Federated Learning via Inexact ADMM
Shenglong Zhou and Geoffrey Ye Li

Abstract—One of the crucial issues in federated learning is how to develop efficient optimization algorithms. Most of the current ones require full device participation and/or impose strong assumptions for convergence. Different from the widely-used gradient descent-based algorithms, in this paper, we develop an inexact alternating direction method of multipliers (ADMM), which is both computation- and communication-efficient, capable of combating the stragglers’ effect, and convergent under mild conditions. Furthermore, it has a high numerical performance compared with several state-of-the-art algorithms for federated learning.

Index Terms—Partial device participation, inexact ADMM, communication and computation-efficiency, global convergence

1 INTRODUCTION

Federated learning (FL), originated from [1], [2], gains its popularity recently due to its ability to address various applications, such as vehicular communications [3], [4], [5], [6], digital health [7], and mobile edge and over-the-air computing [8], [9], [10], [11]. It is still facing many challenges. One of them is how to develop efficient optimization algorithms for different purposes, such as saving communication resources, accelerating the learning process, coping with the stragglers’ effect, just to name a few. We refer to some nice surveys [12], [13], [14] for more challenges.

1.1 Prior arts

Before implementing FL into practical applications, many critical issues need to be addressed. We present a few of them that motivate our research in this paper.

1.1.1 Communication efficiency

When exchanging parameters between clients and a central server, communication efficiency must be taken into account as frequent communications would consume expensive resources (e.g., transmission power, energy, and bandwidth). Popular techniques to improve communication efficiency include data compression and reduction of communication rounds (CR). The former aims to quantize and sparsify the local parameters before the transmission so as to lessen the amount of the transmitted contents [15], [16], [17], [18]. For the latter, communications between the clients and the server occur in a periodic fashion so as to reduce CR [19], [20], [21], [22], [23]. In this paper, we will exploit this tactic.

1.1.2 Computational efficiency

Since an increasing number of clients are engaging in the training, equipping all of them with strong computational capacities apparently is unrealistic. Hence, a desirable FL algorithm is able to reduce the computational complexity to alleviate clients’ computational burdens. To achieve this goal, there are two promising solutions. The first one is the stochastic approximation of some critical items (e.g., the full gradients). This idea has been extensively employed in the stochastic gradient descent (SGD) algorithms, such as the federated averaging (FedAvg [19]), local SGD [22], [23], and those in [24], [25], [26]. The second solution reduces the computational complexity by solving sub-problems inexactly, which has been widely adopted in the inexact ADMM. They allow clients to update their parameters via solving subproblems approximately, thereby accelerating the computational speed exceptionally [27], [28], [29], [30]. We will take advantage of this technique in our algorithmic design.

1.1.3 Partial devices participation

Since the central server is unable to control the local devices and their communication environments, there may have delays/withdraw of sharing parameters by some clients due to inadequate transmission resources or limited computational capacity. This phenomenon is called the stragglers’ effect, namely, everyone waits for the slowest. A remedy for alleviating the effect is to let the central server pick up a portion of clients in good conditions to take part in the training, which is known as the partial device participation [19], [20], [31]. Based on this scheme, FL algorithms can be categorized into two groups as follows.

a) Full device participation. There is an impressive body of work on developing algorithms based on full device participation, such as the non-stochastic gradient descent methods [32], [33], [34], [35], SGD [21], [22], [23], [26], exact ADMM [36], [37], [38], [39], [40], and inexact ADMM [27], [28], [29], [30]. However, due to the full device engagement, those algorithms are at risk of the stragglers’ effect, particularly in scenarios where large numbers of devices are distributed at the edge nodes. It is worth mentioning that ADMM-based methods have shown considerable popularity in solving the distributed optimization [36], [38], [39], [41], [42], [43], [44].

b) Partial device participation. For realistic purpose, plentiful algorithms have been developed to choose partial devices for the training at every iteration, thereby enabling us to eliminate the stragglers’ effect. Popular representatives consist of FedAvg [19], SCAFFOLD [45], FedProx [31], FedAlt [46], [47], [48], FedSim [49], [48], FedDCD [50], and FedSPD-DP [51]. The former five algorithms aim at solving...
the primal optimization problem while the latter two also investigate the dual problem.

1.2 Our contributions
The main contribution of this paper is to develop an inexact ADMM-based FL algorithm (FedADMM, see Algorithm 1) with the following advantages.

a) Communication and computation efficiency. The framework states the global averaging occurs only at certain steps (e.g., at steps $k$ being a multiple of a pre-defined integer $k_0$). This means CR can be affected by setting a proper $k_0$. It is shown that the larger $k_0$ the fewer CR for our algorithm to converge, see Figure 2. In addition to the communication efficiency, FedADMM allows selected clients to solve their sub-problems approximately with a flexible accuracy. In this regard, clients can relax the accuracy to lessen the computational complexity.

b) Eliminating the stragglers’ effect. In FedADMM, at each round of communication, the sever divides all clients into two groups. One group adopts the inexact ADMM to update their parameters, while parameters in the second group remain unchanged, which means the server can put stragglers into the second group to diminish their impact on the training.

c) Convergence under mild conditions. It is noted that most of the aforementioned algorithms in Section 1.1 impose relatively strong assumptions on the model to establish the convergence. Common assumptions comprise the gradient Lipschitz continuity (also known as $L$-smoothness), convexity, or strong convexity. However, we have proven that FedADMM converges to a stationary point of the learning optimization problem with a sub-linear convergence rate only under two mild conditions: gradient Lipschitz continuity and the coerciveness of the objective function, see Theorems 4.2 and 4.3.

d) High numerical performance. The numerical comparison with several state-of-the-art algorithms has demonstrated that FedADMM can learn the parameter using the fewest CR and the shortest computational time.

1.3 Organization
The paper is organized as follows. In the next section, we provide some mathematical preliminaries. In Section 3, we present algorithm FedADMM, followed by highlighting its advantages. We establish its global convergence and convergence rate in Section 4. Numerical comparison and concluding remarks are given in the last two sections.

2 PRELIMINARIES
In this section, we present the notation to be employed throughout this paper and introduce ADMM and FL.

2.1 Notation
We use plain, bold, and capital letters to present scalars, vectors, and matrices, respectively, e.g., $k$ and $\sigma$ are scalars, $w$ and $\pi$ are vectors, $W$ and $\Pi$ are matrices. Let $\lceil \cdot \rceil$ be the largest integer smaller than $\cdot + 1$ (e.g., $\lceil 1.1 \rceil = \lceil 2 \rceil = 2$). Denote $\{1, 2, \cdots, m\}$ with $\cdot := \cdot$ meaning define and $\mathbb{R}^n$ the $n$-dimensional Euclidean space equipped with inner product $\langle \cdot, \cdot \rangle$ defined by $\langle w, z \rangle := \sum_i w_i z_i$. The 2-norm is written as $\| \cdot \|$, i.e., $\| w \|_2^2 = \langle w, w \rangle$. Function $f$ is said to be gradient Lipschitz continuous with a constant $r > 0$ if

$$\| \nabla f(w) - \nabla f(z) \| \leq r \| w - z \|$$

for any two vectors $w$ and $z$, where $\nabla f(w)$ is the gradient of $f$ with respect to $w$. Hereafter, for two groups of vectors $w_i$ and $\pi_i$ in $\mathbb{R}^n$, we denote $W := (w_1, w_2, \cdots, w_m)$, $\Pi := (\pi_1, \pi_2, \cdots, \pi_m)$.
Similar rules are also applied for $W^k, W^+, W^\infty$ and $\Pi^k, \Pi^+, \Pi^\infty$. Here $k, *$ and $\infty$ mean the iteration number, optimality and accumulation, e.g., see Corollary 4.1.

2.2 ADMM
We refer to the earliest work [52] and a nice book [42] for more details of ADMM and briefly introduce it as follows: Given an optimization problem,

$$\min_{w \in \mathbb{R}^n, \pi \in \mathbb{R}^n} f(w) + g(z), \text{ s.t. } Aw + Bz = b = 0,$$

where $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{q \times n}$, and $b \in \mathbb{R}^p$, its corresponding augmented Lagrange function is

$$L(w, z, \pi) := f(w) + g(z) + \langle Aw + Bz - b, \pi \rangle + \frac{\sigma}{2} \| Aw + Bz - b \|^2,$$

where $\pi$ is the Lagrange multiplier and $\sigma$ is a given positive constant. Then starting with an initial point $(w^0, z^0, \pi^0)$, ADMM performs the following steps iteratively,

$$\begin{align*}
W^{k+1} := & \arg\min_{w \in \mathbb{R}^n} L(w, z^k, \pi^k), \\
Z^{k+1} := & \arg\min_{z \in \mathbb{R}^n} L(w^{k+1}, z, \pi^k), \\
\Pi^{k+1} := & \pi^k + \sigma(Aw^{k+1} + Bz^{k+1} - b).
\end{align*}$$

2.3 Federated learning
Suppose we have $m$ local clients (or devices) with datasets $\{D_1, D_2, \cdots, D_m\}$. Each client has the loss

$$f_i(w) := \frac{1}{d_i} \sum_{x \in D_i} \ell_i(w; x),$$

where $\ell_i(\cdot; x) : \mathbb{R}^n \mapsto \mathbb{R}$ is a continuous loss function and bounded from below, $d_i$ is the cardinality of $D_i$, and $w \in \mathbb{R}^n$ is the parameter to be learned. The overall loss function can be defined by

$$f(w) := \sum_{i=1}^m \alpha_i f_i(w),$$

where $\alpha_i$ is a positive weight satisfying $\sum_{i=1}^m \alpha_i = 1$. The task of FL is to learn an optimal parameter $w^*$ that minimizes the overall loss, namely,

$$w^* := \arg\min_{w \in \mathbb{R}^n} f(w).$$

Since $f_i$ is supposed to be bounded from below, we have

$$f^* := f(w^*) > -\infty.$$
Algorithm 1: FL via inexact ADMM (FedADMM)

Initialize an integer \( k_0 > 0 \) and \( \Omega^0 = [m] \). Set \( k = 0 \). Denote \( \tau_k := [k/k_0] \) and \( g^k_i := \alpha_i \nabla f_i(w^k_i) \). All clients \( i \in [m] \) initialize \( \epsilon_i^0, \sigma_i > 0, \nu_i \in [1/2, 1], \) \( w^0_i, \pi_i^0 = -g^0_i, z_i^0 = \sigma_i w_i^0 + \pi_i^0 \) and send the server \( \sigma_i \) to calculate \( \sigma = \sum_{i=1}^m \sigma_i \).

for \( k = 0, 1, 2, \ldots \) do

if \( k \in \mathcal{K} := \{0, k_0, 2k_0, 3k_0, \ldots \} \) then

Weights upload: (Communication occurs) Clients in \( \Omega^k \) send their parameters \( \{z_i^k : i \in \Omega^k \} \) to the server.

Global averaging: The server calculates average parameter \( w^{k+1} \) by

\[
 w^{k+1} = \frac{1}{\sigma} \sum_{i=1}^m z_i^k. \tag{4}
\]

Weights feedback: (Communication occurs) The server randomly selects clients in \([m]\) to form a subset \( \Omega^{k+1} \) and broadcasts them parameter \( w^{k+1} \).

end

for every \( i \in \Omega^{k+1} \) do

Local update: Client \( i \) updates its parameters as follows:

\[
 \epsilon_i^{k+1} \leq \nu_i \epsilon_i^k, \tag{5}
\]

Find \( w_i^{k+1} \) such that \( \| g_i^{k+1} + \pi_i^k + \sigma_i(w_i^{k+1} - w^{k+1}) \|^2 \leq \epsilon_i^{k+1} \) by solving \( \min_{w_i} \mathcal{L}(w^{k+1}, w_i, \pi_i^k) \),

\[
 \pi_i^{k+1} = \pi_i^k + \sigma_i(w_i^{k+1} - w^{k+1}), \tag{6}
\]

\[
 z_i^{k+1} = \sigma_i w_i^{k+1} + \pi_i^{k+1}. \tag{7}
\]

end

for every \( i \notin \Omega^{k+1} \) do

Local invariance: Client \( i \) keeps its parameters by

\[
 (\epsilon_i^{k+1}, w_i^{k+1}, \pi_i^{k+1}, z_i^{k+1}) = (\epsilon_i^k, w_i^k, \pi_i^k, z_i^k). \tag{9}
\]

end

end

3 FL via Inexact ADMM

By introducing auxiliary variables, \( w_i = w, i \in [m] \), problem (2) can be rewritten as the following form,

\[
 \min_{w,\Pi} \sum_{i=1}^m \alpha_i f_i(w_i), \text{ s.t. } w_i = w, i \in [m]. \tag{10}
\]

Throughout the paper, we shall focus on the above optimization problem instead of problem (2) as they are equivalent to each other. For simplicity, we also denote

\[
 F(W) := \sum_{i=1}^m \alpha_i f_i(w_i). \tag{11}
\]

Clearly, \( F(w, w, \ldots, w) = f(w) \).

3.1 Algorithmic design

To implement ADMM for our problem (10), the corresponding augmented Lagrange function can be defined by,

\[
 \mathcal{L}(w, \Pi) := \sum_{i=1}^m \mathcal{L}(w, w_i, \pi_i) \tag{12}
\]

\[
 \mathcal{L}(w, w_i, \pi_i) := \alpha_i f_i(w_i) + \langle w_i - w, \pi_i \rangle + \frac{\alpha_i}{2} \|w_i - w\|^2,
\]

where \( \Pi \) is the Lagrange multiplier, and \( \sigma_i > 0, i \in [m] \). The framework of ADMM for problem (10) is given as follows:

For an initial point \((w^0, W^0, \Pi^0)\) and any \( k \geq 0 \), perform the following updates iteratively,

\[
 \begin{align*}
 w^{k+1} &= \arg\min_w \mathcal{L}(w, W^k, \Pi^k), \\
 W^{k+1} &= \frac{1}{\sum_{i=1}^m (\sigma_i w_i^k + \pi_i^k)}, \\
 \Pi_i^{k+1} &= \arg\min_{\Pi} \mathcal{L}(w^{k+1}, W_i^{k+1}, \Pi, \pi_i^k), \quad i \in [m], \\
 \pi_i^{k+1} &= \pi_i^k + \sigma_i(w_i^{k+1} - w_i^{k+1}), \quad i \in [m].
\end{align*} \tag{13}
\]

To employ the above framework into FL, we treat \( w_i^{k+1} \) as the global parameter updated by a central server and \( (w_i^{k+1}, \pi_i^{k+1}) \) as the local parameters updated by local client \( i \in [m] \). However, this framework encounters several drawbacks. i) It repeats the three updates at every step, which means that the local clients and the central server have to communicate at every step, leading to communication inefficiency. ii) Solving the second sub-problem in (13) may incur an expensive computational cost as it generally does not admit a closed-form solution. iii) Due to inadequate transmission resources or limited computational capacity, some clients may delay sharing parameters (i.e., stragglers’ effect). So it is necessary to avoid selecting these clients.

To overcome these drawbacks, we cast a new algorithm into Algorithm 1 which aims at i) reducing communication rounds by averaging parameters only at certain steps (i.e., (4) occurs when \( k \in \mathcal{K} \)), ii) alleviating the computational burdens for clients by solving their sub-problems inexact (i.e., computing (6)), and iii) diminishing the stragglers’ effect by selecting a portion of clients (i.e., clients in \( \Omega^{k+1} \)).
to join in the training at every step. More precisely, we have the following advantageous properties.

3.2 Communication efficiency

Directly performing framework (13) in an FL setting leads to the communication between local clients and the central server at every step, which would consume large amounts of communication resources (e.g., power and bandwidth). Therefore, in Algorithm 1, we allow a portion of clients (i.e., clients in $\Omega^{k+1}$) to update their parameters a few times (i.e., $k_0$ times) and then upload them to the central server. In other words, the central server collects parameters from local clients only at step $k \in K = \{0, k_0, 2k_0, 3k_0, \cdots \}$. Here, choosing a proper $k_0$ can reduce CR significantly. It is worth mentioning that such an idea has been extensively employed in [21], [22], [23], [24], [25], [26].

3.3 Fast computation

We emphasize that $w_{i}^{k+1}$ in (6) is well defined. In fact, let $v_i^*$ be any optimal solution to $\min_{w_i} L(w_{i}^{k+1}, w_i, \pi^k)$. Then it satisfies the following optimality condition,

$$
\alpha_i \nabla f_i(v_i^*) + \pi_i^k + \sigma_i(v_i^* - w_i^{k+1}) = 0.
$$

This means there always exists a point satisfying the condition in (6). Since $\min_{w_i} L(w_{i}^{k+1}, w_i, \pi^k)$ is an unconstrained optimization problem, many solvers can be used to solve it. However, we are interested in algorithms that can find $w_{i}^{k+1}$ quickly. Particularly, we initialize $v_i^0 = w_i^{k+1}$, and perform the following steps, for $\ell = 0, 1, 2, \cdots, \kappa$,

$$
v_{i}^{\ell+1} = \arg\min_{v_{i}^k} (w_i - w_i^{k+1}, \pi_i^k) + \frac{\sigma_i}{2} \| w_i - w_{i}^{k+1} \|_2^2 + \frac{1}{\alpha_i}(f_i(v_{i}^k) + \langle \nabla f_i(v_{i}^k), w_i - v_{i}^k \rangle + \frac{\sigma_i}{2} \| w_i - v_{i}^k \|_2^2)
$$

$$
= \frac{1}{\alpha_i r_i} (\alpha_i r_i v_i^k + \sigma_i w_i^{k+1} - (\alpha_i \nabla f_i(v_{i}^k) + \pi_i^k)),
$$

where $r_i > 0$ which can be set as the Lipschitz continuous constant if $f_i$ is Lipschitz continuous and $\kappa$ is a given maximum number of steps to update $v_i^k$. The following theorem states that using (15) to find $w_{i}^{k+1} = v_{i}^{k+1}$ can guarantee the condition in (6) within a small number of iterations $\kappa$.

**Theorem 3.1.** Suppose that every $f_i$, $i \in [m]$ is gradient Lipschitz continuous with $r_i > 0$ and Hessian matrix $\nabla^2 f_i \succeq -s_i I$ with $s_i \geq 0$. By setting $\sigma_i = \alpha_i s_i + \rho \alpha_i r_i / 2$ with $\rho > 1$, client $i \in [m]$ can find $w_{i}^{k+1} = v_{i}^{k+1}$ such that (6) through (15) with at most $\kappa$ steps, where

$$
\kappa = \log_{\rho} \left[ \frac{2(|\alpha_i s_i^2 r_i^2 + \sigma_i^2 | w_i^{k+1} - v_i^k |^2)}{\kappa_i^2} \right] - 1.
$$

Here, $\nabla^2 f_i \succeq -s_i I$ stands for $\nabla^2 f_i + s_i I \succeq 0$, a positive semi-definite matrix. All convex and plentiful non-convex functions satisfy this condition. For convex functions, we could choose $s_i = 0$. Based on (16), clients can set a slightly large accuracy $\epsilon_i^{k+1}$ to ensure a small $\kappa$ for the sake of fastening their computation. Therefore, the computational cost can be saved in comparison with solving the second sub-problem of (13) exactly.

3.4 Coping with straggler's effect

The framework of FedADMM integrates partial device participation and hence can deal with the straggler’s effect. According to [20], this can be done as follows: By setting a threshold $m_0 \in [1, m]$, the server collects the outputs of the first $m_0$ responded clients (to form $\Omega^{k+1}$). After collecting $m_0$ outputs, the server stops waiting for the rest, namely the rest clients are deemed as stragglers in this iteration.

Partial device participation has been exploited by FedAvg [19], FedProx [31], FedAlt [48], and so forth. Here, FedAvg presented in Algorithm 2 corresponds to the case of $B = \infty$ in its original version. There are some differences among these algorithms and ours. First of all, the global averaging for FedProx and FedAlt is taken on the selected clients in $\Omega^{k}$, that is,

$$
w_{i}^{k+1} = \frac{1}{|\Omega^{\tau_k}|} \sum_{i \in \Omega^{\tau_k}} w_i^k,
$$

while FedADMM and FedAvg assemble parameters of all clients, see (4) and (18). Moreover, the other three algorithms average parameters $w_{i}^{k}$ directly while FedADMM aggregates $z_{i}^{k}$ which is a combination of primal variable $w_{i}^{k}$ and dual variable $\pi_{i}^{k}$. To this end, it is more secured to protect clients’ data when communicating with the server.

**Algorithm 2: FedAvg.**

```
Initialize an integer $k_0, \gamma > 0$ and $\Omega^{0} = [m]$. Set $k = 0$. All clients $i \in [m]$ initialize $w_{i}^{0} = 0$.

for $k = 0, 1, 2, \cdots$ do

if $k \in K := \{0, k_0, 2k_0, 3k_0, \cdots \}$ then

Weights upload: (Communication occurs)

Clients in $\Omega^{k}$ send $\{w_{i}^{k} : i \in \Omega^{k}\}$ to the server.

Global averaging: The server averages $w_{i}^{k+1}$ by

$$
w_{i}^{k+1} = \frac{1}{m} \sum_{i=1}^{m} w_{i}^{k}.
$$

Weights feedback: (Communication occurs)

The server randomly selects clients to form $\Omega^{k+1}$ and broadcasts them $w_{i}^{k+1}$.

end

for every $i \in \Omega^{k+1}$ do

Local update: Client $i$ updates its parameters by

$$
w_{i}^{k+1} = \begin{cases} w_{i}^{k+1} - \frac{\gamma}{m} \nabla f_{i}(w_{i}^{k+1}), & k \in K, \\ w_{i}^{k} - \frac{\gamma}{m} \nabla f_{i}(w_{i}^{k}), & k \notin K. \end{cases}
$$

for every $i \notin \Omega^{k+1}$ do

Local invariance: Client $i$ keeps $w_{i}^{k+1} = w_{i}^{k}$.

end

end
```

3.5 Local invariance

We would like to point out that clients outside $\Omega^{k}$ do nothing at steps $k, k + 1, \cdots, k + k_0 - 1$. We use (9) for the purpose of notational convenience when conducting convergence analysis. Moreover, (9) also allows the server to record the previous uploaded parameters from clients outside $\Omega^{k}$. Precisely, for any $i \notin \Omega^{k}$, at step $k \in K$, let
4 Convergence analysis

We aim to establish the global convergence and convergence rate for FedADMM in this section, before which we define the optimality conditions of problems (10) and (2) as follows.

4.1 Stationary point

**Definition 4.1.** A point \((w^*, W^*, \Pi^*)\) is a stationary point of problem (10) if it satisfies

\[
\begin{align*}
\alpha_i \nabla f_i(w^*_i) + \pi^*_i &= 0, \quad i \in [m], \\
w^*_i - w^* &= 0, \quad i \in [m], \\
\sum_{i=1}^{m} \pi^*_i &= 0.
\end{align*}
\]

It is not difficult to prove that any locally optimal solution to problem (10) must satisfy (19). If \(f_i\) is convex for every \(i \in [m]\), then a point is a globally optimal solution if and only if it satisfies condition (19). Moreover, a stationary point \((w^*, W^*, \Pi^*)\) of problem (10) indicates

\[
\nabla f(w^*) = \sum_{i=1}^{m} \alpha_i \nabla f_i(w^*_i) = -\sum_{i=1}^{m} \pi^*_i = 0.
\]

That is, \(w^*\) is also a stationary point of problem (2).

4.2 Some assumptions

**Assumption 4.1.** Every \(f_i, i \in [m]\) is gradient Lipschitz continuous with a constant \(r_i > 0\).

**Assumption 4.2.** Function \(f\) is coercive. That is, \(f(w) \to +\infty\) when \(\|w\| \to +\infty\).

**Scheme 4.1.** The sever randomly selects \(\Omega^\tau\) that satisfies

\[
\Omega^\tau = \Omega^\tau + 1 \cup \Omega^\tau + 2 \cup \cdots \cup \Omega^\tau + s_0 = [m], \quad \forall \tau = 0, s_0, 2s_0, \cdots
\]

where \(s_0\) is a pre-defined positive integer.

Such a scheme indicates that for each group of \(s_0\) sets \(\{\Omega^{\tau+1}, \Omega^{\tau+2}, \cdots, \Omega^{\tau+s_0}\}\), all clients should be chosen at least once. In other words, for any client \(i \in [m]\), the maximum gap between its two consecutive selections is no more than \(s_0\), namely,

\[
\max \left\{ u - v : \quad i \in \Omega^\tau, i \in \Omega^\nu, i \notin \Omega^\tau, \tau = v + 1, \cdots, u - 1 \right\} \leq s_0.
\]

**Remark 4.1.** Scheme 4.1 can be satisfied with a high probability. In fact, if \(\Omega^1, \Omega^2, \cdots\) are selected independent and indices in \(\Omega^\tau\) are uniformly sampled from \([m]\) without replacement, then the probability of client \(i\) being selected in \(\{\Omega^\tau, \Omega^{\tau+1}, \cdots, \Omega^{\tau+s_0}\}\) is

\[
p_i = 1 - \prod_{i \notin \Omega^{\tau+1}} (1 - \frac{|\Omega^{\tau+1}|}{m}) \prod_{i \notin \Omega^{\tau+2}} (1 - \frac{|\Omega^{\tau+2}|}{m}) \cdots \prod_{i \notin \Omega^{\tau+s_0}} (1 - \frac{|\Omega^{\tau+s_0}|}{m}),
\]

which tends to 1. For example, \(p_i = 1 - 10^{-5}\) if \(s_0 = 5\) and \(|\Omega^\tau| = 0.9m\) for any \(\tau \geq 1\).

4.3 Global convergence

The sketch of showing the convergence is as follows: by defining sequence \(\{\tilde{L}^k\}\) as

\[
\tilde{L}^k = \tilde{L}^k + \sum_{i=1}^{m} \frac{20c_i}{(1 - c_i)\tau},
\]

we first prove its decreasing property with a descent scale \(\sum_{i=1}^{m} \frac{1}{10} (\|w^\tau_{k+1} - w^\tau_k\|^2 + \|w^f_{k+1} - w^f_k\|^2)\). It allows us to claim the convergence of \(\{\tilde{L}^k\}\) and the vanishing of \(\|w^\tau_{k+1} - w^\tau_k\|, \|w^f_{k+1} - w^f_k\|, \) and \(\|w^f_k - w^f_k\| \). Then these properties enable us to obtain (23) and (24), which together with the optimality conditions shows the convergence of sequence \(\{w^\tau_k, W^\tau_k, \Pi^k\}\) itself. Therefore, we first establish the decreasing property of sequence \(\{\tilde{L}^k\}\).

**Lemma 4.1.** Under Assumption 4.1, it holds that

\[
\tilde{L}^k - \tilde{L}^{k+1} \geq \sum_{i=1}^{m} \sigma_i (\|w^\tau_{k+1} - w^\tau_k\|^2 + \|w^f_{k+1} - w^f_k\|^2).
\]

The above result enables us to show the convergence of three sequences \(\{f(w^\tau_k)\}, \{F(W^\tau_k)\}, \) and \(\{L^k\}\).

**Theorem 4.1.** Suppose that Assumptions 4.1 and 4.2 hold. Every client \(i \in [m]\) chooses \(s_i \geq 3\alpha_i r_i\) and the sever selects \(\Omega^\tau_k\) as Scheme 4.1. Then the following results hold.

a) Sequences \(\{w^\tau_k, W^\tau_k, \Pi^k\}\) is bounded.

b) Three sequences \(\{L^k\}, \{F(W^\tau_k)\}, \) and \(\{f(w^\tau_k)\}\) converge to the same value, namely,

\[
\lim_{k \to \infty} L^k = \lim_{k \to \infty} F(W^\tau_k) = \lim_{k \to \infty} f(w^\tau_k).
\]

c) \(\nabla F(W^\tau_k)\) and \(\nabla f(w^\tau_k)\) eventually vanish, namely,

\[
\lim_{k \to \infty} \nabla F(W^\tau_k) = \lim_{k \to \infty} \nabla f(w^\tau_k) = 0.
\]

Theorem 4.1 establishes the convergence property of the objective function values. In the following theorem, we would like to see the convergence performance of sequence \(\{w^\tau_k, W^\tau_k, \Pi^k\}\) itself, which requires more conditions.

**Theorem 4.2.** Suppose that Assumptions 4.1 and 4.2 hold. Every client \(i \in [m]\) chooses \(s_i \geq 3\alpha_i r_i\) and the sever selects \(\Omega^\tau_k\) as Scheme 4.1. Then the following results hold.

a) Any accumulating point \((w^\infty, W^\infty, \Pi^\infty)\) of sequence \(\{w^\tau_k, W^\tau_k, \Pi^k\}\) is a stationary point of problem (10), where \(w^\infty\) is a stationary point of problem (2).

b) If further assuming that \(w^\infty\) is isolated, then the whole sequence converges to \((w^\infty, W^\infty, \Pi^\infty)\).

We point out that the establishments of Theorems 4.1 and 4.2 do not rely on the choices of \(\Omega^\tau_k\) explicitly due to Scheme 4.1. If the sever generates \(\Omega^\tau_k\) randomly rather than using Scheme 4.1, then the above two theorems are valid with a high probability. In addition, since no convexity of \(f_i\) or \(f\) is imposed, the sequence is guaranteed to converge to the stationary point of problems (10) and (2). If we have the convexity of \(f\), then the sequence converges to the optimal solution to (10) and (2), stated by the following corollary.
Corollary 4.1. Suppose that Assumptions 4.1 and 4.2 hold and 
\( f \) is convex. Every client \( i \in [m] \) chooses \( \sigma_i \geq 3\alpha_ir_i \) and the sever selects \( \Omega^n \) as Scheme 4.1. Then the following results hold.

a) Three sequences converge to the optimal function value of 
problem (2), namely,
\[
\lim_{k \to \infty} F(W^k) = \lim_{k \to \infty} f(W^k) = \lim_{k \to \infty} f(w^*k) = f^*.
\]  
(25)

b) Any accumulating point \( \{w^*, W^*, \Pi^*\} \) of sequence 
\( \{w^k, W^k, \Pi^k\} \) is an optimal solution to problem (10), 
where \( w^* \) is an optimal solution to problem (2).

c) If \( f \) is strongly convex, then whole sequence converges 
the unique optimal solution \( (w^*, W^*, \Pi^*) \) to problem (10), 
where \( w^* \) is the unique optimal solution to problem (2).

Remark 4.2. Regarding assumptions in Corollary 4.1, \( f \) being 
strongly convex does not require that every \( f_i \) \( i \in [m] \) is strongly 
convex. If one of \( f_i \) is strongly convex and the remaining is 
convex, then \( f = \sum_{i=1}^m \alpha_i f_i \) is strongly convex. Moreover, 
the strongly convexity suffices to the coerciveness of \( f \). Therefore, 
under the strongly convexity, Assumption 4.2 can be exempted.

4.4 Convergence rate

We have shown that Algorithm 1 converges. Now, we would 
like to see how fast this convergence is, stated as follows.

Theorem 4.3. Suppose that Assumptions 4.1 holds. Every client 
chooses \( \sigma_i \geq 3\alpha_ir_i \) \( i \in [m] \) and the sever selects \( \Omega^n \) as Scheme 4.1. Then for any \( k > s_0k_0 \) it has
\[
\min_{s=1,2,\ldots,k}\left\| \nabla f(w^{s+1}) \right\|^2 \leq \frac{c_k}{k-s_0k_0},
\]  
(26)

where \( c := \frac{940\rho_m\sigma(\rho_m)\omega}{{9}} \cdot \frac{1}{\max_{i \in [m]} \sigma_i^2} \frac{1}{\epsilon} + \frac{m}{\sum_{i=1}^m \frac{1}{\epsilon} \frac{1}{\epsilon}} .
\)

According to the above theorem, the minimal value among 
\( \left\{ \left\| \nabla f(w^{s+1}) \right\|^2, \sigma \in [k] \right\} \) vanishes with a rate \( O(1/k) \), 
a sub-linear rate. We emphasize that the establishment of 
such a convergence rate requires nothing but the assumption 
of gradient Lipschitz continuity, namely, Assumption 4.1. 
Similar results can be found in many literature. For example, 
in [45] the convergence rate is about \( O(\sqrt{1/K}) \) 
while the rate in [30], [53] is about \( O(1/k) \) but has been 
obtained under the full device participation (corresponding 
to the case of \( \sigma_0 = 1 \) in Scheme 4.1).

Remark 4.3. Theorem 4.3 suggests that Algorithm 1 should be 
terminated if the following condition is satisfied,
\[
\left\| \nabla f(w^{s+1}) \right\|^2 \leq \varepsilon,
\]  
(27)

where \( \varepsilon \) is a given tolerance. Based on (26), after
\[
k = \left\lceil \frac{(c+\rho_m)\omega}{\varepsilon} \right\rceil
\]  
(28)

iterations, Algorithm 1 meets (27) and the total CR is
\[
CR := \left\lceil \frac{2k}{K_k} \right\rceil = \left\lceil \frac{2(c+\rho_m)\omega}{\varepsilon} \right\rceil.
\]  
(29)

The above relation implies that the larger \( s_0 \) the more CR 
required by Algorithm 1 to converge, which is reasonable. In fact, one 
can observe that Scheme 4.1 can be satisfied with a larger \( s_0 \). This is 
because, a larger \( s_0 \) allows us to choose fewer clients to form \( \Omega^n \), 
namely, fewer clients participating in the training at every step, 
which apparently results in slower convergence. As a consequence, 
the algorithm needs higher CR, thereby wasting communication 
resources. Hence, in order to reduce CR, it is essential to set an 
appropriately small \( s_0 \). However, to meet Scheme 4.1, a small \( s_0 \) 
means that more clients take part in the training, which will incur 
higher computational complexity.

5 NUMERICAL EXPERIMENTS

In this section, we conduct some numerical experiments to 
demonstrate the performance of FedADMM (available at 
https://github.com/ShenglongZhou/FedADMM). All numerical 
experiments are implemented through MATLAB (R2019a) on a laptop with 32GB memory and 2.3Ghz CPU.

5.1 Testing example

Example 5.1 (Linear regression with non-i.i.d. data). For this 
problem, local clients have their objective functions as
\[
f_i(w) = \frac{1}{d_i} \sum_{t=1}^{d_i} (a_i(w, w) - b_i)^2,
\]

where \( a_i(w, w) \) is the \( i\)-th sample for client \( i \). We first pick \( m \) integers \( d_1, \ldots, d_m \) randomly from \([50, 100]\) 
and denote \( d := d_1 + \cdots + d_m \). Then we generate \( d/3 \) samples 
(a, b) from the standard normal distribution, \( d/3 \) samples from 
the Student’s t distribution with degree 5, and \( d/3 \) samples from 
the uniform distribution in \([-5, 5]\). Now we shuffle all 
shapes and divide them into \( m \) parts with sizes \( d_1, \ldots, d_m \) 
for \( m \) clients. In the regard, each client has non-i.i.d. data.

Example 5.2 (Logistic regression). For this problem, local 
clients have their objective functions as
\[
f_i(w) = \frac{1}{d_i} \sum_{t=1}^{d_i} \left( \ln(1 + e^{a_i(w, w)}) - b_i(a_i(w, w)) \right) + \frac{\lambda}{2} \| w \|^2,
\]

where \( a_i(w, w) \) is \( i\)-th sample. \( b_i(w, w) \in \{0, 1\} \), and \( \lambda > 0 \) is a penalty parameter 
(e.g., \( \lambda = 0.001 \) in our numerical experiments). We use two 
real datasets described in Table 1 to generate (a, b). Again, we 
randomly split \( d \) samples into \( m \) groups for \( m \) clients.

| Data        | Datasets Source | n    | d    |
|-------------|----------------|------|------|
| qot         | Quasar oral toxicity | UCI | 1024 | 8992 |
| rls         | real-sim        | Libsvm | 20958 | 72309 |

5.2 Implementations

We fix \( \alpha_i = 1/m, i \in [m] \) in model (2) and initialize \( w^0 \) = \( \pi^0 \) = 0. Parameters are set as follows: for each \( i \in [m] \), let \( d_i = 20d, \) and \( \nu_i = 0.95 \), and \( \sigma_i = 0.2r_i/m \), where \( r_i \) is the 
gradient Lipschitz continuous constant for \( f_i \). We terminate 
our algorithm if the following condition is satisfied,
\[
\left\| \nabla f(w^{s+1}) \right\|^2 < \min \left\{ \frac{1}{s} \left\| \nabla f(0) \right\|^2, \frac{\lambda m}{4n} \right\},
\]  
(30)

where \( \varepsilon = 10^{-3} \) for Example 5.1 and \( \varepsilon = 10^{-7} \) for Example 5.2. In the subsequent numerical experiments, instead of 
using Scheme 4.1, we generate \( \Omega^n \) randomly since it is easier 
than Scheme 4.1. As mentioned in Remark 4.1, Scheme 4.1 
can be guaranteed with a high probability in this way. Specifying, 
let \( \Omega^1, \Omega^2, \ldots \) be selected independently with 
\( \left\| \Omega^n \right\| = \rho \) for any \( \tau \geq 1 \), where \( \rho \in \{0, 1\} \). Indices in each 
\( \Omega^n \) are uniformly sampled from \([m]\) without replacement.
5.3 Benchmark algorithms

We will compare FedADMM with FedAvg [19] described in Algorithm 2, FedProx [31], and FedAlt and FedSim [48]. For FedProx, every selected client \(i \in \Omega^k \) needs to approximately solve a sub-problem at each iteration. We adopt the gradient descent method to tackle it using an initial point \( w^k_{\Omega^k} \) if \( k \in \mathcal{K} \) and \( w^k_i \) if \( k \notin \mathcal{K} \). For FedAlt and FedSim, we also employ the strategy that the global averaging only occurs at \( k \in \mathcal{K} \) and exploit a partial model personalization \( h_i \) from [49], i.e.,

\[
h_i(w, v) = (1 - \alpha)f_i(w) + \alpha f_i(v) + \frac{\mu}{2}\|w - v\|^2,
\]

where \( \alpha = 0.5 \) and \( \mu = 0.001 \) are used in the numerical experiments. To ensure fair comparison, we initialize all algorithms with \( w^0 = w^0_i = 0, i \in [m] \). In addition, we first implement FedADMM to solve the problem and terminate it if its solution \( w^t \) satisfies condition (30). Then we employ the other four algorithms to solve the problem if its solution \( w \) meets the following condition,

\[
f(w) - f(w^t) \leq 2(1 + |f(w^t)|)10^{-4}.
\]

This condition allows all algorithms to stop with producing similar objective function values.

5.4 Numerical comparisons

We compare five algorithms by reporting the following factors: objective function values \( f(w^t) \), CR, and computational time (in seconds). It is noted that there are four influential parameters \((n, m, \rho, k_0)\), where \( n \) is the dimension of the solution, \( m \) is the number of clients, \( k_0 \) has the impact on the CR, and \( \rho \in (0, 1] \) is the participation rate (i.e., the bigger \( \rho \) the more clients to be chosen for the training at every iteration). To see the effect of one parameter, we will fix the others in the sequel.

5.4.1 Effect of \( k_0 \)

To see this, we fix \((n, m, \rho) = (100, 100, 0.5)\). Here, \( \rho = 0.5 \) means half clients chosen for the training (i.e., \( |\Omega^t| = 0.5m \)). First, we perform five algorithms to solve Example 5.1 under \( k_0 \in \{1, 10, 30, 50\} \) and report the results in Figure 1. One can observe that with the increasing of CR, all algorithms eventually achieve the same objective function value (i.e., the optimal one). Because of this, we will not report the objective function values in the subsequent numerical comparison. When \( k_0 = 1 \), the objective function values obtained by FedADMM decline slowly at the first several steps but approach the optimal value quickly afterwards. When \( k_0 > 1 \), it always outperforms the others as it uses the lowest CR.

We next generate 20 instances and report the results in terms of the median values in Figure 2, where each instance is solved by one algorithm under different choices of \( k_0 \in \{1, 5, 10, \cdots, 50\} \). For example, when \( k_0 = 10 \), FedADMM solves the 20 instances and obtains 20 values of CR. The data reported in the figure is the median of these 20 values. Based on the results presented in Figure 2, we have the following comments. For Example 5.1, when \( k_0 \) is increasing, there is a descending trend of CR for FedADMM but an ascending trend for the other four algorithms. However, for Example 5.2, CR generated by every algorithm is declining with the rising of \( k_0 \). Apparently, for both examples the larger \( k_0 \) the longer the computational time and FedADMM behaves the best in terms of using the fewest CR and running the fastest.

5.4.2 Effect of participation rate \( \rho \)

To see the effect of participation rate \( \rho \) on the performance of each algorithm, we fix \((n, m, k_0) = (100, 100, 10)\) and alter \( \rho \in (0.1, 0.2, \cdots, 0.9) \). Similarly, we report the median values over 20 instances in Figure 3. First, for Example 5.1, when \( \rho \) is getting bigger (i.e., more and more clients are selected for the training), as expected that every algorithm consumes fewer CR, which results in shorter computational
time. However, the picture for Example 5.2 is slightly different. From the figure, when ρ is varying, CR stabilizes at a certain level for FedADMM while slightly fluctuating for the other four algorithms. Basically, the higher value of ρ the longer computational time spent by every algorithm for this example. Once again, FedADMM outperforms the other algorithms for most scenarios.

5.4.3 Effect of m
Similarly, we fix \((n, ρ, k_0) = (100, 0.5, 10)\) but alter \(m \in \{50, 100, 150, 200\}\) for both examples. In addition, we also fix \(n = 1000\) for Example 5.1. Now according to the generation of Example 5.1, each instance has \(n = 1000\) features and \(d \in \{50000, 600000\}\) total samples. We use dataset \textsc{rils} for Example 5.2 since it has much more features and samples (i.e., \(n = 20958\) and \(d = 72309\)). Again, we report the results in terms of the median values of 20 instances in Figure 5. For Example 5.1, there are five descending trends for CR and ascending trends for the time. However, for Example 5.2, the larger \(m\) the higher CR and the time. We find that FedADMM runs much faster than the others (e.g., when \(m = 4000\), FedADMM, FedProx, FedSim, FedAlt, and FedAvg consume 73, 728, 4711, 3834, and 475 seconds, respectively). As always, FedADMM produces the most desirable results.

6 Conclusion
We developed an inexact ADMM-based FL algorithm. The periodic global averaging allows it to reduce CR so as to save communication resources. Solving sub-problems inexact alleviates clients’ computational burdens significantly, thereby accelerating the learning process. Partial device participation in the algorithm eliminates the stragglers’ effect. Those merits show the strong potential of FedADMM for real-world applications like vehicular communications, mobile edge and over-the-air computing.
[50] Z. Fan, H. Fang, and M. Friedlander, “A dual approach for federated learning,” arXiv preprint arXiv:2201.11183, 2022.
[51] Y. Li, S. Wang, T.-H. Chang, and C.-Y. Chi, “Federated stochastic primal-dual learning with differential privacy,” arXiv preprint arXiv:2204.12284, 2022.
[52] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximation,” Comput. Math. with Appl., vol. 2, no. 1, pp. 17–40, 1976.
[53] S. Zhou and G. Y. Li, “FedGiA: An efficient hybrid algorithm for federated learning,” arXiv preprint arXiv:2110.15318, 2022.
[54] J. Moré and D. Sorensen, “Computing a trust region step,” SIAM J. Sci. Statist. Comput., vol. 4, no. 3, pp. 533–572, 1983.

**APPENDIX A**

**SOME BASICS**

For any $w_1, w_2,$ and $w_i \in \{w_1, w_2\}$, it follows that $w_2 + t(w_1 - w_2) - w_i = (t - 1)(w_1 - w_2)$ or $t(w_1 - w_2)$. If function is gradient Lipschitz continuous with constant $r$, then the Mean Value Theorem suffices to

$$f(w_1) - f(w_2) - \langle \nabla f(w_1), w_1 - w_2 \rangle = \int_0^1 \nabla f(w_2 + t(w_1 - w_2) - w_i) \cdot (w_1 - w_2) dt$$

$$\leq \frac{1}{r} \| w_1 - w_2 \|^2$$

If $\nabla^2 f \succeq -sI$, then the Mean Value Theorem brings out

$$\langle \nabla f(w_1), w_1 - w_2 \rangle = \int_0^1 \langle \nabla^2 f(w_2 + t(w_1 - w_2)), w_1 - w_2 \rangle dt$$

$$\leq -s \| w_1 - w_2 \|^2$$

For any vectors $w_1, w_2$, and $t > 0$, we have

$$2(w_1, w_2) \leq \| w_1 \|^2 + 1/t \| w_2 \|^2$$

$$\| w_1 + w_2 \|^2 \leq (1 + t) \| w_1 \|^2 + (1 + 1/t) \| w_2 \|^2$$

$$\| \sum_{i=1}^m w_i \|^2 \leq m \sum_{i=1}^m \| w_i \|^2$$

For notational simplicity, hereafter, we denote

$$\Delta w^{k+1} := w^{k+1} - w^k$$

$$\Delta w_1^{k+1} := w_1^{k+1} - w_1^k$$

$$\Delta w_2^{k+1} := w_2^{k+1} - w_2^k$$

$$\Delta \pi_1^{k+1} := \pi_1^{k+1} - \pi_1^k$$

$$\Delta \pi_2^{k+1} := \pi_2^{k+1} - \pi_2^k$$

$$\epsilon_i^{k+1} := \epsilon_i^{k+1} - \epsilon_i^k$$

and let $\epsilon^k \rightarrow w$ stand for $\lim_{k \rightarrow \infty} \epsilon^k = w$. In the sequel, for notational simplicity we write

$$\sum_i := \sum_{i=1}^m$$

**APPENDIX B**

**PROOF OF THEOREM 3.1**

The main idea to prove this theorem is to show

$$\| v_i^{k+1} - v_i^* \|^2 \leq \varrho^{-1} \| v_i^* - v_i^* \|^2$$

which can be verified by using the optimality condition of problem (15) that is satisfied by $v_i^{k+1}$.

**Proof.** Since $v_i^{k+1}$ is a solution to problem (15), it satisfies the following optimality condition,

$$\alpha_i \nabla f_i(v_i^*) + \pi_i^k + \sigma_i(v_i^{k+1} - w_i^{k+1}) + \alpha_i r_i(v_i^{k+1} - v_i^*) = 0$$

which subtracting (14) gives rise to

$$- \sigma_i(v_i^{k+1} - v_i^*) - \alpha_i r_i(v_i^{k+1} - v_i^*)$$

$$= \alpha_i (\nabla f_i(v_i^*) - \nabla f_i(v_i^*))$$

$$= \alpha_i (\nabla f_i(v_i^*) - \nabla f_i(v_i^*) + \nabla f_i(v_i^*) - \nabla f_i(v_i^*))$$

Using the condition allows us to obtain

$$- \alpha_i \sigma_i \| v_i^{k+1} - v_i^* \|^2$$

$$\leq \langle v_i^{k+1} - v_i^*, \alpha_i \nabla f_i(v_i^*) - \nabla f_i(v_i^*) \rangle$$

$$= \langle v_i^{k+1} - v_i^*, -\sigma_i(v_i^{k+1} - v_i^*) - \alpha_i r_i(v_i^{k+1} - v_i^*) \rangle$$

$$+ \langle v_i^{k+1} - v_i^*, \alpha_i \nabla f_i(v_i^*) - \nabla f_i(v_i^*) \rangle$$

$$= \langle v_i^{k+1} - v_i^*, -\sigma_i(v_i^{k+1} - v_i^*) - \alpha_i r_i(v_i^{k+1} - v_i^*) \rangle$$

$$+ \alpha_i \langle \nabla f_i(v_i^*) - \nabla f_i(v_i^*) \rangle$$

$$- \sigma_i \| v_i^{k+1} - v_i^* \|^2$$

$$\leq \alpha_i \sigma_i \| v_i^{k+1} - v_i^* \|^2$$

$$+ \alpha_i \| v_i^{k+1} - v_i^* \|^2$$

$$\leq \| v_i^{k+1} - v_i^* \|^2$$

$$\leq \| v_i^{k+1} - v_i^* \|^2$$

$$\leq \alpha_i \sigma_i \| v_i^{k+1} - v_i^* \|^2$$

which immediately results in (35), thereby leading to

$$\| v_i^{k+1} - v_i^* \|^2 \leq \varrho \| v_i^* - v_i^* \|^2$$

$$\leq \cdots \leq \frac{1}{\varrho^k} \| v_i^* - v_i^* \|^2$$

Now letting $w_i^{k+1} = v_i^{k+1}$, we verify the condition in (6) by

$$\| g_i^{k+1} + \pi_i^k + \sigma_i w_i^{k+1} - w_i^{k+1} \|^2$$

$$\leq \| g_i^{k+1} - \alpha_i \nabla f_i(v_i^*) + \sigma_i w_i^{k+1} - v_i^* \|^2$$

$$\leq 2(\alpha_i^2 r_i^2 + \beta_i^2) \| w_i^{k+1} - v_i^* \|^2$$

$$\leq \| \varphi_i^k - \beta_i^2 / \alpha_i^2 \| v_i^* - v_i^* \|^2$$

which by $v_i^k = w_i^{k+1}$ implies that

$$\kappa = \log \frac{2(\alpha_i^2 r_i^2 + \beta_i^2) \| v_i^* - v_i^* \|^2}{\epsilon_i^{k+1}} - 1.$$
c) Under Assumption 4.1, for any $k \geq 0$ and any $i \in [m],$
\[ \| \Delta \pi_i^{k+1} \|^2 \leq \frac{6a_i r_i^2}{5} \| \Delta w_i^{k+1} \|^2 - \frac{24}{1-\nu_i} \epsilon_i^{k+1}. \] (38)

\[ d) \text{ Under Scheme 4.1, for any } i \in [m], \]
\[ \epsilon_i^{k+1} \rightarrow 0. \] (39)

**Proof.** a) For any $i \in [m]$ and at $(k+1)$th iteration, let $k_i$ be the largest integer in $[-1, k]$ such that $i \in \Omega^{k_i+1}$. This implies that client $i$ is not selected in all $\Omega^{k_i+2}, \Omega^{k_i+3}, \ldots, \Omega^{k_i+1}$, which by (9) yields
\[ (\epsilon_i^{k+1}, w_i^{k+1}, \pi_i^{k+1}, z_i^{k+1}) = (\epsilon_i^{k+1}, w_i^{k+1}, \pi_i^{k+1}, z_i^{k+1}) \]
\[ \forall \ell = k_i, k_i + 1, \ldots, k. \] (40)

For any client $i \in \Omega^{k+1}$, we have (8). For any client $i \notin \Omega^{k+1}$, if $k_i \geq 0$, then $(w_i^{k+1}, \pi_i^{k+1}, z_i^{k+1})$ also satisfies (8) due to $i \in \Omega^{k_i+1}$, which by condition (40) implies that $(w_i^{k+1}, \pi_i^{k+1}, z_i^{k+1})$ satisfies (8). If $k_i = -1$, this means that client $i$ has never been selected. Then by (40) and our initialization, we have
\[ x_i^{k+1} = z_i^{k+1} = z_i^0 = \sigma_i w_i^0 + \pi_i^0 = \sigma_i w_i^{k+1} + \pi_i^{k+1}. \]

Hence, (8) is still valid. Overall, we can conclude that (8) holds for every $i \in [m]$ and $k \geq 0$. Now, for any $k \in K$,\n\[ \sum_i (\sigma_i (w_i^k - w_i^{k+1}) + \pi_i^k) = \sum_i (w_i^k - w_i^{k+1}) + \sigma_i \Delta w_i^k \] (41)

b) For any $i \in \Omega^{k+1}$, solution $w_i^{k+1}$ in (6) satisfies
\[ \phi_i^{k+1} = g_i^{k+1} + \pi_i^{k+1} \]
\[ \| \phi_i^{k+1} \|^2 \leq \epsilon_i^{k+1}. \] (42)

For any $i \notin \Omega^{k+1}$, we have
\[ \phi_i^{k+1} = g_i^{k+1} + \pi_i^{k+1}, \]
\[ \| \phi_i^{k+1} \|^2 \leq \epsilon_i^{k+1} \]
due to $i \in \Omega^{k_i+1}$. This together with (40) implies that (37) is still true. So, (37) holds for any $i \in [m]$ and any $k \geq 0$.

c) For any $i \in \Omega^{k+1}$, it follows from (37) and the gradient Lipschitz continuity of $f_i$ that
\[ \| \Delta \pi_i^{k+1} \|^2 \leq \| g_i^{k+1} - g_i^k \|^2 + \| \phi_i^{k+1} - \phi_i^k \|^2 \]
\[ \leq \frac{6a_i r_i^2}{5} \| \Delta w_i^{k+1} \|^2 + 12(\epsilon_i^{k+1} + \epsilon_i^k) \] (43)
\[ \leq \frac{6a_i r_i^2}{5} \| \Delta w_i^{k+1} \|^2 + \frac{24}{1-\nu_i} \epsilon_i^{k+1}. \] (44)

For any $i \notin \Omega^{k+1}$, we have $\Delta \pi_i^{k+1} = 0$ by (9), and thus the above condition is still valid.

d) For sufficiently large $k \in K$, any client $i$ has been selected at least $k/(s_0 k_0)$ times due to (21). This means that (5) (e.g., $\epsilon_i^{k+1} = \epsilon_i^k/2$) occurs at least $k/(s_0 k_0)$ times, thereby leading to $\epsilon_i^{k+1} \rightarrow 0$. □

### C.2 Proof of Lemma 4.1

To estimate the upper bound of gap $(\mathcal{L}^{k+1} - \mathcal{L}^k)$, we first decompose it into three pieces as follows
\[ \mathcal{L}^{k+1} - \mathcal{L}^k = p_1^k + p_2^k + p_3^k, \] (45)

where
\[ p_1^k := \mathcal{L}(w^k \tau + 1, W^k, \Pi^k) - \mathcal{L}(w^k \tau + 1, W^k, \Pi^k), \]
\[ p_2^k := \mathcal{L}(w^k \tau + 1, W^k, \Pi^k) - \mathcal{L}(w^k \tau + 1, W^k, \Pi^k), \]
\[ p_3^k := \mathcal{L}(w^k \tau + 1, W^k, \Pi^k) - \mathcal{L}(w^k \tau + 1, W^k, \Pi^k). \]

Then we apply condition (36) for $w^k \tau + 1$, condition (37) for $w^k \tau + 1$, and conditions (7) and (9) for $\Pi^k$ to derive the upper bounds for $p_1^k$, $p_2^k$, and $p_3^k$, respectively. Finally, adding these bounds yields the upper bound of gap $(\mathcal{L}^{k+1} - \mathcal{L}^k)$.

**Proof.** Since $\sigma_i \geq 3\alpha_i r_i > 0$, it follows
\[ 0 < \alpha_i r_i \leq \frac{\sigma_i}{2}, \quad \forall i \in [m]. \] (46)

**Estimate $p_1^k.$** If $k \in K$, then we have (36) and
\[ \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 = \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 + \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 \]
\[ = \langle \Delta w^k \tau + 1, \sigma_i \Delta w^k \tau + 1 \rangle + \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2. \] (47)

The fact allows us to derive that
\[ p_1^{k+1} = \sum_i (\Delta w^k \tau + 1, \sigma_i \Delta w^k \tau + 1) \]
\[ + \sum_i (\frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 - \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2) \]
\[ = \sum_i (\Delta w^k \tau + 1, -\sigma_i \Delta w^k \tau + 1) \]
\[ \leq \sum_i \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2. \] (48)

If $k \notin K$, then $w^k \tau + 1 = w^k$, thereby also leading to
\[ p_1^k = 0 = \sum_i \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2. \]

**Estimate $p_2^k.$** We consider two cases: $i \in \Omega^{k+1}$ and $i \notin \Omega^{k+1}.$ For case $i \in \Omega^{k+1}$, it follows from (33) that
\[ \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2 - \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 \]
\[ = \langle \Delta w^k \tau + 1, \sigma_i \Delta w^k \tau + 1 \rangle - \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2. \] (49)

Let $u_i$ be defined as follows,
\[ u_i^k := L(w^k \tau + 1, w_i^k, \pi_i^k) - L(w^k \tau + 1, w_i^k, \pi_i^k) \]
\[ = \alpha_i f_i(w_i^k) - \alpha_i f_i(w_i^k) + \langle \Delta w_i^k, \pi_i^k \rangle \]
\[ + \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2 - \frac{\sigma_i}{2} \| w_i^k - w^k \tau + 1 \|^2 \]
\[ = \alpha_i f_i(w_i^k) - \alpha_i f_i(w_i^k) + \langle \Delta w_i^k, \pi_i^k \rangle \]
\[ + \langle \Delta w_i^k, \pi_i^k \rangle - \frac{\sigma_i}{2} \| \Delta w^k \tau + 1 \|^2. \] (50)

**Estimate $p_3^k.$** We consider two cases: $i \in \Omega^{k+1}$ and $i \notin \Omega^{k+1}.$ For case $i \in \Omega^{k+1}$, it follows from (33) that
\[ \frac{\sigma_i}{2} \| \Delta w_i^k \|^2 - \frac{\sigma_i}{2} \| w_i^k - w^k \|^2 \]
\[ = \langle \Delta w_i^k, \pi_i^k \rangle - \frac{\sigma_i}{2} \| \Delta w_i^k \|^2. \] (51)
Now, by using the above condition we have

\[
    u_i^{(3)} \leq \frac{\alpha_i r_i c_i}{2} \|L_i^{k+1}\|^2 + \langle \Delta w_i^{k+1}, g_i^{k+1} + \pi_i^{k+1} \rangle
\]

(45)

\[
    \leq -\frac{\alpha_i}{3} \|L_i^{k+1}\|^2 + \langle \Delta w_i^{k+1}, g_i^{k+1} + \pi_i^{k+1} \rangle
\]

(37)

\[
    \leq -\frac{\alpha_i}{3} \|L_i^{k+1}\|^2 + \|\Delta w_i^{k+1}, \varphi_i^{k+1} \|
\]

(33)

\[
    \leq -\frac{\alpha_i}{3} \|L_i^{k+1}\|^2 + \frac{3}{\alpha_i} \|\Delta w_i^{k+1}\|^2 + \frac{3}{\sigma_i} \|\Delta w_i^{k+1}\|^2
\]

(37)

\[
    \leq -\frac{\alpha_i}{3} \|L_i^{k+1}\|^2 - \frac{\Delta w_i^{k+1}}{1-\nu_i \sigma_i}.
\]

(5)

For case \( i \notin \Omega^{k+1} \), \( \pi_i^{k+1} = \pi_i^k \) and \( e_i^{k+1} = e_i^k \) from (9) indicate the above condition is also valid. Therefore, for both cases, we obtain

\[
    p_i^k = \sum_{i} u_i^k \leq \sum_{i} (\frac{\alpha_i}{3} \|L_i^{k+1}\|^2 - \frac{\Delta w_i^{k+1}}{1-\nu_i \sigma_i}).
\]

(50)

Estimate \( p_i^k \). We consider two cases: \( i \in \Omega^{k+1} \) and \( i \notin \Omega^{k+1} \). For case \( i \in \Omega^{k+1} \), it is easy to see that

\[
    v_i^k \triangleq \langle \Delta w_i^{k+1}, \pi_i^{k+1} \rangle \leq \frac{\alpha_i}{3} \|\Delta w_i^{k+1}\|^2
\]

(38)

\[
    \leq \frac{\alpha_i}{3} \|\Delta w_i^{k+1}\|^2 - \frac{24}{\alpha_i} \|\delta_i^{k+1}\| \|\Delta \epsilon_i^{k+1}\|
\]

(45)

\[
    \leq \frac{\alpha_i}{3} \|\Delta w_i^{k+1}\|^2 - \frac{24}{\alpha_i} \|\delta_i^{k+1}\| \|\Delta \epsilon_i^{k+1}\|
\]

(51)

For case \( i \notin \Omega^{k+1} \), the last inequality in the above condition is also valid due to \( \pi_i^{k+1} = \pi_i^k \) and \( e_i^{k+1} = e_i^k \) from (9). Therefore, for both cases we obtain

\[
    p_i^k = \sum_{i} u_i^k \leq \sum_{i} (\frac{\alpha_i}{3} \|\Delta w_i^{k+1}\|^2 - \frac{24}{\alpha_i} \|\delta_i^{k+1}\| \|\Delta \epsilon_i^{k+1}\|
\]

(51)

Overall, combining conditions (43), (47), (48), and (50), (51), we obtain

\[
    L^{k+1} - L_k = p_i^k + p_2^k + p_3^k
\]

\[
    \leq -\sum_{i} \left( \frac{\alpha_i}{1-\nu_i \sigma_i} \right) \left( \|\Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2 \right) - \frac{24}{\alpha_i} \|\delta_i^{k+1}\| \|\Delta \epsilon_i^{k+1}\|
\]

showing the desired result.

Lemma 4.1 allows us to directly conclude the non-increasing property of sequence \( \{L_k\} \) and vanishing of several gaps \( \Delta w_i^{k+1}, \Delta w_i^{k+1}, \Delta w_i^{k+1}, \) and \( \delta_i^{k+1} \), as shown by the following results.

Lemma C.2. Suppose that Assumptions 4.1 and 4.2 hold. Every client \( i \in [m] \) chooses \( r_i \geq 3 \alpha_i \) and the server selects \( \Omega^k \) as Scheme 4.1. Then the following results hold.

a) Sequence \( \{L_k\} \) is non-increasing.

b) \( L_k \geq f(w^k) \geq f^* \) for any \( k \geq 1 \).

c) The limits of all the following terms are zero, namely,

\[
    (\epsilon_i^{k+1}, \Delta w_i^{k+1}, \Delta w_i^{k+1}, \Delta w_i^{k+1}, \pi_i^{k+1}) \to 0
\]

(52)

Proof.

a) The conclusion follows Lemma 4.1 immediately.

b) It follows from (33) where \( t = \frac{2}{\alpha_i} \) that

\[
    \langle \Delta w_i^k, \varphi_i^k \rangle \leq \frac{\alpha_i}{\alpha_i} \|\Delta w_i^k\|^2 + \frac{3}{\alpha_i} \|\Delta \varphi_i^k\|^2
\]

(53)

The gradient Lipschitz continuity of \( f_i \) implies

\[
    \alpha_i f_i(w^k) - \alpha_i f_i(w_i^k)
\]

\[
    \leq \langle \Delta w_i^k, g_i^k \rangle + \frac{\alpha_i}{\alpha_i} \|\Delta w_i^k\|^2
\]

(45)

\[
    \leq \langle \Delta w_i^k, g_i^k \rangle + \frac{\alpha_i}{\alpha_i} \|\Delta w_i^k\|^2
\]

(45)

\[
    \langle \Delta w_i^k, \pi_i^k + \varphi_i^k \rangle + \frac{3}{\alpha_i} \|\Delta w_i^k\|^2
\]

(37)

\[
    \langle \Delta w_i^k, \pi_i^k + \varphi_i^k \rangle + \frac{\alpha_i}{\alpha_i} \|\Delta w_i^k\|^2
\]

(37)

\[
    \langle \Delta w_i^k, \pi_i^k + \varphi_i^k \rangle + \frac{3}{\alpha_i} \|\Delta w_i^k\|^2
\]

(37)

\[
    \langle \Delta w_i^k, \pi_i^k + \varphi_i^k \rangle + \frac{3}{\alpha_i} \|\Delta w_i^k\|^2
\]

(37)

The above relation and \( 1 > \nu_i \geq 1/2 \) give rise to

\[
    L^{k+1} \geq \frac{1}{L} \sum_i (L(w_i^k, w_i^k) + \frac{2\alpha_i}{\sigma_i})
\]

(12)

\[
    \sum_i (\alpha_i f_i(w_i^k) + \langle \Delta w_i^k, \pi_i^k + \varphi_i^k \rangle + \frac{3}{\alpha_i} \|\Delta w_i^k\|^2)
\]

\[
    \sum_i \alpha_i f_i(w_i^k) = f(w^k) \geq f^* \geq -\infty.
\]

c) Using Lemma 4.1 and \( L \rightarrow -\infty \) enables to show that

\[
    \sum_{k=0}^{\infty} \sum_i \|\Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2
\]

\[
    \leq \sum_{k=0}^{\infty} \frac{1}{L} \left( \|\Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2 \right)
\]

\[
    \leq 0
\]

(54)

where the last relationship is due to \( \Delta \pi_i^k \to 0 \) and \( \Delta w_i^{k+1} \to 0 \). The whole proof is finished.

\[\square]\]

C.3 Proof of Theorem 4.1

The sketch of proving results in Theorem 4.1 is as follows: The boundedness of \( \{w_i^{k+1}\} \) can be ensured by Lemma C.2 b) and the coerciveness of \( f \). Since \( \{L_k\} \) is non-increasing and bounded from below. Therefore, whole sequence \( \{L_k\} \) converges, which by (52) can show (23). Finally, to show \( \nabla F(W^{k+1}) \to 0 \) as \( k \to \infty \), we only need to show \( \nabla F(W^{k+1}) \to 0 \) as \( \ell (k) \to \infty \) due to (52), which can be guaranteed by conditions (36) and (37).

Proof. a) By Lemma C.2 that \( L_k \geq f(w^{k+1}) \) and \( f \) being coercive, we can claim the boundedness of sequence \( \{w_i^{k+1}\} \) immediately. This calls forth the boundedness of sequence \( \{w_i^{k+1}\} \) as \( \Delta w_i^{k+1} \to 0 \) from (52), thereby delivering

\[
    \pi_i^{k+1} \leq \|\pi_i^{k+1} - g_i^{k+1}\|
\]

\[
    \leq \|\pi_i^{k+1} - g_i^{k+1}\| + \|g_i^{k+1} - g_i^0\| + \|g_i^0\|
\]

(37), l)

\[
    \sqrt{\epsilon_i^{k+1} + \alpha_i r_i \|w_i^{k+1} - w_i^0\| + \|g^0\|} < +\infty.
\]
This shows the boundedness of sequence \( \{ \pi_i^{k+1} \} \). Overall, sequence \( \{ (w_i^{k+1}, W^{k+1}, \Pi^{k+1}) \} \) is bounded.

b) It follows from Lemma C.2 that \( \hat{L}_k \) is non-increasing and bounded from below. Therefore, whole sequence \( \{ \hat{L}_k \} \) converges and \( \hat{L}^{k+1} \to L^{k+1} \) due to \( \epsilon_i^{k+1} \to 0 \) in (52). Again by (52) and the boundedness of sequence \( \{ \pi_i^{k+1} \} \), we can prove that

\[
\hat{L}^{k+1} - F(W^{k+1}) \leq \sum_i (\Delta w_i^{k+1}, \pi_i^{k+1}) + \frac{\sigma_i}{2} \| \Delta w_i^{k+1} \|^2 \to 0.
\]

(55)

It follows from Mean Value Theory that

\[
f_i(w_i^{k+1}) = f_i(w_i^{\tau_{k+1}}) + \langle \Delta w_i^{k+1}, \nabla f_i(w_i) \rangle,
\]

where \( w_i := (1-t)w_i^{\tau_{k+1}} + tw_i^{k+1} \) for some \( t \in (0,1) \). Since \( \{ w_i^{\tau_{k+1}}, w_i^{k+1} \} \) is bounded, so is \( w_i \). This calls forth \( f_i(w_i^{k+1}) - f_i(w_i^{\tau_{k+1}}) \to 0 \) due to \( \Delta w_i^{k+1} \to 0 \). Using this condition enables us to obtain

\[
\hat{L}^{k+1} - f(w_i^{\tau_{k+1}}) = \sum_i (\alpha_i f_i(w_i^{k+1}) - \alpha_i f_i(w_i^{\tau_{k+1}}) \leq \sum_i (\Delta w_i^{k+1}, \pi_i^{k+1}) + \frac{\sigma_i}{2} \| \Delta w_i^{k+1} \|^2 \to 0.
\]

(56)

Since \( \ell \in K \), we have

\[
\sum_i \pi_i^{\ell+1} \leq \sum_i (\pi_i^{k+1} - \pi_i^{k+1}) \to 0.
\]

(57)

We note that sequence \( \{ \epsilon_i^{k+1} \} \) is non-increasing and thus obtain \( \epsilon_i^{k+1} \leq \epsilon_i^{k+1} \) from (56), whereby rendering that

\[
\| \pi_i^{k+1} - \pi_i^{k+1} \|^2 \leq 3\epsilon_i^{k+1} + 3\epsilon_i^{k+1} + 3\epsilon_i^{k+1} \| w_i^{k+1} - w_i^{k+1} \|^2 \leq 6\epsilon_i^{k+1} + 6\epsilon_i^{k+1} \| \Delta w_i^{k+1} \|^2 + \| \Delta w_i^{k+1} \|^2 \leq 0.
\]

(58)

Using the above two conditions immediately derives that

\[
\sum_i \pi_i^{k+1} \to 0.
\]

(59)

Taking the limit on both sides of (57) gives us

\[
\nabla F(W^{k+1}) = \sum_i g_i^{k+1} \leq \sum_i (g_i^{k+1} + \pi_i^{k+1}) \leq \sum_i (g_i^{k+1} + \pi_i^{k+1}) \to 0,
\]

(60)

which together with \( \Delta w_i^{k+1} \to 0 \) and the gradient Lipschitz continuity yields that \( \nabla f_i(w_i^{k+1}) = \sum_i \alpha_i \nabla f_i(w_i^{k+1}) \to 0 \).

This completes the whole proof.

C.4 Proof of Theorem 4.2

Proof. a) Let \( (w, W, \Pi) \) be any accumulating point of the sequence, it follows from (37) and (38) that

\[
0 = \alpha_i \nabla f_i(w_i^\infty) + \pi_i^\infty.
\]

By \( (w_i^{k+1} - w_i^{k+1}) \to 0 \) and (57), we have

\[
0 = w_i^\infty - w_i^\infty, 0 = \sum_i \pi_i^\infty.
\]

Therefore, recalling (19), \( (w, W, \Pi) \) is a stationary point of (10) and \( w^\infty \) is a stationary point of (2).

b) It follows from [54, Lemma 4.10], \( \Delta w_i^{k+1} \to 0 \) and \( w^\infty \) being isolated that the whole sequence, \( \{ w_i^{k+1} \} \) converges to \( w^\infty \), which by \( \Delta w_i^{k+1} \to 0 \) implies that \( \{ W_i^{k+1} \} \) also converges to \( W^\infty \). Finally, this together with (37) and (38) results in the convergence of \( \{ \Pi_i^{k+1} \} \).

C.5 Proof of Corollary 4.1

Proof. a) The convexity of \( f \) and the optimality of \( w^* \) yields

\[
f_i(w_i^{k+1}) \geq f_i(w_i^*) \geq f_i(w_i^{k+1}) + \langle \nabla f_i(w_i^{k+1}), w_i^* - w_i^{k+1} \rangle.
\]

(59)

Theorem 4.1 ii) states that

\[
\lim_{k \to \infty} \nabla F(W_k) = \lim_{k \to \infty} \nabla F(w^k) = 0.
\]

Using this and the boundedness of \( \{ w_i^{k+1} \} \) from Theorem 4.2, we take the limit of both sides of (59) to derive that \( f_i(w_i^{k+1}) \to f_i(w^*) \), which recalling Theorem 4.1 i) yields (25).

b) The conclusion follows from Theorem 4.2 ii) and the fact that the stationary points are equivalent to optimal solutions if \( f \) is convex.

c) The strong convexity of \( f \) means that there is a positive constant \( \nu \) such that

\[
f_i(w_i^{k+1}) - f_i(w^*) \geq g_i^{k+1}, 2\| w_i^{k+1} - w^* \|^2 
\]

where the equality is due to (20). Taking limit of both sides of the above inequality shows \( w_i^{k+1} \to w^* \) since \( f_i(w_i^{k+1}) \to f_i(w^*) \). This together with (52) yields \( w_i^{k+1} \to w^* \). Finally, \( \pi_i \to \pi_i^* \) because of

\[
\| \pi_i^* - \pi_i^* \|^2 \leq 2\alpha_i x_i^* - w_i^* \|^2 + \| \pi_i^* \|^2 \to 0,
\]

displaying the desired result.

C.6 Proof of Theorem 4.3

The proof focuses on \( k \in K \) and aims at estimating term \( \| \nabla f(w_i^{k+1}) \|^2 \) by decomposing it as

\[
\| \nabla f(w_i^{k+1}) \|^2 \leq 3\| \nabla f(w_i^{k+1}) \|^2 + \| \sum_i g_i^{k+1} \|^2 + 3\| \sum_i g_i^{k+1} \|^2 \leq 3m \sum_i \| \nabla f_i(w_i^{k+1}) \|^2 + \| \Delta w_i^{k+1} \|^2 + 3\| \sum_i g_i^{k+1} \|^2 \leq 3m \sum_i \| \Delta w_i^{k+1} \|^2 + \| \Delta w_i^{k+1} \|^2 + 3\| \sum_i g_i^{k+1} \|^2,
\]

(60)

where the last two inequalities used the gradient Lipschitz continuity and (45). Then we estimate each term on the right-hand side. The detailed proof is given as follows.
Proof. Estimate \(\sum_i \|\sigma_i \Delta w_i^{k+1}\|^2\). Recalling Lemma 4.1,
\[
\sum_i \sigma_i \|\Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2 \leq \tilde{L}^k - \tilde{L}^{k+1},
\]
which by letting \(\sigma := \max_{i \in [m]} \sigma_i\) results in
\[
\max \{\sum_i \|\sigma_i \Delta w_i^{k+1}\|^2, \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2\} \leq 10\sigma \sum_i \frac{\|\sigma_i \Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2}{\sum_i \frac{\|\sigma_i \Delta w_i^{k+1}\|^2 + \|\Delta w_i^{k+1}\|^2)} \leq 10\sigma (\tilde{L}^k - \tilde{L}^{k+1}).
\]
Estimate \(\sum_i \|\sigma_i \Delta w_i^{k+1}\|^2\). For any \(i \in \Omega^{\tau_{k+1}}\), by the third inequality in (42), we have
\[
\|\Delta \pi^{k+1}\|^2 \leq \frac{6\sigma_i^2}{5} \|\Delta w_i^{k+1}\|^2 + 12(\epsilon_i^k + \epsilon_{i+1}^k + \epsilon_{i+2}^k)
\]
\[
\leq \frac{2\sigma_i^2}{5} \|\Delta w_i^{k+1}\|^2 + 24(\epsilon_i^k + \epsilon_{i+1}^k + \epsilon_{i+2}^k).
\]
Then it follows
\[
\|\sigma_i \Delta w_i^{k+1}\|^2 \leq \frac{2\sigma_i^2}{5} \|\Delta w_i^{k+1}\|^2 + 24\epsilon_i^k.
\]
For \(i \notin \Omega^{\tau_{k+1}}\), let \(k_i\) be defined similarly to that in the proof of Lemma C.1 a) at step \(k\). Then \(i \in \Omega^{\tau_{k+1}}\) and (62) holds for \(k_i\), which allows us to obtain
\[
\|\sigma_i \Delta w_i^{k+1}\|^2 \leq \frac{2\sigma_i^2}{5} \|\Delta w_i^{k+1}\|^2 + 24\epsilon_i^k
\]
\[
\leq \frac{2\sigma_i^2}{5} \|\Delta w_i^{k+1}\|^2 + 24\epsilon_i^k.
\]
Based on (21), we have \(\tau_{k+1} - \tau_{k+1} \leq s_0\) and thus \(k \geq k_i \geq -s_0 \leq s_0\), which by the non-increasing properties of \(\{\tilde{L}^k\}\) and \(\{\epsilon_i^k\}\) imply that
\[
\tilde{L}^k \leq \tilde{L}^{k_i} \leq \tilde{L}^{k-s_0\epsilon_i}, \quad \epsilon_i^k \equiv \epsilon_i^{k_i}, i \notin \Omega^{\tau_{k+1}}.
\]
By (63) and (64) we can obtain
\[
\sum_i \|\sigma_i \Delta w_i^{k+1}\|^2 \leq \sum_i \epsilon_i \leq \tilde{L}^{k-s_0\epsilon_i},
\]
\[
\sum_i \epsilon_i \leq \tilde{L}^{k-s_0\epsilon_i}.\]
Estimate \(\sum_i \|g_i^k\|^2\). For any \(k \in \mathcal{K}\), we have
\[
\sum_i \|g_i^k\|^2 \leq \sum_i (\varphi_i^k - \pi_i^k)
\]
\[
\leq \sum_i (\varphi_i^k + \sigma_i (w_i^k - \tau_{k+1})) \leq \sum_i (\varphi_i^k + \sigma_i (\Delta w_i^{k+1} - \Delta w_i^{k+1})),
\]
which by \(\|\varphi_i^k\|^2 \leq \epsilon_i^k\) from (37) leads to
\[
\sum_i \|g_i^k\|^2 \leq 3m \sum_i \epsilon_i^k
\]
\[
+ 3m \sum_i \|\sigma_i^2 \Delta w_i^{k+1}\|^2 \leq 3m \sum_i \epsilon_i^k
\]
(67)
\[
+ 3m \sum_i \epsilon_i^k \|\Delta w_i^{k+1}\|^2.
\]
Now, combining (60) and (67) yields that
\[
\|\nabla f(w^{\tau_{k+1}})\|^2 \leq \sum_i \frac{10m\sigma_i^2}{9} \|\Delta w_i^{k+1}\|^2 + 12m \sum_i \epsilon_i^k
\]
\[
\leq \frac{4m}{9} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2 + \frac{8m}{9} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2 + 169m \sum_i \epsilon_i^k
\]
\[
\leq \frac{4m}{9} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2 + \frac{8m}{9} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2 + 169m \sum_i \epsilon_i^k
\]
\[
\leq \frac{40m \sigma_i^2}{9} (\tilde{L}^k - \tilde{L}^{k+1}) + \frac{80m \sigma_i^2}{9} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2
\]
\[
+ 169m \sum_i \epsilon_i^k + \frac{200m \sigma_i}{3} \sum_i \|\sigma_i \Delta w_i^{k+1}\|^2
\]
\[
\leq \frac{40m \sigma_i^2}{9} (\tilde{L}^k - \tilde{L}^{k+1}) + 169m \sum_i \epsilon_i^k
\]
\[
+ \frac{200m \sigma_i}{3} (\tilde{L}^k - \tilde{L}^{k+1}) + 169m \sum_i \epsilon_i^k
\]
\[
\leq \frac{40m \sigma_i^2}{9} (\tilde{L}^k - \tilde{L}^{k+1}) + 169m \sum_i \epsilon_i^k
\]
\[
\leq \frac{40m \sigma_i^2}{9} (\tilde{L}^k - \tilde{L}^{k+1}) + 169m \sum_i \epsilon_i^k
\]
The non-increasing property of \(\{\tilde{L}^k\}\) suffices to
\[
\tilde{L}^{(p+1)k_0} - \tilde{L}^{p_{k_0}+1} \leq 0, \quad \forall p \geq 0.
\]
We note that \(\{\epsilon_i^k\}\) is non-increasing. Moreover, under Scheme 4.1, for every \(s_0k_0\) steps, each client \(i \in [m]\) is chosen at least once to update their parameters by (3)-(8), which means that \(\epsilon_i^{(a+1)k_0} \leq \nu_i \epsilon_i^0 k_0\) for all \(i \in [m]\) and \(a = 1, 2, \cdots\), thereby leading to
\[
\sum_{p=s_0}^{t} \epsilon_i^{p_{k_0}} \leq \sum_{p=s_0}^{t} \frac{\epsilon_i^{p_{k_0}}}{t}\]
\[
\leq \sum_{p=s_0}^{t} \frac{\epsilon_i^{p_{k_0}}}{t(s_0k_0)},
\]
\[
\leq \sum_{p=s_0}^{t} \frac{\epsilon_i^{p_{k_0}}}{s_0k_0 - 1/\nu_i},
\]
\[
\leq \sum_{p=s_0}^{t} \frac{\epsilon_i^{p_{k_0}}}{s_0k_0 - 1/\nu_i}.
\]
Finally, using (69) and (70) enables us to derive
\[
(t - s_0 + 1) \min_{p=s_0, s_0+1, \cdots, t} \|\nabla f(w^{\tau_{k+1}})\|^2
\]
\[
\leq \sum_{p=s_0}^{t} \|\nabla f(w^{\tau_{k+1}})\|^2
\]
\[
\leq \frac{940m \sigma_i^2}{9} \sum_{p=s_0}^{t} \|\nabla f(w^{\tau_{k+1}})\|^2
\]
\[
\leq 40m \sigma_i^2 (\tilde{L}^{(t-s_0)k_0+1} + \tilde{L}^{(t-s_0-1)k_0+1} + \cdots + \tilde{L}^{k_0+1})
\]
where the last inequality is due to \(\tilde{L}^0 \geq \tilde{L}^k \geq f^*\) for any \(k \geq 1\) from Lemma C.2 b). Now by letting \(k = t k_0\), we have
\[
\min_{s=1, 3, \cdots, k} \|\nabla f(w^{\tau_{k+1}})\|^2
\]
\[
\leq \min_{p=s_0, s_0+1, \cdots, t} \|\nabla f(w^{\tau_{k+1}})\|^2
\]
\[
\leq \frac{c}{t-s_0+1} \leq \frac{c_0}{k-s_0},
\]
showing the desired result. □