same case where $log(1/\theta_c)\frac{C_n D_n}{f_n^*} + \frac{S_n}{r_n^*} - \gamma_T = 0$. This is the same as solving for case I, since in both cases I and II $\lambda_{3,n} \neq 0$. Thus, the closed form of b_{n,θ_c} remains the same, but in cases I, II the f_{n,θ_c} must be f_n^{max} due to $\lambda_{1,n} \neq 0$.

If cases I or II yield the same b_{n,θ_c} numerically, we can conclude that their f_{n,θ_c} is also equal, since a closed form solution for b_{n,θ_c} with respect to f_{n,θ_c} is derived. Also, that means our proposed solution is still providing us with the optimal answer. For case I or II to be the solution while our proposed solution is not, it means that at least one smaller b_{n,θ_c} exists in those cases. Since both solutions are satisfying $log(1/\theta_c)\frac{C_n D_n}{f_n^*} + \frac{S_n}{r_n^*} - \gamma_T = 0$, then our proposed f_{n,θ_c} must be smaller. We also note that in our proposed solution, $\lambda_{6,n} \neq 0$, meaning constraint (3.12e) is active. Hence, we can conclude that cases I or II with a smaller b_{n,θ_c} and a larger f_{n,θ_c}, will have a larger result for $\frac{1}{1-\theta_c}\left(\log(1/\theta_c)k_n C_n D_n {f^*}_n^2 + p_n \frac{S_n}{r_n^*}\right) - \gamma_E$ than ours, which is 0. This violates the primal feasibility of the problem for constraint (3.12e). Thus, these cases cannot yield any feasible bandwidth that is lower than our proposed solution.

Hence, we can conclude that if a feasible optimal solution exists, it can be obtained by solving for $\lambda_{3,n} \neq 0$ and $\lambda_{6,n} \neq 0$. Therefore, the optimal solution to problem (3.12) can be derived by solving the system of equations consisting of the equality cases of (3.12c) and (3.12e).

$$\log(1/\theta_c)\frac{C_n D_n}{f_n} + \frac{S_n}{r_n} - \gamma_T = 0, \tag{3.23a}$$

$$\frac{1}{1-\theta_c}(\log(1/\theta_c)k_n C_n D_n f_n^2 + p_n \frac{S_n}{r_n}) - \gamma_E = 0 \tag{3.23b}$$

We can derive $\frac{S_n}{r_n}$ from (3.23a) as

$$\frac{\gamma_T f_n \log(2) - C_n D_n \log\left(\frac{1}{\theta_c}\right)}{f \log(2)}. \tag{3.24}$$

Then, we can substitute (3.24) in (3.23b). By solving the obtained equation, we arrive at optimal f_{n,θ_c} as (3.17). Once we have the f_{n,θ_c}, we can substitute it in (3.23a) or (3.23b) to obtain b_{n,θ_c}. From (3.23a) we have

$$b_n \log_2\left(1 + \frac{h_n p_n}{b_n N_0}\right) = \frac{\gamma_T - \log(1/\theta_c)\frac{C_n D_n}{f_n}}{S_n}. \tag{3.25}$$

By further mathematical manipulation, this yields b_{n,θ_c} as shown in (3.18).

□

Algorithm 1 Iterative Local Accuracy Optimization Algorithm

Input: The constraint tolerance δ, the precision ϵ,

Output: θ^*, f^*, b^*

Initialize $\theta_{temp} \leftarrow 0$, $b_{temp} \leftarrow []$,$f_{temp} \leftarrow []$, $b_{sum}^{min} \leftarrow +\infty$

for $i \leftarrow 0$ to $\frac{1}{\epsilon}$ **do**

    $\theta_{temp} \leftarrow 0 + i * \epsilon$

    $f_{temp} \leftarrow$ list of calculated f_n^*s based on (3.17) using θ_{temp}

    $b_{temp} \leftarrow$ list of calculated b_n^*s based on (3.18) using θ_{temp} and f_{temp}

    if f_{temp},b_{temp}, and θ_{temp} pass the constraints (3.13b) - (3.13f) considering given tolerance δ **then**

        if sum(b_{temp}) $< b_{sum}^{min}$ **then**

            $\theta^* \leftarrow \theta_{temp}$, $b^* \leftarrow b_{temp}$, $f^* \leftarrow f_{temp}$

        end if

    end if

end for

Having solved SUB1, we can use (3.17) and (3.18) to address the second sub-problem and obtain θ^*, which minimizes the overall bandwidth usage.

3.3.3 SUB 2 Solution

So far, we have derived f_{n,θ_c} and b_{n,θ_c} which are functions of θ_c. To solve the problem (3.13), we need to obtain the optimal value of θ_c that minimizes the total bandwidth usage while satisfying (3.13b) to (3.13f). Hence, we propose an algorithm that iterates over θ_c in the feasible range in (3.13f) to find the optimal local accuracy θ^*. In each iteration, the algorithm examines if for a given θ_c, constraints (3.13b) to (3.13f) are satisfied. Finally, among all feasible values of θ_c, we choose the one that minimizes the total bandwidth usage. The details of the proposed solution are shown in Algorithm 1. In Algorithm 1, the loop needs to iterate $\frac{1}{\epsilon}$ times over different values of θ. In each iteration, b_n and f_n need to be calculated for the N users. Thus, it can be realized that it has the computational complexity of $\mathcal{O}(\frac{1}{\epsilon}N)$.

Table 3.1: Parameter Values

Parameter	Value
Maximum CPU-cycle frequency, $f_n^{\max}$	2 GHz
Maximum user bandwidth, $B^{\max}$	0.5 MHz
Processing density, C_n	273.5 cycle/bit
Noise power density, N_0	−174 dBm/Hz
Training data size, D_n	40 KB
Local model parameters size, S_n	4 KB
Tolerable maximum delay, γ_T	2 s
Tolerable maximum energy consumption, γ_E	2 J
Effective switched capacitance k	10^{-27}

3.4 Numerical Results

3.4.1 Simulation Setup

Unless otherwise mentioned, we consider an FL system with $N = 4$ participating users, randomly distributed within a 1000 m radius of the BS. We use the log-distance path loss model to model large-scale fading [51]. Thus, the channel gain of user n is calculated as follows:

$$h_n = h_0(d_n/d_0)^{-\alpha} \tag{3.26}$$

where $\alpha = 4$ is the path loss exponent, and $h_0 = -40$ dB is the channel power at reference distance point of $d_0 = 1$ m [16]. The distance between the BS and user n is denoted by d_n. We set the transmit power of each user n to $p_n = 2$ W, $\forall \mathrm{n} \in \mathcal{N}$. The values of other parameters are summarized in Table 3.1 [17]. The setups and results are independent of other FL metrics since the bound on the iterations is a general bound in FL literature for arbitrary local solvers [40]. For simplicity and without loss of generality, $f_n^{\max}$, S_n, C_n, and D_n are considered to be the same among all users.

Results are averaged over 10^3 Monte Carlo realizations.

3.4.2 Performance Evaluation

Here, various numerical results are presented to demonstrate the performance of the proposed algorithm.

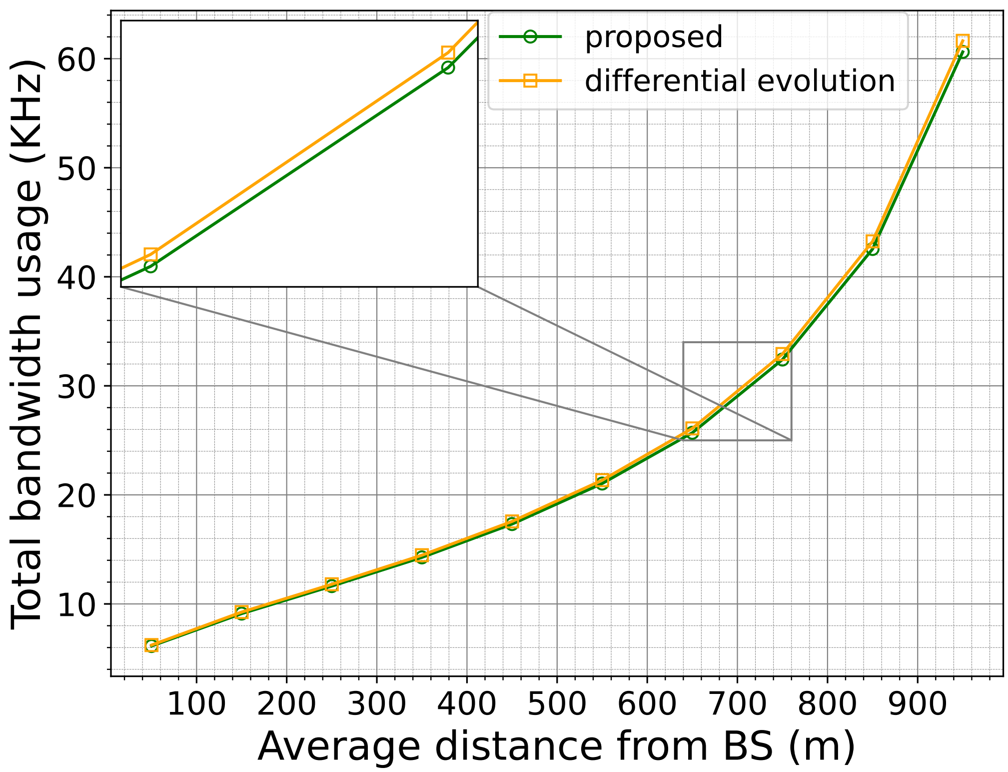


Figure 3.2: Impact of distance on total bandwidth usage.

The impact of the distance between users and the BS on total bandwidth usage is investigated in Fig. 3.2. To examine this, we partitioned the total coverage area into $\{(0, 100), (100, 200) \ldots (900, 1000)\}$, picked one range and positioned all users randomly inside this range. In addition to the proposed algorithm, a global optimization meta-heuristic called Differential Evolution was used for comparison [52]. One can observe our algorithm performs slightly better than DE. However, unlike DE which searches the input space, our proposed algorithm utilizes (3.17) and (3.18) to calculate the optimal solution, which makes it more precise. As expected, one can observe a non-linear gradual increase in the bandwidth usage as users are being positioned further from the BS. The intuition behind that is based on (3.26), (3.11c) and (3.11e), where increasing distance decreases channel gain h_n and transmit rate r_n. Hence, more bandwidth is needed to account for the rate loss and to keep the constraints satisfied.

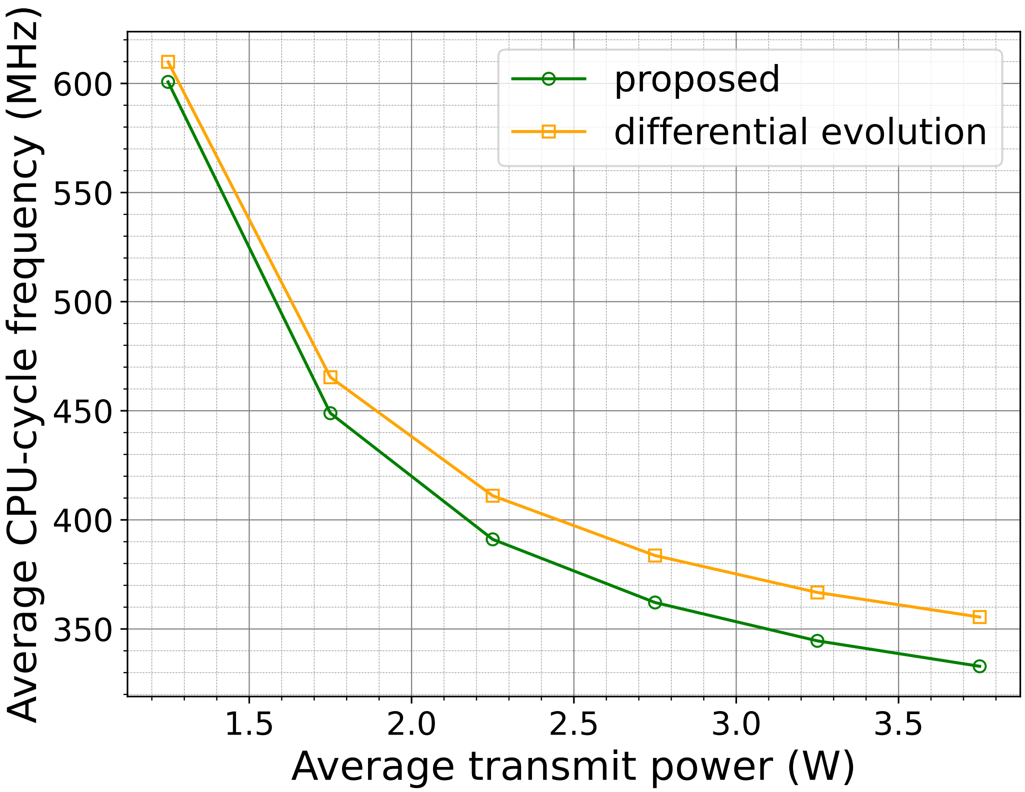


Figure 3.3: Impact of transmit power on the average frequency of each user.

The impact of transmit power on the average CPU-cycle frequency for a user is shown in Fig. 3.3. In this experiment, six different ranges of transmit power are set {(1, 1.5), (1.5, 2), (2, 2.5), (2.5, 3), (3, 3.5), (3.5, 4)} Watts. Each time, all users were assigned a random transmit power from a single range. Other parameters are set as mentioned in Subsection (3.4.1). Our algorithm's optimal solution can result in significant power savings (e.g., about 20% saving around 3 Watts transmit power) compared to DE which also suffers from higher complexity.

The impact of the number of users on the total bandwidth usage is investigated in Fig. 3.4. We have applied our proposed algorithm and DE to {5, 10, 30, 50, 75,

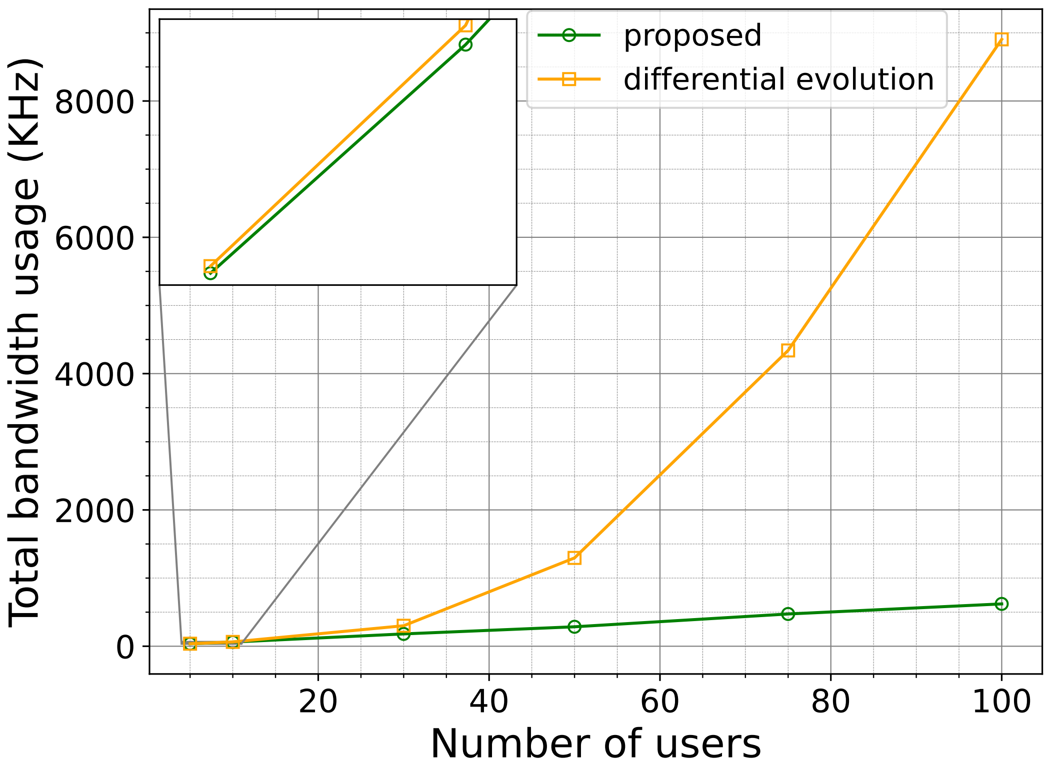


Figure 3.4: Impact of number of users on total bandwidth usage.

100} users and measured the average total bandwidth in each case. One can notice that the average optimal bandwidth usage increases linearly with the number of participating users. As seen, DE does not always find the optimal answer. Moreover, the search space of DE grows exponentially with the number of users resulting in degraded performance and prolonged execution times [54]. In this experiment, in addition to DE's weaker performance, it took on average 7.8 to 46 times longer than our algorithm to execute. Hence, DE cannot be used in time-sensitive applications with a large number of users.

Fig. 3.5 demonstrates the impact of the number of users on the execution time of our proposed algorithm as well as DE. As shown, experimental results confirm that DE suffers from exponential growth in time complexity when increasing the dimensions of the problem, i.e. the number of participating FL users. On the other hand, our proposed algorithm enjoys a linear increase in its execution time with respect to the

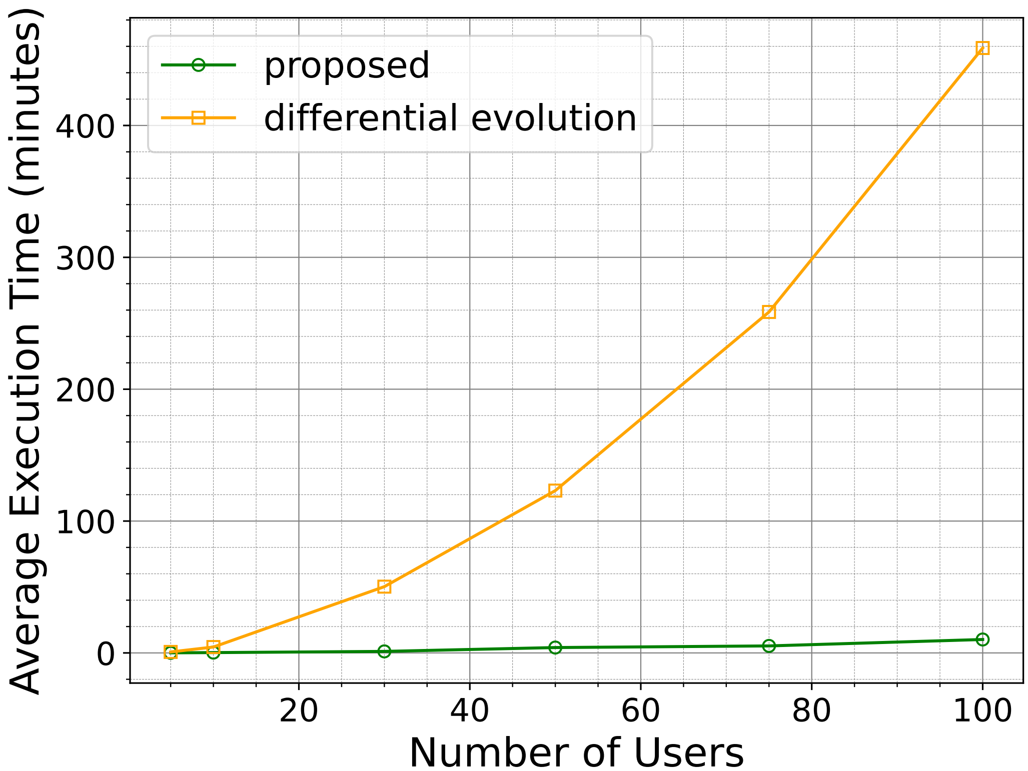


Figure 3.5: Impact of number of users on execution time.

number of users, as expected. This is in addition to its weaker performance shown in Fig. 3.4. Hence, the proposed algorithm not only outperforms DE in terms of complexity but also provides consistently optimal performance. It should be noted that wireless FL systems usually have a large number of participating users [55]. Hence, most FL systems in practice work in the region where the performance gap in Fig. 3.4 and Fig. 3.5 is significant.

In Fig. 3.6, the effect of the transmit power of FL users on the solution of minimum bandwidth usage is investigated. One can observe that increasing the transmit power results in higher bandwidth usage. Although this may seem counter-intuitive at first, it happens as a result of the p_n coefficient in (3.11e). This essentially means that by increasing transmit power, users are consuming more energy for the same time frame. Hence, they need to decrease their transmission time by using more channel bandwidth, to satisfy the energy consumption constraint. Note that the p_n in the

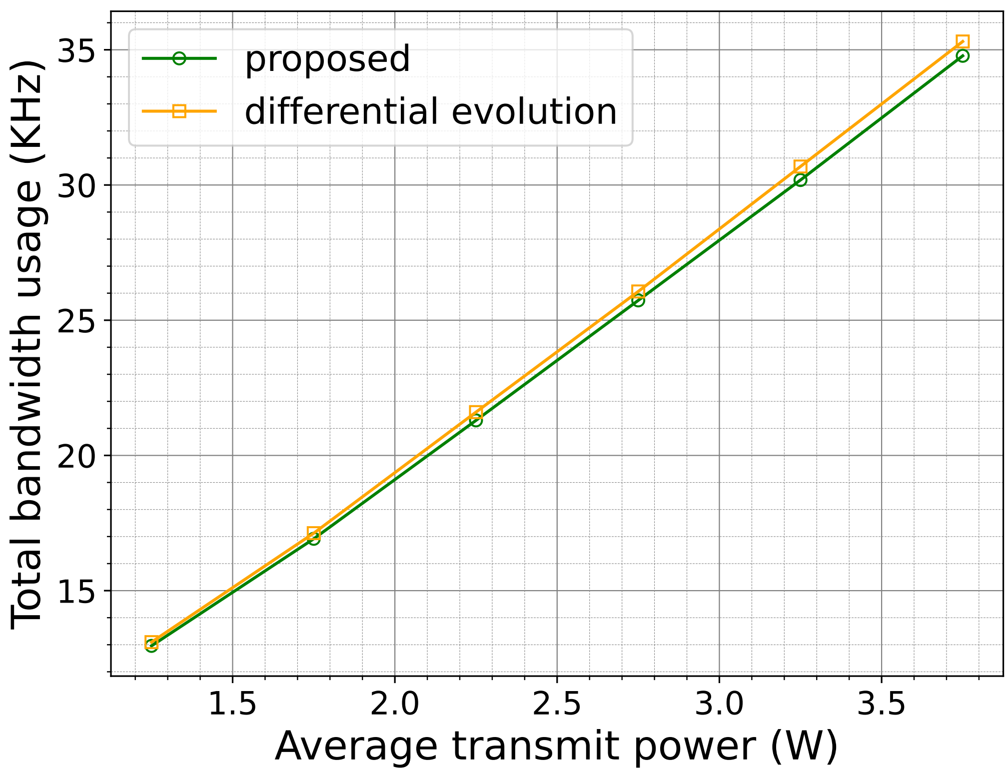


Figure 3.6: Impact of transmit power on total bandwidth usage.

transmit rate (3.8) is much less influential in the total calculated energy consumption than the one in (3.7). It can also be noted that despite running this simulation for a limited number of users, there still exists a gap in performance that shows our proposed algorithm is outperforming DE.

3.5 Solving the Problem in Dual Accuracy Form

In the previous sections, we discussed the problem with the assumption of a fixed global accuracy. While this assumption is true in many real-world applications in practice, there may be cases where the global accuracy of an FL system can be chosen with flexibility to optimize the usage of other resources such as time, energy, or bandwidth. In this case, we can consider both local accuracy and global accuracy to be variables of our problem of minimizing bandwidth usage. Hence, in this section, we will reformulate the optimization problem to a new form that includes both of them. We will create it by turning the previous problem into a multi-objective optimization problem, where the cost function consists of a trade-off between bandwidth usage and global accuracy. Then, we will offer the optimal solution for the dual-accuracy form of the problem. Finally, we will run experiments and demonstrate the results of the proposed algorithms.

3.5.1 Problem Formulation

In this part, we will rewrite the optimization problem in the aforementioned new form, i.e. the dual accuracy form. Before doing that, we should update our model of the total energy consumption from (3.10). Since in this section, the global accuracy is not fixed, the bound on the total number of global iterations can be normalized to $\frac{\log(1/\varepsilon)}{1-\theta}$. Thus, the equation for the total energy consumption of a user will be

$$E_n^{total,dual} = \frac{\log(1/\varepsilon)}{1-\theta} E_n^{(m)} = \frac{\log(1/\varepsilon)}{1-\theta} (\log(1/\theta) E_n^{cmp} + E_n^{com}). \tag{3.27}$$

Now, using (3.27) and adding global accuracy, we formulate our optimization problem as follows.

$$\min_{\varepsilon,\theta,b,f} \sum_{n=1}^{N} b_n + \kappa\varepsilon \tag{3.28a}$$

$$\text{s.t. } 0 \leq f_n \leq f_n^{\max}, \forall n, \tag{3.28b}$$

$$\log(1/\theta)\frac{C_n D_n}{f_n} + \frac{S_n}{r_n} \leq \gamma_T, \forall n \tag{3.28c}$$

$$0 \leq b_n \leq B_{max}, \forall n, \tag{3.28d}$$

$$\frac{\log(1/\varepsilon)}{1-\theta}(\log(1/\theta)k_n C_n D_n f_n^2 + p_n \frac{S_n}{r_n}) \leq \gamma_E, \forall n \tag{3.28e}$$

$$0 \leq \theta \leq 1 \tag{3.28f}$$

$$0 \leq \varepsilon \leq \varepsilon_{max}. \tag{3.28g}$$

In (3.28), κ is the trade-off parameter that captures the penalty for compromising global accuracy. One can adjust their tolerance or willingness to compromise the accuracy by adjusting this coefficient accordingly. Constraint (3.28g) sets the bounds on the global accuracy. While the domain is between 0 to 1, one can set a lower ε_{max}, to meet their minimum accuracy requirements. This problem, compared to (3.11), has the extra decision variable of global accuracy ε. and the extra constraint on its bounds, in addition to the change made to the energy consumption constraint and the objective function. Other elements of the problem are similar to what was explained for (3.11). Next, we will analyze the problem and discuss the optimal solution.

3.5.2 Solution

Similar to before, the optimization problem in (3.28) is non-convex in its original formulated form. Hence, we will once again perform a decomposition of the problem into two subproblems, namely SUB1 and SUB2. First, assuming θ and ε are given constant values θ_c and ε_c, we formulate SUB1, and by solving it we obtain f and b in terms of those values. Then, we find the optimal values for θ_c and ε_c which minimize the total cost by solving SUB2. SUB1 and SUB2 are shown in (3.29) and (3.30), respectively. Please refer to page 32 for a similar justification of why this approach yields the optimal global solution.

$$\text{SUB1: } \min_{b,f} \sum_{n=1}^{N} b_n + \kappa \varepsilon_c \tag{3.29a}$$

$$\text{s.t. } 0 \le f_n \le f_n^{\max}, \forall n, \tag{3.29b}$$

$$\log(1/\theta_c)\frac{C_n D_n}{f_n} + \frac{S_n}{r_n} \le \gamma_T, \forall n \tag{3.29c}$$

$$0 \le b_n \le B_{max}, \forall n, \tag{3.29d}$$

$$\frac{\log(1/\varepsilon_c)}{1-\theta_c}(\log(1/\theta_c) k_n C_n D_n f_n^2 + p_n \frac{S_n}{r_n}) \le \gamma_E, \forall n, \tag{3.29e}$$

$$\text{SUB2: } \min_{\varepsilon_c, \theta_c} \sum_{n=1}^{N} b_{n,\theta_c,\varepsilon_c} \tag{3.30a}$$

$$\text{s.t. } 0 \le f_{n,\theta_c,\varepsilon_c} \le f_n^{\max}, \forall n, \tag{3.30b}$$

$$\log(1/\theta_c)\frac{C_n D_n}{f_{n,\theta_c,\varepsilon_c}} + \frac{S_n}{r_{n,\theta_c,\varepsilon_c}} \le \gamma_T, \forall n \tag{3.30c}$$

$$0 \le b_{n,\theta_c,\varepsilon_c} \le B_{max}, \forall n, \tag{3.30d}$$

$$\frac{\log(1/\varepsilon_c)}{1-\theta_c}(\log(1/\theta_c) k_n C_n D_n f_{n,\theta_c,\varepsilon_c}^2 + p_n \frac{S_n}{r_{n,\theta_c,\varepsilon_c}}) \le \gamma_E, \forall n \tag{3.30e}$$

$$0 \le \theta_c \le 1 \tag{3.30f}$$

$$0 \le \varepsilon_c \le \varepsilon_{max} \tag{3.30g}$$

We can solve subproblem (3.29) by transforming it into the subproblem in (3.12). Given that κ and ε_c are constants in this subproblem, they do not affect the objective function's solution $\{f^*, b^*\}$. Hence, this is the same as minimizing $\sum_{n=1}^{N} b_n$, which is the objective function in (3.12). The other difference of these subproblems is between (3.12e) and (3.29e). To transform (3.29d), we move $\log(1/\varepsilon_c)$ to the right-hand side of the inequality. Thus, the subproblem in (3.29) is reduced to the subproblem in (3.12), with $\frac{\gamma_E}{\log(1/\varepsilon_c)}$ replacing γ_E in (3.12e). Therefore, we can reuse the solutions obtained in Theorem 1.

After solving SUB1, we can use (3.17) and (3.18) to find the optimal solution θ^* and ε^* for SUB2. Note that as pointed out above, we have to replace γ_E in those equations with $\frac{\gamma_E}{\log(1/\varepsilon_c)}$. Having done that that, we have derived $f_{n,\theta_c,\varepsilon_c}$ and $b_{n,\theta_c,\varepsilon_c}$, which are functions of θ_c and $,\varepsilon_c$. To solve the problem (3.30), we need to find the optimal value of local accuracy θ_c and global accuracy $,\varepsilon_c$ that minimizes the cost function while satisfying the constraints. Therefore, we propose an algorithm that iterates over θ_c and $,\varepsilon_c$ within their feasible ranges specified in (3.30g) and (3.30f), and with the provided precisions, to find their optimal values. In each iteration, the algorithm checks whether for a given θ_c and $,\varepsilon_c$, constraints (3.30b) to (3.30g) are satisfied. Finally, among all feasible values for θ_c and $,\varepsilon_c$, the algorithm selects the one that minimizes the total cost function as output. The details of the proposed solution are presented in Algorithm 2. In Algorithm 2, the loop iterates $\frac{1}{\epsilon_1 * \epsilon_2}$ times over different values of $,\varepsilon$ and θ. In each iteration, b_n and f_n need to be calculated for the N users. Hence, the computational complexity is $\mathcal{O}\left(\frac{1}{\epsilon_1 * \epsilon_2} N\right)$.

Algorithm 2 Iterative Dual Accuracy Optimization Algorithm

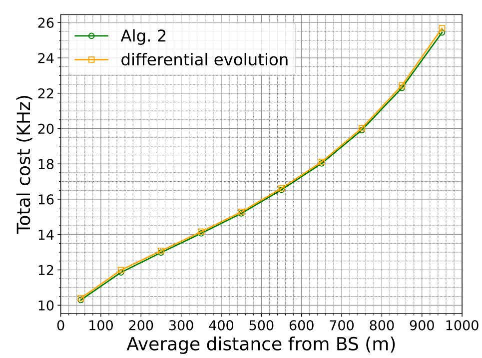
Input: The constraint tolerance δ, the global accuracy precision ϵ_1, the local accuracy precision ϵ_2, trade-off parameter κ

Output: ε^*, θ^*, f^*, b^*

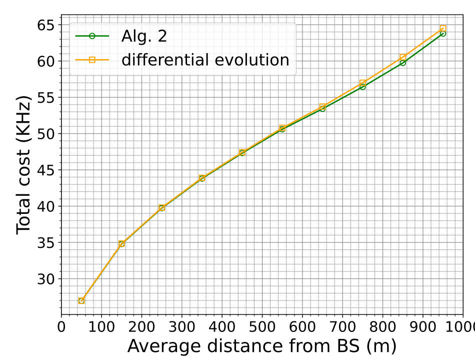
```
Initialize ε_temp ← 0, θ_temp ← 0, b_temp ← [],f_temp ← [],
cost_min ← +∞
for i ← 0 to ε_max/ε_1 do
    ε_temp ← 0 + i * ε_1
    for j ← 0 to 1/ε_2 do
        θ_temp ← 0 + i * ε_2
        f_temp ← list of calculated f*_n s using θ_temp and ε_temp
        b_temp ← list of calculated b*_n s using θ_temp, ε_temp, and f_temp
        if f_temp,b_temp, and θ_temp pass the constraints (3.30b) - (3.30g) considering
        given tolerance δ then
            cost_temp ← Σ_n b_temp + κ ε_temp,
            if cost_temp < cost_min then
                ε* ← ε_temp, θ* ← θ_temp, b* ← b_temp, f* ← f_temp
            end if
        end if
    end for
end for
```

3.5.3 Numerical Results

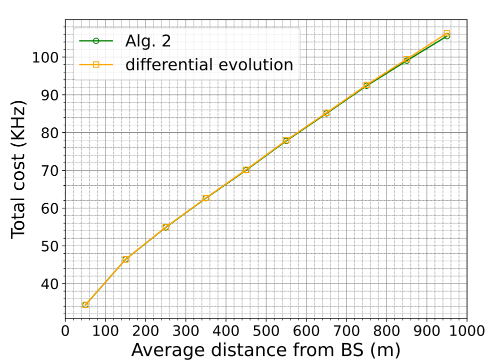
In this subsection, we present diverse numerical results to demonstrate the performance of the proposed algorithm. Compared to the previous problem, this problem has an extra parameter κ, which is the trade-off parameter of the global accuracy in the cost function in (3.28). other setup and parameters for the experiments in this section are exactly as described in subsection 3.4.1 and Table 3.1. We will explore the numerical results for four different values of κ, i.e. $\{10000, 60000, 120000, 200000\}$. That translates to each 0.01 global accuracy being worth $\{100, 600, 1200, 2000\}$Hz of bandwidth to us, starting from 0. In this sense, we can consider the unit of κ to be $\frac{\text{Hz}}{1}$, since ε is unitless.



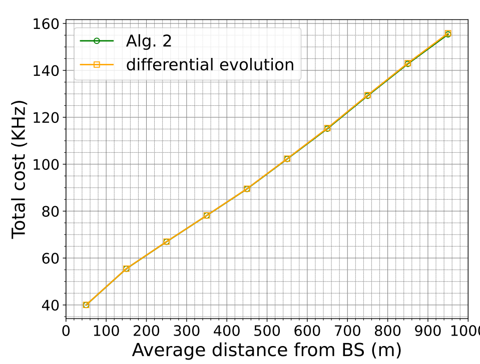
(a) $\kappa = 10000$



(b) $\kappa = 60000$



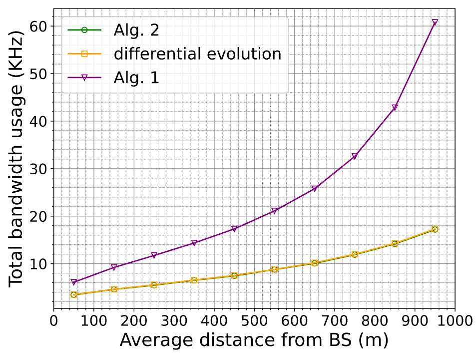
(c) $\kappa = 120000$



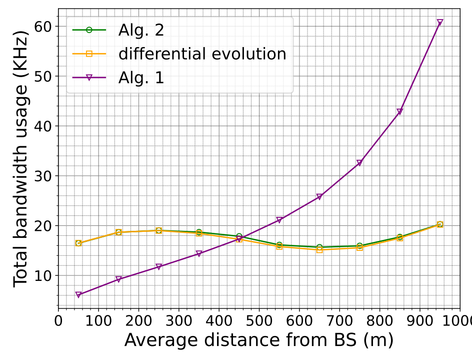
(d) $\kappa = 200000$

Figure 3.7: Diagrams of the total cost with respect to the average distance from BS

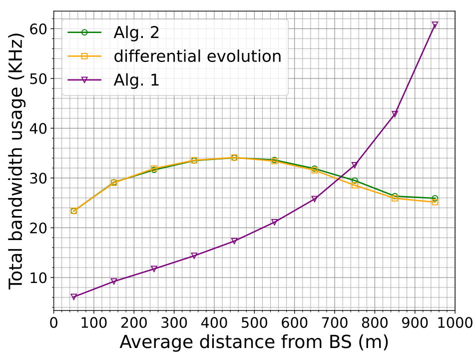
.



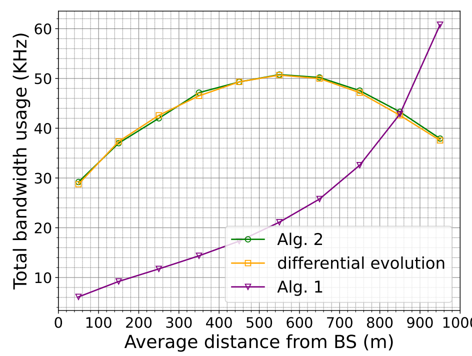
(a) $\kappa = 10000$



(b) $\kappa = 60000$



(c) $\kappa = 120000$



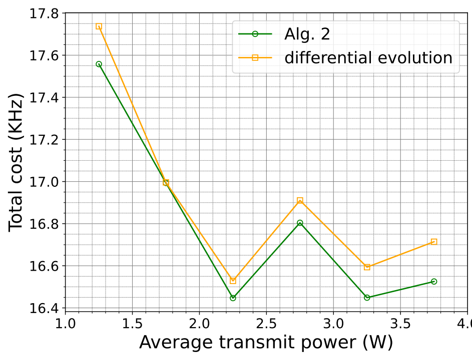
(d) $\kappa = 200000$

Figure 3.8: Diagrams of total bandwidth usage with respect to average distance from BS

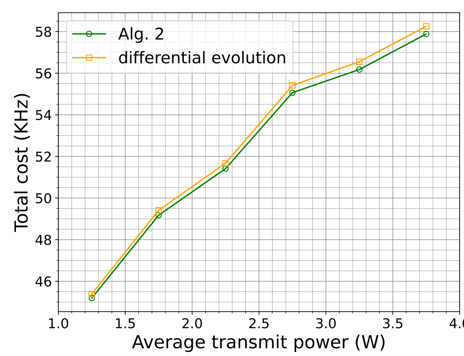
.

Impact of Distance

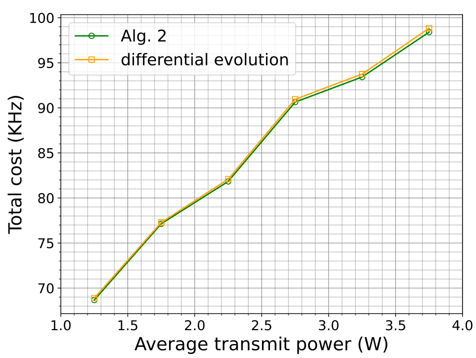
In Figures 3.8 and 3.7, we investigate the impact of users' distance from the BS on the results. As shown in 3.7, placing users further from the BS, always results in a higher cost in general. However, Fig. 3.8 shows that this does not necessarily mean a constant bandwidth usage. In cases where κ is smaller, it is easier to compromise accuracy to reduce bandwidth usage. Also, in cases with higher κ, when bandwidth usage increases past a certain point due to distance from BS, compromising the accuracy yields better solutions from using more bandwidth, in terms of the total cost. Thus, the bandwidth usage starts to drop again. Fig. 3.8 also compares the results with the ones from Alg. 1. In Alg. 1, we considered a fixed global accuracy and normalized $\mathcal{O}(\log(1/\varepsilon))$ to 1. It could be translated to the same problem in (3.28), where $\log(1/\varepsilon) = 1$ and $\kappa = 0$. Our algorithm is consistently showing better performance, although the gap for a small number of users is not large. That will be further investigated in the subsection on the impact of the number of users.



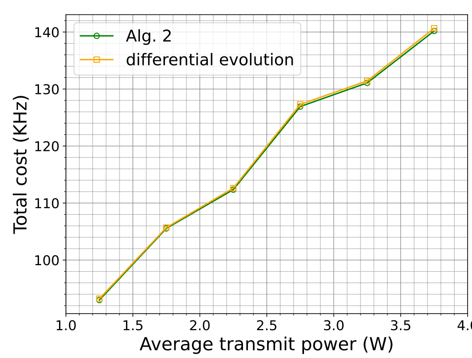
(a) $\kappa = 10000$



(b) $\kappa = 60000$



(c) $\kappa = 120000$



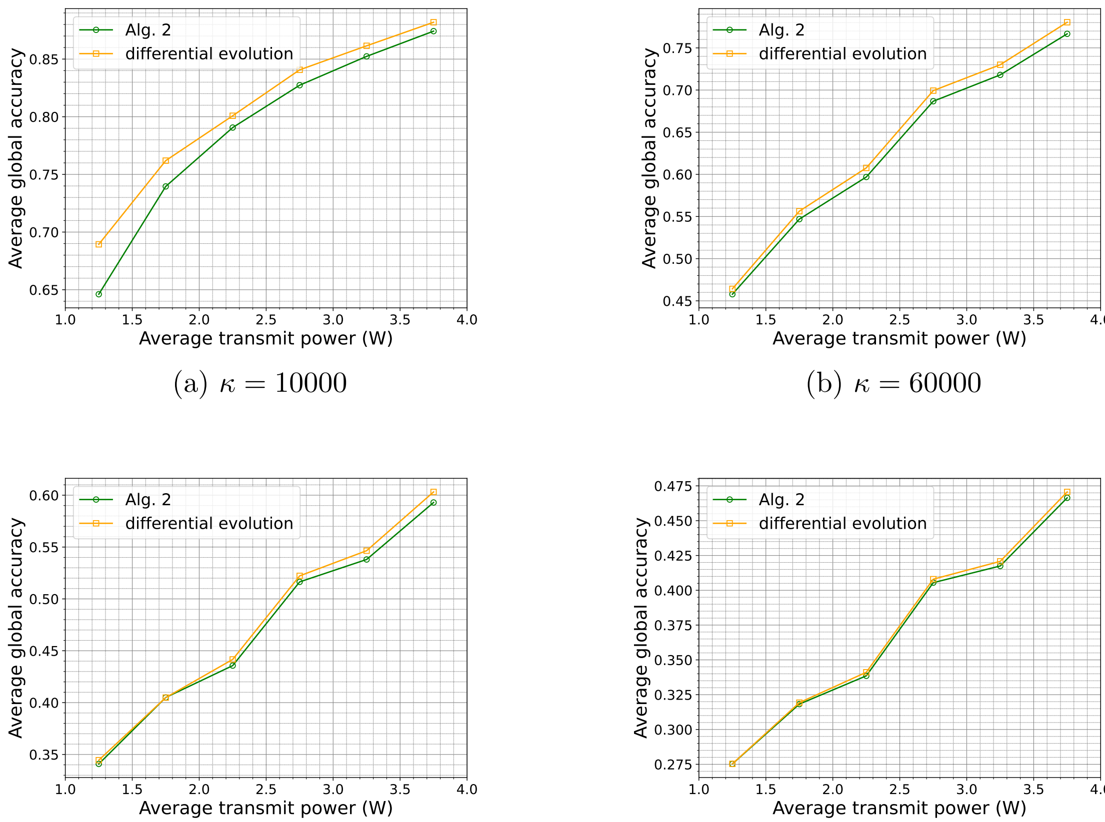
(d) $\kappa = 200000$

Figure 3.9: Diagrams of the total cost with respect to the average transmit power

.

Impact of Transmit Power

In Figure 3.11, the impact of users' transmit powers on the bandwidth usage is explored. As shown in 3.11, when κ is small, it is possible for us to compromise accuracy and reduce bandwidth usage. However, increasing κ increases our bandwidth usage as the accuracy becomes more and more valuable in the cost function. Thus, the bandwidth usage surges. We can In Fig. 3.9, the resulting cost function values with



(a) $\kappa = 10000$ (b) $\kappa = 60000$

(c) $\kappa = 120000$ (d) $\kappa = 200000$

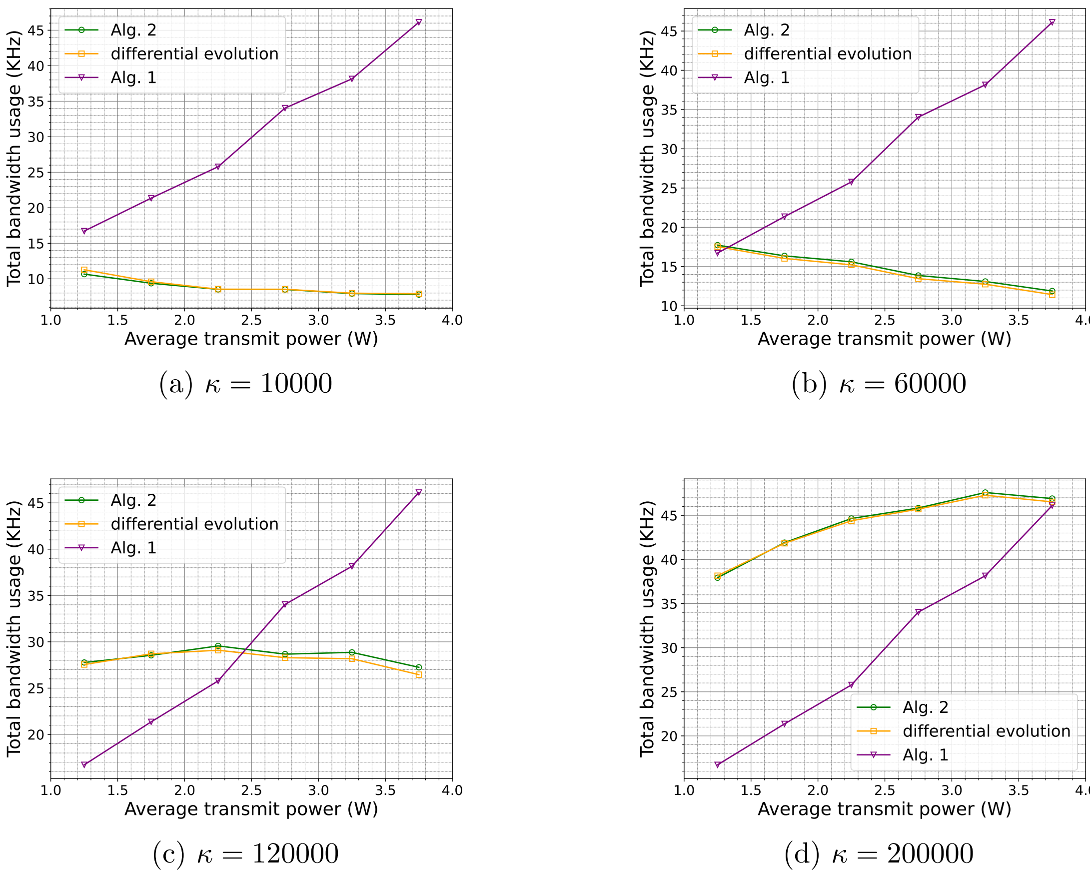
Figure 3.10: Diagrams of the global accuracy with respect to the average transmit power

.

respect to the average transmit power are shown. Again, our algorithm is consistently showing better performance. Figures 3.10a to 3.10d demonstrate the average optimal global accuracy with respect to the transmit power. It can be seen that our algorithm is maintaining better accuracy, yielding a better cost result in Fig. 3.9.

Impact of Number of Users

Last but not least, we observe the impact of the number of users on each algorithm's performance and execution time. In this part, we have applied our proposed algorithms and the differential evolution to setups with $\{5, 10, 30, 50, 75, 100\}$ users. We compare Algorithm 2 which was introduced in this section with Algorithm 1 and differential evolution.



(a) $\kappa = 10000$

(b) $\kappa = 60000$

(c) $\kappa = 120000$

(d) $\kappa = 200000$

Figure 3.11: Diagrams of the total bandwidth usage with respect to the average transmit power

.

The impact of the number of users on the total bandwidth usage and the total cost is investigated in Figures 3.13 and 3.12, respectively. It can be noticed that for our proposed algorithms, the average optimal bandwidth usage increases linearly with the number of participating users. As expected, Algorithm 2 can offer lower bandwidth compared to Algorithm 1, due to the flexibility of compromising accuracy. The figure also shows that DE does not always find the optimal answer. As the search space of DE grows exponentially with the number of users, a significantly degraded performance can be observed in terms of both cost and bandwidth usage. Although for $\mathcal{N} = 5$ bandwidth usage of Algorithm 2 is slightly more, it still has a lower cost. Fig. 3.12 shows that Algorithm 2 yields a promising result that is consistently better

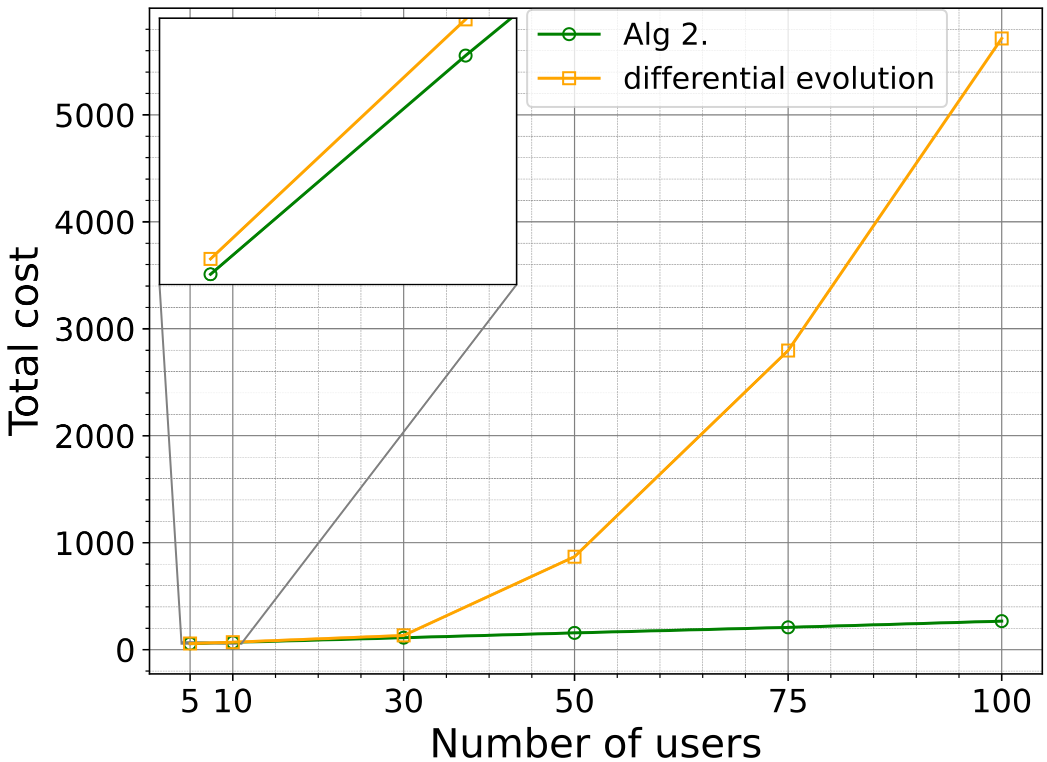


Figure 3.12: Impact of number of users on total cost in dual accuracy form.

than differential evolution.

Fig. 3.5 demonstrates the impact of the number of users on the execution time of Algorithm 1, differential evolution, and Algorithm 2. As shown, experimental results confirm that DE suffers from exponential growth in time complexity when increasing the dimensions of the problem, i.e., the number of participating FL users. As previously discussed, Algorithm 1, on the other hand, enjoys a linear increase in its execution time with respect to the number of users. As expected, Algorithm 2 also exhibits a linear growth pattern similar to Algorithm 1; however, it incurs a higher execution time than Algorithm 1. That is due to the increase of complexity from $\mathcal{O}\left(\frac{1}{\epsilon_2}N\right)$ to $\mathcal{O}\left(\frac{1}{\epsilon_1 * \epsilon_2}N\right)$. Despite this, Algorithm 2 significantly outperforms DE, particularly when the number of users increases, which is typical in most practical scenarios [55]. Hence, both Algorithm 1 and Algorithm 2 not only outperform DE

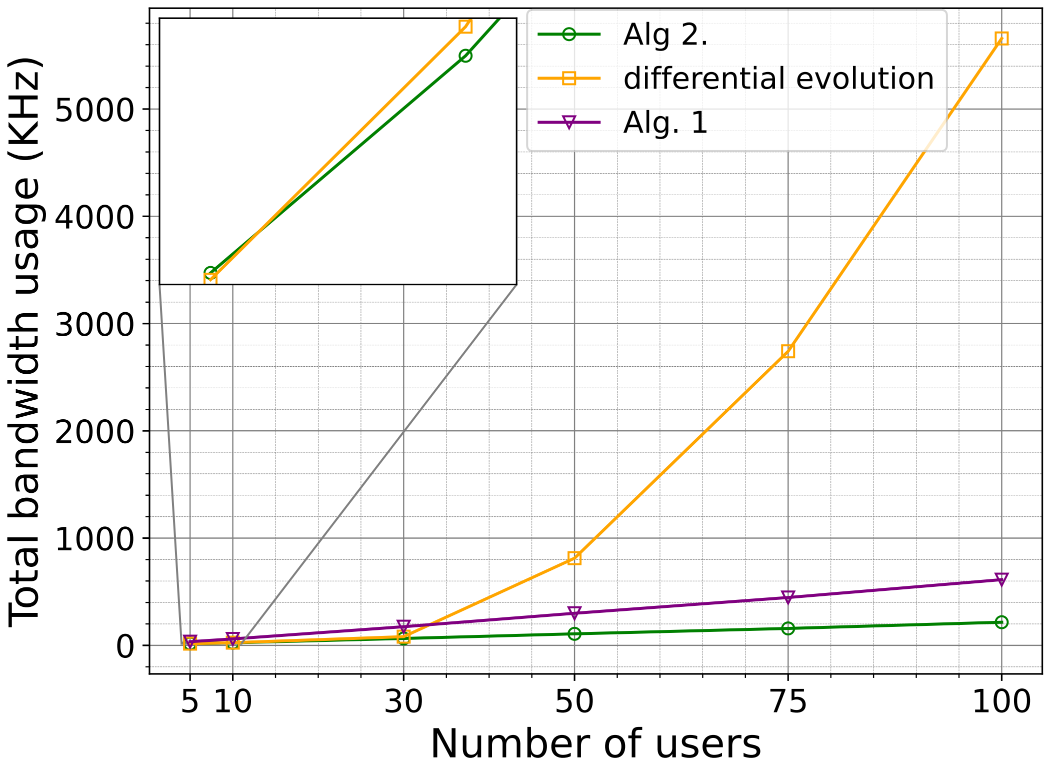


Figure 3.13: Impact of number of users on total bandwidth usage in dual accuracy form.

in terms of complexity but also provide consistently optimal performance. While Algorithm 1 remains the most time-efficient, Algorithm 2 offers a robust alternative with superior flexibility.

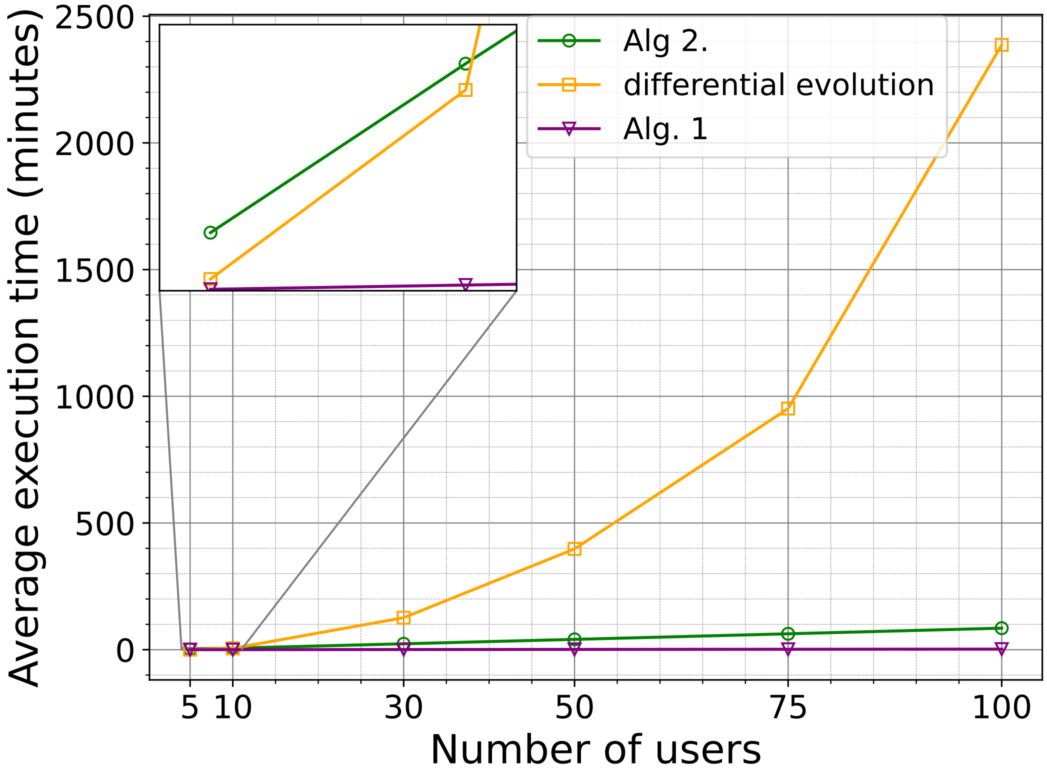


Figure 3.14: Impact of number of users on execution time in dual accuracy form.

Chapter 4

Conclusion and Future Works

4.1 Conclusion

In this paper, we thoroughly explored the problem of minimizing the total bandwidth usage in a wireless federated learning system, taking into account the critical constraints of total energy consumption and the completion time required by users. The complexity and significance of this problem are underscored by the need to balance efficient bandwidth usage with the practical limitations of computational resources and energy availability in wireless environments. The challenge is further heightened in scenarios where timely communication is essential, making it imperative to devise solutions that not only minimize resource usage but also ensure that user devices can complete their tasks within acceptable time frames.

To address these challenges, we formulated the problem as a non-convex optimization problem and explored it in two distinct forms. In the first form, the global accuracy of the FL model was treated as a fixed parameter. This approach is relevant in scenarios where a certain level of model accuracy is mandated, such as in applications where the performance of the model must meet predefined standards for deployment. In this case, the optimization focused solely on minimizing the total bandwidth usage while adhering to the constraints on energy consumption and completion time, and only using bandwidth, local accuracy, and CPU-cycle frequency as variables. By treating the global accuracy as a fixed variable, we ensured that the resulting solution would be applicable in situations where model performance cannot be compromised.

In the second form, we expanded the problem by introducing global model accuracy as an additional decision variable. We formulated the problem as a multi-objective, non-convex, problem for joint optimization of global accuracy and bandwidth usage. This approach recognizes that in many real-world applications, there may be some flexibility in the trade-off between model accuracy and resource usage. By allowing accuracy to be adjusted within acceptable bounds, we were able to explore more efficient solutions that dynamically balance the competing demands of accuracy and bandwidth. This formulation is particularly valuable in scenarios where achieving the highest possible accuracy is less critical than optimizing resource usage, or where the available resources fluctuate over time.

For both forms of the problem, we proposed an optimal solution that leverages convex optimization techniques to navigate the non-convex landscape of the problem. By decomposing the original problem and applying convex optimization methods, we derived solutions that efficiently allocate bandwidth while respecting the constraints of the problem. Our approach ensures that the FL process can be sustained in resource-constrained environments without sacrificing either the quality of the global model or the feasibility of real-time deployment.

Extensive simulations and comparisons with existing algorithms demonstrated the superiority of our proposed solutions. Our algorithm consistently outperformed the benchmarks in terms of bandwidth efficiency, achieving significant reductions in total bandwidth usage. This advantage is particularly important in wireless environments, where bandwidth is a precious and limited resource. Moreover, our solution maintained a much lower computational complexity, making it more practical for deployment on devices with limited processing power and energy reserves.

The ability to balance accuracy with resource constraints while minimizing bandwidth usage is a key contribution of this work. Our approach offers a flexible framework that can be tailored to the specific needs of the deployment environment, providing a robust and scalable solution for a wide range of FL applications in wireless networks.

In conclusion, this paper not only advances the state of the art in optimizing bandwidth usage in wireless FL systems but also provides a versatile framework that can adapt to various deployment scenarios.

4.2 Future Works

While this research provides significant advancements in the optimization of bandwidth usage for wireless federated learning systems, there remain multiple areas for future exploration that could further enhance the applicability and efficiency of FL in diverse environments [56].

One potential area for future work is the exploration of different, more complex wireless networks. In real-world deployments, wireless communication can become more complicated and involve various other factors and elements such as interference or RIS. Extension of this work to more advanced wireless communication setups with real-world applications can be a promising direction for future works. This can also include new communication technologies, such as 5G, 6G, and low Earth orbit (LEO) satellite networks. Future work could explore how these technologies can be integrated into the proposed optimization framework to further enhance the efficiency and reliability of FL in next-generation networks.

The current work focuses primarily on minimizing bandwidth usage while considering energy consumption and completion time constraints. It also includes a multi-objective optimization that captures the trade-off between bandwidth usage and model accuracy. However, many other scenarios may be beneficial that adopt a multi-objective optimization approach and simultaneously optimize multiple criteria, such as energy efficiency, latency, or computational load along with bandwidth usage or model accuracy. Future research could explore advanced multi-objective optimization techniques to balance these competing demands more effectively, providing a more comprehensive solution that caters to a broader range of application requirements.

As FL continues to gain traction in applications involving vast numbers of devices, such as smart cities or large-scale IoT networks, scalability becomes a critical concern. Future work could investigate the scalability of the proposed optimization framework to support thousands or even millions of devices. This may involve hierarchical or decentralized approaches to FL, where decisions are made at multiple levels of the network to reduce the computational burden and communication overhead on any single entity. Various network topologies and their communication can be explored to form new, efficient FL frameworks.

Bibliography

[1] 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.

[2] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, *Federated optimization: Distributed machine learning for on-device intelligence*, 2016. arXiv: 1610.02527 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1610.02527.

[3] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 909–910. DOI: 10.1109/ALLERTON.2015.7447103.

[4] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, Sep. 2020, ISSN: 2398-6352. DOI: 10.1038/s41746-020-00323-1. [Online]. Available: http://dx.doi.org/10.1038/s41746-020-00323-1.

[5] D. C. Nguyen *et al.*, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (Csur)*, vol. 55, no. 3, pp. 1–37, 2022.

[6] D. Byrd and A. Polychroniadou, "Differentially private secure multi-party computation for federated learning in financial applications," in *Proceedings of the First ACM International Conference on AI in Finance*, ser. ICAIF '20, New York, New York: Association for Computing Machinery, 2021, ISBN: 9781450375849. DOI: 10.1145/3383455.3422562. [Online]. Available: https://doi.org/10.1145/3383455.3422562.

[7] S. Pandya *et al.*, "Federated learning for smart cities: A comprehensive survey," *Sustainable Energy Technologies and Assessments*, vol. 55, p. 102 987, 2023, ISSN: 2213-1388. DOI: https://doi.org/10.1016/j.seta.2022.102987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138822010359.

[8] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021. DOI: 10.1109/COMST.2021.3075439.

[9] J. Ren, W. Ni, G. Nie, and H. Tian, *Research on resource allocation for efficient federated learning*, 2022. arXiv: 2104.09177 [eess.SP]. [Online]. Available: https://arxiv.org/abs/2104.09177.

[10] 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*, 2022, pp. 5075–5080. DOI: 10.1109/GLOBECOM48099.2022.10001492.

[11] Y. Ji, X. Zhong, Z. Kou, S. Zhang, H. Li, and Y. Yang, "Efficiency-boosting federated learning in wireless networks: A long-term perspective," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 9434–9447, 2023. DOI: 10.1109/TVT.2023.3250273.

[12] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2687–2700, 2022. DOI: 10.1109/TPDS.2021.3131654.

[13] 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, 2021. DOI: 10.1109/TWC.2020.3037554.

[14] J. Kuang, M. Yang, H. Zhu, and H. Qian, "Client selection with bandwidth allocation in federated learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06. DOI: 10.1109/GLOBECOM46510.2021.9685090.

[15] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 073–11 087, 2022. DOI: 10.1109/TITS.2021.3099597.

[16] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1387–1395. DOI: 10.1109/INFOCOM.2019.8737464.

[17] X. Ji, J. Tian, H. Zhang, D. Wu, and T. Li, "Joint device selection and bandwidth allocation for cost-efficient federated learning in industrial internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 9148–9160, 2023. DOI: 10.1109/JIOT.2022.3233595.

[18] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over uav-enabled wireless powered communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4977–4990, 2022. DOI: 10.1109/TVT.2022.3150004.

[19] M. A. Hossain and N. Ansari, "Energy-efficient federated edge learning in multi-tier noma-enabled hetnet," *IEEE Transactions on Cloud Computing*, vol. 11, no. 4, pp. 3355–3366, 2023. DOI: 10.1109/TCC.2023.3285534.

[20] N. Yang, S. Wang, M. Chen, C. G. Brinton, and C. Yin, "Energy efficient collaborative federated learning design: A graph neural network based approach," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 164–169. DOI: 10.1109/GLOBECOM54140.2023.10437172.

[21] 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. DOI: 10.1109/TWC.2020.3002988.

[22] S. Zhang, Y. Xu, J. Liu, H. Takakura, L. Chen, and N. Shiratori, "Bandwidth allocation for low-latency wireless federated learning: An evolutionary game approach," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 1628–1633. DOI: 10.1109/ICC45041.2023.10279623.

[23] C. T. Dinh *et al.*, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, 398–409, Feb. 2021, ISSN: 1558-2566. DOI: 10.1109/tnet.2020.3035770. [Online]. Available: http://dx.doi.org/10.1109/TNET.2020.3035770.

[24] 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.

[25] A. M. Elbir and S. Coleri, "Federated learning for hybrid beamforming in mm-wave massive mimo," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2795–2799, 2020. DOI: 10.1109/LCOMM.2020.3019312.

[26] A. M. Elbir and S. Coleri, "Federated learning for channel estimation in conventional and ris-assisted massive mimo," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4255–4268, 2022. DOI: 10.1109/TWC.2021.3128392.

[27] L. Li *et al.*, "Enhanced reconfigurable intelligent surface assisted mmwave communication: A federated learning approach," *China Communications*, vol. 17, no. 10, pp. 115–128, 2020. DOI: 10.23919/JCC.2020.10.008.

[28] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2021. DOI: 10.1109/TWC.2020.3042530.

[29] R. Hamdi, A. B. Said, A. Erbad, A. Mohamed, M. Hamdi, and M. Guizani, "Hierarchical federated learning over hetnets enabled by wireless energy transfer," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06. DOI: 10.1109/GLOBECOM46510.2021.9685344.

[30] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2021. DOI: 10.1109/RBME.2020.3013489.

[31] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*, Q. Yang, L. Fan, and H. Yu, Eds. Cham: Springer International Publishing, 2020, pp. 240–254, ISBN: 978-3-030-63076-8. DOI: 10.1007/978-3-030-63076-8_17. [Online]. Available: https://doi.org/10.1007/978-3-030-63076-8_17.

[32] Y. Li, X. Tao, X. Zhang, J. Liu, and J. Xu, "Privacy-preserved federated learning for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8423–8434, 2021.

[33] T Manoj, K. Makkithaya, and V. Narendra, "A federated learning-based crop yield prediction for agricultural production risk management," in *2022 IEEE Delhi Section Conference (DELCON)*, IEEE, 2022, pp. 1–7.

[34] B. Nour, S. Cherkaoui, and Z. Mlika, "Federated learning and proactive computation reuse at the edge of smart homes," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3045–3056, 2022. DOI: 10.1109/TNSE.2021.3131246.

[35] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.

[36] T. Yang *et al.*, *Applied federated learning: Improving google keyboard query suggestions*, 2018. [Online]. Available: https://arxiv.org/abs/1812.02903.

[37] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[38] J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, p. 6230, 2020.

[39] M. Jaggi *et al.*, "Communication-efficient distributed dual coordinate ascent," *Advances in neural information processing systems*, vol. 27, 2014.

[40] C. Ma *et al.*, *Distributed optimization with arbitrary local solvers*, 2016. arXiv: 1512.04039 [cs.LG].

[41] J. Konečný, Z. Qu, and P. Richtárik, *Semi-stochastic coordinate descent*, 2014. arXiv: 1412.6293 [cs.NA]. [Online]. Available: https://arxiv.org/abs/1412.6293.

[42] J. Konečný and P. Richtárik, *Semi-stochastic gradient descent methods*, 2015. arXiv: 1312.1666 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1312.1666.

[43] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., ser. Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 2021–2031. [Online]. Available: https://proceedings.mlr.press/v108/reisizadeh20a.html.

[44] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, "Adaptive quantization of model updates for communication-efficient federated learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3110–3114. DOI: 10.1109/ICASSP39728.2021.9413697.

[45] W. Feng and X. Zhang, "Wireless federated learning with dynamic quantization and bandwidth adaptation," *IEEE Wireless Communications Letters*, vol. 11, no. 11, pp. 2335–2339, 2022. DOI: 10.1109/LWC.2022.3202645.

[46] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2020. DOI: 10.1109/TNNLS.2019.2944481.

[47] Z. Jiang, Y. Xu, H. Xu, Z. Wang, and C. Qian, "Heterogeneity-aware federated learning with adaptive client selection and gradient compression," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10229029.

[48] Y. Lin, S. Han, H. Mao, Y. Wang, and W. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2018. [Online]. Available: https://openreview.net/pdf?id=SkhQHMW0W.

[49] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering). Springer New York, 2006, ISBN: 9780387227429. [Online]. Available: https://books.google.ca/books?id=7wDpBwAAQBAJ.

[50] S. Boyd and L. Vandenberghe, *Convex Optimization* (Berichte über verteilte messysteme pt. 1). Cambridge University Press, 2004, ISBN: 9780521833783.

[51] T. Rappaport, *Wireless Communications: Principles and Practice* (Electrical engineering). Prentice Hall PTR, 1996, ISBN: 9780133755367.

[52] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 23, Jan. 1995.

[53] J. H. Lambert, "Observationes variae in mathesin puram," *Acta Helvetica*, vol. 3, no. 1, pp. 128–168, 1758.

[54] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011. DOI: 10.1109/TEVC.2010.2059031.

[55] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, *Heterogeneous federated learning: State-of-the-art and research challenges*, 2023. arXiv: 2307.10616 `[cs.LG]`.

[56] P. Kairouz *et al.*, *Advances and open problems in federated learning*, 2021. arXiv: 1912.04977 `[cs.LG]`. [Online]. Available: https://arxiv.org/abs/1912.04977.