A Sharp Convergence Rate Analysis for Distributed Accelerated Gradient Methods

Huan Li, Cong Fang, Wotao Yin, Zhouchen Lin, Fellow;

Abstract—In this paper, we study the computation and communication costs in decentralized distributed optimization and give a sharp complexity analysis for the proposed distributed accelerated gradient methods. We present two algorithms based on the framework of the accelerated penalty method with increasing penalty parameters. Our first algorithm achieves the $O\left(\sqrt{1 - \sigma_2(W)}\right)$ complexities for both computation and communication, which match the communication complexity lower bound for non-smooth distributed optimization, where $\sigma_2(W)$ denotes the second largest singular value of the weight matrix $W$ associated to the network. Our second algorithm employs a double-loop and obtains the near optimal $O\left(\sqrt{L/\epsilon(1 - \sigma_2(W))}\right)$ communication complexity and the optimal $O\left(\sqrt{L/\epsilon}\right)$ computation complexity for $L$-smooth distributed optimization. When the problem is $\mu$-strongly convex and $L$-smooth, our second algorithm has the near optimal $O\left(\sqrt{L/\mu(1 - \sigma_2(W)))\log^2 \epsilon^{-1}}\right)$ complexity for communication and the optimal $O\left(\sqrt{L/\mu\log \epsilon^{-1}}\right)$ computation complexity. Our communication complexities are only worse by a factor of $\log \epsilon^{-1}$ than the lower bounds for the smooth distributed optimization.

Index Terms—Distributed accelerated gradient algorithms, accelerated penalty method, optimal computation complexity, near optimal communication complexity.

I. INTRODUCTION

In this paper, we consider the decentralized distributed optimization, where $m$ agents form a connected and undirected network $G = (V, E)$ and cooperatively solve the convex problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} f_i(x),$$  \hfill (1)

where $V = \{1, 2, \cdots, m\}$ is the set of agents, $f_i$ is the local objective function only available to agent $i$ and $E \subset V \times V$ is the set of edges in the network. We consider the distributed algorithms using only local computation and communication, i.e., each agent $i$ makes his decision only based on the local function $f_i$ and the information obtained from his neighbors. A pair of agents can exchange information if and only if they are connected in the network $G$.

Decentralized distributed computation has been widely used in automatic control, signal processing and machine learning.

Among the classical decentralized first order algorithms, two different types of methods have been proposed, namely, the dual based methods and primal-only methods. Typical examples of the first type of methods include the dual subgradient ascent [1], dual gradient ascent and its accelerated version [2], [3], the primal-dual method [4], [5], [6], [7] and ADMM [8], [9], [10], [11], [12], [13]. Examples of the primal-only methods include the distributed subgradient/gradient method [14], [15], [16], [17], [18], the augmented distributed gradient method [19], [20], [21], the distributed accelerated gradient method [22], [23], [24] and EXTRA [25], [26]. In general, most dual based methods require the evaluation of Fenchel conjugate of the local objective function $f_i$ and thus have a larger computational cost per-iteration than the primal-only algorithms. Among the primal-only algorithms, the distributed accelerated gradient method [22] obtains the optimal computation complexity and thus seems to have greater potential in theory.

A. Proposed Algorithms

In this paper, we study the distributed accelerated gradient method and give a sharp complexity analysis for both computation and communication. Before briefly introducing the proposed algorithms, we describe the notations and assumptions. Let $x_i \in \mathbb{R}^n$ be the local copy of the variable $x$ for agent $i$ and we introduce the aggregate variable $x$, aggregate objective function $f(x)$ and aggregate gradient $\nabla f(x)$ as

$$x = \begin{pmatrix} x_1^T \\ \vdots \\ x_m^T \end{pmatrix}, \quad f(x) = \sum_{i=1}^{m} f_i(x_i), \quad \nabla f(x) = \begin{pmatrix} \nabla f_1(x_1)^T \\ \vdots \\ \nabla f_m(x_m)^T \end{pmatrix},$$

where $x \in \mathbb{R}^{mn \times n}$. We use $\| \cdot \|$ as the $l_2$ Euclidean norm for a vector, $\| \cdot \|_F$ as the Frobenius norm and $\| \cdot \|_2$ as the spectral norm for a matrix. Define $1 = (1, 1, \cdots, 1)^T \in \mathbb{R}^m$ to be the vector with all ones and $I \in \mathbb{R}^{m \times m}$ to be the identity matrix. Define

$$\Pi = I - \frac{1}{m} 11^T,$$  \hfill (2)

$$U = \sqrt{I - W},$$  \hfill (3)

and

$$\alpha(x) = \frac{1}{m} \sum_{i=1}^{m} x_i$$  \hfill (4)

as the average across the rows of $x$.

We make the following assumptions for the aggregate function:

Assumption 1:
Algorithm 1 APM

Initialize $x^0 = x^1$.
for $k = 0, 1, 2, \ldots, K$ do
  $y^k = x^k + \frac{\theta_k}{\theta_{k+1}}(x^k - x^{k-1})$,
  $x^{k+1} = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{\theta_k}{2\theta_{k+1}} \langle \nabla f(y^k), x - y^k \rangle + \frac{\theta_k}{2} \|x - y^k\|^2_{\frac{\theta_k}{\theta_{k+1}}}$.
end for

1) $f(x)$ is convex: $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2_{\frac{1}{\theta}}$. Specially, $\mu$ can be zero. If $\mu > 0$, then $f(x)$ is strongly convex.
2) $f(x)$ is $L$-smooth: $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2_{\theta}$.

Assume that the set of minimizers is non-empty. Denote $x^*$ as one minimizer of problem (1) and $x^* = (x^*)^T$. We make the following assumptions for the weight matrix $W$ associated to the network:

**Assumption 2:**

1) $W \in \mathbb{R}^{n \times n}$ is a symmetric matrix with $W_{i,j} \neq 0$ if and only if agents $i$ and $j$ are neighbors or $i = j$. Otherwise, $W_{i,j} = 0$.
2) $I \preceq W \preceq 0$, $W^1 = 1$, and $W^T = W^T$.
3) $\sigma_2(W) < 1$, where $1 = \sigma_1(W) > \sigma_2(W) \geq \cdots \geq \sigma_m(W)$ are the singular values of $W$.

Examples satisfying Assumption 2 include: 1, $W = \frac{1+\beta}{\beta} L$ where $L$ is the Laplacian matrix. 2, $W = \frac{1+\beta}{\beta} I - \frac{T}{\sigma_2(L)}$ where $M$ is the Metropolis weight matrix [27] with $M_{i,j} = \begin{cases} 1/(1 + \max \|d_i, d_j\|), & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{if } (i, j) \notin \mathcal{E} \text{ and } i \neq j, \end{cases}$ where $\mathcal{N}_i$ is the neighborhood of agent $i$ and $d_i$ is its degree, i.e., the number of its immediate neighbors.

Now we are ready to present the proposed algorithms. The first algorithm solves a more general problem of

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} f_i(x), \quad \text{where } f_i(x) = f_i(x) + h_i(x).$$

and is described in Algorithm 1, where $h(x) = \sum_{i=1}^{m} h_i(x_i)$ is convex (not necessarily strongly convex or differentiable) and we specify $\theta_k = \theta_k = 1 + \frac{\mu}{\rho_k+1}$ and $\theta_0 = 0$. For many families of undirected graph, we can give order-accurate estimate on $1 - \sigma_2(W)$ [28, Proposition 5]. Since $U^2 = I - W$, we know that computing $U^T y^k$ involves sending messages to adjacent agents. The remaining operations in Algorithm 1 are local computation of each agent. Therefore, Algorithm 1 is a decentralized algorithm.

Our second algorithm solves problem (1), which ignores $h(x)$ in (5). It has double loops and is described in Algorithm 2. The inner loop improves the consensus of $x^k$ by multiplying it with $W$ for $T_k$ times, which requires sending messages to adjacent agents $T_k$ times. Specifically, we use the accelerated average consensus [29] in the inner loop. The outer loop let the agents perform local computation. In Algorithm 2, $\beta_0$ is suggested to be a large constant and $\theta_k$, $\delta_k$ and $T_k$ are given for two regularity cases:

1) $\mu = 0$ ($f(x)$ is non-strongly convex). Then $\delta_k = \theta_k^2$.

$$T_k = O \left( \frac{\log k}{\sqrt{1 - \sigma_2(W)}} \right)$$

and $\theta_0 = 1$. 2) $\mu > 0$ ($f(x)$ is $\mu$-strongly convex). Then $\theta_k = \sqrt{2}, \forall k$.

$\delta_k = (1 - \sqrt{2})^{k+1}$ and $T_k = O \left( \frac{k\sqrt{2}}{1 - \sigma_2(W)} \right)$.

B. Complexities

We study both computation and communication complexities and they are presented as the number of computation or communication to find an $\epsilon$-optimal solution $x$ such that $\frac{1}{m} \sum_{i=1}^{m} f_i(\alpha(x)) - \frac{1}{m} \sum_{i=1}^{m} f_i(x^*) \leq \epsilon$ for problem (5) or $\frac{1}{m} \sum_{i=1}^{m} f_i(\alpha(x)) - \frac{1}{m} \sum_{i=1}^{m} f_i(x^*) \leq \epsilon$ for problem (1). We define one communication to be one application of $Wx$, i.e., the agents receive information from all their neighbors once. One computation is defined to be the cost of computing $\nabla f_i(x_i)$ once for all agents. The computation and communication complexities are identical for Algorithm 1 and thus we do not distinguish them. However, because of the inner loop, they are different for Algorithm 2.

Under Assumptions 1 and 2, we establish the complexity of $O \left( \frac{1}{\frac{\log k}{\sqrt{1 - \sigma_2(W)}}} \right)$ complexity for Algorithm 1 with $\mu = 0$, which matches the communication complexity lower bound for the non-smooth distributed optimization [5]. For Algorithm 2, the complexities are given for two cases. When $\mu = 0$, we establish the near optimal communication complexity of $\tilde{O} \left( \frac{L}{\sqrt{1 - \sigma_2(W)}} \log \frac{2}{\epsilon} \right)$ and the optimal computation complexity of $\tilde{O} \left( \frac{L}{\sqrt{2}} \log \frac{2}{\epsilon} \right)$. When $\mu > 0$, we establish the near optimal $\tilde{O} \left( \frac{L}{\mu(1 - \sigma_2(W)) \log \frac{2}{\epsilon}} \right)$ communication complexity and the optimal $\tilde{O} \left( \frac{L}{\mu(1 - \sigma_2(W)) \log \frac{2}{\epsilon}} \right)$ computation complexity.

Between Algorithms 1 and 2, the former is simpler and works for more general problems. But the latter has better complexities. They are improvements over two algorithms of D-NG and D-NC in [22]. Algorithms 1 and D-NG both use Nesterov’s acceleration technique, have a single loop and compute $\nabla f(x)$ and apply $W$ in once each iteration, but Algorithm 1 uses new parameters. Also, Algorithms 1 handles a possible
Complexity of gradient computation

| Methods                                      | Complexity of gradient computation | Complexity of communication |
|----------------------------------------------|-----------------------------------|----------------------------|
| Accelerated Distributed Nesterov Gradient Descent | $O\left(\frac{1}{\sqrt{\tau}}\right)$ [24] | $O\left(\frac{1}{\sqrt{\tau}}\right)^2$ [24] |
| Accelerated Dual Ascent                      | $O\left(\frac{L}{\sqrt{1 - \sigma_2(W)}}\log^2 \frac{1}{\tau}\right)$ [3] | $O\left(\sqrt{\frac{L}{\mu(1 - \sigma_2(W))}}\log^2 \frac{1}{\tau}\right)$ [3] |
| Our APM-C                                    | $O\left(\frac{L}{\mu\log \frac{1}{\tau}}\right)$ | $O\left(\sqrt{\frac{L}{\mu(1 - \sigma_2(W))}}\log^2 \frac{1}{\tau}\right)$ |
| Lower Bound                                  | $O\left(\frac{L}{\mu\log \frac{1}{\tau}}\right)$ [30] | $O\left(\sqrt{\frac{L}{\mu(1 - \sigma_2(W))}}\log^2 \frac{1}{\tau}\right)$ [2] |

Complexity comparisons between the accelerated dual ascent, accelerated distributed Nesterov gradient descent and Algorithm 2 for smooth convex problems.

Algorithm 2 uses the accelerated average consensus while Algorithms 1 and 2 and their parameters are well motivated. Their complexity is better than the result of D-NC, which is

$$O\left(\frac{L}{\mu\sqrt{1 - \sigma_2(W)}}\log^2 \frac{1}{\tau}\right)$$

Thus our communication complexity is better than the result of D-NC, which is

$$O\left(\frac{L}{\mu\log \frac{1}{\tau}}\right)$$

Moreover, Algorithm 2 can solve both the strongly convex and non-strongly convex problems. Algorithms 1 and 2 and their parameters are well motivated by a constraint-penalty approach.

Let us compare the complexities of Algorithm 2 to the state-of-art decentralized optimization algorithms, namely, the accelerated distributed Nesterov gradient descent and accelerated dual ascent algorithm, as well as the complexity lower bounds. They are summarized in Table I. Our complexities match the lower bounds except that the communication ones have an extra factor of $\log \frac{1}{\tau}$. The communication complexity of [3] matches ours for $\mu = 0$ and is optimal for $\mu > 0$ (thus better than ours by $\log \frac{1}{\tau}$). On the other hand, our computation complexities match the lower bounds and they are better than those in [3], [24].

C. Prior Art

We review prior literature that we group in two categories of primal-only algorithms and dual based algorithms.

1) Primal-only Algorithms: The early work of the distributed subgradient method can be found in [14], while its stochastic version in [15] and asynchronous variant in [16].

2) Dual Based Algorithms: The dual based methods introduce the Lagrangian function and work in the dual space. Many classical methods can be used to solve the dual problem, e.g., the subgradient method [1] and the accelerated gradient method [2], [3]. The primal-dual method [4], [5] and ADMM [8], [9], [10], [11], [12], [13] solve the reformulated problem with liner constraint. Generally, the dual based methods are computation-efficient. Their computation cost can be reduced by introducing the augmented Lagrangian function with a special weighted norm [6], [21]. Moreover, [6] and [21] established the connection of EXTRA [25] and DIGing [21] with the primal-dual approaches, respectively. Specifically, EXTRA and DIGing can be explained by using the basic gradient method in the Gauss-Seidel-like order for computing the saddle of the augmented Lagrangian function. [4], [5] studied the communication-efficient primal-dual method for the nonsmooth problems. Specifically, they used the classical primal-dual method [32] in the outer iteration and the subgradient
method in the inner iteration to compute the proximal mapping. Chebyshev acceleration [33] is used to reduce the computation complexity. The subgradient based algorithm is beyond the scope of this paper.

D. Paper Organization

Section II develops the accelerated penalty method with increasing penalty parameters for the decentralized distributed optimization problem. Specifically, Section II-A studies Algorithm 1 while Section II-B studies Algorithm 2. Section III establishes the complexities for Algorithms 1 and 2 while Section IV provides simulation experiments to verify the efficiency of the proposed algorithms. Finally, we conclude in Section V with future work.

II. DEVELOPMENT OF THE ACCELERATED PENALTY METHOD

A. Accelerated Penalty Method

In this section, we consider problem (5) with the assumption that the proximal mapping of \( h_i(x) \), i.e., \( \text{argmin}_{x \in \mathbb{R}} h_i(x) + \frac{1}{2}\|x - x^*\|^2 \), has closed form solution or can be easily computed. We rewrite the problem as:

\[
\min_{x_1=x_2=\cdots=x_m \in \mathbb{R}^n} \sum_{i=1}^m (f_i(x_i) + h_i(x_i)).
\]

From Assumption 2 and the definition in (3) we know that \( I \geq \mathbf{U} \geq 0 \) and \( x_1 = \cdots = x_m \) is equivalent to \( \mathbf{U}x = 0 \). Thus we can reformulate the problem as:

\[
\min_{x \in \mathbb{R}^{m \times n}} F(x) = f(x) + h(x), \quad \text{s.t.} \quad \mathbf{U}x = 0.
\]

A lot of literatures reformulate the decentralized consensus problem as problem (7) [2], [5], [3], [4] and many algorithms can be used to solve it, e.g., the primal-dual method [4], [5], [6], [7] and dual ascent [1], [2], [3]. In this paper, we follow [18] to use the penalty method to solve problem (7). Specifically, the penalty method solves the following problem instead of [18, Equ. (7)]:

\[
\min_{x \in \mathbb{R}^{m \times n}} f(x) + h(x) + \frac{\beta}{2} \|\mathbf{U}x\|^2_F,
\]

where \( \beta \) is a large constant. However, one big issue of the penalty method is that problems (7) and (8) are not equivalent for finite \( \beta \). When solving problem (8), we can only obtain an approximate solution of (7) with small \( \|\mathbf{U}x\|_F \), rather than \( \|\mathbf{U}x\|_F \to 0 \), and the algorithm only converges to a neighborhood of the solution set [18]. Moreover, to find an \( \epsilon \)-optimal solution of (7), we need to pre-define a large \( \beta \) in the order of \( \frac{1}{\epsilon^2} \) [18]. Thus, the parameter setting depends on the precision \( \epsilon \). To solve these problems, we use the gradually increasing penalty parameters to make the solution of (8) approximate that of (7), i.e., at the \( k \)-th iteration, we use \( \beta = \frac{\theta_k}{\theta_k} \) with diminishing \( \theta_k \) to 0.

We use the standard forward-backward operation with extrapolation [34] to minimize the penalized problem (8), i.e., at the \( k \)-th iteration, we first compute the gradient of \( f(x) + \frac{\beta_k}{2} \|\mathbf{U}x\|^2_F \) at some extrapolated point and then compute the proximal mapping of \( h(x) \), which leads to Algorithm 1 named as the Accelerated Penalty Method (APM).

Consider the simple case with \( h(x) = 0 \) and \( \frac{\beta_k}{\theta_k} = \frac{k+1}{c} \), then the second step of Algorithm 1 becomes

\[
x^{k+1} = \frac{L_y^k + (k+1)Wy^k/c}{L + (k+1)/c} - \frac{\nabla f(y^k)}{L + (k+1)/c}.
\]

Thus, when \( (k+1)/c \gg L \), we have \( x^{k+1} \approx \frac{Wy^k - \beta_k}{
\frac{c}{k+1} \nabla f(y^k)} \) and it approximates the D-NG [22]. It gives a different explanation of the D-NG from the perspective of the accelerated penalty method with increasing penalty parameters.

Introduce the saddle point problem \( \min_x \max_{\lambda} F(x) + \langle \lambda, \mathbf{U}x \rangle \) and let \( (\ast, \lambda^\ast) \) be a KKT point. Then we can describe the convergence rate for Algorithm 1 in Theorem 1.

**Theorem 1:** Assume that Assumptions 1 and 2 hold with \( \mu = 0 \) and \( h(x) \) is convex. Let sequences \( \{\theta_k\} \) and \( \{\vartheta_k\} \) satisfy \( \theta_0 = 1 \), \( \frac{\theta_k}{\theta_{k+1}} = \frac{1}{\theta_{k-1}} \) and \( \vartheta_k = \theta_k \). Then, for Algorithm 1, we have

\[
F(x^{K+1}) - F(\ast) \leq \frac{C_1}{K+1} + \|\lambda^\ast\|_F \|\mathbf{U}x^{K+1}\|_F,
\]

\[
\|\mathbf{U}x^{K+1}\|_F \leq \frac{1}{\beta_0(K+1)} \left( \sqrt{2\beta_0 C_1} + \|\lambda^\ast\|_F \right),
\]

where \( C_1 = \frac{1}{2\beta_0} \|\lambda^\ast\|_F + \left( \frac{L}{\beta_0} + \frac{\mu}{\beta_0} \right) \|x^0 - x^\ast\|^2_F \).

**Theorem 1** establishes the complexity for an \( \epsilon \)-optimal solution of problem (7). In the following Corollary, we establish how to transform an \( \epsilon \)-optimal solution of problem (7) to an \( \epsilon \)-optimal solution of problem (5). Let \( \|x^0 - x^\ast\| \leq R, \forall i \) and assume \( R \geq 1 \) for simplicity. Then we have

**Corollary 1:** Assume that each \( F_i(x) \) is \( M \)-Lipschitz continuous, i.e., \( |F_i(x) - F_i(y)| \leq M||x - y|| \). Then, under the settings in Theorem 1 with \( \beta_0 = \frac{\max\{L, M\}}{\sqrt{1 - \sigma_2(W)}} \), we have

\[
\frac{1}{m} \sum_{i=1}^m F_i(a(x^{K+1})) - \frac{1}{m} \sum_{i=1}^m F_i(x^\ast) \leq O \left( \frac{\beta_0 R^2}{K+1} \right).
\]

When the network parameter of \( \sigma_2(W) \) is unknown, we can set \( \beta_0 \) to be any constant and the complexity becomes \( O \left( \frac{1}{C(1 - \sigma_2(W))} \right) \). Accordingly, the complexity of D-NG [22] becomes \( O \left( \frac{1}{C(1 - \sigma_2(W))} \right) \log \frac{1}{\epsilon} \). For any directed connected graph, \( \sigma_2(W) = O(m^2) \) in the worst case [28, Proposition 5]. Thus Corollary 1 establishes the \( O \left( \frac{1}{C} \right) \) complexity for any graph, which scales linearly in the number of agents and improves the result of \( O \left( \frac{1}{C} \right) \) in [35].

When \( f(x) \) is strongly convex, we can establish a faster \( O \left( \frac{1}{C^2} \right) \) convergence rate for Algorithm 1 with \( \theta_k = \frac{2}{k+2} \) and \( \vartheta_k = \theta_k^2 \). However, the quickly diminishing step-size makes the algorithm slow in practice. So we omit the discussion for the strongly convex case.

B. Accelerated Penalty Method with Consensus

In this section, we consider problem (6) with \( h(x) = 0 \). Different from the reformulation in (7), we use the constraint of \( \Pi x = 0 \), rather than \( \mathbf{U}x = 0 \), because the proximal
mapping of $\|\Pi x\|^2_F$ has a closed form solution, which we use to develop Algorithm 2. Specifically, we reformulate the problem as

$$\min_{x \in \mathbb{R}^{m \times n}} f(x), \quad s.t. \quad \Pi x = 0$$

and solve the following penalized problem instead:

$$\min_{x \in \mathbb{R}^{m \times n}} f(x) + \frac{\beta}{2} \|\Pi x\|^2_F.$$  \hspace{1cm} (10)

Due to the same reason as explained in Section II-A, we also use the increasing penalty parameters of $\beta = \frac{\beta_0}{m}$ and the standard forward-backward operation with extrapolation. However, we inexactly compute the proximal operation of $\frac{\partial}{\partial x} \|\Pi x\|^2_F$ at the $k$-th iteration this time. Specifically, the algorithm framework consists of the following steps:

$$y^k = x^k + \frac{L \theta_k - \mu}{L - \mu} \frac{1}{\theta_k} (x^k - x^{k-1}),$$

$$z^k = y^k - \frac{1}{L} \nabla f(y^k),$$

$$x^{k+1} \approx \arg\min_{x \in \mathbb{R}^{m \times n}} \frac{\beta}{2} \|\Pi x\|^2_F + \frac{L}{2} \|x - z^k\|^2_F,$$  \hspace{1cm} (11c)

where the sequences of $\{\theta_k\}$ and $\{\theta_k\}$ and the precision in step (11c) will be specified in Theorems 2 and 3. Now we consider how to solve the subproblem in procedure (11c). Due to the special form of $\Pi$, we know that the exact solution of subproblem in (11c) is $\frac{L \theta_k + \beta_0}{L \theta_k + \beta_0} \alpha(x^k)^T$, where $\alpha(x)$ is defined in (4). Thus we only need to approximate $\alpha(z^k)$, which can be obtained by the classical average consensus [36] and accelerated average consensus [29]. The following Lemma establishes the required approximate precision for an $\varepsilon_k$-optimal solution of the subproblem (11c).

**Lemma 1:** If $x^{k+1} = \frac{L \theta_k + \beta_0}{L \theta_k + \beta_0} \alpha(x^k)^T$ and $\|z^k, T_k + 1 - \frac{1}{m} 1^T z^k \|^2_F \leq 2 \frac{\beta_0}{\theta_k}$, then we have

$$\frac{L}{2} \|x^{k+1} - z^k\|^2_F + \frac{\beta_0}{2 \theta_k} \|\Pi x^{k+1}\|^2_F$$

$$\leq \min_x \frac{L}{2} \|x - z^k\|^2_F + \frac{\beta_0}{2 \theta_k} \|\Pi x\|^2_F + \varepsilon_k,$$  \hspace{1cm} (12)

where $z^{k+1}$ can be obtained by any average consensus procedure.

We only consider the accelerated average consensus and name Algorithm 2 as the Accelerated Penalty Method with Consensus (APM-C). The convergence of Algorithm 2 is established in the following theorem for strongly convex problems.

**Theorem 2:** Assume that Assumptions 1 and 2 hold with $\mu > 0$, Setting $\theta_k = \theta = \sqrt{\frac{\mu}{\epsilon}}$, then $\theta_k = (1 - \theta)^{k+1}$ and $T_k = O \left( \frac{k \sqrt{\frac{\mu}{\epsilon}}}{\sqrt{1 - \sigma_2(W)}} \right)$. Then, Algorithm 2 needs $O \left( \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$ gradient computations and $O \left( \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$ communications to achieve an $\epsilon$-optimal solution such that $\frac{1}{m} \sum_{i=1}^{m} f_i(x^*) \leq \epsilon$. When we drop the strong-convexity assumption, we can have the following theorem.

**Theorem 3:** Assume that Assumptions 1 and 2 hold with $\mu = 0$. Let sequences $\{\theta_k\}$ and $\{\theta_k\}$ satisfy $\theta_0 = 1$, $\frac{1}{\sigma_2(W)} = \frac{1}{\sigma_2(W)}$ and $\theta_k = \theta_0^2$. Setting $T_k = O \left( \frac{\log k}{\sqrt{1 - \sigma_2(W)}} \right)$ and $\beta_0 \geq L \|\nabla f(x^*)\|^2_F$. Then, Algorithm 2 needs $O \left( \frac{L}{\mu} \right)$ gradient computations and $O \left( \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$ communications to achieve an $\epsilon$-optimal solution such that $\frac{1}{m} \sum_{i=1}^{m} f_i(x^*) \leq \epsilon$.

Theorems 2 and 3 avoid the assumption that $\nabla f(x)$ is bounded for all $x$, and, which is required in [22]. In fact, we only need the boundedness of $\nabla f(x^*)$, which is satisfied in general. In the worst case of $\frac{1}{m} \sum_{i=1}^{m} f_i(x) = O(m^2)$, the communication complexities of Algorithm 2 also scale linearly in the total number of agents $m$.

**Difference from the classical penalty method.** To the best of our knowledge, most traditional work analyzed the penalty method with a fixed penalty parameter [37], [38]. Authors in [37], [38] also studied the adaptive penalty method with an increasing sequence of penalty parameters $\{\beta_0, \beta_1, \ldots\}$. However, at each outer iteration, they solved the subproblem accurately, which is not the case with Algorithm 2. Theoretical advantage of our strategy is that we can guarantee the linear convergence when minimizing the objective, they compute the proximal mapping of $\frac{\beta_0}{\theta_k} \|\Pi x\|^2_F$ inexact, which is a few iterations of some first order method. The $O \left( \frac{1}{\epsilon} \right)$ complexity is proved in [37] for smooth problems. When using the regularization strategy, the complexity can be improved to $O \left( \frac{1}{\epsilon \log \frac{1}{\epsilon}} \right)$ [37]. For the smooth and strongly convex objective, the $O \left( \frac{1}{\epsilon \log \frac{1}{\epsilon}} \right)$ complexity can be proved [37, Equ. (50)]. Only these sublinear rates are given in [37], [38].

Different from the adaptive strategy in [37], [38], our accelerated penalty method increases the penalty parameter at each iteration of the algorithm. Algorithm 1 linearizes the smooth terms and thus avoids the inner loop. The $O \left( \frac{1}{\epsilon^2} \right)$ complexity of Algorithm 1 improves the $O \left( \frac{1}{\epsilon} \log \frac{1}{\epsilon} \right)$ one in [37]. As for Algorithm 2, the theoretical advantage of our strategy is that we can guarantee the linear convergence when minimizing $\frac{\beta_0}{\theta_k} \|\Pi x\|^2_F + \frac{L}{2} \|x - z^k\|^2_F$ in the inner loop even if $\frac{\beta_0}{\theta_k}$ is huge, and thus avoid the problem of ill-conditioning. However, this is not the case when minimizing $f(x) + \frac{\beta_0}{\theta_k} \|\Pi x\|^2_F$ in [37] due to its ill-conditioning, even if $f(x)$ is $\mu$-strongly convex. Specifically, when solving the subproblem, our strategy needs $O \left( \log \frac{\beta_0}{\theta_k \epsilon} \right)$ inner iterations to obtain an $\varepsilon_k$-optimal solution. The strategy in [37] needs $O \left( \frac{1}{\sqrt{\frac{\beta_0}{\theta_k \epsilon}}} \right)$ inner iterations.

**Difference from the classical accelerated first order algorithms.** We extend the classical accelerated gradient method [39], [40], [34], [41], [42] from the unconstrained problems to solve the linearly constrained problems via the perspective of the penalty method. However, since we use the increasing penalty parameters at each iteration, i.e., the penalized objective varies at different iterations, the conclusion in [34], [41], [42] for the unconstrained problems cannot be directly used for procedure (11a)-(11c). The increasing penalty parameters make the convergence analysis more challenging.

Our strategy also differs from the accelerated dual ascent [2], accelerated ADMM [43] and accelerated primal-dual method [32]. Although these three methods need $O \left( \frac{L}{\mu} \log \frac{1}{\epsilon} \right)$ iterations (ignoring the dependence on $1 - \sigma_2(W)$) for an $\epsilon$-optimal solution, they compute the proximal mapping of $f(x)$ or the
gradient of its Fenchel conjugate at each iteration, rather than the gradient of \( f(x) \), which needs an inner loop or require \( f(x) \) to have a special structure. The authors in [44], [45] developed the accelerated linearized augmented Lagrangian method with the complexity of \( O \left( \frac{1}{\epsilon^2} \right) \) for the general linearly constrained problems. They linearized \( f(x) \), but not the augmented term. Thus their methods also need multi-consensus when applied to the distributed setting. The augmented Lagrangian method is a different framework from our algorithms. Moreover, only sublinear rates are studied in [44], [45]. Many algorithms based on the proximal method of multipliers can be used to solve the non-smooth problem (7) with the \( O \left( \frac{1}{\epsilon} \right) \) computation and communication complexities, please see [46] for a unified introduction. In this paper, we study the penalty method to give a new perspective of D-NG and simplify the proof in [22], although the same \( O \left( \frac{1}{\epsilon} \right) \) complexities can be obtained by other algorithms.

III. Complexity Analysis

Before providing a comprehensive convergence analysis for Algorithms 1 and 2, we first present some useful technical lemmas.

**Lemma 2:** For any \( x, y, z, w \in \mathbb{R}^{m \times n} \), we have the following two identities:

\[
\begin{align*}
2(x - z, y - z) &= \|x - z\|_F^2 + \|y - z\|_F^2 - \|x - y\|_F^2, \\
2(x - z, y - w) &= \|y - z\|_F^2 - \|w - z\|_F^2 + \|x - w\|_F^2 - \|x - y\|_F^2.
\end{align*}
\]

In the following Lemma, we bound the Lagrange multiplier, which is useful in the distributed optimization community.

**Lemma 3:** Assume that Assumptions 1 and 2 hold, \( h(x) \) is convex and each \( F_i(x) \) is \( M \)-Lipschitz continuous. Then, we have the following properties:

1. If \((x^*, \lambda^*)\) is a KKT point of the saddle point problem \( \min_{x} \max_{\lambda} F(x) + \langle \lambda, \Pi x \rangle \), then we have \( \|\lambda^*\|_F \leq \|\nabla F(x^*)\|_F \).

2. If \((x^*, \lambda^*)\) is a KKT point of the saddle point problem \( \min_{x} \max_{\lambda} F(x) + \langle \lambda, Ux \rangle \), then we have \( \|\lambda^*\|_F \leq \frac{\|\nabla F(x^*)\|_F}{\sqrt{1 - \sigma_2(W)}} \).

3. If \((x^*, \lambda^*)\) is a KKT point of the saddle point problem \( \min_{x} \max_{\lambda} F(x) + \langle \lambda, Ux \rangle \), then we have \( \|\lambda^*\|_F \leq \frac{\|\nabla F(x^*)\|_F}{\sqrt{mM}} \).

The proof can be found in [4, Theorem 3]. The following Lemma can be used to analyze the Lagrangian function.

**Lemma 4:** If \( F(x) \) is convex and \((x^*, \lambda^*)\) is a KKT point of the saddle point problem \( \min_{x} \max_{\lambda} F(x) + \langle \lambda, Ax \rangle \), then we have \( F(x) - F(x^*) + \langle \lambda^*, Ax \rangle \geq 0, \forall x \).

The following Lemma bounds the consensus violation of \( \|\Pi x\|_F \) from \( \|Ux\|_F \).

**Lemma 5:** Assume that Assumption 2 holds. Then, we have \( \|\Pi x\|_F \leq \frac{1}{\sqrt{1 - \sigma_2(W)}} \|Ux\|_F \).

**Proof 1:** From Assumption 2, we know \( U1 = 0, U = U^T \) and \( \text{rank}(U) = m - 1 \). For any \( x \in \mathbb{R}^{m \times n} \), denote \( \Xi = x - \frac{1}{m} 11^T x \). Since \( 1^T \Xi = 0 \), we know \( \Xi \) is orthogonal to the null space of \( U \) and thus it belongs to the row (i.e., column) space of \( U \). Let \( UV^T = U \) be its economical SVD with \( V \in \mathbb{R}^{m \times (m - 1)} \). Then we can have

\[
\|UX\|_F^2 = \|UX\|_F^2 = \sum_{i=1}^{n} \xi_i^T U^T \xi_i = \sum_{i=1}^{n} (V^T \Xi, i)^T \Sigma^2 (V^T \Xi, i) \\
\geq (1 - \sigma_2(W)) \sum_{i=1}^{n} \|V^T \Xi, i\|_F^2 = (1 - \sigma_2(W)) \|V^T \Xi\|_F^2 \\
= (1 - \sigma_2(W)) \|\Xi\|_F^2 = (1 - \sigma_2(W)) \|\Pi x\|_F^2,
\]

where we denote \( \Xi, i \) to be the \( i \)-th column of \( \Xi \).

At last, we present the following Lemma, which can be used to analyze the algorithms with inexact subproblem computation.

**Lemma 6:** [42] Assume that \( (s_k) \) is a sequence with increasing scalars and \( (\nu_k), (\alpha_i) \) are sequences with nonnegative scalars, \( \nu_0^2 \leq s_0 \). If \( \nu_k^2 \leq s_k + \sum_{i=1}^{k} \alpha_i \nu_i^2 \), then we have \( \nu_k \leq \frac{1}{\beta} \sum_{i=1}^{k} \alpha_i + \sqrt{\left( \frac{1}{\beta} \sum_{i=1}^{k} \alpha_i \right)^2 + s_k} \).

A. Complexity Analysis for Algorithm 1

Now we prove Theorem 1. The proof uses the following Lemma, which is a standard technique when analyzing first-order methods.

**Lemma 7:** Assume that Assumption 1 holds. Then, for Algorithm 1, we have

\[
F(x^{k+1}) - F(x) \\
\leq \beta_0 \frac{\partial}{\partial x} (Uy^k, Ux - Ux^{k+1}) + L \frac{\partial}{\partial x} (y^k, x - y^k) \\
+ \left( \frac{L}{2} + \frac{\beta_0}{\nu_k} \right) \|x^{k+1} - y^k\|_F^2
\]

for any \( x \).

**Proof 2:** From the smoothness and convexity of \( f(x) \), we have

\[
f(x^{k+1}) \\
\leq f(y^k) + \langle \nabla f(y^k), x^{k+1} - y^k \rangle + \frac{L}{2} \|x^{k+1} - y^k\|_F^2 \\
= f(y^k) + \langle \nabla f(y^k), x - y^k \rangle + \langle \nabla f(y^k), x^{k+1} - x \rangle + \frac{L}{2} \|x^{k+1} - y^k\|_F^2 \tag{13}
\]

\[
\leq f(x) + \langle \nabla f(y^k), x^{k+1} - x \rangle + \frac{L}{2} \|x^{k+1} - y^k\|_F^2.
\]

From the optimality condition of the second step in Algorithm 1 and the convexity of \( h(x) \), we obtain

\[
0 \in \partial h(x^{k+1}) + \nabla f(y^k) + \frac{\beta_0}{\nu_k} \U^2 y^k + \left( \frac{L}{2} + \frac{\beta_0}{\nu_k} \right) (x^{k+1} - y^k)
\]

and

\[
h(x) - h(x^{k+1}) \\
\geq - \langle \nabla f(y^k) + \frac{\beta_0}{\nu_k} \U^2 y^k + \left( \frac{L}{2} + \frac{\beta_0}{\nu_k} \right) (x^{k+1} - y^k), x - x^{k+1} \rangle.
\]
Adding it and (13) together, we have
\[
F(x^{k+1}) - F(x)
\leq \left( \frac{\beta_0}{\theta_k} U^2 y^k + \left( L + \frac{\beta_0}{\theta_k} \right) (x^{k+1} - y^k), x - x^{k+1} \right) + \frac{L}{2} \|x^{k+1} - y^k\|_F^2
+ \frac{\beta_0}{\theta_k} \langle Uy^k, Ux - Ux^{k+1} \rangle + \left( L + \frac{\beta_0}{\theta_k} \right) \langle x^{k+1} - y^k, x - y^k \rangle
- \left( \frac{L}{2} + \frac{\beta_0}{\theta_k} \right) \|x^{k+1} - y^k\|_F^2.
\]

We introduce the Lagrangian function and let
\[
\rho_{k+1} = F(x^{k+1}) - F(x^*) + \langle \lambda^*, Ux^{k+1} \rangle.
\]

The proof of Theorem 1 is based on the following Lyapunov function
\[
\ell_{k+1} = \frac{\rho_{k+1}}{\theta_k} + \frac{1}{2\beta_0} \left\| \frac{\beta_0}{\theta_k} Ux^{k+1} - \lambda^* \right\|_F^2 + \frac{L\theta_k}{2} + \frac{\beta_0}{\theta_k} \|w^{k+1} - x^*\|_F^2,
\]
where \(w^{k+1} = \frac{x^{k+1}}{\theta_k} - \frac{1}{\theta_k} x^k\) and \(w^0 = x^0\). From the definitions of \(w^{k+1}\) and \(y^k\), we can have the following easy-to-identify identities.

**Lemma 8**: For Algorithm 1, we have
\[
\theta_k x^* + (1 - \theta_k) x^k - y^k = \theta_k (x^* - w^k),
\]
\[
\theta_k x^* + (1 - \theta_k) x^k - x^{k+1} = \theta_k (x^* - w^{k+1}).
\]

We will show \(\ell_{k+1} \leq \ell_k\) for all \(k = 0, 1, \cdots\) in the following proof and complete the proof of Theorem 1.

**Proof 3**: When we apply Lemma 7 first with \(x = x^k\) and then with \(x = x^*\), we obtain two inequalities. Multiplying the first inequality by \((1 - \theta_k)\), multiplying the second by \(\theta_k\), adding them together with \(\langle \lambda^*, Ux^{k+1} - (1 - \theta_k) Ux^k \rangle\) to both sides and using \(Ux^* = 0\), we can have
\[
F(x^{k+1}) - (1 - \theta_k) F(x^k) - \theta_k F(x^*) + \langle \lambda^*, Ux^{k+1} - (1 - \theta_k) Ux^k \rangle
\leq \left( \frac{\beta_0}{\theta_k} Uy^k - \lambda^*, (1 - \theta_k) Ux^k - Ux^{k+1} \right)
+ \left( L + \frac{\beta_0}{\theta_k} \right) \langle x^{k+1} - y^k, \theta_k x^* + (1 - \theta_k) x^k - y^k \rangle
- \left( \frac{L}{2} + \frac{\beta_0}{\theta_k} \right) \|x^{k+1} - y^k\|_F^2.
\]

where we use \(\frac{1}{\theta_{k-1}} = \frac{1}{\theta_k} - \frac{\beta_0}{\theta_k} \). Applying the identities in Lemma 2 to the two inner products, using

\[
\frac{\beta_0}{\theta_k} \|Uy^k - \lambda^*, (1 - \theta_k) Ux^k - Ux^{k+1}\|_F^2 \leq \frac{\beta_0}{\theta_k} \|y^k - x^{k+1}\|_F^2
\]
and dropping the negative terms, we can have
\[
\rho_{k+1} - (1 - \theta_k) \rho_k
\leq \frac{\beta_0}{2\beta_0} \left[ \left\| \frac{\beta_0}{\theta_k} Ux^k - \lambda^* \right\|_F^2 - \left\| \frac{\beta_0}{\theta_k} Ux^{k+1} - \lambda^* \right\|_F^2 \right]
+ \left( \frac{L}{2} + \frac{\beta_0}{2\theta_k} \right) \|\theta_k x^* + (1 - \theta_k) x^k - y^k\|_F^2
- \|\theta_k x^* + (1 - \theta_k) x^k - x^{k+1}\|_F^2.
\]

Dividing both sides of this inequality by \(\theta_k\), letting \(\theta_k = \theta\) and using \(\frac{1}{\theta_{k-1}} = \frac{1}{\theta_k} - \frac{\beta_0}{\theta_k} \), \(\theta_{k-1} \leq \theta_k\) and the identities in Lemma 8, we obtain
\[
\rho_{k+1} - \frac{\rho_k}{\theta_k} - \frac{\rho_k}{\theta_{k-1}}
\leq \frac{1}{2\beta_0} \left[ \left\| \frac{\beta_0}{\theta_{k-1}} Ux^k - \lambda^* \right\|_F^2 - \left\| \frac{\beta_0}{\theta_k} Ux^{k+1} - \lambda^* \right\|_F^2 \right]
+ \left( \frac{L\theta_k}{2} + \frac{\beta_0}{2\theta_k} \right) \|w^{k+1} - x^*\|_F^2 - \left( \frac{L\theta_{k-1}}{2} + \frac{\beta_0}{2\theta_k} \right) \|w^k - x^*\|_F^2,
\]
which reorganizes to \(\ell_{k+1} \leq \ell_k\) and thus
\[
\ell_{k+1} \leq \ell_0 = \frac{1}{2\beta_0} \|\lambda^*\|_F^2 + \left( \frac{L\theta_0}{2} + \frac{\beta_0}{2\theta_k} \right) \|w^0 - x^*\|_F^2 \equiv C_1,
\]
where we use \(\frac{1}{\theta_{k-1}} = \frac{1}{\theta_k} - \frac{\beta_0}{\theta_k} = 0\). From Lemma 4 and \(\theta_K = \frac{1}{\beta_0}\) we get the second conclusion. From \(F(x^{k+1}) - F(x^*) \leq \rho_{k+1} + \|\lambda^*\|_F \|Ux^{k+1}\|_F\) we obtain the first conclusion. □

From Lemma 3 and Theorem 1, we can prove Corollary 1.

**Proof 4**: Since \(F(x)\) is \(M\)-Lipschitz continuous, we can have
\[
F \left( \frac{1}{m} 11^T x^{K+1} \right) - F(x^*)
= F \left( \frac{1}{m} 11^T x^{K+1} \right) - F(x^{K+1}) + F(x^{K+1}) - F(x^*)
\leq \sqrt{m} \|\Pi x^{K+1}\|_F + F(x^{K+1}) - F(x^*)
\leq \sqrt{m} \|\Pi x^{K+1}\|_F + F(x^{K+1}) - F(x^*)
\leq \sqrt{m} \|\Pi x^{K+1}\|_F + F(x^{K+1}) - F(x^*)
\]
where we use Lemma 5 in the last inequality. From Lemma 3, we have \(\|\lambda^*\|_F \leq \frac{\sqrt{m} M}{1 - \sigma_2(W)} \equiv \frac{1}{\chi}\). From the setting of \(\beta_0\), we have \(\beta_0 \geq L, \beta_0 \sqrt{m} \geq \frac{1}{\chi}\) and
\[
C_1 \leq \frac{1}{2\beta_0} \left( \frac{L + \beta_0 m R^2}{2} \right) \leq \frac{\beta_0 m R^2}{2} \leq \frac{5m \beta_0 R^2}{K + 1},
\]
\[
\|U x^{K+1}\|_F \leq \frac{1}{K + 1} \left( \frac{3m R}{\sqrt{K + 1}} \right) \leq \frac{3m R}{K + 1},
\]
\[
F(x^{K+1}) - F(x^*) \leq \frac{1}{K + 1} \left( \frac{15m \beta_0 R^2}{K + 1} + 3m R^2 / K + 1 \right) \leq \frac{4.5m \beta_0 R^2}{K + 1},
\]
\[
F \left( \frac{1}{m} 11^T x^{K+1} \right) - F(x^*) \leq \frac{1}{K + 1} \left( \frac{15m \beta_0 R^2}{K + 1} + 3m R^2 / K + 1 \right) \leq \frac{7.5m \beta_0 R^2}{K + 1}.
\]

□
B. Complexity Analysis for Algorithm 2

Before proving the convergence of the outer iterations of procedure (11a)-(11c), we first establish the property when the proximal mapping of $\frac{\beta_0}{\partial \beta_k} \Pi x$ is inexactly computed. When the proximal mapping is exactly computed, i.e., $\epsilon_k = 0$ in (12), we have $L(x^{k+1} - z^k) + \frac{\beta_0}{\partial \beta_k} \Pi x^{k+1} = 0$. However, when the proximal mapping is computed inexactly, we should modify the conclusion accordingly. Specifically, we give the following lemma.

Lemma 9: Assume that (12) holds. Then, there exists $\delta_k$ with $\|\delta_k\|_F \leq \sqrt{\frac{\beta_0}{\partial \beta_k}}$ and $\frac{\beta_0}{\partial \beta_k} \|\Pi x\|_F^2 \leq 2\epsilon_k$ such that

$$L(x^{k+1} - z^k + \delta_k) = 0. \quad (14)$$

Proof 5: Define $x^{k,*} = \arg\min_x \frac{L}{2}\|x - z^k\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi x\|_F^2$. From the optimality condition, we have

$$0 = L(x^{k,*} - z^k) + \frac{\beta_0}{\partial \beta_k} \Pi_2 x^{k,*}. \quad (15)$$

From (12), we have

$$\epsilon_k \geq \frac{L}{2} \|x^{k+1} - z^k\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi x^{k+1}\|_F^2$$

$$= \frac{L}{2} \|x^{k,*} - z^k\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi x^{k,*}\|_F^2$$

$$= L(\langle x^{k,*} - z^k, x^{k+1} - x^{k,*}\rangle + \frac{L}{2} \|x^{k+1} - x^{k,*}\|_F^2)$$

$$+ \frac{\beta_0}{\partial \beta_k} \|\Pi(x^{k+1} - x^{k,*})\|_F^2$$

$$= \frac{L}{2} \|x^{k+1} - x^{k,*}\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi(x^{k+1} - x^{k,*})\|_F^2,$$

where we use Lemma 2 in the first equality and (15) in the second equality. Define $\delta_k = x^{k,*} - x^{k+1}$. Then we have $\|\delta_k\|_F \leq \sqrt{\frac{\beta_0}{\partial \beta_k}}$ and $\frac{\beta_0}{\partial \beta_k} \|\Pi x^{k+1}\|_F^2 \leq 2\epsilon_k$. From (15) we can have (14).

Proof 6: Define $\tilde{x}^k = \frac{1}{m} 11T z^k$ and $\tilde{x}^{k,*} = \frac{1}{m} 11T x^{k,*}$. From (15) and $\Pi_2 = \Pi$, we have $0 = L(\tilde{x}^{k,*} - z^k) + \frac{\beta_0}{\partial \beta_k} \tilde{x}^{k,*} - \tilde{x}^{k,*} = \frac{L \beta_0 z^k + \beta_0 \tilde{x}^{k,*}}{L \beta_0 + \beta_0}$, which leads to $\tilde{x}^{k,*} = \frac{L \beta_0 z^k + \beta_0 \tilde{x}^{k,*}}{L \beta_0 + \beta_0}$. From $1T \tilde{x}^{k,*} = \tilde{x}^{k,*} = \frac{L \beta_0 z^k + \beta_0 \tilde{x}^{k,*}}{L \beta_0 + \beta_0}$, we have $\tilde{x}^{k,*} = \tilde{x}^k$ and $\tilde{x}^{k,*} = \frac{L \beta_0 z^k + \beta_0 \tilde{x}^{k,*}}{L \beta_0 + \beta_0}$. From (16) and the definition of $x^{k+1}$, we can have

$$\frac{L}{2} \|x^{k+1} - z^k\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi x^{k+1}\|_F^2$$

$$= \frac{L}{2} \|x^{k,*} - z^k\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi x^{k,*}\|_F^2$$

$$= \frac{L}{2} \|x^{k+1} - x^{k,*}\|_F^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi(x^{k+1} - x^{k,*})\|_F^2$$

$$= \frac{\beta_0}{L \beta_0 + \beta_0} (L \beta_0 z^k - \tilde{x}^k)^2 + \frac{\beta_0}{\partial \beta_k} \|\Pi(z^{k+1} - \tilde{x}^k)\|_F^2$$

$$\leq \frac{\beta_0}{L \beta_0 + \beta_0} \|z^{k+1} - \tilde{x}^k\|_F^2 \leq \frac{\beta_0}{\partial \beta_k} \|z^{k+1} - \tilde{x}^k\|_F^2.$$ 

Now we are ready to analyze procedure (11a)-(11c). Similar to Lemma 7, we can have the following lemma for Algorithm 2.

Lemma 10: Assume that Assumption 1 holds. Then, for Algorithm 2, we have

$$f(x^{k+1}) - f(x) \leq \frac{\beta_0}{\partial \beta_k} (\Pi x^{k+1} + \Pi \delta_k) + L (x^{k+1} - y^k, x - y^k) + \langle \delta_k, x - x^{k+1} \rangle - \frac{\mu}{2} \|x - y^k\|^2_F - \frac{L}{2} \|x^{k+1} - y^k\|^2_F,$$

for any $x$.

The conclusion can be obtained by (13), (14), $z^k = y^k - \frac{1}{\mu} \nabla f(y^k)$ and a similar induction to the proof of Lemma 7. So we omit the details.

Define $w^{k+1} = \frac{x^{k+1} - 1 - \theta_k}{\theta_k} x^k$ with $w^0 = x^0$. Similar to Lemma 8, we can have the following identifies from the definition of $y^k$ in Algorithm 2.

**Lemma 11:** For Algorithm 2, we have

$$x^* + (1 - \theta_k) L x^k - \frac{L}{L \theta_k - \mu} x^k = x^* - w^k,$$

$$\theta_k x^* + (1 - \theta_k) x^k - x^{k+1} = \theta_k (x^* - w^{k+1}).$$

Define $\rho_{k+1} = f(x^{k+1}) - f(x^*) + \langle \lambda^*, \Pi x^{k+1} \rangle$ and let $(x^*, \lambda^*)$ be a KKT point of the saddle point problem $\min_x \max_{\lambda} f(x) + \langle \lambda, \Pi x \rangle$. Then similar to the proof of Theorem 1, we can have the following Lemma, which gives a progress in one iteration of Algorithm 2. We use the same notations of $\rho_{k+1}$ with Section III-A for easy analogy.

**Lemma 12:** Assume that Assumption 1 holds. Let sequences \{\theta_k\} and \{\theta_k\} satisfy $\frac{1}{\theta_k} = \frac{1}{\theta_{k-1}}$ and $\theta_k \geq \frac{2}{\beta_0}$. Then, under the assumption of (12), we have

$$\rho_{k+1} + \frac{\theta_k}{2 \beta_0} \|\theta_k - \lambda^*\|^2_F + \frac{L \theta_k^2}{2} \|w^{k+1} - x^*\|^2_F$$

$$\leq (1 - \theta_k) \rho_k + \frac{\theta_k}{2 \beta_0} \|\Pi x^k - \lambda^*\|^2_F + \epsilon_k$$

$$+ \frac{(L \beta_0 - \mu) \theta_k}{2} \|w^k - x^*\|^2_F + L \theta_k \sqrt{\frac{2 \epsilon_k}{L}} \|w^{k+1} - x^*\|^2_F.$$

Proof 7: From Lemma 10 and a similar induction to the proof of Theorem 1, we can have

$$f(x^{k+1}) - (1 - \theta_k) f(x^k) - \theta_k f(x^*) + \langle \lambda^*, \Pi x^{k+1} - (1 - \theta_k) \Pi x^k \rangle$$

$$\leq \frac{\beta_0}{\partial \beta_k} (\Pi x^{k+1} + \Pi \delta_k) - \lambda^*, \frac{\beta_0}{\partial \beta_k} - \Pi x^k - \frac{\beta_0}{\partial \beta_k}$

$$+ L (x^{k+1} - y^k, (1 - \theta_k) x^k + \theta_k x^* - y^k) + L \langle \delta_k, (1 - \theta_k) x^k + \theta_k x^* - x^{k+1} \rangle$$

$$- \frac{\mu}{2} \|x^* - y^k\|^2_F - \frac{L}{2} \|x^{k+1} - y^k\|^2_F.$$
Applying the identities in Lemma 2 to the two inner products, we can have
\[ \rho_{k+1} - (1 - \theta_k)\rho_k \]
\[ \leq \frac{\delta_k}{2\beta_0} \left( \left\| \frac{\beta_0}{\beta_0} \Pi x^k - \lambda^* \right\|_F^2 - \left\| \frac{\beta_0}{\theta_k} \Pi x^{k+1} + \Pi \delta^k \right\|_F^2 \right) \]
\[ + \left( \frac{L}{2} \right) \left[ \left\| (1-\theta_k) x^k + \theta_k x^* - y_k \right\|_F^2 - \left\| (1-\theta_k) x^k + \theta_k x^* - x^{k+1} \right\|_F^2 \right] \]
\[ + L \left\langle \delta^k, (1-\theta_k) x^k + \theta_k x^* - x^{k+1} \right\rangle - \frac{\mu \theta_k}{2} \left\| x^* - y^k \right\|_F^2 \]
\[ \leq \frac{\delta_k}{2\beta_0} \left( \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F^2 - \left\| \frac{\beta_0}{\theta_k} \Pi x^{k+1} - \lambda^* \right\|_F^2 + \frac{\beta_0^2}{\theta_k} \Pi \delta^k \right) \]
\[ + \frac{L}{2} \left[ \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F^2 - \left\| \frac{\beta_0}{\theta_k} \Pi x^{k+1} - \lambda^* \right\|_F^2 \right] \]
\[ + L \left\langle \delta^k, (1-\theta_k) x^k + \theta_k x^* - x^{k+1} \right\rangle - \frac{\mu \theta_k}{2} \left\| x^* - y^k \right\|_F^2 \]
where the last inequality follows from the second identity in Lemma 11. By reorganizing the terms in \( \frac{\delta_k}{\beta_0} = \frac{1-\theta_k}{\theta_k} x^k - x^* \) carefully, we can have
\[ \frac{L \theta_k^2}{2} \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F^2 \]
\[ = \frac{L \theta_k^2}{2} \left\| \frac{\mu}{L \theta_k} \left( x^k - x^* \right) + \frac{\left( 1 - \theta_k \right) L \theta_k - \mu}{L \theta_k} \right\|_F^2 \]
\[ \leq \frac{\mu \theta_k}{2} \left\| x^k - x^* \right\|_F^2 + \frac{2 \left( 1 - \theta_k \right) L \theta_k - \mu}{2} \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F^2 \]
where we let \( \frac{\mu}{\theta_k} \leq 1 \), use Jensen’s inequality for \( \left\| \cdot \right\|_F \) in the first inequality and the first identity in Lemma 11 in the last equality. Plugging it into the above inequality and using the bounds for \( \left\| \delta^k \right\|_F \) and \( \Pi \delta^k \) in Lemma 9, we can get (17).

Due to the term \( \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F \) on the right hand side of (17), recursion (17) cannot be directly telescoped unless we assume the boundness of \( \left\| \frac{\beta_0}{\theta_k} \Pi x^k - \lambda^* \right\|_F \). Lemma 6 can avoid such boundness assumption. Now, we use Lemmas 12 and 6 to analyze the outer iterations of procedure (11a)- (11c). The following theorem shows the convergence for strongly convex problems.

**Theorem 4:** Assume that Assumptions 1, 2 and (12) hold with \( \mu > 0 \) and \( \varepsilon_k \leq (1 - (1 + \tau)^k)^{1+1} \), where \( 1 > \tau > 0 \) can be any small constant. Let sequences \( \{ \theta_k \} \) and \( \{ \delta_k \} \) satisfy \( \theta_k = \theta = \sqrt{\frac{1}{2} \varepsilon} \) and \( \delta_k = (1 - \theta_k)^{1+1} \). Then, we have
\[ F(x^{k+1}) - F(x^*) \leq C_2 (1 - \theta) K^{1+1}, \]
\[ \left\| \Pi x^{k+1} \right\|_F \leq C_3 (1 - \theta) K^{1+1}, \]
\[ \left\| x^{k+1} - x^* \right\|_2^2 \leq C_4 (1 - \theta) K^{1+1}, \]
\[ f\left( \frac{1}{m} 11^T x^{k+1} \right) - f(x^*) \leq C_5 (1 - \theta) K^{1+1} + \frac{L C_2^2}{2} (1 - \theta)^2 K^{1+2}, \]
where \( C_2 = C_0 + \left\| \lambda^* \right\|_F C_1, \)
\[ C_3 = \frac{2C_0^2}{\varepsilon_0}, \]
\[ C_4 = \frac{2C_0^2}{\varepsilon_0}, \]
\[ C_5 = \left( \frac{\Pi x^{k+1} \cdot x^*}{F} + L \right), \]
\[ C_6 = \frac{64}{\varepsilon_0} + 2 \left( f(x^0) - f(x^*) + \left\langle \lambda^*, \Pi x^0 \right\rangle \right) + \frac{1}{\varepsilon_0} \left\| \beta_0 \Pi x^0 - \lambda^* \right\|_F^2 + \left\| \mu x^0 - x^* \right\|_F^2. \]

Proof: Letting \( L(\theta - \mu) = \theta_0 (1 - \theta) \), then we have \( \theta = \sqrt{\frac{1}{2} \varepsilon} \) and sequences \( \{ \theta_k \} \) and \( \{ \delta_k \} \) satisfy the requirement in Lemma 12. Define the Lyapunov function \( \ell_{k+1} \) satisfying
\[ (1 - \theta) K^{1+1} \ell_{k+1} = (1 - \theta) K^{1+1} \ell_k \leq \varepsilon_k + \theta_0 \left\| \frac{\beta_0}{\theta_k} \Pi x^{k+1} - \lambda^* \right\|_F^2 \]
\[ + \frac{L \theta^2}{2} \left\| w^{k+1} - x^* \right\|_F^2. \]

From (17) and the special choice of \( \theta_k \) and \( \delta_k \), we have
\[ (1 - \theta) K^{1+1} \ell_{k+1} \leq \varepsilon_k + L \theta_0 \left\| \frac{\beta_0}{\theta_k} \Pi x^{k+1} - x^* \right\|_F. \]

Dividing both sides by \( (1 - \theta) K^{1+1} \) and summing over \( k = 0, 1, \ldots, K \), we have
\[ \ell_{K+1} - \ell_0 \]
\[ \leq \sum \frac{K}{k} \left\| x^k - x^* \right\|_F \]
\[ + \sum \frac{K}{k} \left\| \triangle k \right\| + \frac{K}{k} \left\| \Pi x^k \right\|_F \]
\[ \leq \sum \frac{K}{k} \left\| x^k - x^* \right\|_F + \sum \frac{K}{k} \left\| \triangle k \right\| + \frac{K}{k} \left\| \Pi x^k \right\|_F \]
\[ \leq \sum \frac{K}{k} \left\| x^k - x^* \right\|_F + \sum \frac{K}{k} \left\| \triangle k \right\| + \frac{K}{k} \left\| \Pi x^k \right\|_F \]
\[ \leq \frac{64}{\varepsilon_0^2} + 2 \ell_0 = C_0. \]

From the definition of \( \ell_{K+1} \) and a similar induction to the proof of Theorem 1, we can have the first two conclusions. Since \( f(x) + \left\langle \lambda^*, \Pi x \right\rangle \) is \( \mu \)-strongly convex over \( x \) and \( x^* = \arg\min_x f(x) + \left\langle \lambda^*, \Pi x \right\rangle \), we have \( \frac{\mu}{2} \left\| x^{k+1} - x^* \right\|_2^2 \leq f(x^{k+1}) + \left\langle \lambda^*, \Pi x^{k+1} \right\rangle - f(x^*) - \left\langle \lambda^*, \Pi x^* \right\rangle \leq C_6 (1 - \theta) K^{1+1}. \)

From the smoothness of \( f(x) \), we can have
\[ f\left( \frac{1}{m} 11^T x^{k+1} \right) - f(x^*) \]
\[ \leq \frac{L}{2} \left\| \frac{1}{m} 11^T x^{k+1} - x^* \right\|_F \]
\[ + f(x^{k+1}) - f(x^*) \]
\[ \leq \left( \frac{\Pi x^k \cdot x^*}{F} + L \right), \]
\[ \Pi x^{k+1} \right\|_F + \frac{L}{2} \left\| \Pi x^{k+1} \right\|_F \]
\[ + f(x^{k+1}) - f(x^*) \]
and the forth conclusion. \( \square \)
In the following theorem, we consider the case that \( f(x) \) is non-strongly convex.

**Theorem 5**: Assume that Assumptions 1, 2 and (12) hold with \( \mu = 0 \) and \( \varepsilon_k \leq (k+1)\tau \). Let sequences \( \{\theta_k\} \) and \( \{\bar{\theta}_k\} \) satisfy \( \theta_0 = 1, \frac{1-\theta_0}{\sigma_0} = \frac{1}{\theta_{k+1}} \) and \( \bar{\theta}_0 = \theta_k^2 \). Then, we have
\[
f(x^{K+1}) - f(x^*) \leq \frac{C_7}{(K+2)^2},
\]
\[
\|\Pi x^{K+1}\|_F \leq \frac{C_8}{(K+2)^2},
\]
\[
\|x^{K+1} - x^*\|_F^2 \leq C_9,
\]
\[
f \left( \frac{1}{m} \mathbf{1}^T x^{K+1} \right) - f(x^*) \leq \frac{C_{10}}{(K+2)^2} + \frac{LC_8^2}{2(K+2)^2},
\]
where
\[
C_7 = 4C_{11} + \frac{\|\nabla f(x^*)\|_F C_8, \quad C_9 = \frac{4\sqrt{\mu c_0} c_{11} + \|\nabla f(x^*)\|_F}{\sigma_0}, \quad C_{10} = \left( \|\nabla f(x^*)\|_F + L\sqrt{\sigma_0} \right) C_8 \text{ and } C_{11} = 6 + \|\nabla f(x^*)\|_F^2 + L\|x^0 - x^*\|_F^2.
\]

**Proof**: Define the following Lyapunov function \( \ell_{k+1} \).
\[
\ell_{k+1}^2 = \frac{\theta_{k+1}}{\theta_k} + 1 + \frac{1}{\theta_k} \Pi x^{k+1} - x^* ||^2 + \frac{L}{2} \|x^{k+1} - x^*\|_F^2.
\]
Dividing both sides of (17) by \( \theta_k^2 \), using \( \theta_0 = \theta_2 \) and \( \frac{1-\theta_0}{\sigma_0} = \frac{1}{\sigma_k^{k+1}} \), we have
\[
\ell_{k+1}^2 - \ell_k^2 \leq \frac{\varepsilon_k}{\theta_k^2} + \frac{2\varepsilon_k}{\theta_k} \sqrt{L} \|x^{k+1} - x^*\|_F.
\]
Similar to the proof of Theorem 4, we can obtain
\[
\ell_{k+1}^2 - \ell_k^2 \leq \sum_{k=0}^{K+1} \frac{\varepsilon_k}{\theta_k^2} + \frac{2\varepsilon_k}{\theta_k} \sqrt{L} \|x^{k+1} - x^*\|_F.
\]
and
\[
\ell_{k+1}^2 \leq \left( \sum_{k=0}^{K+1} \frac{2\varepsilon_k}{\theta_k} \right)^2 + \sum_{k=0}^{K+1} \frac{2\varepsilon_k}{\theta_k} \sqrt{L} \|x^{k+1} - x^*\|_F.
\]
where we use \( \frac{K+1}{\theta_k} \leq \theta_0 \leq \frac{2}{\theta_k} \) from \( \frac{1-\theta_0}{\sigma_0} = \frac{1}{\sigma_k^{k+1}} \), and \( \theta_0 = 1 \).

Letting \( \varepsilon_k \leq (k+1)\tau \), then \( \sum_{k=0}^{K} 2\varepsilon_k (k+1)^2 \leq \frac{\tau^2}{2+\tau} \), and \( \sum_{k=1}^{K+1} 2\varepsilon_k \leq \frac{\tau^2}{2} \). So
\[
\ell_{k+1}^2 \leq \frac{\tau^2}{4} + \frac{4}{1+2\tau} + \frac{\|x^*\|_F^2}{\beta_0} + L\|x^0 - x^*\|_F^2 \leq C_{11},
\]
where we let \( \tau = 1 \) for simplicity. From the definition of \( w^{k+1} = x^{k+1} - \frac{1-\varepsilon_k}{\sigma_k} x^k \), we have \( \|x^{k+1} - x^*\|_F = \|\theta_k^2 w^{k+1} + (1-\theta_k^2) x^k - x^*\|_F \leq \theta_k \|w^{k+1} - x^*\|_F + (1-\theta_k^2) \|x^k - x^*\|_F \). By induction, we can prove \( \|x^{k+1} - x^*\|_F \leq \frac{2C_{11}}{L}, \forall k \). Similar to the proof of Theorem 4 and using Lemma 3, we can have the remaining conclusions.

Now we consider the subproblem in procedure (11c). At the \( k \)-iteration, we want to find an approximate solution \( x^{k+1} \) such that (12) is satisfied. From [29, Proposition 3], we can have
\[
\|z_k^T z_k^* - \frac{1}{m} \mathbf{1}^T x^{k+1} \|_F \leq \left( \frac{\sigma_2(W)}{1+\sqrt{1-\sigma_2(W)}} \right)^T_k \|\Pi z^k\|_F \leq \left( 1 - \sqrt{1-\sigma_2(W)} \right)^T_k \|\Pi z^k\|_F
\]
for the accelerated average consensus. Thus from Lemma 1, we only need
\[
T_k = -\frac{1}{2\log \left( 1 - \sqrt{1-\sigma_2(W)} \right)} \frac{1}{2\theta_k \varepsilon_k}
\]
such that (12) is satisfied.

Based on Theorems 4, 5 and the analysis for the inner loop, we can establish the computation and communication complexities for Algorithm 2. We first consider the strongly convex case and prove Theorem 2.

**Proof**: We first prove that \( \|\Pi z^k\|_F \) is bounded for any \( k \) given
\[
T_k = -\frac{1}{2\log \left( 1 - \sqrt{1-\sigma_2(W)} \right)} \frac{1}{2\theta_k \varepsilon_k},
\]
where \( C_4 \) is defined in Theorem 4. We prove \( \|\Pi z^k\|_F \) is bounded for all \( k \leq K \). Then from (18) we know that (12) holds for \( k \leq K \). From Theorem 4, we have \( \|x^K - x^*\|_F \leq \sqrt{C_4} \) and \( \|x^{K+1} - x^*\|_F \leq \sqrt{C_4} \). Thus,
\[
\|\Pi z^k\|_F \leq \|\Pi y^{K+1}\|_F + \frac{1}{L} \|\nabla f(y^{K+1})\|_F
\]
\[
\leq \|\Pi (y^{K+1} - x^*\|_F + \frac{1}{L} \|\nabla f(x^*)\|_F + L\|y^{K+1} - x^*\|_F
\]
\[
\leq \frac{1}{L} \|\nabla f(x^*)\|_F + 4\|x^{K+1} - x^*\|_F + 2\|x^K - x^*\|_F
\]
\[
\leq \frac{1}{L} \|\nabla f(x^*)\|_F + 6\sqrt{C_4}.
\]
So we can get the conclusion.

From Theorem 4, we know that the number of gradient computation, i.e., the number of outer iterations, is \( O \left( \sqrt{\frac{L}{\mu}} \log \frac{1}{\tau} \right) \). From (18), we have
\[
T_k = O \left( \frac{\log \frac{1}{\sqrt{1-\sigma_2(W)}}}{-\log \left( 1 - \sqrt{1-\sigma_2(W)} \right)} \frac{1}{\log \left( 1 - \sqrt{1-\sigma_2(W)} \right)} \frac{1}{\mu L} \right)
\]
\[
= O \left( \frac{k \log \left( 1 - \sqrt{1-\sigma_2(W)} \right)}{\log \left( 1 - \sqrt{1-\sigma_2(W)} \right)} \right) = O \left( \frac{k \sqrt{\mu L}}{\sqrt{1-\sigma_2(W)}} \right),
\]
where we use log \( (1 - \sqrt{1-\sigma_2(W)}) \sim -\sqrt{1-\sigma_2(W)} \) and \( \log \left( \sqrt{1-\sigma_2(W)} \right) \approx -\sqrt{\mu L} \) from Taylor expansion when \( \sqrt{1-\sigma_2(W)} \) and \( \sqrt{\mu L} \) are small. Thus, the total number of communication, i.e., the total number of inner iterations, is
\[
\sum_{k=0}^{K} O \left( k \sqrt{\frac{\mu L}{(1-\sigma_2(W))}} \right) = O \left( \sqrt{\frac{L}{\mu \left( 1 - \sigma_2(W) \right)}} \log^2 \frac{1}{\tau} \right).
\]
Now we consider the non-strongly case and prove Theorem 3.

**Proof 11:** Similar to the above proof of Theorem 2, we know that \(\|z^k\|_F\) is also bounded for all \(k\). Let \(\beta_0 \geq L \|\nabla f(x^0)\|_F^2\) and assume \(L \geq 1\). \(\|x^0 - x^*\|_F \geq 1\) and \(\|\nabla f(x^*')\|_F \geq 1\) for simplicity. Then \(C_T = O(L\|x^0 - x^*\|_F^2)\), \(C_8 = O((\|x^0 - x^*\|_F^2)\), and \(C_{10} = O(L\|x^0 - x^*\|_F^2)\).

From Theorem 5, we know that the number of gradient computations is \(O(\sqrt{\ell/\epsilon})\). From (18), we have

\[
T_k = O\left(-\frac{1}{\sqrt{1 - \sigma_2(W)}}\right) = O\left(-\frac{1}{\sqrt{1 - \sigma_2(W)}}\right).
\]

Thus, the total number of communication is

\[
\sum_{k=0}^{K} T_k = O\left(-\frac{L}{\epsilon(1 - \sigma_2(W))}\log\frac{1}{\epsilon}\right).
\]

\(\square\)

IV. NUMERICAL EXPERIMENTS

We test the performance of the proposed algorithms on the following least square regression problem

\[
\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} f_i(x) \quad \text{with} \quad f_i(x) = \frac{1}{2}\|A_i^T x - b_i\|^2 + \frac{\mu_i}{2}\|x\|^2.
\]

We generate \(A_i \in \mathbb{R}^{n \times N/m}\) from the uniform distribution with each entry in \([0, 1]\) and normalize each column of \(A_i\) to be 1, where \(N\) is the sample size. We set \(N = 1000\), \(n = 500\) and \(m = 100\). We consider both the non-strongly convex aggregate objective \((\mu = 0)\) and strongly convex aggregate objective \((\mu > 0)\). We consider the random network that each pair of agents has a connection with the probability of \(p\) and set \(W = \frac{MM}{W}\), where \(M\) is the Metropolis weight matrix [27]. This network is known as the Erdős-Rényi random graph. Almost all Erdős-Rényi random graph with \(p = \frac{2\log m}{m}\) is connected and \(\frac{1}{\sigma_2(W)} = O(1)\) [28, Proposition 5]. We test the performance with \(p = 0.1\) and \(p = 0.5\) and observe that \(1 - \sigma_2(W) = 0.13\) and \(1 - \sigma_2(W) = 0.33\), respectively.

For the non-strongly convex aggregate objective, we test the performance of APM, APM-C, D-NG [22], D-NC [22], Accelerated Distributed Nesterov Gradient Descent (Acc-DNGD) [24], EXTRA [25] and the accelerated dual ascent (ADA) [3]. We set the inner iteration number \(T_k\) as \(\left[\frac{\log(k+1)^8}{5\sqrt{1 - \sigma_2(W)}}\right]\) and \(\left[\frac{-\log(k+1)^8}{5\sqrt{1 - \sigma_2(W)}}\right]\) for APM-C and D-NC, respectively, where \([\cdot]\) means the top integral function. We set the stepsize as \(\frac{F}{\epsilon}\) for the two algorithms and \(\beta_0 = 100\) for APM-C. We set \(\beta_0 = \frac{k+1}{5\epsilon}\) with \(c = 50\) for APM and tune the best \(c = 1\) for D-NG. Larger \(c\) makes D-NG diverge. We tune the best stepsize as \(\frac{F}{\epsilon}\) for EXTRA, \(\frac{F}{\epsilon}\) for Acc-DNGD with \(p = 0.1\) and \(\frac{F}{\epsilon}\) for Acc-DNGD with \(p = 0.5\), respectively. For ADA, we follow [3] to add a small regularizer of \(\frac{\epsilon}{2}\|x\|^2\) to each \(f_i(x)\) and solve a regularized problem with \(\epsilon = 10^{-7}\). We follow the theory in [3] to set the inner iteration number as \(T_k = \left[\frac{F}{\epsilon}\log\frac{F}{\epsilon}\right]\). We initialize \(x^0\) at 0 for all the compared methods.

Figure 1 plots the comparisons. We can see that APM-C has the lowest computation cost to obtain an \(\epsilon\)-optimal solution, which means that APM-C suits for the environment that computation is the bottleneck of the overall performance. APM performs better than D-NG because APM allows to use a larger stepsize in practice, which can reduce the negative impacts from the diminishing stepsize. APM suits for the environment that high precision is not required, otherwise, the diminishing stepsize makes the algorithm slow. ADA has the lowest communication cost. However, ADA needs to predefined \(\epsilon\) to set the algorithm parameter and thus it only achieves an approximate optimal solution in the precision of \(\epsilon\) due to the weakness of the regularization trick. Due to the large \(T_k\) for ADA, it only performs one outer iteration after 20000 gradient computations and thus has almost no decreasing in the first and third plots of Figure 1.

For the strongly convex aggregate objective, we compare APM-C with ADA [2], Acc-DNGD [24], EXTRA [25] and NEAR-DGD+ [31], NEAR-DGD+ can be seen as a counterpart of APM-C without Nesterov’s acceleration scheme and accelerated average consensus. We test on different condition number by setting \(\mu = 0.001\) and \(\mu = 0.0001\), respectively. We set \(T_k = \left[\frac{k\epsilon}{\sqrt{\mu/L}}\right]\), \(\beta_0 = 100\) and the the stepsize as \(\frac{1}{L}\) for APM-C. For ADA, we set the inner iteration number as \(\left[\frac{\sqrt{L}}{\epsilon p\log\frac{F}{\epsilon}}\right]\) [3] and the stepsize as \(\mu\). We tune the best stepsize as \(\frac{1}{L}\) and \(\frac{0.5}{L}\) for EXTRA and Acc-DNGD, respectively. We follow [31] to set \(T_k = k\) for NEAR-DGD+.

From figure 2, we can see that APM-C also has the lowest computation cost and ADA has the lowest communication cost, which matches the theory. APM-C has a slight higher communication cost than Acc-DNGD but a lower computation cost. APM-C performs better than NEAR-DGD+ and it verifies that Nesterov’s acceleration scheme is critical to improve the performance. When preparing the experiments, we empirically observe that APM-C suits for the network with small \(\frac{\sqrt{L}}{\epsilon}\) and otherwise, the communication costs will be high. In fact, when \(\frac{1}{\sigma_2(W)}\) is small and the objective function is ill-conditioned, \(\frac{\sqrt{\mu/L}}{\sqrt{1 - \sigma_2(W)}}\) will be very small, e.g., 0.01 in our experiment with \(\mu = 0.0001\) and \(p = 0.1\). Thus the required \(T_k\) is small, e.g., \(T_{3000} = 11\) in our experiment. As a comparison, NEAR-DGD+ suggests \(T_k = k\) and thus it increases quickly, which leads to almost no decreasing in the second and forth plot of Figure 2. In practice, we can use the expander graph [47] which satisfies \(\frac{1}{\sigma_2(W)} = O(1)\) [28]. The Erdős-Rényi random graph is a special case of expander graph and can be easily implemented.

V. CONCLUSION

In this paper, we study the distributed accelerated gradient method from the perspective of the accelerated penalty method with increasing penalty parameters. Two algorithms are proposed. The first one obtains the optimal computation complexity for non-smooth distributed optimization. The second algorithm achieves the optimal computation complexities and near optimal communication complexities for smooth...
we achieve both the optimal computation and communication (right two).

[0x0]0.5
[0x0]0.5
[0x0]0.5
[0x0]0.5
[0x0]1000
[0x0]600
[0x0]2
[0x0]2000
[0x0]1
[0x0]200
[0x0]200
[0x0]1000
[0x0]800
[0x0]800
[0x0]400
[0x0]600
[0x0]1
[0x0]1000
[0x0]3000
[0x0]400
[0x0]1
[0x0]600
[0x0]1.5
[0x0]1
[0x0]200
[0x0]1000
[0x0]400
[0x0]3000
[0x0]200
[0x0]2000
[0x0]2
[0x0]400
[0x0]3000
[0x0]1.5
[0x0]1000
[0x0]2000
[0x0]800
[0x0]1000
[49x196]we achieve both the optimal computation and communication (right two).

environment with large that high precision is not required and APM-C works for the of APM-C partially answers this question. APM and APM-C the lower bounds. It remains an open problem that how can

Fig. 2. Comparisons on the strongly convex problem and random network with

\[
\frac{1}{\sqrt{1-\sigma_2(W)}}
\]
distributed optimization. Our communication complexities of the second algorithm are only worse by a factor of \(\log \frac{1}{\mu}\) than the lower bounds. It remains an open problem that how can we achieve both the optimal computation and communication complexities for distributed algorithms? Our second algorithm of APM-C partially answers this question. APM and APM-C may not fit all applications. APM suits for the environment that high precision is not required and APM-C works for the environment with large \(\frac{1}{\mu}\) and small \(\sqrt{1-\sigma_2(W)}\).

REFERENCES

[1] H. Terelius, U. Topcu, and R. Murray. Decentralized multi-agent optimization via dual decomposition. IFAC proceedings volumes, 44(1):11245–11251, 2011.

[2] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In ICML, 2017.

[3] C. Uribe, S. Lee, A. Gasnikov, and A. Nedić. A dual approach for optimal algorithms in distributed optimization over networks. arxiv:1809.00710, 2018.

[4] G. Lan, S. Lee, and Y. Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. arxiv:1701.03961, 2017.

[5] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulié. Optimal algorithms for non-smooth distributed optimization in networks. NIPS, 2018.

[6] M. Hong, D. Hajinezhad, and M. Zhao. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In ICML, 2017.

[7] D. Jakovetić. A unification and generalization of exact distributed first order methods. arxiv:1709.01317, 2017.

[8] T. Erseghe, D. Zennaro, E. Dall’Anese, and L. Vangelista. Fast consensus by the alternating direction multipliers method. IEEE Trans. on Signal Processing, 59(11):5523–5537, 2011.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin. On the linear convergence of the ADMM in decentralized consensus optimization. IEEE Trans. on Signal Processing, 62(2):1750–1761, 2014.

[10] E. Wei and A. Ozdaglar. On the \(o(1/k)\) convergence of asynchronous distributed alternating direction method of multipliers. In IEEE Global
