fractal and fractional

MDPI

Article

A Flexible Framework for Decentralized Composite Optimization with Compressed Communication

Zhongyi Chang [1], Zhen Zhang [1], Shaofu Yang [1,*] and Jinde Cao [2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; zychang@seu.edu.cn (Z.C.); zhang_zhen@seu.edu.cn (Z.Z.)
[2] School of Mathematics, Southeast University, Nanjing 211189, China; jdcao@seu.edu.cn
* Correspondence: sfyang@seu.edu.cn

Abstract: This paper addresses the decentralized composite optimization problem, where a network of agents cooperatively minimize the sum of their local objective functions with non-differentiable terms. We propose a novel communication-efficient decentralized ADMM framework, termed as CE-DADMM, by combining the ADMM framework with the three-point compressed (3PC) communication mechanism. This framework not only covers existing mainstream communication-efficient algorithms but also introduces a series of new algorithms. One of the key features of the CE-DADMM framework is its flexibility, allowing it to adapt to different communication and computation needs, balancing communication efficiency and computational overhead. Notably, when employing quasi-Newton updates, CE-DADMM becomes the first communication-efficient second-order algorithm based on compression that can efficiently handle composite optimization problems. Theoretical analysis shows that, even in the presence of compression errors, the proposed algorithm maintains exact linear convergence when the local objective functions are strongly convex. Finally, numerical experiments demonstrate the algorithm's impressive communication efficiency.

Keywords: decentralized composite optimization; ADMM; quasi-Newton; communication-efficient mechanism

Citation: Chang, Z.; Zhang, Z.; Yang, S.; Cao, J. A Flexible Framework for Decentralized Composite Optimization with Compressed Communication. *Fractal Fract.* **2024**, *8*, 721. https://doi.org/10.3390/fractalfract8120721

Academic Editor: Carlo Cattani

Received: 9 October 2024
Revised: 25 November 2024
Accepted: 4 December 2024
Published: 5 December 2024

1. Introduction

The recent increase in the number of mobile devices with enhanced computing and communication capabilities has led to significant development of multiagent systems [1,2]. Within these systems, many applications, including smart grid management [3,4], wireless communications [5], multi-robot coordination [6,7], large-scale machine learning [8], etc., can be cast to decentralized optimization problems, in which a network of nodes cooperatively solve a finite-sum optimization problem using local information.

A vast of decentralized optimization algorithms have been proposed, since the pioneer work DGD [9], in which each node performs gradient descent and simultaneously communicates decision vector with its neighbors for consensus. As DGD requires diminishing stepsize, which might slower the convergence rate, gradient tracking (GT) based algorithms using constant stepsizes are then developed [10,11] and have been extensively investigated under various scenarios [12–15], to name a few. However, GT-based methods require to transmit both the decision vector and an additional gradient estimation vector, which increase the communication cost. In parallel, another type of decentralized algorithms based on alternating direction method of multipliers (ADMM) are proposed and analyzed [16,17]. Compared with GT-based algorithms, ADMM-type algorithms can achieve the same convergence rate but require the transmission of only decision vector, which can be more communication-efficient. Following this line, some decentralized optimization algorithms are proposed for accelerating the convergence rate by introducing second-order information [18–21]. More recently, Ref. [22] proposed a family of decentralized curvature aided primal dual algorithms, which can include gradient, Newton, and BFGS type of updates.

In the decentralized algorithms, it is of great significance to improve communication-efficiency. The methods can be classified into two types. One method is to adopt compressed communication scheme, using quantization [23] or sparsification [24,25] techniques to reduce communication overhead per transmission. Recently, the compressed communication scheme has been combined with DGD [26,27], GT-based algorithms [28,29], and ADMM-type algorithms [30,31]. Another method is to employ intermittent communication scheme which aims to reduce the communication frequency. Such type of methods includes event-triggered communication [32–34], lazy aggregation scheme (LAG) [35], etc. Besides, there are also some works combining the both methods [36–38]. In particular, ref. [39] combined event-triggered and compressed communication scheme with an ADMM-type algorithm and proposed a communication-efficient decentralized second-order optimization algorithm, which improve both computation and communication efficiency. It is worthy noting that the *information distortion* arisen by compressed communication scheme may have a negative effect on the convergence performance of the decentralized optimization algorithms. To overcome this shortage, Ref. [40] developed an error-feedback communication scheme (EF21) to avoid the negative effect of *information distortion*. Recently, a more general efficient communication scheme termed as three point compressor (3PC) is proposed in [41], which provides a unified method including EF21 and LAG as special cases. However, 3PC scheme is only investigated in distributed gradient descent algorithms under the parameter-server framework.

Despite of the progress, the development of communication-efficient decentralized optimization algorithms over the general networks is still lack an in-deep analysis, especially for the objective with non-differentiable part, i.e., decentralized composite optimization problems. Note that such problems have widely applications in the field of machine learning due to the existence of the non-differentiable regularization terms. Currently, some decentralized composite optimization algorithms have been proposed [22,42–44], but without employing efficient-communication schemes. Moreover, to the best of our knowledge, no work has been reported on communication-efficient decentralized composite optimization using second-order information. To fill this gap, in this paper, we incorporate the general efficient communication scheme 3PC [41] into the ADMM-based decentralized optimization framework [22], which result in a family of communication-efficient decentralized composite optimization algorithms with theoretical guarantees. It is worthy noting that such incorporation is not trivial as we need to overcome the negative effect arisen by the propagation of communication error over networks. *The main contribution of this work can be summarized in the following two aspects:*

- First, we propose a flexible framework termed as CE-DADMM for communication-efficient decentralized composite optimization problems. The framework not only encompasses some existing algorithms, such as COLA [32] and CC-DQM [39], but also introduces several new algorithms. Specifically, by incorporating quasi-Newton updates into CE-DADMM, we derive CE-DADMM-BFGS, the first communication-efficient decentralized second-order algorithm for composite optimization. Compared with CC-DQM, it avoids computing the Hessian matrix and its inversion, significantly reducing the computational cost. Compared with DRUID [22], CE-DADMM can reduce the communication cost due to the efficient communication scheme.
- Second, we theoretically prove that CE-DADMM can achieve exact linear convergence under the assumption of strong convexity by carefully analyzing the mixing error arisen by the efficient communication scheme and the disagreement of decision vectors. The dependency of the convergence rate on the parameters of the compression mechanism is also established. Additionally, sufficient numerical experiments are presented to substantiate the superior performance of our algorithms in terms of the communication efficiency.

Notation. If not specified, $\|\mathbf{x}\|$ and $\|\mathbf{A}\|$ represent the Euclidean norm and the spectral norm, respectively. For a positive definite matrix $\mathbf{P} \succ 0$, let $\|\mathbf{x}\|_{\mathbf{P}} := \sqrt{\mathbf{x}^\top \mathbf{P}\mathbf{x}}$. Use $[n]$ to denote the set $\{1, \cdots, n\}$. The proximal mapping for a function $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is defined by

$\mathbf{prox}_{g/\mu}(\mathbf{v}) = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\{g(\boldsymbol{\theta}) + \frac{\mu}{2}\|\boldsymbol{\theta}-\mathbf{v}\|^2\}$. Let $\mathbf{I}_d$ represent the d-dimensional identity matrix, and $\mathbf{A}\otimes\mathbf{B}$ represent the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$.

2. Problem Setting

In this paper, we study the decentralized composite optimization problem on an undirected connected network with n agents (or, *nodes*), which takes the form

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left\{\sum_{i=1}^{n} f_i(\mathbf{x}) + g(\mathbf{x})\right\}, \tag{1}$$

where $\mathbf{x}$ refers to the decision vector and $f_i(\cdot):\mathbb{R}^d\to\mathbb{R}$ is a convex and smooth function accessible only by node i and $g(\cdot):\mathbb{R}^d\to\mathbb{R}$ is a convex (possibly non-smooth) regularizer.

Next, we equivalently reformulate problem (1) into a compact form in terms of the whole network following the same idea in [22]. Denote the communication graph as $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$, where $\mathcal{V}:=[n]$ is the set of agents and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the set of edges containing the pair (i,j) if and only if agent i can communicate with agent j. There is no self-loops in $\mathcal{G}$, i.e., $(i,i)\notin\mathcal{E}$ for any $i\in[n]$. Note that the edges in $\mathcal{E}$ are enumerated in arbitrary order, with $e_k:=(i,j)\in\mathcal{E}$ denoting the k-th edge, where $k\in[m]$ and $m:=|\mathcal{E}|$ is the number of edges. The neighbor set of agent i is $\mathcal{N}_i:=\{j\in\mathcal{V}:(i,j)\in\mathcal{E}\}$. Let $\mathbf{x}_i\in\mathbb{R}^d$ and $\mathbf{z}_k\in\mathbb{R}^d$ be the local decision vectors corresponding the ith-node kth-edge, respectively. We assume that $\mathcal{G}$ is connected. Then, problem (1) is equivalent to the following constrained form:

$$\begin{aligned}\min_{\{\mathbf{x}_i\},\boldsymbol{\theta},\{\mathbf{z}_{ij}\}}\quad & \left\{\sum_{i=1}^{n} f_i(\mathbf{x}_i) + g(\boldsymbol{\theta})\right\},\\ \text{s.t.}\quad & \mathbf{x}_i=\mathbf{x}_j=\mathbf{z}_k \quad \forall e_k=(i,j)\in\mathcal{E},\\ & \mathbf{x}_l=\boldsymbol{\theta}, \text{ for one arbitrary } l\in\mathcal{V},\end{aligned} \tag{2}$$

where $\boldsymbol{\theta}\in\mathbb{R}^d$ is an auxiliary variable for decoupling the smooth and non-smooth functions. Denote the optimal solution of problem (1) and (2) as $\mathbf{x}^\star$ and $\{\mathbf{x}_i^\star,\mathbf{z}_k^\star,\boldsymbol{\theta}^\star\}$, respectively. It is straightforward to verify that $\mathbf{x}^\star=\mathbf{x}_i^\star=\mathbf{z}_k^\star=\boldsymbol{\theta}^\star$ for all $i\in[n]$ and $k\in[m]$.

In what follows, define $\tilde{\mathbf{x}}:=[\mathbf{x}_1;\mathbf{x}_2;\ldots;\mathbf{x}_n]\in\mathbb{R}^{nd}$ and $\tilde{\mathbf{z}}:=[\mathbf{z}_1;\mathbf{z}_2;\ldots;\mathbf{z}_m]\in\mathbb{R}^{md}$. In addition, define two matrices $\hat{\mathbf{A}}_s$ and $\hat{\mathbf{A}}_d\in\mathbb{R}^{m\times n}$ as follows: the k-th row of both $\hat{\mathbf{A}}_s$ and $\hat{\mathbf{A}}_d$ represents the k-th edge e_k. Specifically, the entries $[\hat{\mathbf{A}}s]_{ki}$ and $[\hat{\mathbf{A}}d]_{kj}$ are both equal to 1 if and only if the edge $e_k=(i,j)$; otherwise, they are 0. We also define $\mathbf{S}:=(\mathbf{s}_l\otimes\mathbf{I}_d)\in\mathbb{R}^{nd\times d}$, where $\mathbf{s}_l\in\mathbb{R}^n$ is a vector with a 1 at its l-th position and 0 elsewhere. Clearly, the matrix $\mathbf{S}^\top$ extracts the component of $\tilde{\mathbf{x}}$ that corresponds to agent l, meaning that $\mathbf{S}^\top\tilde{\mathbf{x}}=\mathbf{x}_l$. Let $F(\tilde{\mathbf{x}}):=\sum_{i=1}^n f_i(\mathbf{x}_i)$. Then, problem (2) can be written as

$$\begin{aligned}\min_{\tilde{\mathbf{x}},\boldsymbol{\theta},\tilde{\mathbf{z}}}\quad & F(\tilde{\mathbf{x}})+g(\boldsymbol{\theta})\\ \text{s.t.}\quad & \mathbf{A}\tilde{\mathbf{x}}=\mathbf{B}\tilde{\mathbf{z}},\ \ \mathbf{S}^\top\tilde{\mathbf{x}}=\boldsymbol{\theta}, \quad \text{where } \mathbf{A}=\begin{bmatrix}\hat{\mathbf{A}}_s\otimes\mathbf{I}_d\\ \hat{\mathbf{A}}_d\otimes\mathbf{I}_d\end{bmatrix} \text{ and } \mathbf{B}=\begin{bmatrix}\mathbf{I}_{md}\\ \mathbf{I}_{md}\end{bmatrix}.\end{aligned} \tag{3}$$

Note that problem (3) is written from the network level, which will be the basis for designing our algorithm.

3. Algorithm Formulation

In this section, we first introduce the basic iterations of our algorithm based on ADMM method. Then, by combining compressed communication techniques with the ADMM-based algorithm, we will devise our algorithm and discuss its relationship with existing algorithms.

3.1. Background: ADMM-Based Algorithm

ADMM is a powerful tool to solve an optimization problem with several blocks of variables. To apply ADMM for solving problem (3), define its augmented Lagrangian as:

$$\begin{aligned}\mathcal{L}(\tilde{\mathbf{x}}, \boldsymbol{\theta}, \tilde{\mathbf{z}}; \boldsymbol{\nu}, \boldsymbol{\lambda}) := F(\tilde{\mathbf{x}}) + g(\boldsymbol{\theta}) + \boldsymbol{\nu}^\top(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{B}\tilde{\mathbf{z}}) + \boldsymbol{\lambda}^\top(\mathbf{S}^\top\tilde{\mathbf{x}} - \boldsymbol{\theta}) \\ + \frac{\mu_z}{2}\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{B}\tilde{\mathbf{z}}\|^2 + \frac{\mu_\theta}{2}\left\|\mathbf{S}^\top\tilde{\mathbf{x}} - \boldsymbol{\theta}\right\|^2,\end{aligned} \tag{4}$$

where μ_z and μ_θ are positive constants, $\boldsymbol{\nu} \in \mathbb{R}^{2md}$ and $\boldsymbol{\lambda} \in \mathbb{R}^d$ are Lagrange multipliers. Then, the kth-iteration in ADMM is written as

$$\tilde{\mathbf{x}}_{t+1} = \arg\min_{\tilde{\mathbf{x}}} \mathcal{L}(\tilde{\mathbf{x}}, \boldsymbol{\theta}_t, \tilde{\mathbf{z}}_t; \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t) \tag{5a}$$

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\tilde{\mathbf{x}}_{t+1}, \boldsymbol{\theta}, \tilde{\mathbf{z}}_t; \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t) \tag{5b}$$

$$\tilde{\mathbf{z}}_{t+1} = \arg\min_{\tilde{\mathbf{z}}} \mathcal{L}(\tilde{\mathbf{x}}_{t+1}, \boldsymbol{\theta}_{t+1}, \tilde{\mathbf{z}}; \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t) \tag{5c}$$

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \mu_z(\mathbf{A}\tilde{\mathbf{x}}_{t+1} - \mathbf{B}\tilde{\mathbf{z}}_{t+1}) \tag{5d}$$

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu_\theta(\mathbf{S}^\top\tilde{\mathbf{x}}_{t+1} - \boldsymbol{\theta}_{t+1}). \tag{5e}$$

Define $\hat{\mathbf{E}}_s = \hat{\mathbf{A}}_s - \hat{\mathbf{A}}_d$, $\hat{\mathbf{E}}_u = \hat{\mathbf{A}}_s + \hat{\mathbf{A}}_d$, $\hat{\mathbf{L}}_s = \hat{\mathbf{E}}_s^\top\hat{\mathbf{E}}_s$, $\hat{\mathbf{L}}_u = \hat{\mathbf{E}}_u^\top\hat{\mathbf{E}}_u$, $\hat{\mathbf{D}} = \frac{1}{2}(\hat{\mathbf{L}}_s + \hat{\mathbf{L}}_u)$. In addition, define $\mathbf{E}_s = \hat{\mathbf{E}}_s \otimes \mathbf{I}_d$, and similarly for $\mathbf{E}_u, \mathbf{L}_s, \mathbf{L}_u$, and $\mathbf{D}$. Similar as [22], if we initialize the multiplier $\boldsymbol{\nu}_t := [\boldsymbol{\alpha}_t; \boldsymbol{\beta}_t] \in \mathbb{R}^{2md}$ with $\boldsymbol{\alpha}_0 = -\boldsymbol{\beta}_0$, $\mathbf{E}_u\tilde{\mathbf{x}}_0 = 2\tilde{\mathbf{z}}_0$, there is $\mathbf{E}_u\tilde{\mathbf{x}}_t = 2\tilde{\mathbf{z}}_t$. Let $\boldsymbol{\phi}_t = \mathbf{E}_s^\top\boldsymbol{\alpha}_t$, and approximate the augmented Lagrangian $\mathcal{L}(\cdot)$ in (5a) by employing a second-order expansion at $\mathbf{x}_t$ as

$$\hat{\mathcal{L}}(\tilde{\mathbf{x}}, \boldsymbol{\theta}_t, \tilde{\mathbf{z}}_t; \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t) = \mathcal{L}_t(\tilde{\mathbf{x}}_t) + (\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_t)\nabla_{\tilde{\mathbf{x}}}\mathcal{L}_t(\tilde{\mathbf{x}}_t) + \frac{1}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_t\|_{\mathbf{H}_t}^2,$$

where $\mathcal{L}_t(\tilde{\mathbf{x}}_t)$ is short for $\mathcal{L}(\tilde{\mathbf{x}}_t, \boldsymbol{\theta}_t, \tilde{\mathbf{z}}_t; \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t)$, and $\mathbf{H}_t$ is an invertible matrix representing the approximated Hessian of the augmented Lagrangian, then the iteration (5) can be simplified as

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \mathbf{H}_t^{-1}\left(\nabla F(\tilde{\mathbf{x}}_t) + \boldsymbol{\phi}_t + \mathbf{S}\boldsymbol{\lambda}_t + \frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{x}}_t + \mu_\theta\mathbf{S}\left(\mathbf{S}^\top\tilde{\mathbf{x}}_t - \boldsymbol{\theta}_t\right)\right) \tag{6a}$$

$$\boldsymbol{\theta}_{t+1} = \mathbf{prox}_{g/\mu_\theta}\left(\mathbf{S}^\top\tilde{\mathbf{x}}_{t+1} + \mu_\theta^{-1}\boldsymbol{\lambda}_t\right) \tag{6b}$$

$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{x}}_{t+1} \tag{6c}$$

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu_\theta(\mathbf{S}^\top\tilde{\mathbf{x}}_{t+1} - \boldsymbol{\theta}_{t+1}). \tag{6d}$$

Compared with (5), the iteration (6) contains fewer vectors by eliminating the vector $\tilde{\mathbf{z}}$ and replacing $\boldsymbol{\nu}$ by $\boldsymbol{\phi}$ to halve the dimension of $\boldsymbol{\nu}$. Note that the iteration (6) is written in terms of the whole network. To implement (6) in a decentralized manner, we require the matrix $\mathbf{H}_t$ be block-diagonal, so that each block can be computed independently by each agent. Here, we assume that $\mathbf{H}_t = \mathrm{diag}\{\mathbf{H}_{1,t}, \mathbf{H}_{2,t}, \cdots, \mathbf{H}_{n,t}\}$. The choice of $\mathbf{H}_{i,t}$ will be discussed later. Then, agent i will perform the following iteration:

$$\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} - \mathbf{H}_{i,t}^{-1}\left(\nabla f(\mathbf{x}_{i,t}) + \boldsymbol{\phi}_{i,t} + \frac{\mu_z}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{x}_{i,t} - \mathbf{x}_{j,t}) + \delta_{il}\left(\boldsymbol{\lambda}_t + \mu_\theta\left(\mathbf{x}_{i,t} - \boldsymbol{\theta}_t\right)\right)\right) \tag{7a}$$

$$\boldsymbol{\phi}_{i,t+1} = \boldsymbol{\phi}_{i,t} + \frac{\mu_z}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{x}_{i,t+1} - \mathbf{x}_{j,t+1}) \tag{7b}$$

$$\boldsymbol{\theta}_{t+1} = \delta_{il}\,\mathbf{prox}_{g/\mu_\theta}\left(\mathbf{x}_{i,t+1} + \mu_\theta^{-1}\boldsymbol{\lambda}_t\right) \tag{7c}$$

$$\boldsymbol{\lambda}_{t+1} = \delta_{il}\left(\boldsymbol{\lambda}_t + \mu_\theta(\mathbf{x}_{i,t+1} - \boldsymbol{\theta}_{t+1})\right), \tag{7d}$$

where $\delta_{il} = 1$ if $i = l$, otherwise $\delta_{il} = 0$.

3.2. Communication-Efficient Decentralized ADMM

Recalling the iteration (7a) and (7b), it can be seen that agent i will communicate the information on $\mathbf{x}$ to its neighbors at each iteration. However, such communication might not be realized for scenarios with limited communication resources. To reduce the communication overhead, we introduce the idea of compressed communication scheme into iteration (7), which results in our algorithm, termed as communication-efficient decentralized ADMM for composite optimization (CE-DADMM).

To compress the communication, we first give a definition of compressor.

Definition 1 (Compressor)**.** *A randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is called a compressor if there exists a constant $\delta \in [0, 1)$ such that $\mathbb{E}\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|^2 \leq (1-\delta)\|\mathbf{x}\|^2$ holds for any $\mathbf{x} \in \mathbb{R}^d$.*

In Definition 1, the compressor is characterized using the relationship between the compression error and the original state. Clearly, δ refers to the compression ratio. We can also call $\mathcal{C}$ a δ-compressor. In our algorithm, we do not apply $\mathcal{C}(\cdot)$ directly to the transmitted $\mathbf{x}$, as it will lead to a non-dismissing compression error. Instead, we will adopt a general compressor termed as *Three Point Compressor* (3PC) [41], whose definition is given below.

Definition 2 (Three Point Compressor, see [41])**.** *A randomized map $\mathcal{C}_{\mathbf{h},\mathbf{y}} : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is called a three point compressor (3PC) if there exist constants $0 < A \leq 1$ and $B \geq 0$ such that*

$$\mathbb{E}\left[\left\|\mathcal{C}_{\mathbf{h},\mathbf{y}}(\mathbf{x}) - \mathbf{x}\right\|^2\right] \leq (1-A)\|\mathbf{h}-\mathbf{y}\|^2 + B\|\mathbf{x}-\mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{h} \in \mathbb{R}^d, \tag{8}$$

where $\mathbf{y}$ and $\mathbf{h}$ are parameters of the compressor.

3PC can be realized using δ-compressor $\mathcal{C}$. Two examples of $\mathcal{C}_{\mathbf{h},\mathbf{y}}$ are given below [41]:

$$\text{EF21}: \qquad \mathcal{C}_{\mathbf{h},\mathbf{y}}(\mathbf{x}) := \mathbf{h} + \mathcal{C}(\mathbf{x}-\mathbf{h}), \tag{9}$$

$$\text{CLAG}: \qquad \mathcal{C}_{\mathbf{h},\mathbf{y}}(\mathbf{x}) := \begin{cases} \mathbf{h} + \mathcal{C}(\mathbf{x}-\mathbf{h}), & \text{if } \|\mathbf{x}-\mathbf{h}\|^2 > \varsigma\|\mathbf{x}-\mathbf{y}\|^2 \\ \mathbf{h}, & \text{otherwise} \end{cases}. \tag{10}$$

It can be checked that $A := 1 - \sqrt{\delta}$ and $B := \frac{\delta}{1-\sqrt{\delta}}$ for EF21, and $A := 1 - \sqrt{\delta}$ and $B := \max\left\{\frac{\delta}{1-\sqrt{\delta}}, \varsigma\right\}$ for CLAG.

Next, we formulate our algorithm based on 3PC, whose pseudo code is presented in Algorithm 1. We introduce a new state $\mathbf{y}_{i,t}$ relating to $\mathbf{x}_{i,t}$, which refers to the estimation on $\mathbf{x}_{i,t}$ of agent i's neighbors. The computation of $\mathbf{H}_{i,t}(y_{i,t})^{-1}$ relies on $\mathbf{y}_{i,t}$, and since $\mathbf{y}_{i,t}$ consists of compressed information, this significantly reduces the computational cost associated with the inverse of $\mathbf{H}_{i,t}(y_{i,t})$. Then, at iteration t, agent i transmits the compressed vector $\mathcal{C}_{\mathbf{y}_{i,t-1},\mathbf{x}_{i,t-1}}(\mathbf{x}_{i,t})$ rather than $\mathbf{x}_{i,t}$ to its neighbors, which lead to the new iterations as below:

$$\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} - \left(\mathbf{H}_{i,t}(\mathbf{y}_{i,t})\right)^{-1}\Big(\nabla f(\mathbf{x}_{i,t}) + \boldsymbol{\phi}_{i,t} + \frac{\mu_z}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{y}_{i,t} - \mathbf{y}_{j,t}) + \delta_{il}\left(\boldsymbol{\lambda}_t + \mu_\theta(\mathbf{y}_{i,t} - \boldsymbol{\theta}_t)\right)\Big) \tag{11a}$$

$$\boldsymbol{\phi}_{i,t+1} = \boldsymbol{\phi}_{i,t} + \frac{\mu_z}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{y}_{i,t+1} - \mathbf{y}_{j,t+1}) \tag{11b}$$

$$\boldsymbol{\theta}_{t+1} = \delta_{il}\,\mathbf{prox}_{g/\mu_\theta}\Big(\mathbf{y}_{l,t+1} + \mu_\theta^{-1}\boldsymbol{\lambda}_t\Big) \tag{11c}$$

$$\boldsymbol{\lambda}_{t+1} = \delta_{il}\Big(\boldsymbol{\lambda}_t + \mu_\theta(\mathbf{y}_{i,t+1} - \boldsymbol{\theta}_{t+1})\Big). \tag{11d}$$

Algorithm 1 CE-DADMM

Initialization: $\mathbf{x}_{i,0} = \mathbf{0}$, $\mathbf{y}_{i,0} = \mathbf{0}$, $\boldsymbol{\phi}_{i,0} = \mathbf{0}$, $\boldsymbol{\theta}_0 = \mathbf{0}$, $\boldsymbol{\lambda}_0 = \mathbf{0}$, $\mathbf{p}_{i,t} = \nabla f_i(\mathbf{0})$.
for $t = 0,1,\ldots$ **do**
  for agent i **do**
    Compute $\mathbf{H}_{i,t}^{-1}$ using (13), (14), or (15) according to its choice;
    Compute $\mathbf{x}_{i,t+1}$ using (11a);
    $\mathbf{y}_{i,t+1} = \mathcal{C}_{\mathbf{y}_{i,t},\mathbf{x}_{i,t}}(\mathbf{x}_{i,t+1})$ // Compressing information
    Broadcast $\mathbf{y}_{i,t+1}$ to neighbors
    $\boldsymbol{\phi}_{i,t+1} = \boldsymbol{\phi}_{i,t} + \frac{\mu_z}{2}\sum_{j\in\mathcal{N}_i}(\mathbf{y}_{i,t+1} - \mathbf{y}_{j,t+1})$
    if $i = l$ **then** // Dealing with the non-smooth function
      $\boldsymbol{\theta}_{t+1} = \mathbf{prox}_{g/\mu_\theta}(\mathbf{y}_{l,t+1} + \frac{1}{\mu_\theta}\boldsymbol{\lambda}_t)$
      $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu_\theta(\mathbf{y}_{l,t+1} - \boldsymbol{\theta}_{t+1})$
    end if
  end for
end for

3.3. Discussion

Our algorithm CE-DADMM presents a flexible framework that accommodates gradient updates, Newton updates, and quasi-Newton updates, depending on the choice of matrix $\mathbf{H}_{i,t}^{-1}$. Here, $\mathbf{H}_{i,t}^{-1}$ has the following general structure:

$$\mathbf{H}_{i,t}^{-1} = \left(\mathbf{J}_{i,t} + (\mu_z|\mathcal{N}_i| + \delta_{il}\mu_\theta + \epsilon)\mathbf{I}_d\right)^{-1}, \tag{12}$$

where $\epsilon > 0$ is used to provide additional robustness and $\mathbf{J}_{i,t}$ is a matrix to be determined. A detailed discussion is presented below.

Case 1: Gradient Updates. By choosing $\mathbf{J}_{i,t} \equiv \mathbf{0}$, (12) equals to

$$\mathbf{H}_{i,t}^{-1} = \left((\mu_z|\mathcal{N}_i| + \delta_{il}\mu_\theta + \epsilon)\mathbf{I}_d\right)^{-1}. \tag{13}$$

Clearly, $\mathbf{H}_t$ is diagonal. The computation of $\mathbf{H}_t^{-1}$ requires $\mathcal{O}(d)$ computational cost. Compared with COLA [45], CE-DADMM considers the presence of the non-smooth term $g(\cdot)$ and allows for more options in the choice of the compression mechanism $\mathcal{C}$. When $g(\cdot)$ is excluded, only the lazy aggregation compression mechanism is applied, and Gradient Updates are used, CE-DADMM aligns with the form of COLA.

Case 2: Newton Updates. By choosing $\mathbf{J}_{i,t} = \nabla^2 f_i(\mathbf{y}_{i,t})$, (12) equals to

$$\mathbf{H}_{i,t}^{-1} = \left(\nabla^2 f_i(\mathbf{y}_{i,t}) + (\mu_z|\mathcal{N}_i| + \delta_{il}\mu_\theta + \epsilon)\mathbf{I}_d\right)^{-1}. \tag{14}$$

According to the definition of $F(\tilde{\mathbf{y}}_t)$, $\nabla^2 F(\tilde{\mathbf{y}}_t)$ is a block diagonal matrix with the ith block being $\nabla^2 f_i(\mathbf{y}_{i,t})$. The computation of $\mathbf{H}_t^{-1}$ incurs $\mathcal{O}(d^3)$ computational cost. When CE-DADMM uses Newton updates, excludes $g(\cdot)$, and adopts the same communication compression approach as CC-DQM [39], it recovers the form consistent with CC-DQM.

Case 3: Quasi-Newton Updates. Inspire by the distributed BFGS scheme in [22], we can derive a novel decentralized algorithm termed as CE-DADMM-BFGS, which combines the BFGS method with communication-efficient mechanisms. According to secant condition, each agent i constructs a model of the inverse Hessian directly using the pairs $\{\mathbf{q}_{i,t}, \mathbf{s}_{i,t}\}_{i=1}^n$ defined as

$$\mathbf{q}_{i,t} := (\nabla f_i(\mathbf{y}_{i,t+1}) - \nabla f_i(\mathbf{y}_{i,t})) + (\mu_z|\mathcal{N}_i| + \delta_{il}\mu_\theta + \epsilon)\mathbf{s}_{i,t} \quad \text{and} \quad \mathbf{s}_{i,t} := \mathbf{y}_{i,t+1} - \mathbf{y}_{i,t}.$$

The Hessian inverse approximation is then iteratively updated as:

$$\mathbf{H}_{i,t+1}^{-1} = \frac{\mathbf{s}_{i,t}(\mathbf{s}_{i,t})^\top}{\mathbf{q}_{i,t}^\top\mathbf{s}_{i,t}} + \left(\mathbf{I}_d - \frac{\mathbf{s}_{i,t}(\mathbf{q}_{i,t})^\top}{\mathbf{q}_{i,t}^\top\mathbf{s}_{i,t}}\right)(\mathbf{H}_{i,t})^{-1}\left(\mathbf{I}_d - \frac{\mathbf{q}_{i,t}(\mathbf{s}_{i,t})^\top}{\mathbf{q}_{i,t}^\top\mathbf{s}_{i,t}}\right). \tag{15}$$

Notably, the explicit inverse of $\mathbf{H}_{i,t+1}$ is unnecessary, as this expression serves merely as a formal representation. Consequently, the computational cost for each agent is reduced from $\mathcal{O}(d^3)$ to $\mathcal{O}(d^2)$.

4. Convergence Analysis

In this section, we propose a unified framework to analyze the proposed algorithms that incorporate gradient, Newton, and BFGS updates, along with a communication-efficient mechanism. First, we make the following assumptions throughout of the paper.

Assumption 1. *Each f_i is twice continuously differentiable, m_f-strongly convex, and M_f–smooth, i.e., $m_f\mathbf{I}_d \preceq \nabla^2 f_i(\mathbf{x}_i) \preceq M_f\mathbf{I}_d$, where $M_f \geq m_f > 0$. $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is proper, closed, and convex, i.e., $(\mathbf{x}-\mathbf{y})^\top(\mathbf{s}_\mathbf{x}-\mathbf{s}_\mathbf{y}) \geq 0$ holds for any subgradients $\mathbf{s}_\mathbf{x} \in \partial g(\mathbf{x})$ and $\mathbf{s}_\mathbf{y} \in \partial g(\mathbf{y})$.*

Assumption 2. *Each $\nabla^2 f_i$ is Lipschitz continuous with constant L_f, i.e., $\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\| \leq L_f\|\mathbf{x}-\mathbf{y}\|$ holds for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Assumption 3. *Each $\mathbf{H}_{i,t}$ is uniformly upper bounded, i.e., for any $t \geq 0$, there exists a constant $\psi > 0$ such that $\mathbf{H}_{i,t} \preceq \psi\mathbf{I}_d$.*

It is worthy noting that Assumption 3 is only required for the quasi-Newton update case. Next, we introduce the optimal condition of problem (3), which is independent of the algorithm and has been proved in [22]. The result is given below.

Lemma 1 (optimal condition, see Lemma 2 in [22])**.** *Suppose $(\tilde{\mathbf{x}}^\star, \tilde{\mathbf{z}}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)$ is a primal-dual optimal pair of problem (3), if and only if the following holds:*

$$\nabla F(\tilde{\mathbf{x}}^\star) + \mathbf{E}_s^\top \boldsymbol{\alpha}^\star + \mathbf{S}\boldsymbol{\lambda}^\star = 0, \tag{16a}$$
$$\partial g(\boldsymbol{\theta}^\star) - \boldsymbol{\lambda}^\star \ni 0, \tag{16b}$$
$$\mathbf{E}_s\tilde{\mathbf{x}}^\star = 0, \tag{16c}$$
$$\mathbf{E}_u\tilde{\mathbf{x}}^\star = 2\tilde{\mathbf{z}}^\star, \tag{16d}$$
$$\mathbf{S}^\top\tilde{\mathbf{x}}^\star = \boldsymbol{\theta}^\star. \tag{16e}$$

Moreover, there exists a unique dual optimal pair $[\boldsymbol{\alpha}^\star; \boldsymbol{\lambda}^\star] \in \mathbb{R}^{(m+1)d}$ that lies in the column space of $\mathbf{C} := \begin{bmatrix} \mathbf{E}_s \\ \mathbf{S}^\top \end{bmatrix} \in \mathbb{R}^{(m+1)d \times nd}$.

Now, we are ready to analyze our algorithm CE-DADMM. First, we write (11) into a compact form:

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - (\mathbf{H}_t)^{-1}\Big(\nabla F(\tilde{\mathbf{x}}_t) + \boldsymbol{\phi}_t + \frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{y}}_t + \mathbf{S}\Big(\boldsymbol{\lambda}_t + \mu_\theta(\mathbf{S}^\top\tilde{\mathbf{y}}_t - \boldsymbol{\theta}_t)\Big)\Big) \tag{17a}$$
$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{y}}_{t+1} \tag{17b}$$
$$\boldsymbol{\theta}_{t+1} = \mathbf{prox}_{g/\mu_\theta}(\mathbf{S}^\top\tilde{\mathbf{y}}_{t+1} + \frac{1}{\mu_\theta}\boldsymbol{\lambda}_t) \tag{17c}$$
$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu_\theta(\mathbf{S}^\top\tilde{\mathbf{y}}_{t+1} - \boldsymbol{\theta}_{t+1}). \tag{17d}$$

According to the discussion in Section 3.1, we have $\mathbf{E}_u\tilde{\mathbf{x}}_t = 2\tilde{\mathbf{z}}_t$ and $\boldsymbol{\phi}_t = \mathbf{E}_s^\top\boldsymbol{\alpha}_t$ hold in (17). Then, it follows from Lemma 1 that the convergence of CE-DADMM can be obtained by showing $(\tilde{\mathbf{x}}^t, \tilde{\mathbf{z}}^t, \boldsymbol{\alpha}^t, \boldsymbol{\theta}^t, \boldsymbol{\lambda}^t)$ converges to $(\tilde{\mathbf{x}}^\star, \tilde{\mathbf{z}}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)$.

Due to the existence of the efficient communication scheme, we need to analyze the impact of communication error on the convergence of CE-DADMM. Define $\tilde{\mathbf{e}} := [\mathbf{e}_1; \mathbf{e}_2; \cdots; \mathbf{e}_n]$, where $\mathbf{e}_i := \mathbf{y}_i - \mathbf{x}_i$ for all agent i. Clearly, $\tilde{\mathbf{e}}$ describes the error caused by efficient communication scheme. Regarding $\tilde{\mathbf{e}}$, the following result holds.

Lemma 2. *The error* $\tilde{\mathbf{e}}_{t+1}$ *in CE-DADMM satisfies* $\|\tilde{\mathbf{e}}_{t+1}\|^2 \leq (1-A)\|\tilde{\mathbf{e}}_t\|^2 + B\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|^2$, *where A and B are the parameters of 3PC.*

Proof. According to the definition of 3PC, we have

$$\|\tilde{\mathbf{e}}_{t+1}\|^2 = \|\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{x}}_{t+1}\|^2 \leq (1-A)\|\tilde{\mathbf{e}}_t\|^2 + B\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|^2,$$

which completes the proof. □

Clearly, if 3PC is set as EF21 (9), it follows from Lemma 2 that

$$\|\tilde{\mathbf{e}}_{t+1}\|^2 \leq \sqrt{\delta}\|\tilde{\mathbf{e}}_t\|^2 + \frac{\delta}{1-\sqrt{\delta}}\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|^2.$$

If 3PC is set as CLAG (10), we have

$$\|\tilde{\mathbf{e}}_{t+1}\|^2 \leq \sqrt{\delta}\|\tilde{\mathbf{e}}_t\|^2 + \max\left\{\frac{\delta}{1-\sqrt{\delta}}, \varsigma\right\}\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|^2.$$

Then, to characterize the suboptimality of the iterates when (5a) is replaced by (17a), we introduce the following error term:

$$\mathbf{r}_t := \nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{x}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t). \tag{18}$$

The bound of the error term (18) is give below, which is important for our main result.

Lemma 3. *It holds that* $\|\mathbf{r}_t\| \leq \tau_t\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| + \gamma_t(\|\tilde{\mathbf{e}}_{t+1}\| + \|\tilde{\mathbf{e}}_t\|)$, *where* τ_t *and* γ_t *correspond to the update case as below:*

$$\textit{Case 1:} \quad \tau_t = M_f, \qquad \gamma_t = 0. \tag{19}$$

$$\textit{Case 2:} \quad \tau_t = \min\left\{2M_f, \frac{L_f}{2}\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| + L_f\|\tilde{\mathbf{e}}_t\|\right\}, \qquad \gamma_t = 0. \tag{20}$$

$$\textit{Case 3:} \quad \tau_t \leq 2\psi, \qquad \gamma_t \leq 2(M_f + \psi). \tag{21}$$

Proof. See Appendix A. □

Lemma 3 extends the results in [17,18] by providing an upper bound on the error introduced when replacing the exact sub-optimization step (5a) with a one-step update using the compressed variable (17a). Under Newton updates, as the error $\|\tilde{\mathbf{e}}_t\|$ approaches zero, the term $\frac{L_f}{2}\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| + L_f\|\tilde{\mathbf{e}}_t\|$ becomes smaller than $2M_f$ in (20). In the case of Quasi-Newton updates, the error $\|\mathbf{r}_t\|$ remains bounded by a constant, ensuring it do not grow indefinitely.

Let $\sigma_{\max}^{\mathbf{L}_u}$ and $\sigma_{\min}^{\mathbf{L}_u}$ denote the maximum and minimum eigenvalues of $\mathbf{L}_u$, respectively. Let $\sigma_{\max}^{\mathbf{L}_s}$ denote the maximum eigenvalue of $\mathbf{L}_s$. Denote by $\sigma_{\min}^{+}$ the smallest positive eigenvalue of $\mathbf{C}\mathbf{C}^\top$, where $\mathbf{C}$ is given in Lemma 1. Define $\tilde{\mathbf{v}}_t := [\tilde{\mathbf{x}}_t, \tilde{\mathbf{z}}_t, \boldsymbol{\alpha}_t, \boldsymbol{\theta}_t, \boldsymbol{\lambda}_t]^\top$, $\tilde{\mathbf{v}}^\star := [\mathbf{x}_\star, \mathbf{z}_\star, \boldsymbol{\alpha}_\star, \boldsymbol{\theta}_\star, \boldsymbol{\lambda}_\star]^\top$, and $\mathcal{H}_1 = \mathrm{diag}[\epsilon, 2\mu_z, \frac{2}{\mu_z}, \mu_\theta, \frac{1}{\mu_\theta}]$. Consider the following Lyapunov function:

$$V_t = \|\tilde{\mathbf{v}}_t - \tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 + \xi\|\tilde{\mathbf{e}}_t\|^2. \tag{22}$$

Clearly, V_t converges to zero implies that $\tilde{\mathbf{v}}_t$ converges to the optimal solution. We will use V_t to establish the convergence result of our algorithm, which is given below.

Theorem 1. *Suppose Assumptions 1–3 hold. Let* $\mu_z = 2\mu_\theta$, $c_1 = 7\sigma_{\max}^{\mathbf{L}_s}\mu_\theta + 25\mu_\theta$, $c_2 = 3\sigma_{\max}^{\mathbf{L}_s}\mu_\theta + 17\mu_\theta$, *and* $c_3 = 2\mu_\theta^2(1 + (\sigma_{\max}^{\mathbf{L}_s})^2) + 4\gamma_t^2$. *If* ζ *satisfies*

$$\zeta \in \left(\frac{2m_f M_f}{m_f + M_f} - \mu_\theta, \min\left\{ \frac{\epsilon + B(6 - c_1 - \mu_\theta)}{\tau_t^2 + 2B\gamma_t^2}, \frac{4 - c_2 + (1 - A)(6 - c_1 - \mu_\theta)}{2(2 - A)\tau_t^2} \right\} \right),$$

then the iterates generated by CE-DADMM satisfy $V_{t+1} \leq \frac{1}{1+\eta_t} V_t$, *where*

$$\eta_t = \min\left\{ \frac{\epsilon + B(6 - c_1 - \mu_\theta) - \zeta(\tau_t^2 + 2B\gamma_t^2)}{8B + \frac{7}{\mu_\theta \sigma_{\min}^+}(\epsilon^2 + 2\tau_t^2 + c_3 B)}, \frac{\frac{2m_f M_f}{m_f + M_f} - \frac{1}{\zeta} - \mu_\theta}{\epsilon + 4\mu_\theta + \sigma_{\max}^{\mathbf{L}_u}\mu_\theta}, \frac{3\sigma_{\min}^+}{\sigma_{\max}^{\mathbf{L}_u}}, \right.$$
$$\left. \frac{4 - c_2 + (1 - A)(6 - c_1 - \mu_\theta) - 2(2 - A)\zeta\tau_t^2}{(1 - A)8 + \frac{7}{\mu_\theta \sigma_{\min}^+}(2 - A)c_3}, \frac{\mu_\theta \sigma_{\min}^+}{7(m_f + M_f)}, \frac{\sigma_{\min}^+}{7}, \frac{3}{16} \right\}. \tag{23}$$

Proof. See Appendix B. □

Clearly, Theorem 1 implies that CE-DADMM can achieve exact linear convergence at a rate of $\mathcal{O}(\rho^t)$ with $\rho = \frac{1}{1+\eta_t}$. The larger η_t is, the faster CE-DADMM converges. To ensure positive η_t in (23), it can be obtain that the step size μ_θ should not be too large. Besides, excessive compression of $\mathbf{x}_k$ should be avoided. This is because, when the compression ratio A is very small, B approaches infinity. According to (23), as $B \to \infty$, the first term in (23) becomes negative. It is worth noting that when $\eta_t, \gamma_t = 0$, the compression ratio A can be arbitrarily small, making the approach highly applicable in scenarios with extremely limited bandwidth. Also, at this situation, we can get the fastest convergence rate and the smallest communication cost. However, setting $\eta_t, \gamma_t = 0$ implies solving the subproblem (5a) exactly, which may result in high computational cost. This shows a trade-off between communication cost and computation cost in decentralized optimization.

5. Numerical Experiments

In this section, CE-DADMM is compared with existing state-of-the-art algorithms, including DRUID [22], PG-EXTRA [42], P2D2 [43], and CC-DQM [39], in distributed logistic/ridge/lasso regression problems. Noting that CC-DQM do not support non-smooth terms.

Datasets. We use real-world datasets from the LIBSVM library (Available Online: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html, accessed on 3 December 2024): a9a (32,561 samples, 123 dimensions) and ijcnn1 (49,990 samples, 22 dimensions). The samples are evenly distributed across n agents. The distribution of samples across agents for the a9a dataset (left) and ijcnn1 (right) is shown in Figure 1. Our experiments are implemented in Python 3.10.13.

Experimental setting. The communication graph is randomly generated, with connections based on a Bernoulli distribution ($p = 0.5$) among $n = 10$ agents, as shown in Figure 2. We evaluate performance based on the total communication bits and the number of iterations. CE-DADMM employs compression mechanisms EF21 and CLAG, using a Top-K compressor to reduce dimensions to 30 dimensions for the a9a dataset and 6 dimensions for the ijcnn1 dataset. In the experiment, we examine the algorithms from two perspectives: the number of iterations and total communication bits. The number of iterations refers to the number of times the algorithm runs, while total communication bits is calculated based on the cumulative number of bits of variable $\mathbf{y}$ transmitted between agents. Additionally, we define $err_t := \frac{\|\mathbf{x}_t - \mathbf{x}^\star\|}{\|\mathbf{x}_0 - \mathbf{x}^\star\|}$ to measure the algorithm's convergence.

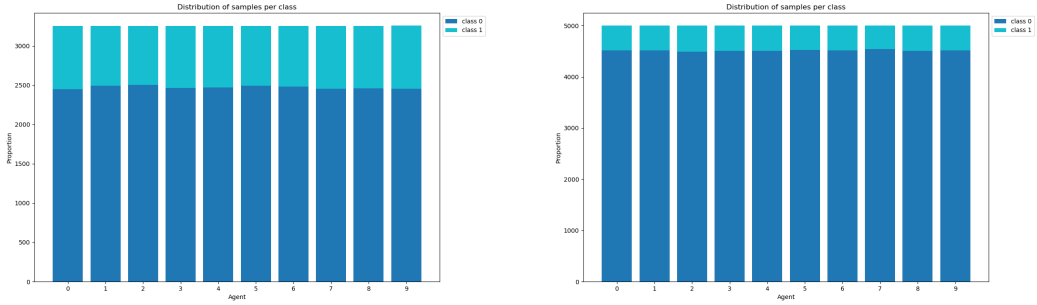

Figure 1. Distribution of samples across agents for the a9a dataset (**left**) and ijcnn1 (**right**).

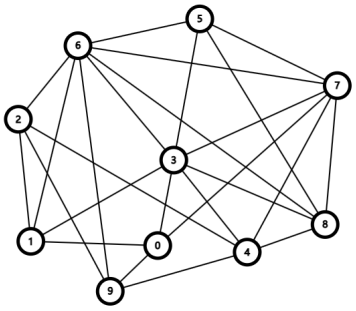

Figure 2. Random communication graph of network with 10 agents.

5.1. Distributed Logistic Regression

The distributed logistics regression solves problem (1) with $g(\mathbf{x}) = \gamma_2\|\mathbf{x}\|_1$, $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ defined as:

$$f_i(\mathbf{x}) := \frac{1}{m_i}\sum_{j=1}^{m_i}\log\left(1+e^{-b_{ij}\mathbf{a}_{ij}^\top\mathbf{x}}\right)+\frac{\gamma_1}{2}\|\mathbf{x}\|^2, \tag{24}$$

where $\mathbf{a}_{ij} \in \mathbb{R}^d$ represents the feature vector, $b_{ij} \in \{-1, 1\}$ denotes the label, and m_i is the number of sample data for agent i. The parameters $\gamma_1 = 10^{-2}$ and $\gamma_2 = 10^{-6}$ are regularization terms.

In Figures 3 and 4, when measured by the number of iterations, CE-DADMM with EF21 and CLAG compression mechanisms performs on par with DRUID without compression, and surpasses P2D2 and PG-EXTRA in both convergence speed and accuracy. Additionally, we observe that CE-DADMM, when using (quasi) Newton methods, significantly reduces the number of iterations required to reach a given accuracy compared to first-order methods. When measured by total communication bits, the introduction of EF21 and CLAG in CE-DADMM allows for a substantial reduction in communication overhead compared to DRUID, even under the same update scheme. Notably, when using quasi-Newton updates, CE-DADMM requires fewer communication bits to achieve the same accuracy than DRUID-Newton updates, and also outperforms P2D2 and PG-EXTRA in this regard. When achieving the same convergence accuracy as shown in Table 1, the detailed numerical results are presented in Tables 2 and 3. Note that for P2D2 and PG-EXTRA, since they fail to reach the predefined convergence accuracy, we use the number of iterations required for them to converge to their optimal values as a benchmark.

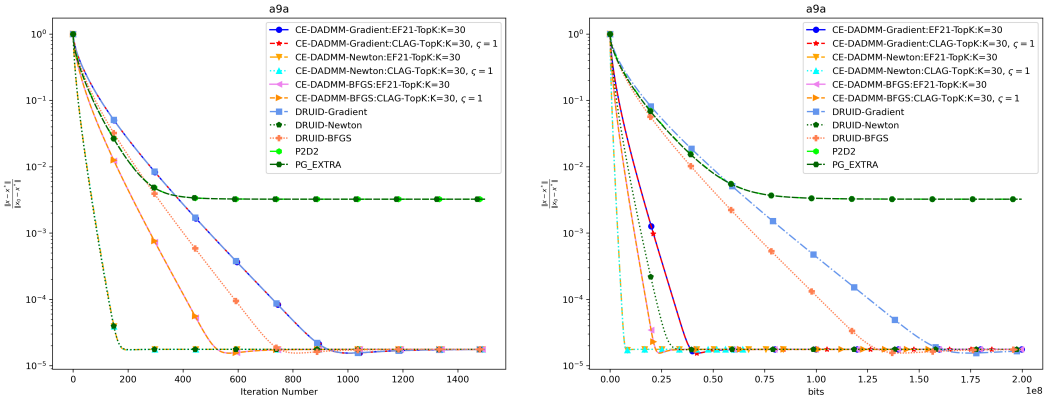

Figure 3. Performance comparison of distributed logistic regression the on a9a dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

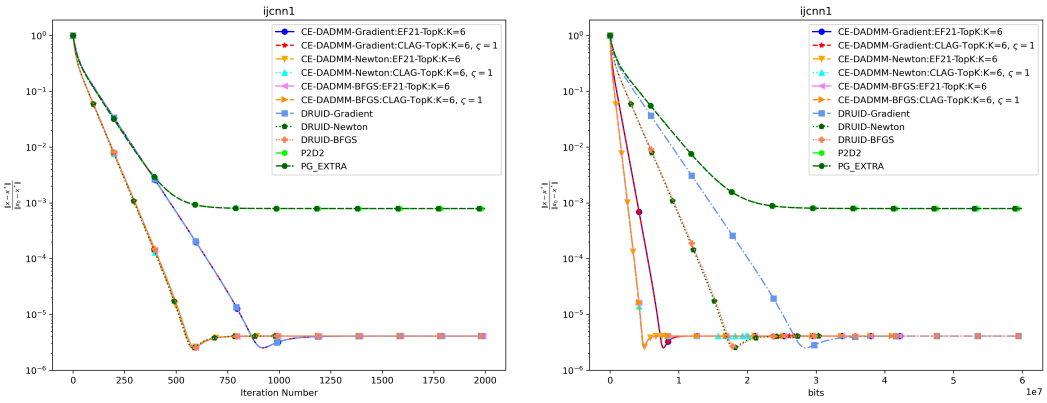

Figure 4. Performance comparison of distributed logistic regression the on ijcnn1 dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

Table 1. Convergence accuracy (err_t) for different experiments.

	Distributed Logistic Regression		**Distributed Ridge Regression**		**Distributed LASSO**	
	a9a	**ijcnn1**	**a9a**	**ijcnn1**	**a9a**	**ijcnn1**
Convergence Error (err_t)	$1 \times 10^{-4.5}$	$1 \times 10^{-5.5}$	$1 \times 10^{-6.8}$	$1 \times 10^{-6.8}$	$1 \times 10^{-4.5}$	$1 \times 10^{-5.5}$

Table 2. Comparison of iterations.

Method	**Distributed Logistic Regression**		**Distributed Ridge Regression**		**Distributed LASSO**	
	a9a	**ijcnn1**	**a9a**	**ijcnn1**	**a9a**	**ijcnn1**
P2D2	468	811	-	-	4230	409
PG-EXTRA	470	813	-	-	4231	410
CC-DQM	-	-	246	79	-	-
DRUID-Gradient	845	884	977	176	3348	445
CE-DADMM-Gradient:EF21	845	884	977	171	3348	446
CE-DADMM-Gradient:CLAG	845	884	980	170	3349	446
DRUID-Newton	154	559	197	60	1910	318
CE-DADMM-Newton:EF21	155	559	252	60	1912	318
CE-DADMM-Newton:CLAG	155	559	194	64	1912	318
DRUID-BFGS	684	566	330	77	2890	399
CE-DADMM-BFGS:EF21	478	567	325	73	2890	400
CE-DADMM-BFGS:CLAG	478	566	327	69	2890	399

Table 3. Comparison of communication bits.

Method	Distributed Logistic Regression		Distributed Ridge Regression		Distributed LASSO	
	a9a	**ijcnn1**	**a9a**	**ijcnn1**	**a9a**	**ijcnn1**
P2D2	116,056,832	38,539,264	-	-	862,720,000	29,344,768
PG-EXTRA	125,088,000	30,080,000	-	-	861,520,000	20,101,504
CC-DQM	-	-	20,782,080	1,219,680	-	-
DRUID-Gradient	146,340,480	27,382,784	169,200,768	5,451,776	579,820,032	13,784,320
CE-DADMM-Gradient:EF21	35,692,800	7,468,032	41,268,480	1,444,608	141,419,520	3,767,808
CE-DADMM-Gradient:CLAG	35,650,560	7,459,584	41,352,960	1,427,712	141,419,520	3,759,360
DRUID-Newton	26,670,336	17,315,584	34,117,248	1,858,560	330,781,440	9,850,368
CE-DADMM-Newton:EF21	6,547,200	4,722,432	10,644,480	506,880	80,762,880	2,686,464
CE-DADMM-Newton:CLAG	6,504,960	4,713,984	8,152,320	532,224	80,720,640	2,678,016
DRUID-BFGS	118,457,856	17,532,416	57,150,720	2,385,152	500,501,760	12,359,424
CE-DADMM-BFGS:EF21	20,190,720	4,790,016	13,728,000	616,704	122,073,600	3,379,200
CE-DADMM-BFGS:CLAG	20,148,480	4,773,120	13,770,240	574,464	122,031,360	3,362,304

5.2. Distributed Ridge Regression

The distributed ridge regression solves problem (1), whose $f_i(\cdot)$ is the same as in (24) but with $g(x) = 0$.

In Figures 5 and 6, we observe that when measured by the number of iterations, CE-DADMM with EF21 and CLAG compression mechanisms performs similarly to DRUID without compression. However, CE-DADMM using (quasi) Newton updates converges faster compared to CC-DQM, which also employs an communication-efficient mechanism. On the other hand, when CE-DADMM employs first-order method, its convergence is slower than second-order method, CC-DQM, due to the latter benefiting from additional Hessian information. When measured by total communication bits, CE-DADMM with (quasi) Newton updates requires fewer bits to achieve the same accuracy compared to CC-DQM. Additionally, CE-DADMM benefits from communication-efficient mechanisms, resulting in a substantial reduction in communication bits needed to achieve the same convergence accuracy compared to DRUID. Similarly, the corresponding numerical results are also presented in Tables 2 and 3.

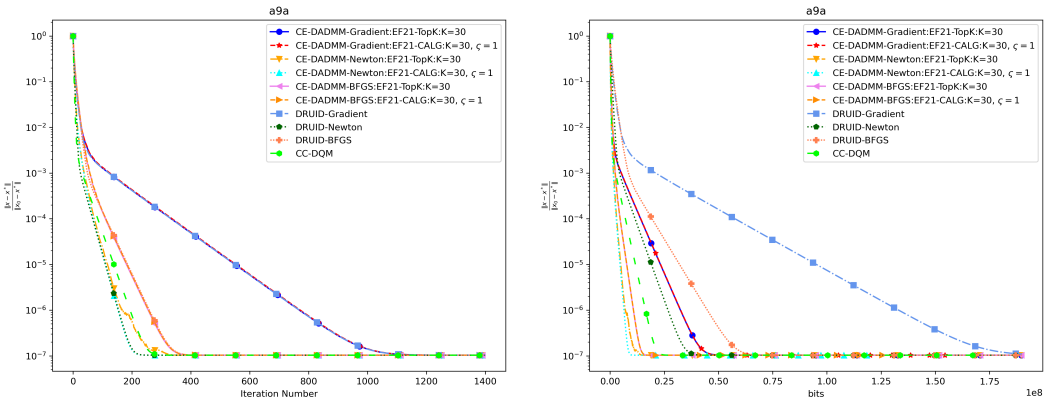

Figure 5. Performance comparison of distributed ridge regression on the a9a dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

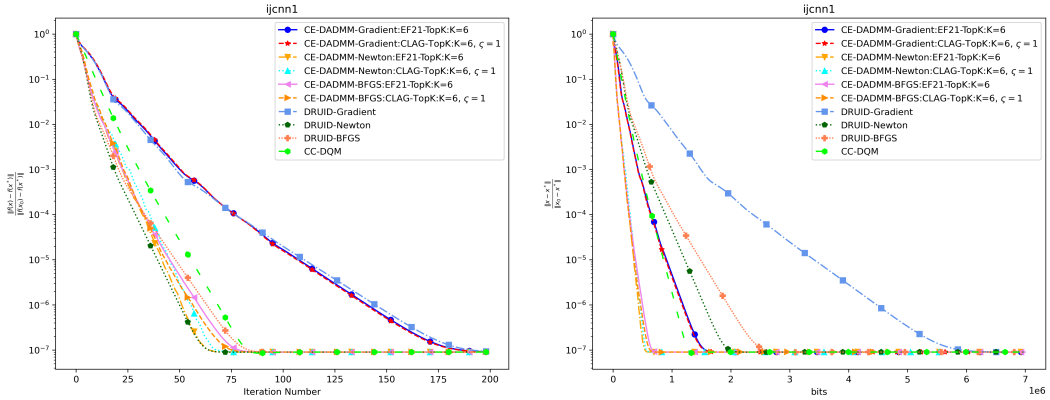

Figure 6. Performance comparison of distributed ridge regression on the ijcnn1 dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

5.3. Distributed LASSO

The distributed LASSO solves problem (1) with $g(\mathbf{x}) = \gamma_2\|\mathbf{x}\|_1$, $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ defined as:

$$f_i(\mathbf{x}) := \frac{1}{2m_i}\sum_{j=1}^{m_i}\|\mathbf{a}_{ij}\mathbf{x} - b_j\|^2 + \frac{\gamma_1}{2}\|\mathbf{x}\|^2, \tag{25}$$

where $\mathbf{a}_{ij}$, b_{ij}, and m_i are as defined in Section 5.1. The parameters $\gamma_1 = 10^{-2}$ and $\gamma_2 = 10^{-6}$ are regularization terms.

In Figures 7 and 8, when measured by the number of iterations, CE-DADMM performs similarly to DRUID and outperforms P2D2 and PG-EXTRA in terms of convergence speed and accuracy, indicating that the introduction of the 3PC compression mechanism does not negatively affect the convergence speed of the CE-DADMM algorithm. When measured by total communication bits, after introducing the EF21 and CLAG compression mechanisms, CE-DADMM significantly reduces communication overhead compared to DRUID, while also outperforming P2D2 and PG-EXTRA, demonstrating that the communication compression mechanisms can significantly lower communication costs between agents. Similarly, the corresponding numerical results are also presented in Tables 2 and 3.

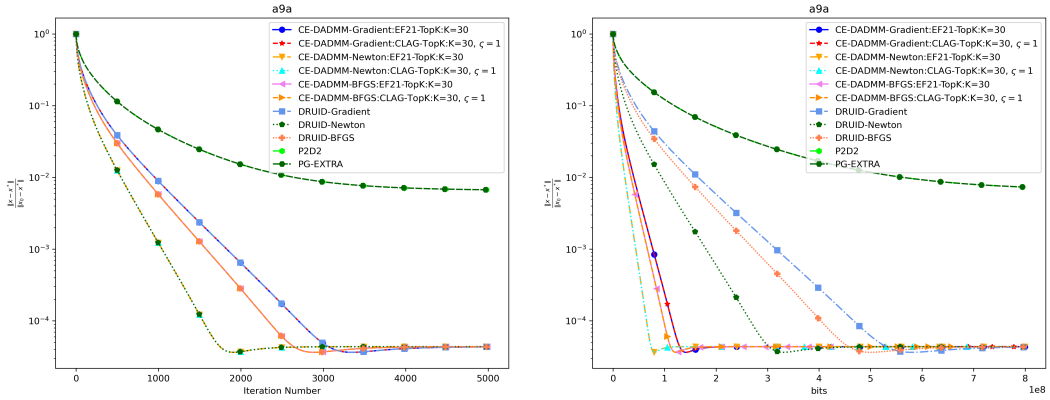

Figure 7. Performance comparison of distributed LASSO on the a9a dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

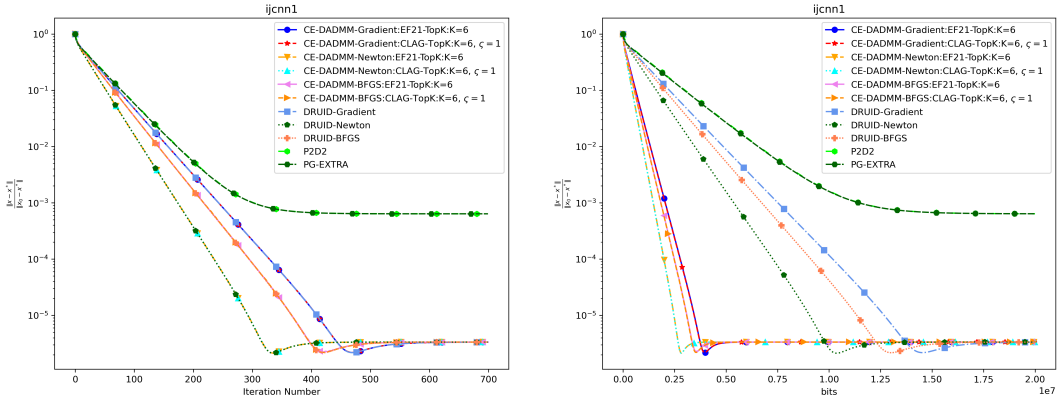

Figure 8. Performance comparison of distributed LASSO on the ijcnn1 dataset: Plots of iteration number (**left**) and total communication bits (**right**) versus distance error.

6. Conclusions

This paper presents a communication-efficient ADMM algorithm for composite optimization, named as CE-DADMM. The algorithm utilizes the ADMM framework with the 3PC communication mechanism, effectively adapting to various communication and computational demands while balancing communication efficiency and computational cost. Notably, when employing quasi-Newton updates, CE-DADMM becomes the first compression-based second-order communication-efficient algorithm. Theoretical analysis demonstrates that the proposed algorithm achieves linear convergence when the local objective functions are strongly convex. Numerical experiments further validate the effectiveness and superior performance of the algorithm. Future work will focus on extending the algorithm to fully asynchronous settings and stochastic problems.

Author Contributions: Z.C.: Conceptualization, writing—original draft and methodology; Z.Z.: Conceptualization, writing—original draft and methodology; S.Y.: writing—review and editing and supervision; J.C.: writing—review and editing and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62176056, and in part by the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (CAST) under Grant 2021QNRC001.

Data Availability Statement: The data supporting the findings of this study are openly available in a9a and ijcnn1 at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html (accessed on 3 December 2024).

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this article.

Appendix A. Proof of Lemma 3

Proof of Lemma 3. First, according to (18), it follows from the triangle inequality and Cauchy–Schwartz inequality that

$$\|\mathbf{r}_t\| \leq \|\nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{x}}_{t+1})\| + \|\mathbf{J}_t\|\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|. \tag{A1}$$

For case 1 (gradient updates), there is $\mathbf{J}_t \equiv 0$. By Assumption 1, we obtain $\|\mathbf{r}_t\| \leq \|\nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{x}}_{t+1})\| \leq M_f\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|$.

For case 2 (Newton updates), there is $\mathbf{J}_t = \nabla^2 F(\tilde{\mathbf{y}}_t)$. Applying Assumption 1 and (A1) yields

$$\|\mathbf{r}_t\| \leq 2M_f\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|. \tag{A2}$$

In parallel, by the fundamental theorem of calculus, we can obtain

$$\begin{aligned}
&\nabla F(\tilde{\mathbf{x}}_{t+1}) - \nabla F(\tilde{\mathbf{x}}_t) \\
=& \int_0^1 \nabla^2 F(s\tilde{\mathbf{x}}_{t+1} + (1-s)\tilde{\mathbf{x}}_t)(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t) ds \\
=& \int_0^1 (\nabla^2 F(s\tilde{\mathbf{x}}_{t+1} + (1-s)\tilde{\mathbf{x}}_t) - \nabla^2 F(\tilde{\mathbf{y}}_t))(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t) ds + \nabla^2 F(\tilde{\mathbf{y}}_t)(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t),
\end{aligned} \tag{A3}$$

which implies that

$$\begin{aligned}
&\|\nabla F(\tilde{\mathbf{x}}_{t+1}) - \nabla F(\tilde{\mathbf{x}}_t) - \nabla^2 F(\tilde{\mathbf{y}}_t)(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t)\| \\
=& \left\| \int_0^1 (\nabla^2 F(s\tilde{\mathbf{x}}_{t+1} + (1-s)\tilde{\mathbf{x}}_t) - \nabla^2 F(\tilde{\mathbf{y}}_t))(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t) ds \right\| \\
\leq& \int_0^1 \|\nabla^2 F(s\tilde{\mathbf{x}}_{t+1} + (1-s)\tilde{\mathbf{x}}_t) - \nabla^2 F(\tilde{\mathbf{y}}_t)\| \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| ds \\
\leq& \int_0^1 L_f \|s(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t) + (\tilde{\mathbf{x}}_t - \tilde{\mathbf{y}}_t)\| \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| ds \\
\leq& \int_0^1 L_f s \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|^2 ds + L_f \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{y}}_t\| \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| \\
=& \left(\frac{L_f}{2} \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\| + L_f \|\tilde{\mathbf{e}}_t\| \right) \|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|.
\end{aligned} \tag{A4}$$

The result for case 2 can be obtained by comparing (A2) and (A4).

For case 3 (quasi-Newton updates), according to Assumption 3, the secant condition $\mathbf{H}_{t+1}\mathbf{s}_t = \mathbf{q}_t$, and the definition of $\{\mathbf{q}_t, \mathbf{s}_t\}$, we have

$$\begin{aligned}
\|\mathbf{r}_t\| &= \|\nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{x}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t)\| \\
&= \|\nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{y}}_t) + \mathbf{J}_t(\tilde{\mathbf{y}}_t - \tilde{\mathbf{x}}_t) - \left(\nabla F(\tilde{\mathbf{x}}_{t+1}) - \nabla F(\tilde{\mathbf{y}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{x}}_{t+1})\right) \\
&\quad + \nabla F(\tilde{\mathbf{y}}_t) - \nabla F(\tilde{\mathbf{y}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{y}}_t)\| \\
&\leq \|\nabla F(\tilde{\mathbf{x}}_t) - \nabla F(\tilde{\mathbf{y}}_t) + \mathbf{J}_t(\tilde{\mathbf{y}}_t - \tilde{\mathbf{x}}_t)\| + \|\nabla F(\tilde{\mathbf{x}}_{t+1}) - \nabla F(\tilde{\mathbf{y}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{x}}_{t+1})\| \\
&\quad + \|\nabla F(\tilde{\mathbf{y}}_t) - \nabla F(\tilde{\mathbf{y}}_{t+1}) + \mathbf{J}_t(\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{y}}_t)\| \\
&\leq 2(M_f + \psi)(\|\tilde{\mathbf{e}}_{t+1}\| + \|\tilde{\mathbf{e}}_t\|) + 2\psi\|\tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_t\|,
\end{aligned} \tag{A5}$$

which completes the proof of case 3. □

Appendix B. Proof of Theorem 1

Lemma A1. *Let $(\boldsymbol{\alpha}_\star, \boldsymbol{\lambda}_\star)$ be the unique dual optimal pair which lies in the column space of $\mathbf{C}$ as established in Lemma 1. The following inequality holds:*

$$\sigma_{\min}^{+}\left(\|\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_\star\|^2 + \|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_\star\|^2\right) \leq \|\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_\star) + \mathbf{S}(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_\star)\|^2. \tag{A6}$$

Proof. We rewrite (17b) and (17d) as

$$\begin{bmatrix} \boldsymbol{\alpha}_{t+1} \\ \boldsymbol{\lambda}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ \boldsymbol{\lambda}_t \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\mu_z}{2}\mathbf{E}_s \\ \mu_\theta \mathbf{S}^\top \end{bmatrix}}_{:=\mathbf{N}} \tilde{\mathbf{y}}_{t+1} - \underbrace{\begin{bmatrix} \mathbf{0} \\ \mu_\theta \mathbf{I}_d \end{bmatrix}}_{:=\mathbf{M}} \boldsymbol{\theta}_{t+1}.$$

First, we show that the column space of $\mathbf{M}$ belongs to the column space of $\mathbf{N}$. We fix all $\mathbf{y}_i$ as $\mathbf{y}'$, i.e., $\mathbf{y} = [\mathbf{y}'; \cdots; \mathbf{y}']$, then it holds that $\mathbf{N}\mathbf{y} = \mathbf{M}\mathbf{y}'$, which shows $\mathrm{col}(\mathbf{M}) \subset \mathrm{col}(\mathbf{N})$. By setting $\mu_z = 2\mu_\theta$, we conclude that $[\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_\star; \boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_\star]$ lies in the column space of $\mathbf{C}$. □

Lemma A2. *The following two inequalities hold:*

$$(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\geq 0, \tag{A7}$$

$$(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star)\geq 0. \tag{A8}$$

Proof. From the definition of the proximal operator, it holds that

$$\boldsymbol{\theta}_{t+1}=\arg\min_{\boldsymbol{\theta}}\Big\{g(\boldsymbol{\theta})+\frac{\mu_\theta}{2}\Big\|\mathbf{S}^\top\tilde{\mathbf{y}}_{t+1}+\frac{1}{\mu_\theta}\boldsymbol{\lambda}_t-\boldsymbol{\theta}\Big\|^2\Big\}. \tag{A9}$$

By the optimal condition of (A9) and the dual update (17d), we obtain

$$0\in\partial g(\boldsymbol{\theta}_{t+1})-\mu_\theta\Big(\mathbf{S}^\top\tilde{\mathbf{y}}_{t+1}+\frac{1}{\mu_\theta}\boldsymbol{\lambda}_t-\boldsymbol{\theta}_{t+1}\Big)=\partial g(\boldsymbol{\theta}_{t+1})-\boldsymbol{\lambda}_{t+1},$$

which implies that $\boldsymbol{\lambda}_t\in\partial g(\boldsymbol{\theta}_t)$. Then, it follows from the convexity of $g(\cdot)$ that

$$(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\in(\partial g(\boldsymbol{\theta}_{t+1})-\partial g(\boldsymbol{\theta}_t))^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\geq 0.$$

Similarly, using (16b), we also have

$$(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star)\in(\partial g(\boldsymbol{\theta}_{t+1})-\partial g(\boldsymbol{\theta}_\star))^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star)\geq 0.$$

The proof is completed. □

Proof of Theorem 1. To make the proof more concise, let $\Theta_{t+1}=\frac{m_fM_f}{m_f+M_f}\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2+\frac{1}{m_f+M_f}\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2$. As $F(\mathbf{x})$ is strongly convex with Lipschitz continuous gradient, the following inequality holds:

$$\begin{aligned}\Theta_{t+1}&\leq(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top(\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star))\\&\leq-(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{r}_t-\underbrace{\epsilon(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t)}_{\Xi_1}-\underbrace{(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star)}_{\Xi_2}\\&\quad-\underbrace{\mu_z(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{E}_u^\top(\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t)}_{\Xi_3}-\underbrace{(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{S}\big(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star+\mu_\theta(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\big)}_{\Xi_4}\\&\quad+\underbrace{\frac{\mu_z}{2}(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)}_{\Xi_5}+\underbrace{\mu_\theta(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)}_{\Xi_6}.\end{aligned} \tag{A10}$$

For Ξ_1, we have

$$-2\Xi_1\leq\epsilon(\|\tilde{\mathbf{x}}_t-\tilde{\mathbf{x}}_\star\|^2-\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2-\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2). \tag{A11}$$

For Ξ_2, since $\mathbf{E}_s^\top\boldsymbol{\alpha}_{t+1}=\mathbf{E}_s^\top\boldsymbol{\alpha}_t+\frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{y}}_{t+1}$, it follows from (16c) that

$$\begin{aligned}-2\Xi_2&=-2(\tilde{\mathbf{y}}_{t+1}-\tilde{\mathbf{e}}_{t+1})^\top\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star)\\&=-\frac{4}{\mu_z}(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t)^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star))+2e_{t+1}^\top\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star)\\&=\frac{2}{\mu_z}(\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2)+2e_{t+1}^\top\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star)\\&\leq\frac{2}{\mu_z}(\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2)\\&\quad+4\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{E}_s^\top\mathbf{E}_s\tilde{\mathbf{e}}_{t+1}+\frac{1}{4\mu_\theta}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2.\end{aligned} \tag{A12}$$

For Ξ_3, using $\tilde{\mathbf{z}}_t = \frac{1}{2}\mathbf{E}_u\tilde{\mathbf{x}}_t$ and (16d), there is

$$\begin{aligned} -2\Xi_3 =& -4\mu_z(\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star)^\top(\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t) \\ =& 2\mu_z(\|\tilde{\mathbf{z}}_t-\tilde{\mathbf{z}}_\star\|^2-\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star\|^2-\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2). \end{aligned} \tag{A13}$$

For Ξ_4, using (16e), we have

$$\begin{aligned} -2\Xi_4 =& -2(\tilde{\mathbf{y}}_{t+1}-\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{S}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star+\mu_\theta(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)) \\ =& -2\Big(\frac{1}{\mu_\theta}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)+\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\Big)^\top\big(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star+\mu_\theta(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\big) \\ &+2\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\big(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star+\mu_\theta(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\big) \\ =& -\frac{2}{\mu_\theta}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)^\top(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)\underbrace{-2(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)}_{\le 0,\text{ using (A7)}} \\ &-2\mu_\theta(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star)^\top(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\underbrace{-2(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star)^\top(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)}_{\le 0,\text{ using (A8)}} \\ &+2\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\Big(\frac{1}{\mu_\theta}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)+\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\Big) \\ \le& \frac{1}{\mu_\theta}(\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}_\star\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2) \\ &+\mu_\theta(\|\boldsymbol{\theta}_t-\boldsymbol{\theta}_\star\|^2-\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2-\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2) \\ &+8\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\mathbf{S}^\top\tilde{\mathbf{e}}_{t+1}+\frac{1}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2+\frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2. \end{aligned} \tag{A14}$$

For Ξ_5, using $\mathbf{E}_s^\top\boldsymbol{\alpha}_{t+1} = \mathbf{E}_s^\top\boldsymbol{\alpha}_t + \frac{\mu_z}{2}\mathbf{L}_s\tilde{\mathbf{y}}_{t+1}$ and (16c), we have

$$\begin{aligned} 2\Xi_5 &= \mu_z(\tilde{\mathbf{y}}_{t+1}-\tilde{\mathbf{e}}_{t+1})^\top\mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &= \mu_z\tilde{\mathbf{y}}_{t+1}^\top\mathbf{E}_s^\top\mathbf{E}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-\mu_z\tilde{\mathbf{e}}_{t+1}^\top\mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &= 2(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t)^\top\mathbf{E}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-\mu_z\tilde{\mathbf{e}}_{t+1}^\top\mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &\le \mu_\theta(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)^\top\mathbf{E}_s^\top\mathbf{E}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)+\mu_\theta^{-1}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2+\mu_z\tilde{\mathbf{e}}_{t+1}^\top\mathbf{L}_s\tilde{\mathbf{e}}_t. \end{aligned} \tag{A15}$$

For Ξ_6, using (17d) and (16e), we have

$$\begin{aligned} 2\Xi_6 &= 2\mu_\theta\tilde{\mathbf{y}}_{t+1}^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-2\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-2\mu_\theta\tilde{\mathbf{x}}_\star^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &= 2\mu_\theta\Big(\frac{1}{\mu_\theta}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)+\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\Big)^\top\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-2\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &\le 8\mu_\theta(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)^\top\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) \\ &\quad+\frac{1}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2+\frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2+2\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\mathbf{S}\mathbf{S}^\top\tilde{\mathbf{e}}_t. \end{aligned} \tag{A16}$$

Substituting (A11)–(A16) into (A10) yields

$$\begin{aligned} 2\Theta_{t+1} \le& \epsilon(\|\tilde{\mathbf{x}}_t-\tilde{\mathbf{x}}_\star\|^2-\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2-\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2) \\ &+2\mu_z(\|\tilde{\mathbf{z}}_t-\tilde{\mathbf{z}}_\star\|^2-\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star\|^2-\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2) \\ &+\frac{2}{\mu_z}(\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2-\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2) \\ &+\mu_\theta(\|\boldsymbol{\theta}_t-\boldsymbol{\theta}_\star\|^2-\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2-\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2) \\ &+\frac{1}{\mu_\theta}(\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}_\star\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2-\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2) \end{aligned}$$

$$
\begin{aligned}
&+\underbrace{4\mu_\theta \tilde{\mathbf{e}}_{t+1}^\top \mathbf{L}_s \tilde{\mathbf{e}}_{t+1} + \mu_\theta(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)^\top \mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) + \mu_z \tilde{\mathbf{e}}_{t+1}^\top \mathbf{L}_s \tilde{\mathbf{e}}_t}_{\Xi_7} \\
&+\underbrace{8\mu_\theta \tilde{\mathbf{e}}_{t+1}^\top \mathbf{S}\mathbf{S}^\top \tilde{\mathbf{e}}_{t+1} + 8\mu_\theta(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)^\top \mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t) + 2\mu_\theta \tilde{\mathbf{e}}_{t+1}^\top \mathbf{S}\mathbf{S}^\top \tilde{\mathbf{e}}_t}_{\Xi_8} \\
&+\frac{1}{4\mu_\theta}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2 + \frac{1}{\mu_\theta}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2 + \frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2 + \frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2 \\
&+\frac{1}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2 + 4\frac{1}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2 - 2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top \mathbf{r}_t, \qquad \text{(A17)}
\end{aligned}
$$

where Ξ_7 and Ξ_8 can be further estimated as

$$
\begin{aligned}
\Xi_7 &\le \sigma_{\max}^{\mathbf{L}_s}(4\mu_\theta\|\tilde{\mathbf{e}}_{t+1}\|^2 + \mu_\theta\|\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t\|^2 + \mu_z\tilde{\mathbf{e}}_{t+1}^\top\tilde{\mathbf{e}}_t) \\
&\le \sigma_{\max}^{\mathbf{L}_s}(\frac{\mu_z}{2}+6\mu_\theta)\|\tilde{\mathbf{e}}_{t+1}\|^2 + \sigma_{\max}^{\mathbf{L}_s}(\frac{\mu_z}{2}+2\mu_\theta)\|\tilde{\mathbf{e}}_t\|^2, \qquad \text{(A18)}
\end{aligned}
$$

and

$$
\begin{aligned}
\Xi_8 &\le 8\mu_\theta\|\tilde{\mathbf{e}}_{t+1}\|^2 + 8\mu_\theta\|\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t\|^2 + 2\mu_\theta\tilde{\mathbf{e}}_{t+1}^\top\tilde{\mathbf{e}}_t \\
&\le 25\mu_\theta\|\tilde{\mathbf{e}}_{t+1}\|^2 + 17\mu_\theta\|\tilde{\mathbf{e}}_t\|^2, \qquad \text{(A19)}
\end{aligned}
$$

where (A19) uses the fact that the largest eigenvalue of $\mathbf{S}\mathbf{S}^\top$ is 1. Finally, substituting (A18) and (A19) into (A17), we obtain

$$
\begin{aligned}
2\Theta_{t+1} \le\ & \epsilon(\|\tilde{\mathbf{x}}_t-\tilde{\mathbf{x}}_\star\|^2 - \|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2 - \|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2) \\
&+2\mu_z(\|\tilde{\mathbf{z}}_t-\tilde{\mathbf{z}}_\star\|^2 - \|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star\|^2 - \|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2) \\
&+\frac{2}{\mu_z}\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_\star\|^2 - (\frac{2}{\mu_z}-\frac{1}{4\mu_\theta})\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2 - (\frac{2}{\mu_z}-\frac{1}{\mu_\theta})\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_t\|^2 \\
&+\mu_\theta\|\boldsymbol{\theta}_t-\boldsymbol{\theta}_\star\|^2 - \frac{3\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2 - \frac{3\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2 \\
&+\frac{1}{\mu_\theta}\|\boldsymbol{\lambda}_t-\boldsymbol{\lambda}_\star\|^2 - \frac{3\mu_\theta}{4}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2 - \frac{3\mu_\theta}{4}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2 - 2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{r}_t \\
&+c_1\|\tilde{\mathbf{e}}_{t+1}\|^2 + c_2\|\tilde{\mathbf{e}}_t\|^2. \qquad \text{(A20)}
\end{aligned}
$$

Recall the definitions of $\tilde{\mathbf{v}}$ and $\mathcal{H}_1$, and define $\mathcal{H}_2 = \mathrm{diag}[\epsilon, 4\mu_\theta, \frac{3}{4\mu_\theta}, \frac{3\mu_\theta}{4}, \frac{3}{4\mu_\theta}]$ and $\mathcal{H}_3 = \mathrm{diag}[\epsilon, 4\mu\theta, 0, \frac{3\mu_\theta}{4}, \frac{3}{4\mu_\theta}]$. The inequalities $\mathcal{H}_1 \succ \mathcal{H}_2$ and $\mathcal{H}_1 \succ \mathcal{H}_3$ hold, along with the assumption that $\xi > 0$. Thus, (A20) can be rewritten as

$$
\begin{aligned}
&2\Theta_{t+1} + \|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_t\|_{\mathcal{H}_3}^2 + 2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{r}_t + (\xi-c_1)\|\tilde{\mathbf{e}}_{t+1}\|^2 + (\xi-c_2)\|\tilde{\mathbf{e}}_t\|^2 \\
&-(\frac{1}{4\mu_\theta}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2 + \frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2 + \frac{1}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2) \\
&\le \|\tilde{\mathbf{v}}_t-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 - \|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 + \xi\|\tilde{\mathbf{e}}_t\|^2 - \xi\|\tilde{\mathbf{e}}_{t+1}\|^2. \qquad \text{(A21)}
\end{aligned}
$$

To establish linear convergence, we need to show the following holds for some $\eta_t > 0$:

$$
(1+\eta_t)(\|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 + \xi\|\tilde{\mathbf{e}}_{t+1}\|^2) \le \|\tilde{\mathbf{v}}_t-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 + \xi\|\tilde{\mathbf{e}}_t\|^2. \qquad \text{(A22)}
$$

Note that

$$
\begin{aligned}
&\eta_t(\|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2 + \xi\|\tilde{\mathbf{e}}_{t+1}\|^2) \\
&= \eta_t\Big(\epsilon\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2 + 2\mu_z\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star\|^2 + \frac{2}{\mu_z}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2 + \mu_\theta\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2 \\
&\quad + \frac{1}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2 + \xi\|\tilde{\mathbf{e}}_{t+1}\|^2\Big). \qquad \text{(A23)}
\end{aligned}
$$

Next, we establish an upper bound for each component of (A23), primarily using the inequality $(\sum_{i=1}^{n}\alpha_i)^2 \le \sum_{i=1}^{n} n\alpha_i^2$ and $\mu_z = 2\mu_\theta$. First, according to Lemma A1, we obtain

$$\begin{aligned}
&\frac{2}{\mu_z}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2+\frac{1}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2\\
\le&\frac{1}{\mu_\theta\sigma_{\min}^{+}}\|\mathbf{E}_s^\top(\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star)+\mathbf{S}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star)\|^2\\
=&\frac{1}{\mu_\theta\sigma_{\min}^{+}}\|-\Big(\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)+\epsilon(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t)+2\mu_\theta\mathbf{E}_u^\top(\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t)+\mu_\theta\mathbf{S}(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\\
&-\mu_\theta\mathbf{L}_s(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)-\mu_\theta\mathbf{S}\mathbf{S}^\top(\tilde{\mathbf{e}}_{t+1}-\tilde{\mathbf{e}}_t)+\mathbf{r}_t\Big)\|^2\\
\le&\frac{7}{\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+\epsilon^2\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2\\
&+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+2\mu_\theta^2\big(1+(\sigma_{\max}^{\mathbf{L}_s})^2\big)\big(\|\tilde{\mathbf{e}}_{t+1}\|^2+\|\tilde{\mathbf{e}}_t\|^2\big)+\|\mathbf{r}_t\|^2\Big).
\end{aligned}\tag{A24}$$

Next, since $\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star=\frac{1}{2}\mathbf{E}_u(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)$, we have

$$2\mu_z\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_\star\|^2\le\mu_\theta\sigma_{\max}^{\mathbf{L}_u}\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2.\tag{A25}$$

Then, from (17d) and (16e), we obtain

$$\begin{aligned}
\mu_\theta\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2&=\mu_\theta\|\mathbf{S}^\top(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star+\tilde{\mathbf{e}}_{t+1})+\frac{1}{\mu_\theta}(\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t)\|^2\\
&\le 4\mu_\theta\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2+4\mu_\theta\|\tilde{\mathbf{e}}_{t+1}\|^2+\frac{2}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2.
\end{aligned}\tag{A26}$$

Finally, substituting (A24)–(A26) into (A23), applying Lemma 2, we obtain

$$\begin{aligned}
&\eta_t(\|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_\star\|_{\mathcal{H}_1}^2+\xi\|\tilde{\mathbf{e}}_{t+1}\|^2)\\
\le&\eta_t\Big\{\frac{7}{\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+\epsilon^2\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2\\
&+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+2\mu_\theta^2(1+(\sigma_{\max}^{\mathbf{L}_s})^2)(\|\tilde{\mathbf{e}}_{t+1}\|^2+\|\tilde{\mathbf{e}}_t\|^2)+\|\mathbf{r}_t\|^2\Big)\\
&+\frac{2}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2+(\epsilon+4\mu_\theta+\sigma_{\max}^{\mathbf{L}_u}\mu_\theta)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2+(4+\xi)\|\tilde{\mathbf{e}}_{t+1}\|^2\Big\}\\
\le&\eta_t\Big\{\frac{7}{\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2\Big)\\
&+\frac{2}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2+\Big((1-A)(4+\xi)+\frac{7c_3(2-A)}{\mu_\theta\sigma_{\min}^{+}}\Big)\|\tilde{\mathbf{e}}_t\|^2\\
&+\Big(B(4+\xi)+\frac{7}{\mu_\theta\sigma_{\min}^{+}}\big(\epsilon^2+2\tau_t^2+c_3B\big)\Big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2\\
&+(\epsilon+4\mu_\theta+\mu_z\sigma_{\max}^{\mathbf{L}_\theta})\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2\Big\}\\
\le&2\Theta_{t+1}+\|\tilde{\mathbf{v}}_{t+1}-\tilde{\mathbf{v}}_t\|_{\mathcal{H}_3}^2+2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{r}_t+(\xi-c_1)\|\tilde{\mathbf{e}}_{t+1}\|^2+(\xi-c_2)\|\tilde{\mathbf{e}}_t\|^2\\
&-\frac{1}{4\mu_\theta}\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2-\frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2-\frac{1}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2\\
\le&2\Theta_{t+1}+\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+\big(\epsilon+(\xi-\Xi_7)B\big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+4\mu_\theta\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2\\
&+\big(\xi-\Xi_8+(1-A)(\xi-\Xi_7)\big)\|\tilde{\mathbf{e}}_t\|^2+2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top\mathbf{r}_t\\
&-\frac{1}{4\mu_\theta}\big(\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2+\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2\big)-\frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2\\
&+\frac{3\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+\frac{3}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2.
\end{aligned}\tag{A27}$$

To ensure that both sides of inequality (A27) hold, we need to separately consider and determine the coefficient of η_t. We consider the terms in (A27) separately:

$$\begin{aligned}
&-2(\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star)^\top \mathbf{r}_t \\
&\le \zeta\|\mathbf{r}_t\|^2+\frac{1}{\zeta}\|\tilde{\mathbf{x}}_{t+1}+\tilde{\mathbf{x}}_\star\|^2 \\
&\le \zeta(\tau_t^2+2B\gamma_t^2)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+2(2-A)\zeta\tau_t^2\|\tilde{\mathbf{e}}_t\|^2+\frac{1}{\zeta}\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2,
\end{aligned} \tag{A28}$$

and

$$\begin{aligned}
&\frac{1}{4\mu_\theta}\big(\|\boldsymbol{\alpha}_{t+1}-\boldsymbol{\alpha}_\star\|^2+\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_\star\|^2\big) \\
&\le \frac{7}{4\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+\epsilon^2\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2 \\
&\quad+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+2\mu_\theta^2\big(1+(\sigma_{\max}^{\mathbf{L}_s})^2\big)\big(\|\tilde{\mathbf{e}}_{t+1}\|^2+\|\tilde{\mathbf{e}}_t\|^2\big)+\|\mathbf{r}_t\|^2\Big),
\end{aligned} \tag{A29}$$

and

$$\frac{\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_\star\|^2\le\mu_\theta\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2+\mu_\theta\|\tilde{\mathbf{e}}_{t+1}\|^2+\frac{1}{2\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2. \tag{A30}$$

Substituting (A28)–(A30) into (A27) and by properly choosing η, we obtain

$$\begin{aligned}
&\big(B\mu_\theta+\zeta(\tau_t^2+2B\gamma_t^2)\big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+\big((1-A)\mu_\theta+2(2-A)\zeta\tau_t^2\big)\|\tilde{\mathbf{e}}_t\|^2 \\
&+(\mu_\theta+\frac{1}{\zeta})\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2+\frac{1}{2\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2+\frac{7}{4\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2 \\
&+\Big(\epsilon^2+2\tau_t^2+c_3B\Big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2 \\
&+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+2(2-A)\big(\mu_\theta^2+(\sigma_{\max}^{\mathbf{L}_s})^2\mu_\theta^2+2\gamma_t^2\big)\|\tilde{\mathbf{e}}_t\|^2\Big) \\
&+\eta_t\Big\{\frac{7}{\mu_\theta\sigma_{\min}^{+}}\Big(\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+4\sigma_{\max}^{\mathbf{L}_u}\mu_\theta^2\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2+\mu_\theta^2\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2\Big) \\
&+\frac{2}{\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2+\big((1-A)(4+\xi)+\frac{7(2-A)}{\mu_\theta\sigma_{\min}^{+}}(2\mu_\theta^2+4\gamma_t^2+\frac{\sigma_{\max}^{\mathbf{L}_s^2}\mu_z^2}{2})\big)\|\tilde{\mathbf{e}}_t\|^2 \\
&+\Big(B(4+\xi)+\frac{7}{\mu_\theta\sigma_{\min}^{+}}\big(\epsilon^2+2\tau_t^2+c_3B\big)\Big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2 \\
&+(\epsilon+4\mu_\theta+\sigma_{\max}^{\mathbf{L}_u}\mu_\theta)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_\star\|^2\Big\} \\
&\le 2\Theta_{t+1}+\|\nabla F(\tilde{\mathbf{x}}_{t+1})-\nabla F(\tilde{\mathbf{x}}_\star)\|^2+4\mu_\theta\|\tilde{\mathbf{z}}_{t+1}-\tilde{\mathbf{z}}_t\|^2 \\
&+\big(\epsilon+(\xi-c_1)B\big)\|\tilde{\mathbf{x}}_{t+1}-\tilde{\mathbf{x}}_t\|^2+\frac{3\mu_\theta}{4}\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\|^2+\frac{3}{4\mu_\theta}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2 \\
&+(\xi-c_2+(1-A)(\xi-c_1))\|\tilde{\mathbf{e}}_t\|^2.
\end{aligned} \tag{A31}$$

To make (A31) hold, η_t is chosen such that

$$\begin{aligned}
&\eta_t\Big(B(4+\xi)+\frac{7}{\mu_\theta\sigma_{\min}^{+}}(\epsilon^2+2\tau_t^2+B\Xi_3)\Big) \\
&\le \epsilon+(\xi-\Xi_1-\mu_\theta)B-\zeta(\tau_t^2+2B\gamma_t^2)-\frac{7}{4\mu_\theta\sigma_{\min}^{+}}(\epsilon^2+2\tau_t^2+B\Xi_3), \\
&\eta_t\Big((1-A)(4+\xi)+\frac{7}{\mu_\theta\sigma_{\min}^{+}}(2-A)\Xi_3\Big)
\end{aligned}$$

$$\leq \xi - \Xi_2 + (1-A)(\xi - \Xi_1 - \mu_\theta) - 2(2-A)\zeta\tau_t^2 - \frac{7}{4\mu_\theta\sigma_{\min}^+}(2-A)\Xi_3,$$

$$\eta_t\left(\epsilon + 4\mu_\theta + \sigma_{\max}^{\mathbf{L}_u}\mu_\theta\right) \leq \frac{2m_f M_f}{m_f + M_f} - \frac{1}{\zeta} - \mu_\theta, \quad \eta_t\frac{7}{\mu_\theta\sigma_{\min}^+} \leq \frac{2}{m_f + M_f} - \frac{7}{4\mu_\theta\sigma_{\min}^+},$$

$$\eta_t\frac{28\sigma_{\max}^{\mathbf{L}_u}\mu_\theta}{\sigma_{\min}^+} \leq 4\mu_\theta - \frac{7\sigma_{\max}^{\mathbf{L}_u}\mu_\theta}{\sigma_{\min}^+}, \quad \eta_t\frac{7\mu_\theta}{\sigma_{\min}^+} \leq \frac{3\mu_\theta}{4} - \frac{7\mu_\theta}{4\sigma_{\min}^+}, \quad \eta_t\frac{2}{\mu_\theta} \leq \frac{3}{8\mu_\theta},$$

which implies (23). Then, we can prove that (A22) holds. The proof is completed. □

References

1. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [CrossRef]
2. Yoo, S.J.; Park, B.S. Dynamic event-triggered prescribed-time consensus tracking of nonlinear time-delay multiagent systems by output feedback. *Fractal Fract.* **2024**, *8*, 545. [CrossRef]
3. Liu, H.J.; Shi, W.; Zhu, H. Distributed voltage control in distribution networks: Online and robust implementations. *IEEE Trans. Smart Grid* **2017**, *9*, 6106–6117. [CrossRef]
4. Molzahn, D.K.; Dorfler, F.; Sandberg, H.; Low, S.H.; Chakrabarti, S.; Baldick, R.; Lavaei, J. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Trans. Smart Grid* **2017**, *8*, 2941–2962. [CrossRef]
5. Liu, Y.F.; Chang, T.H.; Hong, M.; Wu, Z.; So, A.M.C.; Jorswieck, E.A.; Yu, W. A survey of recent advances in optimization methods for wireless communications. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 2992–3031. [CrossRef]
6. Huang, J.; Zhou, S.; Tu, H.; Yao, Y.; Liu, Q. Distributed optimization algorithm for multi-robot formation with virtual reference center. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 732–734. [CrossRef]
7. Yang, X.; Zhao, W.; Yuan, J.; Chen, T.; Zhang, C.; Wang, L. Distributed optimization for fractional-order multi-agent systems based on adaptive backstepping dynamic surface control technology. *Fractal Fract.* **2022**, *6*, 642. [CrossRef]
8. Liu, J.; Zhang, C. Distributed learning systems with first-order methods. *Found. Trends Databases* **2020**, *9*, 1–100. [CrossRef]
9. Nedic, A.; Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61. [CrossRef]
10. Nedic, A.; Olshevsky, A.; Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.* **2017**, *27*, 2597–2633. [CrossRef]
11. Xu, J.; Zhu, S.; Soh, Y.C.; Xie, L. Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Trans. Autom. Control* **2018**, *63*, 434–448. [CrossRef]
12. Wen, X.; Luan, L.; Qin, S. A continuous-time neurodynamic approach and its discretization for distributed convex optimization over multi-agent systems. *Neural Netw.* **2021**, *143*, 52–65. [CrossRef]
13. Feng, Z.; Xu, W.; Cao, J. Alternating inertial and overrelaxed algorithms for distributed generalized Nash equilibrium seeking in multi-player games. *Fractal Fract.* **2021**, *5*, 62. [CrossRef]
14. Che, K.; Yang, S. A snapshot gradient tracking for distributed optimization over digraphs. In Proceedings of the CAAI International Conference on Artificial Intelligence, Beijing, China, 27–28 August 2022; pp. 348–360.
15. Zhou, S.; Wei, Y.; Liang, S.; Cao, J. A gradient tracking protocol for optimization over Nabla fractional multi-agent systems. *IEEE Trans. Signal Inf. Process. Over Netw.* **2024**, *10*, 500–512. [CrossRef]
16. Shi, W.; Ling, Q.; Wu, G.; Yin, W. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.* **2015**, *25*, 944–966. [CrossRef]
17. Ling, Q.; Shi, W.; Wu, G.; Ribeiro, A. DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Trans. Signal Process.* **2015**, *63*, 4051–4064. [CrossRef]
18. Mokhtari, A.; Shi, W.; Ling, Q.; Ribeiro, A. DQM: Decentralized quadratically approximated alternating direction method of multipliers. *IEEE Trans. Signal Process.* **2016**, *64*, 5158–5173. [CrossRef]
19. Eisen, M.; Mokhtari, A.; Ribeiro, A. A primal-dual quasi-Newton method for exact consensus optimization. *IEEE Trans. Signal Process.* **2019**, *67*, 5983–5997. [CrossRef]
20. Mansoori, F.; Wei, E. A fast distributed asynchronous Newton-based optimization algorithm. *IEEE Trans. Autom. Control* **2019**, *65*, 2769–2784. [CrossRef]
21. Jiang, X.; Qin, S.; Xue, X.; Liu, X. A second-order accelerated neurodynamic approach for distributed convex optimization. *Neural Netw.* **2022**, *146*, 161–173. [CrossRef] [PubMed]
22. Li, Y.; Voulgaris, P.G.; Stipanović, D.M.; Freris, N.M. Communication efficient curvature aided primal-dual algorithms for decentralized optimization. *IEEE Trans. Autom. Control* **2023**, *68*, 6573–6588. [CrossRef]
23. Alistarh, D.; Grubic, D.; Li, J.Z.; Tomioka, R.; Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In Proceedings of the 30th NeurIPS, Long Beach, CA, USA, 4–9 December 2017; pp. 1710–1721.
24. Wangni, J.; Wang, J.; Liu, J.; Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In Proceedings of the 31st NeurIPS 2018, Montreal, QC, Canada, 2–8 December 2018; pp. 1306–1316.

25. Stich, S.U.; Cordonnier, J.B.; Jaggi, M. Sparsified SGD with memory. In Proceedings of the 31st NeurIPS 2018, Montreal, QC, Canada, 2–8 December 2018; pp. 4447–4458.
26. Doan, T.T.; Maguluri, S.T.; Romberg, J. Fast convergence rates of distributed subgradient methods with adaptive quantization. *IEEE Trans. Autom. Control* **2020**, *66*, 2191–2205. [CrossRef]
27. Taheri, H.; Mokhtari, A.; Hassni, H.; Pedarsani, R. Quantized decentralized stochastic learning over directed graphs. In Proceedings of the 37th ICML, Virtual, 13–18 July 2020; pp. 9324–9333.
28. Song, Z.; Shi, L.; Pu, S.; Yan, M. Compressed gradient tracking for decentralized optimization over general directed networks. *IEEE Trans. Signal Process.* **2022**, *70*, 1775–1787. [CrossRef]
29. Xiong, Y.; Wu, L.; You, K.; Xie, L. Quantized distributed gradient tracking algorithm with linear convergence in directed networks. *IEEE Trans. Autom. Control* **2022**, *68*, 5638–5645. [CrossRef]
30. Zhu, S.; Hong, M.; Chen, B. Quantized consensus ADMM for multi-agent distributed optimization. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4134–4138.
31. Elgabli, A.; Park, J.; Bedi, A.S.; Issaid, C.B.; Bennis, M.; Aggarwal, V. Q-GADMM: Quantized group ADMM for communication efficient decentralized machine learning. *IEEE Trans. Commun.* **2020**, *69*, 164–181. [CrossRef]
32. Li, W.; Liu, Y.; Tian, Z.; Ling, Q. Communication-censored linearized ADMM for decentralized consensus optimization. *IEEE Trans. Signal Inf. Process. Over Netw.* **2020**, *6*, 18–34. [CrossRef]
33. Gao, L.; Deng, S.; Li, H.; Li, C. An event-triggered approach for gradient tracking in consensus-based distributed optimization. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 510–523. [CrossRef]
34. Zhang, Z.; Yang, S.; Xu, W.; Di, K. Privacy-preserving distributed ADMM with event-triggered communication. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 2835–2847. [CrossRef]
35. Chen, T.; Giannakis, G.; Sun, T.; Yin, W. LAG: Lazily aggregated gradient for communication-efficient distributed learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 5050–5060.
36. Sun, J.; Chen, T.; Giannakis, G.; Yang, Z. Communication-efficient distributed learning via lazily aggregated quantized gradients. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3370–3380.
37. Singh, N.; Data, D.; George, J.; Diggavi, S. SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization. *IEEE Trans. Autom. Control* **2022**, *68*, 721–736. [CrossRef]
38. Yang, X.; Yuan, J.; Chen, T.; Yang, H. Distributed adaptive optimization algorithm for fractional high-order multiagent systems based on event-triggered strategy and input quantization. *Fractal Fract.* **2023**, *7*, 749. [CrossRef]
39. Zhang, Z.; Yang, S.; Xu, W. Decentralized ADMM with compressed and event-triggered communication. *Neural Netw.* **2023**, *165*, 472–482. [CrossRef]
40. Richtárik, P.; Sokolov, I.; Fatkhullin, I. EF21: A new, simpler, theoretically better, and practically faster error feedback. In Proceedings of the 34th NeurIPS, Virtual, 6–14 December 2021; pp. 4384–4396.
41. Richtarik, P.; Sokolov, I.; Fatkhullin, I.; Gasanov, E.; Li, Z.; Gorbunov, E. 3PC: Three point compressors for communication-efficient distributed training and a better theory for Lazy aggregation. In Proceedings of the 39th ICML, Baltimore, MD, USA, 17–23 July 2022; pp. 18596–18648.
42. Shi, W.; Ling, Q.; Wu, G.; Yin, W. A proximal gradient algorithm for decentralized composite optimization. *IEEE Trans. Signal Process.* **2015**, *63*, 6013–6023. [CrossRef]
43. Alghunaim, S.; Yuan, K.; Sayed, A.H. A linearly convergent proximal gradient algorithm for decentralized optimization. In Proceedings of the 32nd NeurIPS, Vancouver, BC, Canada, 8–14 December 2019.
44. Guo, L.; Shi, X.; Yang, S.; Cao, J. DISA: A dual inexact splitting algorithm for distributed convex composite optimization. *IEEE Trans. Autom. Control* **2024**, *69*, 2995–3010. [CrossRef]
45. Li, W.; Liu, Y.; Tian, Z.; Ling, Q. COLA: Communication-censored linearized ADMM for decentralized consensus optimization. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 5237–5241.