A Distributed Proximal Method for Composite Convex Optimization

N. S. Aybat∗  G. Iyengar†  Z. Wang *

Abstract

We propose a distributed first-order augmented Lagrangian (FAL) algorithm to minimize the sum of composite convex functions, where each term in the sum is a private function known only to one of the nodes, and only nodes connected by an edge can communicate to each other. This model of computation applies to a number of applications in distributed machine learning. We show that any limit point of FAL iterates is optimal; and for any ε > 0, an ε-optimal and ε-feasible solution can be computed within \( O(\log(1/\epsilon)) \) FAL iterations, which require \( O(\frac{\psi_1^{1.5}}{d_{\min} \epsilon}) \) communications steps, where \( \psi_1 \) is the largest eigenvalue of the graph Laplacian, and \( d_{\min} \) is the degree of smallest degree node. For a connected graph with \( N \) nodes, the second smallest eigenvalue of the Laplacian, \( \psi_{N-1} > 0 \), shows the strength of connectivity and is called the spectral gap. Since \( \psi_{N-1} \leq d_{\min} \), our result also implies that the number of communication steps can be bounded above by \( O(\frac{\psi_1^{1.5}}{\psi_{N-1} \epsilon^{-1}}) \). We also propose an asynchronous version of FAL by incorporating randomized block coordinate descent methods; and demonstrate the efficiency of FAL on large scale sparse-group LASSO problems.

Keywords. distributed optimization, composite convex function, augmented Lagrangian, first-order method, spectral gap.
1 Introduction

Let $G = (\mathcal{N}, \mathcal{E})$ denote a connected undirected graph of $N$ computing nodes where nodes $i$ and $j$ can communicate information only if $(i, j) \in \mathcal{E}$. Each node $i \in \mathcal{N} := \{1, \ldots, N\}$ has a private (local) cost function

$$F_i(x) := \rho_i(x) + \gamma_i(x), \quad (1.1)$$

where $\rho_i : \mathbb{R}^n \to \mathbb{R}$ is a possibly non-smooth convex function, and $\gamma_i : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function. We propose a distributed augmented Lagrangian algorithm for efficiently computing a solution for the convex optimization problem:

$$F^* := \min_{x \in \mathbb{R}^n} F(x) := \sum_{i=1}^{N} F_i(x). \quad (1.2)$$

Clearly, (1.2) can be solved in a centralized fashion by communicating all the private functions $F_i$ to a central node, and solving the overall problem at this node. However, such approach can be very expensive both from a communication and computation perspective. As an example, suppose $F_i(x) = \|A_i x - b_i\|_2^2 + \lambda \|x\|_1$ where $(A_i, b_i) \in \mathbb{R}^{m \times (n+1)}$ with $m \ll n$. Then (1.2) is a very large scale LASSO problem where the data is decentralized. In order to solve (1.2) in a centralized fashion the data $\{(A_i, b_i) : i \in \mathcal{N}\}$ needs to be communicated to the central node. This can be prohibitively expensive, and may also violate privacy constraints in the sense that node $i$ may not want to reveal the details of the data it owns. In addition, the central node will need to have large enough memory to be able to accommodate all the data. We propose decentralized algorithms that can compute solutions to (1.2) using only local computations; thereby, circumventing both the communication and memory issues. In order to facilitate the design of decentralized algorithms, we take advantage of the fact that graph $G$ is connected, to reformulate (1.2) as follows:

$$\min_{x_i \in \mathbb{R}^n, i \in \mathcal{N}} \left\{ \sum_{i=1}^{N} \left( F_i(x_i) : x_i = x_j, \, \forall (i, j) \in \mathcal{E} \right) \right\}. \quad (1.3)$$

Optimization problems of form (1.3) model a variety of very important applications, e.g. distributed linear regression [1], distributed control [2], machine learning [3], and estimation using sensor networks [4]. Consequently, a number of different distributed optimization algorithms have been proposed to solve (1.3). Duchi et al. [5] proposed a dual averaging algorithm to solve (1.2) in a distributed fashion over $G$ when each $F_i$ is only assumed to be convex. This algorithm computes $\epsilon$-optimal solution $\left\{ \bar{x}_i \right\}_{i \in \mathcal{N}}$, i.e. $F(x_i) - F^* \leq \epsilon$ for all $i \in \mathcal{N}$, in $O(1/\epsilon^2)$ iterations; however, they do not provide any guarantees on the consensus violation $\max\{\|x_i - x_j\|_2 : (i, j) \in \mathcal{E}\}$. Nedić et al. [6] developed a subgradient method with constant step size $\alpha > 0$ for distributed minimization
of (1.2) where the network topology is time-varying. Setting \( \alpha = \mathcal{O}(\epsilon) \) in their method guarantees that consensus violation and suboptimality is \( \mathcal{O}(\epsilon) \) in \( \mathcal{O}(1/\epsilon^2) \) iterations; however, since the step size is constant both errors are not guaranteed to decrease further with more iterations. Wei and Ozdaglar [7] proposed an alternating direction method of multipliers (ADMM) algorithm that computes an \( \epsilon \)-optimal and \( \epsilon \)-feasible (consensus violation is at most \( \epsilon \)) solution in \( \mathcal{O}(1/\epsilon) \) proximal map evaluations for each \( F_i \); thus, the overall complexity of the method depends on the complexity of the proximal map evaluations. Note that there are many problems where one can compute the proximal map for \( \rho_i \) easily; but the proximal map for \( F_i = \rho_i + \gamma_i \) is hard. When each \( F_i \) is smooth and has bounded gradients over \( \mathbb{R}^n \), Dušan et al. [8] developed a fast distributed gradient methods with \( \mathcal{O}(1/\sqrt{\gamma}) \) convergence rate. Note that boundedness assumption in [8] does not hold for most commonly used quadratic loss functions. Chen et al. [9] proposed an inexact proximal-gradient method for distributed minimization of (1.2) over a time-varying network topology when \( F_i = \rho + \gamma_i \) where the possibly non-smooth term \( \rho \) is the same at all nodes, and \( \nabla \gamma_i \) is bounded for all \( i \in \mathcal{N} \). The method in [9] can compute \( \epsilon \)-feasible and \( \epsilon \)-optimal solution in \( T = \mathcal{O}(\epsilon^{-1}) \) iterations which require \( \mathcal{O}(T^2) = \mathcal{O}(\epsilon^{-2}) \) communication steps. In contrast, the method proposed in this paper is able to compute an \( \epsilon \)-optimal \( \epsilon \)-feasible solution in \( \mathcal{O}(\epsilon^{-1}) \) communication steps, allowing node specific non-smooth functions \( \rho_i \), and without assuming bounded \( \nabla \gamma_i \) for any \( i \in \mathcal{N} \).

Aybat and Iyengar [10] proposed an efficient first-order augmented Lagrangian (FAL) algorithm for the basis pursuit problem \( \min_{x \in \mathbb{R}^n} \{ \|x\|_1 : Ax = b \} \) to compute an \( \epsilon \)-optimal and \( \epsilon \)-feasible solution to within \( \mathcal{O}(\kappa^2(A)/\epsilon) \) matrix-vector multiplications, where \( A \in \mathbb{R}^{m \times n} \) such that \( \text{rank}(A) = m \), and \( \kappa(A) := \sigma_{\max}(A)/\sigma_{\min}(A) \) denotes the condition number of \( A \). In this work, we extend their FAL algorithm to solve a more general version of (1.3) in Section 2.2.1 and 2.2.2 and establish the following result for (1.3) in Section 2.2.3.

**Main Theorem.** Let \( \{x_i^{(k)}\}_{k \in \mathbb{Z}_+} \) denote the sequence of FAL iterates for \( i \in \mathcal{N} \). Then \( F^* = \lim_{k \in \mathbb{Z}_+} \sum_{i \in \mathcal{N}} F_i(x_i^{(k)}) \). Furthermore, for all \( \epsilon > 0 \),

\[
\left| \sum_{i \in \mathcal{N}} F_i(x_i^{(k)}) - F^* \right| \leq \epsilon, \quad \text{and} \quad \max_{(i,j) \in \mathcal{E}} \left\{ \|x_i^{(k)} - x_j^{(k)}\|_2 \right\} \leq \epsilon,
\]

within \( \mathcal{O}(\log(1/\epsilon)) \) FAL iterations, requiring at most \( \mathcal{O}(\frac{\psi_1^{1/5}}{d_{\min}} \epsilon^{-1}) \) communication steps, where \( \psi_1 \) denotes the largest eigenvalue of the Laplacian corresponding to \( \mathcal{G} \), and \( d_{\min} \) denotes the degree of smallest degree node in \( \mathcal{G} \).

Given a connected graph \( \mathcal{G} = (\mathcal{N}, \mathcal{E}) \) with \( N \) nodes, let \( \psi_1 \geq \psi_2 \geq \ldots \geq \psi_{N-1} \geq \psi_N = 0 \) denote the eigenvalues of the Laplacian corresponding to \( \mathcal{G} \). Since \( \mathcal{G} \) is connected the second smallest eigenvalue of the Laplacian, \( \psi_{N-1} > 0 \) shows the strength of connectivity and is called
the spectral gap of $G$. Let $d_i$ denote the degree of node $i \in \mathcal{N}$. Since $\psi_{N-1} \leq d_{\min} := \min_{i \in \mathcal{N}} d_i$, the main theorem also implies that the number of communication steps can be bounded above by $O\left(\frac{\psi_{N-1}^4}{\psi_{N-1} \epsilon^{-1}}\right)$.

In Section 2.2.4, we also propose an asynchronous version of FAL, where a given node does not need to wait for every other node to finish updating their local variables. It is important to note that FAL can be easily extended to solve (1.3) when there are global side constraints of the form $Ex - q \in \mathcal{K}$, where $\mathcal{K}$ is a proper cone. Note that none of the other algorithms discussed above can accommodate such conic constraints efficiently. However, due to space limitations, we do not discuss this extension in this paper.

## 2 Methodology

**Definition 1.** Let $\Gamma$ denote the set of convex functions $\gamma : \mathbb{R}^n \to \mathbb{R}$ such that $\gamma$ has Lipschitz continuous gradient $\nabla \gamma$ with constant $L_\gamma$, and $\gamma$ is bounded below, i.e. there exists $\gamma \in \mathbb{R}$ such that $\gamma(x) \geq \gamma$ for all $x \in \mathbb{R}^n$. Let $\mathcal{R}$ denote the set of convex functions $\rho : \mathbb{R}^n \to \mathbb{R}$ such that subdifferential of $\rho$ is uniformly bounded on $\mathbb{R}^n$, i.e. there exists $B > 0$ s.t. $\|q\|_2 \leq B$ for all $q \in \partial \rho(x)$, $x \in \mathbb{R}^n$, and $\rho(x)$ is lower bounded by the $\ell_2$-norm, i.e. there exists $\tau > 0$ such that $\tau \|x\|_2 \leq \rho(x)$ for all $x \in \mathbb{R}^n$.

**Assumption 1.** For all $i \in \mathcal{N}$, we assume that $\gamma_i \in \Gamma$ and $\rho_i \in \mathcal{R}$ with corresponding constants $L_{\gamma_i}, \gamma_i, B_i$ and $\tau_i$.

Note that many important regularizers and loss functions used in the machine learning and statistics literature lie in $\mathcal{R}$ and $\Gamma$, respectively. In particular, $\rho \in \mathcal{R}$ can be any norm in a finite dimensional vector space, e.g. $\|\cdot\|_\alpha$ with $\alpha \in \{1, 2, \infty\}$, group norm (see Section 3), nuclear norm, etc., weighted sum of these norms, e.g. sparse group norm (see Section 3); $\gamma \in \Gamma$ can be quadratic-loss $\|Ax - b\|_2^2$, Huber-loss $\sum_{i=1}^m h(a_{i}^T x - b_i)$ (see Section 3), logistic-loss $\sum_{i=1}^m \log \left(1 + e^{-b^T a_i^T x}\right)$, or fair-loss [11] functions for given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

For a convex function $\rho : \mathbb{R}^n \to \mathbb{R}$, we define $\text{prox}_\rho(x) := \arg\min_{y \in \mathbb{R}^n} \rho(y) + \frac{1}{2}\|y - x\|^2_2$. Throughout the paper, we adopt the notation $x = (x_i; x_{-i})$ with $x_{-i} = (x_j)_{j \neq i}$ to denote a vector where $x_i$ and $x_{-i}$ are treated as variable and parameter sub-vectors of $x$, respectively. Given $f : \mathbb{R}^{nN} \to \mathbb{R}$, $\nabla x_i f(x) \in \mathbb{R}^n$ denotes the sub-vector of $\nabla f(x) \in \mathbb{R}^{nN}$ corresponding to components of $x_i \in \mathbb{R}^n$.

### 2.1 Proximal Gradient Algorithm for Centralized Model

In this section, we consider the centralized version of (1.2) where all the functions $F_i$ are available at a centralized node, and all computations are carried out at this node. Suppose $\{\rho_i\}_{i \in \mathcal{N}}$
and \( \{\gamma_i\}_{i \in \mathcal{N}} \) satisfy Assumption \( \ref{assumption} \). Let \( \rho(x) := \sum_{i=1}^{N} \rho_i(x) \) and \( \gamma(x) := \sum_{i=1}^{N} \gamma_i(x) \). Lipschitz continuity of each \( \nabla \gamma_i \) with constant \( L_{\gamma_i} \) implies that \( \nabla \gamma \) is also Lipschitz continuous with constant \( L_\gamma = \sum_{i=1}^{N} L_{\gamma_i} \). When \( \text{prox}_{\rho/L_\gamma} \) can be computed efficiently, the accelerated proximal gradient (APG) algorithm proposed in \[12\] guarantees that

\[
0 \leq F(x^{(\ell)}) - F^* \leq \frac{2L_\gamma}{(\ell + 1)^2} \|x^{(0)} - x^*\|_2^2,
\]

where \( x^{(0)} \) is the initial APG iterate and \( x^* \in \text{argmin}_{x \in \mathbb{R}^n} F(x) \) (see, Corollary 3 in \[13\], and Theorem 4.4 in \[12\]). Thus, using APG one can compute an \( \epsilon \)-optimal to \( (1.2) \) within \( O(\sqrt{L_\gamma \epsilon^{-1}}) \) iterations.

As mentioned above, the centralized APG algorithm cannot be applied when the nodes are unwilling or unable to communicate the privately known functions \( \{F_i\}_{i \in \mathcal{N}} \) to a central node. There are many other setting where one may want to solve \( (1.2) \) as a “distributed” problem. For instance, although \( \text{prox}_{t\rho_i} \) can be computed efficiently for all \( t > 0 \) and \( i \in \mathcal{N} \), \( \text{prox}_{\rho/L_\gamma} \) may be hard to compute. As an example, consider a problem with \( \rho_1(X) = \sum_{i,j} |X_{ij}| \) and \( \rho_2 = \sum_{i=1}^{\text{rank}(X)} \sigma_i(X) \), where \( \sigma(X) \) denotes the vector of singular values for \( X \in \mathbb{R}^{n_1 \times n_2} \). Then \( \text{prox}_{t\rho_i} \) is easy to compute for all \( t > 0 \) and \( i \in \{1, 2\} \); however, \( \text{prox}_{\rho_i} \) is hard to compute. Thus, the “centralized” APG algorithm cannot be applied. In the rest of this paper, we focus on decentralized algorithms.

### 2.2 FAL Algorithm for Decentralized Model

Let \( x = (x_1^T, \ldots, x_N^T)^T \in \mathbb{R}^{nN} \) denotes a vector formed by concatenating \( \{x_i\}_{i \in \mathcal{N}} \subset \mathbb{R}^n \) as a long column vector. Consider the following optimization problem of the form

\[
\bar{F}^* := \min_{x \in \mathbb{R}^{nN}} \bar{F}(x) := \bar{\rho}(x) + \bar{\gamma}(x) \quad \text{s.t.} \quad Ax = b;
\]

where \( \bar{\rho}(x) := \sum_{i=1}^{N} \rho_i(x_i) \), \( \bar{\gamma}(x) := \sum_{i=1}^{N} \gamma_i(x_i) \), and \( A \in \mathbb{R}^{m \times nN} \) such that \( \text{rank}(A) = m \), i.e. the linear map is surjective. Later in Section 2.2.3 we will show that the distributed optimization problem in \( (1.3) \) is a special case of \( (2.2) \). In the rest of the section, we will use the following notation: Let \( \{A_i\}_{i \in \mathcal{N}} \subset \mathbb{R}^{m \times n} \) such that \( A = [A_1, A_2, \ldots, A_N] \); \( \bar{L} := \max_{i \in \mathcal{N}} L_{\gamma_i} \), \( \bar{\tau} := \min_{i \in \mathcal{N}} \tau_i \).

We propose to solve \( (2.2) \) by inexactly solving a sequence of subproblems of the form,

\[
x_i^{(k)} \in \text{argmin}_{x \in \mathbb{R}^n} P^{(k)}(x) := \lambda^{(k)} \bar{\rho}(x) + f^{(k)}(x),
\]

\[
(2.3)
\]
for appropriately chosen sequences of penalty parameters \( \{\lambda^{(k)}\} \) and dual variables \( \{\theta^{(k)}\} \) such that \( \lambda^{(k)} \searrow 0 \), where \( f^{(k)}(x) := \lambda^{(k)} \gamma(x) + \frac{1}{2} \|Ax - b - \lambda^{(k)} \theta^{(k)}\|^2_2 \). Note that \( \nabla f^{(k)}(x) \) is Lipschitz continuous in \( x \in \mathbb{R}^{nN} \) with constant \( \lambda^{(k)} \tilde{L} + \sigma^2_{\max}(A) \). In particular, given \( \{\alpha^{(k)}, \xi^{(k)}\} \) such that \( \alpha^{(k)} \searrow 0 \) and \( \xi^{(k)} \searrow 0 \), the iterate sequence \( \{x^{(k)}\} \) is constructed such that every \( x^{(k)} \) satisfies one of the following conditions:

\[
\begin{align*}
(a) \quad & P^{(k)}(x^{(k)}) - P^{(k)}(x^*_k) \leq \alpha^{(k)}, \\
(b) \quad & \exists g_i^{(k)} \in \partial x_i P^{(k)}(x)|_{x=x^{(k)}} \text{ s.t. } \max_{i \in N} \|g_i^{(k)}\|_2 \leq \xi^{(k)}/\sqrt{N},
\end{align*}
\]

where \( \partial x_i P^{(k)}(x)|_{x=x^{(k)}} := \lambda^{(k)} \partial \rho_i(x_i)|_{x_i=\bar{x}_i} + \nabla x_i f^{(k)}(\bar{x}) \). Given \( \{x^{(0)}, \lambda^{(0)}, \alpha^{(0)}, \xi^{(0)}\} \) and \( 0 < c < 1 \), we choose the sequence \( \{\lambda^{(k)}, \alpha^{(k)}, \xi^{(k)}, \theta^{(k)}\} \) according to the FAL algorithm shown in Fig. 1.

**Algorithm FAL \((\lambda^{(1)}, \alpha^{(1)}, \xi^{(1)})\)**

**Step 0:** Set \( \theta^{(1)} = 0, k = 1 \)

**Step k:** \( (k \geq 1) \)

1. Compute \( x^{(k)} \) such that (2.4a) or (2.4b) holds

2. \( \theta^{(k+1)} = \theta^{(k)} - \frac{Ax^{(k)} - b}{\lambda^{(k)}} \)

3. \( \lambda^{(k+1)} = c\lambda^{(k)}, \quad \alpha^{(k+1)} = c^2 \alpha^{(k)}, \quad \xi^{(k+1)} = c^2 \xi^{(k)} \)

Figure 1: First-order Augmented Lagrangian (FAL) algorithm

In Section 2.2.1 we show that FAL computes an \( \epsilon \)-optimal and \( \epsilon \)-feasible solution \( x_\epsilon \) to (2.2), i.e., \( \|Ax_\epsilon - b\|_2 \leq \epsilon \) and \( |\bar{F}(x_\epsilon) - F^*| \leq \epsilon \), in at most \( O(\log(1/\epsilon)) \) iterations. Next, in Section 2.2.2 we show that computing an \( \epsilon \)-optimal, \( \epsilon \)-feasible solution requires at most \( O\left(\frac{\sigma^2_{\max}(A)}{\min_{i \in N} \sigma^2_{\min}(A_i)} \epsilon^{-1}\right) \) floating point operations. Using this result, in Section 2.2.3 we establish the main theorem given in Section 1. Finally, in Section 2.2.4 we show how to modify FAL for an asynchronous computation setting.

### 2.2.1 FAL iteration complexity

Here, we show that \( \{x^{(k)}\} \) is a bounded sequence, and then argue that this also implies boundedness of \( \{\theta^{(k)}\} \). First, we start with technical lemma that will be used in establishing the main results of this section.

**Lemma 1.** Let \( \bar{\rho} : \mathbb{R}^{nN} \to \mathbb{R} \) be defined as \( \bar{\rho}(x) = \sum_{i \in N} \rho_i(x_i) \), where for all \( i \in N \), \( \rho_i : \mathbb{R}^n \to \mathbb{R} \) is a convex function such that for some \( B_i > 0 \), \( \|q\|_2 \leq B_i \) for all \( q \in \partial \rho_i(x) \) and for all \( x \in \mathbb{R}^n \). Let \( f : \mathbb{R}^{nN} \to \mathbb{R} \) be a convex function such that for some \( \{L_i\}_{i=1}^N \subset \mathbb{R}_{++} \), \( f(y) \leq f(\bar{y}) + \nabla f(\bar{y})^T(y - \bar{y}) + \sum_{i=1}^N \frac{L_i\|y_i - \bar{y}_i\|^2_2}{2} \) for all \( y, \bar{y} \in \mathbb{R}^{nN} \). Given \( \alpha, \lambda \geq 0 \), and
\( \bar{x} \in \mathbb{R}^{nN} \) such that \( \lambda \rho(\bar{x}) + f(\bar{x}) - \min_{x \in \mathbb{R}^{nN}} \{ \lambda \rho(x) + f(x) \} \leq \alpha \), it follows that for all \( i \in \mathcal{N} \)
\[
\| \nabla x_i f(\bar{x}) \|_2 \leq \sqrt{2L_i \alpha + \lambda B_i}.
\]

**Proof.** Let \( x \in \mathbb{R}^{nN} \) and \( g_i \in \partial \rho_i(x_i) \) for all \( i \in \mathcal{N} \). From convexity of \( \rho_i \) and Cauchy-Schwarz, it follows that \( \rho_i(x_i) \leq \rho(\bar{x}_i) + \|g_i\|_2 + \|x_i - \bar{x}_i\|_2 \) for all \( i \in \mathcal{N} \). Hence, we have
\[
\lambda \rho(x) + f(x) \leq \lambda \rho(\bar{x}) + f(\bar{x}) + \sum_{i \in \mathcal{N}} \left( \lambda B_i \|x_i - \bar{x}_i\|_2 + \nabla x_i f(\bar{x})^T (x_i - \bar{x}_i) + \frac{L_i}{2} \|x_i - \bar{x}_i\|_2^2 \right).
\]

Minimizing on both sides and using the separability of the right side, we have \( \min_{x \in \mathbb{R}^{nN}} \lambda \rho(x) + f(x) \leq \lambda \rho(\bar{x}) + f(\bar{x}) + \sum_{i \in \mathcal{N}} \left( \lambda B_i \|x_i - \bar{x}_i\|_2 + \nabla x_i f(\bar{x})^T (x_i - \bar{x}_i) + \frac{L_i}{2} \|x_i - \bar{x}_i\|_2^2 \right) \).

Let \( \mathcal{I} := \{ i \in \mathcal{N} : \| \nabla x_i f(\bar{x}) \|_2 \leq \lambda B_i \} \). For each \( i \in \mathcal{N} \), there are two possibilities.

**Case 1:** Suppose that \( i \in \mathcal{I} \), i.e. \( \| \nabla x_i f(\bar{x}) \|_2 \leq \lambda B_i \). Since \( \min_{x \in \mathbb{R}^{nN}} h_i(x_i) \) has a unique solution, and \( -\nabla x_i f(\bar{x}) \in \lambda B_i \partial \|x_i - \bar{x}_i\|_2 \) when \( \| \nabla x_i f(\bar{x}) \|_2 \leq \lambda B_i \), it follows that \( \bar{x}_i = x_i \)

From the first-order optimality condition, we have \( \nabla x_i f(\bar{x}) + L_i (\bar{x}_i - x_i) + \lambda B_i \partial \|x_i - \bar{x}_i\|_2 \) = 0.

Let \( s_i := x_i - \bar{x}_i \) and \( t_i := \| x_i - \bar{x}_i \|_2 \), then \( s_i = -\frac{\nabla x_i f(\bar{x})}{\| \nabla x_i f(\bar{x}) \|_2} \). Since \( \| s_i \|_2 = 1 \), it follows that \( t_i = \frac{\| \nabla x_i f(\bar{x}) \|_2 - \lambda B_i}{L_i} \) > 0, and \( s_i = -\frac{\nabla x_i f(\bar{x})}{\| \nabla x_i f(\bar{x}) \|_2} \). Hence, \( \bar{x}_i = x_i - \frac{\| \nabla x_i f(\bar{x}) \|_2 - \lambda B_i}{L_i} \| \nabla x_i f(\bar{x}) \|_2 \), and

\[
\lambda \rho(x) + f(x) \leq \lambda \rho(\bar{x}) + f(\bar{x}) + \min_{x \in \mathbb{R}^{nN}} \lambda \rho(x) + f(x) \leq \alpha,
\]

which implies that \( \| \nabla x_i f(\bar{x}) \|_2 \leq \sqrt{2L_i \alpha + \lambda B_i} \) for all \( i \in \mathcal{I} \). Moreover, \( \| \nabla x_i f(\bar{x}) \|_2 \leq \lambda B_i \) for all \( i \in \mathcal{I}^c \). Hence, the result follows from these two inequalities.

In the following lemma, we show that \( f^{(k)} \) in the subproblem (2.3) satisfies the condition given in Lemma 1.

**Lemma 2.** Given \( k \geq 1 \), let \( L_i^{(k)} := \lambda^{(k)} L_{\gamma_i} + \sigma_{\max}^2 (A) \) for all \( i \in \mathcal{N} \). For all \( x, \bar{x} \in \mathbb{R}^{nN} \), \( f^{(k)} \) in (2.3) satisfies \( f^{(k)}(x) \leq f^{(k)}(\bar{x}) + \nabla f^{(k)}(x - \bar{x}) + \sum_{i=1}^{nN} \frac{L_i^{(k)}}{2} \|x_i - \bar{x}_i\|_2^2 \).

**Proof.** For all \( i \in \mathcal{N} \), since \( \nabla \gamma_i \) is Lipschitz continuous with constant \( L_{\gamma_i} \), for any \( x, \bar{x} \in \mathbb{R}^{nN} \),
Lemma 3. Let \( \{x^{(k)}\} \) be the FAL iterate sequence, i.e. at least one of the conditions in (2.4) hold for all \( k \geq 1 \). Then every element of the dual sequence satisfies

\[
\|\theta^{(k+1)}\|_2 \leq \min_{i \in N} \left\{ \frac{\max \left\{ \sqrt{2L_i^{(k)}} \frac{\alpha_i^{(k)}}{\lambda_i^{(k)}}^2, \frac{1}{\sqrt{N}} \xi_i^{(k)} \right\}}{\sigma_{\min}(A_i)} + B_i + \|\nabla \gamma_i(x_i^{(k)})\|_2 \right\}. 
\]

Proof. Fix \( k \geq 1 \). Suppose that \( x^{(k)} \) satisfies (2.4)(a). Then Lemma 1 implies that for all \( i \in N \)

\[
\|\nabla f_i^{(k)}(x^{(k)})\|_2 = \|\lambda_i^{(k)} \nabla \gamma_i(x_i^{(k)}) + A_i^T (Ax^{(k)} - b - \lambda_i^{(k)} \theta^{(k)})\|_2 \leq 2L_i^{(k)} (\alpha_i^{(k)} + \lambda_i^{(k)} B_i).
\]

Now, suppose that \( x^{(k)} \) satisfies (2.4)(b). Then triangular inequality immediately implies that \( \|\nabla f_i^{(k)}(x^{(k)})\|_2 \leq \xi_i^{(k)}/\sqrt{N} + \lambda_i^{(k)} B_i \) for all \( i \in N \). Combining the two inequalities, and further using triangular Cauchy-Schwarz inequalities, it follows for all \( i \in N \) that \( \|Ax^{(k)} - b - \lambda_i^{(k)} \theta^{(k)}\|_2 \leq \max \{ \sqrt{2L_i^{(k)} \alpha_i^{(k)} \xi_i^{(k)} / \sqrt{N}} + \lambda_i^{(k)} (B_i + \|\nabla \gamma_i(x_i^{(k)})\|_2) \} / \sigma_{\min}(A_i) \). Hence, we conclude by diving the above inequality by \( \lambda_i^{(k)} \) and using the definition of \( \theta^{(k+1)} \).

The next result establishes that the FAL iterate sequence \( \{x^{(k)}\} \) is bounded whenever \( \{\rho_i, \gamma_i\}_{i \in N} \) satisfy Assumption 1; therefore, the sequence of dual variables \( \{\theta^{(k)}\} \) is bounded according to Lemma 5.

Theorem 1. Suppose Assumption 7 holds. Then there exist constants \( B_x, B_\theta, \bar{\lambda} > 0 \) such that \( \max \{\|x^*_i\|_2, \|x^{(k)}\|_2\} \leq B_x \) and \( \|\theta^{(k)}\|_2 \leq B_\theta \) for all \( k \geq 1 \), whenever \( \lambda^{(1)} \) and \( \xi^{(1)} \) are chosen such that \( 0 < \lambda^{(1)} \leq \bar{\lambda} \) and \( \xi^{(1)} \bar{\lambda} < \bar{r} \).

Proof. Let \( A = A_1, A_2, \ldots, A_N \) \( \in \mathbb{R}^{m \times nN} \) such that \( A_i \in \mathbb{R}^{m \times n} \) for all \( i \in N \). Throughout the proof we assume that \( \sigma_{\max}(A) \geq \sqrt{\max_{i \in N} d_i + 1} \), and \( \sigma_{\min}(A_i) = \sqrt{d_i} \geq 1 \) for all \( i \in N \),
where \( d_i \geq 1 \) is the degree of \( i \in \mathcal{N} \). Indeed, when \( A \) is chosen as described in Section 2.2.3 corresponding to graph \( \mathcal{G} \), we showed that \( \sigma_{\max}^2(A) = \psi_1 \), where \( \psi_1 \) is the largest eigenvalue of the Laplacian \( \Omega \) corresponding to \( \mathcal{G} \). It is shown in [14] that when \( \mathcal{G} \) is connected, one has \( \psi_1 \geq \max_{i \in \mathcal{N}} d_i + 1 > 1 \). Hence, \( \sigma_{\max}(A) \geq \sqrt{\max_{i \in \mathcal{N}} d_i + 1} > 1 \). Moreover, for \( A \) chosen as described in Section 2.2.3 corresponding to graph \( \mathcal{G} \), we also showed that \( \sigma_{\min}(A_i) = \sqrt{d_i} \) for all \( i \in \mathcal{N} \).

To keep notation simple, without loss of generality, we assume that \( \gamma_i = 0 \) for all \( i \in \mathcal{N} \). Hence, \( \gamma(x) \geq 0 \) for all \( x \in \mathbb{R}^{nN} \). Let \( x^* \) be a minimizer of (2.2). By Lipschitz continuity of \( \nabla \gamma_i \), we have for all \( i \in \mathcal{N} \)

\[
\| \nabla \gamma_i(x_i) \|_2 \leq L_{\gamma_i} \| x_i - x_i^* \|_2 + \| \nabla \gamma_i(x_i^*) \|_2. \tag{2.6}
\]

We prove the theorem using induction. We show that, for an appropriately chosen bound \( R \), \( \| x^{(k)} - x^* \|_2 \leq R \) implies that \( \| x^{(k+1)} - x^* \|_2 \leq 2 R \) for all \( k \geq 1 \). Fix \( k \geq 1 \). First, suppose that \( x^{(k+1)} \) satisfies (2.4)(a), i.e. \( P^{(k+1)}(x^{(k+1)}) \leq P^{(k+1)}(x^*) + \alpha^{(k+1)} \). By dividing both sides by \( \lambda^{(k+1)} \), it follows from Assumption 1, \( Ax^* = b \), and \( f^{(k+1)}(\cdot) \geq 0 \) that

\[
\bar{\tau} \| x^{(k+1)} \|_2 \leq \bar{\rho}(x^*) + \bar{\gamma}(x^*) + \frac{\lambda^{(k+1)}}{2} \left( \| \theta^{(k+1)} \|_2^2 + \frac{\alpha^{(k+1)}}{\lambda^{(k+1)}} \right). \tag{2.7}
\]

Next, suppose \( x^{(k+1)} \) satisfies (2.4)(b). It follows from convexity of \( P^{(k+1)} \) and Cauchy-Schwarz inequality that \( P^{(k+1)}(x^{(k+1)}) \leq P^{(k+1)}(x^*) + \xi^{(k+1)} \| x^{(k+1)} - x^* \|_2 \). Again, dividing both sides by \( \lambda^{(k+1)} \), we get

\[
\bar{\tau} \| x^{(k+1)} \|_2 \leq \bar{\rho}(x^*) + \bar{\gamma}(x^*) + \frac{\lambda^{(k+1)}}{2} \| \theta^{(k+1)} \|_2^2 + \frac{\xi^{(k+1)}}{\lambda^{(k+1)}} \| x^{(k+1)} - x^* \|_2. \tag{2.8}
\]

Combining the bounds for both cases, (2.7) and (2.8), and using triangular inequality, we have

\[
\left( \bar{\tau} - \frac{\xi^{(k+1)}}{\lambda^{(k+1)}} \right) \| x^{(k+1)} - x^* \|_2 \leq F^{(k+1)} + \bar{\tau} \| x^* \|_2 + \frac{\lambda^{(k+1)}}{2} \left( \| \theta^{(k+1)} \|_2^2 + \frac{\alpha^{(k+1)}}{\lambda^{(k+1)}} \right), \tag{2.9}
\]

for all \( k \geq 0 \). Note that \( \{ \lambda^{(k)}, \alpha^{(k)}, \xi^{(k)} \} \) is chosen in FAL such that \( \frac{\alpha^{(k)}}{\lambda^{(k)}} = \frac{\alpha^{(1)}}{\lambda^{(1)}} \) for all \( k > 1 \), and both \( \frac{\xi^{(k)}}{\lambda^{(k)}} \searrow 0 \) and \( \lambda^{(k)} \searrow 0 \) monotonically. Since \( \sigma_{\min}(A_i) \geq 1 \) for all \( i \in \mathcal{N} \), the inductive assumption \( \| x^{(k)} - x^* \|_2 \leq R \), (2.6), and Lemma 3 together imply that

\[
\| \theta^{(k+1)} \|_2 \leq \min_{i \in \mathcal{N}} \left\{ \max \left\{ \sqrt{2 \left( \frac{L_i^{(1)}}{\lambda^{(1)}} \right)}, \frac{\xi^{(1)}}{\lambda^{(1)}} \right\} + B_i + \| \nabla \gamma_i(x_i^*) \|_2 + L_{\gamma_i} R \right\}. \tag{2.10}
\]
To simplify bounds further, choose \( \alpha^{(1)} = \frac{1}{4N} \left( \lambda^{(1)} \bar{\tau} \right)^2 \), and \( \xi^{(1)} = \frac{1}{2} \lambda^{(1)} \bar{\tau} \) for \( \lambda^{(1)} \leq \sigma^2_{\max}(A)/\bar{L} \), where \( \bar{L} = \max_{i \in \mathcal{N}} \{ L_{\gamma_i} \} \). Let \( \bar{B} := \max_{i \in \mathcal{N}} B_i \) and \( \bar{G} := \max \{ \| \nabla \gamma_i(x^*_i) \| : i \in \mathcal{N} \} \). Together with (2.9), (2.10) and \( \sigma_{\max}(A) \geq 1 \), this choice of parameters implies that

\[
\frac{\bar{\tau}}{2} \| x^{(k+1)} - x^* \|_2 \leq \bar{F}^* + \bar{\tau} \| x^* \|_2 + \frac{\lambda^{(1)}}{2} \left[ \left( \frac{\bar{\tau} \sigma_{\max}(A)}{\sqrt{N}} + \bar{B} + \bar{G} + \bar{L}R \right)^2 + \frac{\bar{\tau}^2}{4N} \right].
\]

Define \( \beta_1 := \frac{2}{\bar{\tau}} \left( \bar{F}^* + \bar{\tau} \| x^* \|_2 \right) \), \( \beta_2 := \frac{\bar{\tau} \sigma_{\max}(A) / \sqrt{N} + \bar{B} + \bar{G}}{\bar{\tau}} \), \( \beta_3 := \frac{\bar{\tau}}{2} \), and \( \beta_4 := \frac{\bar{\tau}}{4N} \). Then we have that

\[
\| x^{(k+1)} - x^* \|_2 \leq \beta_1 + \lambda^{(1)} \left[ \left( \beta_2 + \beta_3 R \right)^2 + \beta_4 \right].
\]

Note that we are free to choose any \( \lambda^{(1)} > 0 \) satisfying \( \lambda^{(1)} \leq \sigma^2_{\max}(A)/\bar{L} \). Our objective is to show that by appropriately choosing \( \lambda^{(1)} \), we can guarantee that \( \beta_1 + \lambda^{(1)} \left[ \left( \beta_2 + \beta_3 R \right)^2 + \beta_4 \right] \leq R \), which would then complete the inductive proof. This is indeed true if the above quadratic inequality in \( R \), has a solution, or equivalently if the discriminant

\[
\Delta = (2 \lambda^{(1)} \beta_2 \beta_3 - 1)^2 - 4 \lambda^{(1)} \beta_3 \left[ \lambda^{(1)} \beta_2^2 + \beta_4 + \beta_1 \right]
\]

is non-negative. Note that \( \Delta \) is continuous in \( \lambda^{(1)} \), and \( \lim_{\lambda^{(1)} \to 0} \Delta = 1 \). Thus, for all sufficiently small \( \lambda^{(1)} > 0 \), we have \( \Delta \geq 0 \). Hence, we can set \( R = \frac{1 - 2 \lambda^{(1)} \beta_2 \beta_3 - \sqrt{\Delta}}{2 \lambda^{(1)} \beta_2 \beta_3^2} \) for some \( \lambda^{(1)} > 0 \) such that \( \Delta \geq 0 \), and this will imply that \( \| x^{(k+1)} - x^* \|_2 \leq R \) whenever \( \| x^{(k)} - x^* \|_2 \leq R \) for all \( k \geq 1 \).

The induction will be complete if we can show that \( \| x^{(1)} - x^* \|_2 \leq R \). Note that in FAL we set \( \theta^{(1)} = 0 \). Hence, for \( k = 0 \), (2.9) implies that \( \| x^{(1)} - x^* \|_2 \leq \beta_1 + \lambda^{(1)} \beta_4 \). Hence, our choice of \( R \) guarantees that \( \| x^{(1)} - x^* \|_2 \leq R \). This completes the induction.

Following the same arguments leading to (2.9), it can also be shown that for all \( k \geq 0 \)

\[
\left( \bar{\tau} - \frac{\xi^{(k+1)}}{\lambda^{(k+1)}} \right) \| x^{*(k+1)} - x^* \|_2 \leq \bar{F}^* + \bar{\tau} \| x^* \|_2 + \frac{\lambda^{(k+1)}}{2} \| \theta^{(k+1)} \|_2.
\]

Therefore, we can conclude that \( \| x^{*(k)} - x^* \| \leq R \) for all \( k \geq 1 \) holds for the same \( R \) we selected above.

Note that \( \Delta \) is a concave quadratic of \( \lambda^{(1)} \) such that \( \Delta = 1 \) when \( \lambda^{(1)} = 0 \); hence, one of its roots is positive and the other one is negative. Moreover, \( R \leq \frac{1}{2 \lambda^{(1)} \beta_2 \beta_3^2} - \frac{\beta_2}{\beta_3} \) and the bound on \( R \) is decreasing in \( \lambda^{(1)} > 0 \). Hence, in order to get a smaller bound on \( R \), we will choose \( \lambda^{(1)} \) as the positive root of \( \Delta \). In particular, we set \( \lambda^{(1)} = \sqrt{(\beta_2 + \beta_3 \beta_4)^2 + \beta_4 - (\beta_2 + \beta_3 \beta_4)} / 2 \beta_3 \beta_4 \).

We are now ready to state the main result that will give the iteration complexity of FAL.

**Theorem 2.** Suppose Assumption [7] holds and \( \lambda^{(1)} \) and \( \xi^{(1)} \) are chosen according to Theorem [7]. Then the primal-dual iterate sequence \( \{ x^{(k)}, \theta^{(k)} \} \) generated by FAL satisfy
\[
\begin{align*}
(a) \quad & \|A x^{(k)} - b\|_2 \leq 2B_\theta \lambda^{(k)}, \\
(b) \quad & -\lambda^{(k)} \frac{\|\theta^{(k)}\|_2 + B_\theta^2}{2} \leq \bar{F}(x^{(k)}) - F^* \leq \lambda^{(k)} \left( \frac{B_\theta^2}{2} + \frac{\max \{a^{(1)}, \xi^{(1)}B_\theta\}}{\lambda^{(k)}} \right),
\end{align*}
\]

where \( \theta^* \) denotes any optimal dual solution to (2.2).

**Proof.** The proof directly follows from Theorem 3.3 in [15]. For the sake of completeness, we also provide the proof here. Let \( x^* \) denote an optimal solution to (2.2).

Note that (a) follows immediately from Cauchy-Schwarz and the definition of \( \theta^{(k+1)} \).

First, we prove the second inequality in (b). Suppose that \( x^{(k)} \) satisfies (2.4)(a), which implies that \( \bar{F}(x^{(k)}) + \frac{\lambda^{(k)}}{2} \|\theta^{(k)}\|_2^2 \leq \bar{F}(x^*) + \frac{\lambda^{(k)}}{2} \|\theta^{(k)}\|_2^2 + \frac{a^{(k)}}{\lambda^{(k)}}. \) Now, suppose that \( x^{(k)} \) satisfies (2.4)(b). From the convexity of \( P^{(k)} \) and Cauchy-Schwarz, it follows that

\[
\bar{F}(x^{(k)}) + \frac{\lambda^{(k)}}{2} \|\theta^{(k)}\|_2^2 \leq \bar{F}(x^*) + \frac{\lambda^{(k)}}{2} \|\theta^{(k)}\|_2^2 + \frac{\xi^{(k)}\|x^{(k)} - x^*\|_2}{\lambda^{(k)}}. 
\]

Therefore, for all \( k \geq 1 \), \( x^{(k)} \) satisfies the second inequality in (b) since it also satisfies

\[
\bar{F}(x^{(k)}) - F^* \leq \lambda^{(k)} \left( \frac{\|\theta^{(k)}\|_2^2 - \|\theta^{(k+1)}\|_2^2}{2} + \frac{\max \{a^{(k)}, \xi^{(k)}\|x^{(k)} - x^*\|_2\}}{\lambda^{(k)^2}} \right).
\]

Now, in order to prove the first inequality in (b), we will exploit the primal-dual relations of the following two pairs of problems:

\[
(P) : \min_{x \in \mathbb{R}^N} \{ \bar{F}(x) : Ax = b \}, \quad (D) : \max_{\theta \in \mathbb{R}^m} b^T \theta - \bar{F}^*(A^T \theta),
\]

\[
(P_k) : \min_{x \in \mathbb{R}^N} \lambda^{(k)} \bar{F}(x) + \frac{1}{2} \|Ax - b_k\|_2^2, \quad (D_k) : \max_{\theta \in \mathbb{R}^m} \lambda^{(k)} (b^T \theta - F^*(A^T \theta)) - \frac{\lambda^{(k)^2}}{2} h(\theta),
\]

where \( b_k := b + \lambda^{(k)} \theta^{(k)} \), \( h(\theta) := \|\theta - \theta^{(k)}\|_2^2 - \|\theta^{(k)}\|_2^2 \), and \( F^* \) denotes the convex conjugate of \( F \). Note that problem \((P_k)\) is nothing but the subproblem in (2.3). Therefore, from weak-duality between \((P_k)\) and \((D_k)\), it follows that

\[
P^{(k)}(x^{(k)}) = \lambda^{(k)} \bar{F}(x^{(k)}) + \frac{1}{2} \|Ax^{(k)} - b_k\|_2^2 \geq \lambda^{(k)} (b^T \theta^* - \bar{F}^*(A^T \theta^*)) - \frac{\lambda^{(k)^2}}{2} h(\theta^*).
\]

Note that from strong duality between \((P)\) and \((D)\), it follows that \( \bar{F}^* = \bar{F}(x^*) = b^T \theta^* - \bar{F}^*(A^T \theta^*) \). Therefore, dividing the above inequality by \( \lambda^{(k)} \), we obtain

\[
\bar{F}(x^{(k)}) - \bar{F}^* \geq -\frac{\lambda^{(k)}}{2} (\|\theta^*\|_2^2 - 2(\theta^*)^T \theta^{(k)}) + \|\theta^{(k+1)}\|_2^2) \geq -\frac{\lambda^{(k)}}{2} (\|\theta^*\|_2^2 + B_\theta^2).
\]

**Corollary 1.** The \( \text{FAL} \) iterates \( x^{(k)} \) are \( \epsilon \)-feasible, i.e. \( \|Ax^{(k)} - b\|_2 \leq \epsilon \), and \( \epsilon \)-optimal, \( |\bar{F}(x^{(k)}) - \bar{F}^*| \leq \epsilon \) for all \( k \geq N(\epsilon) \) and \( N(\epsilon) = \mathcal{O}(\log(\frac{1}{\epsilon})) \).
2.2.2 Overall computational complexity for the synchronous algorithm

Efficiency of FAL depends on the complexity of the oracle for Step 1 in Fig.1. In this section, we construct an oracle MS-APG that computes an \( x^{(k)} \) satisfying (2.4) within \( \mathcal{O}(1/\lambda^{(k)}) \) gradient and prox computations. This result together with Theorem 2 guarantees that for any \( \epsilon > 0 \), FAL can compute an \( \epsilon \)-optimal and \( \epsilon \)-feasible iterate within \( \mathcal{O}(\epsilon^{-1}) \) floating point operations. Following lemma gives the iteration complexity of the oracle MS-APG displayed in Fig.2.

**Lemma 4.** Let \( \bar{\rho} : \mathbb{R}^{nN} \to \mathbb{R} \) such that \( \bar{\rho}(x) = \sum_{i \in \mathcal{N}} \rho_i(x_i) \), where \( \rho_i : \mathbb{R}^n \to \mathbb{R} \) is a convex function for all \( i \in \mathcal{N} \), and \( f : \mathbb{R}^{nN} \to \mathbb{R} \) be a convex function such that for some \( \{L_i\}_{i=1}^N \subset \mathbb{R}_{++}^L \), \( f(y) \leq f(\bar{y}) + \nabla f(\bar{y})^T(y - \bar{y}) + \sum_{i=1}^N \frac{L_i\|y_i - \bar{y}_i\|^2}{2} \) for all \( y, \bar{y} \in \mathbb{R}^{nN} \). Suppose that \( y^* \in \text{argmin} \Phi(y) := \bar{\rho}(y) + f(y) \). Then the MS-APG iterate sequence \( \{y^{(l)}\}_{l \in \mathbb{Z}_+} \) satisfies

\[
0 \leq \Phi(y^{(l)}) - \min_{y \in \mathbb{R}^{nN}} \Phi(y) \leq \sum_{i=1}^N \frac{2L_i\|y_i^{(0)} - y_i^*\|^2}{(l + 1)^2}.
\]

(2.11)

**Proof.** (2.11) can be shown by adapting the proof of Theorem 4.4 in [12] for the case here.

**Algorithm MS-APG (\( \rho, f, y^{(0)} \))**

Step 0: Take \( \bar{y}^{(1)} = y^{(0)}, t^{(1)} = 1 \)

Step \( l: (l \geq 1) \)

1. \( y_i^{(l+1)} = \text{prox}_{\rho_i/L_i} \left( \frac{\bar{y}^{(l)}}{t^{(l)}} - \nabla y_i f(\bar{y}^{(l)})/L_i \right) \) \( \forall i \in \mathcal{N} \)

2. \( t^{(l+1)} = (1 + \sqrt{1 + 4\left(\frac{t^{(l)}}{2}\right)^2})/2 \)

3. \( \bar{y}^{(l+1)} = y^{(l)} + \frac{t^{(l+1)} - t^{(l)}}{t^{(l+1)}} \left( y^{(l)} - y^{(l-1)} \right) \)

![Figure 2: Multi Step - Accelerated Proximal Gradient (MS-APG) algorithm](image)

Consider the problem \( \Phi^* = \min \Phi(y) := \bar{\rho}(y) + f(y) \) defined in Lemma 4. Note that \( \nabla f \) is Lipschitz continuous with constant \( L = \max_{i \in \mathcal{N}} L_i \). Within MS-APG algorithm, for each \( i \in \mathcal{N} \), the step size taken in the \( i \)-th block-coordinate is \( 1/L_i \geq 1/L \). Suppose, on the other hand, that the APG algorithm [12, 13] is used as the oracle. Then the step size is determined by the global Lipschitz constant leading to a sub-optimality bound, \( \Phi(y^{(l)}) - \Phi^* \leq \frac{2L\|y^{(0)} - y^*\|^2}{(l + 1)^2} \), that is worse than the bound in (2.11). When \( \{L_i\}_{i \in \mathcal{N}} \) are close to each other, the performance of MS-APG and APG are almost the same; however, when \( \min_{i \in \mathcal{N}} L_i \gg 1 \), APG can only take very tiny steps; hence, MS-APG is likely to converge much faster in practice.

Since the subproblem (2.3) is in the form given in Lemma 4, the following lemma shows that we can efficiently compute \( x^{(k)} \) satisfying (2.4), i.e. Step 1 in FAL, by running MS-APG on the \( k \)-th subproblem in (2.3) starting from \( x^{(k-1)} \). The result immediately follows from Lemma 4.
Lemma 5. The iterate sequence \( \{y^{(f)}\}_{f \in \mathbb{Z}_+} \) generated when we call MS-APG(\( \lambda^{(k)} \), \( \tilde{\rho}, f^{(k)}, x^{(k-1)} \)) satisfies \( P^{(k)}(y^{(f)}) - P^{(k)}(x^{(k)}) \leq \alpha^{(k)} \), for all \( f \geq \sqrt{\frac{\sum_{i=1}^{N} 2L_{i}^{(k)} \|x_{i}^{(k-1)} - x_{i}^{(k)}\|_{2}^{2}}{\alpha^{(k)}}} - 1 \), where \( L_{i}^{(k)} \) is defined in Lemma 2 and \( [x_{i}^{(k)}]_{i} \) represents the \( i \)-th block of \( x^{(k)} \). Hence, one can compute \( x^{(k)} \) satisfying (2.4) within \( O(1/\lambda^{(k)}) \) MS-APG iterations.

Theorem 2 and Lemma 5 together imply that FAL can compute an \( \epsilon \)-feasible, and \( \epsilon \)-optimal solution to (2.2) within \( O(1/\epsilon) \) MS-APG iterations. Due to space considerations, we will only state and prove this result for the case where \( \nabla \gamma \) is bounded in \( \mathbb{R}^{n,N} \) as the bounds \( B_{y} \) and \( B_{x} \) are more simple for this case. Note that Huber-loss, logistic-loss, and fair-loss functions indeed have bounded gradients.

Theorem 3. Suppose that \( \exists G_{i} > 0 \) such that \( \|\nabla \gamma_{i}(x)\|_{2} \leq G_{i} \) for all \( x \in \mathbb{R}^{n} \) and for all \( i \in N \). Let \( N_{FAL}^{\sigma}(\epsilon) \) and \( N_{FAL}^{\lambda}(\epsilon) \) denote the number of FAL-iterations to compute an \( \epsilon \)-optimal, and an \( \epsilon \)-feasible solutions to (2.2). Let \( N^{(k)} \) denote MS-APG iteration number required to compute \( x^{(k)} \) satisfying (2.4). Then \( \sum_{k=1}^{N_{FAL}^{\sigma}(\epsilon)} N^{(k)} \leq O (\Theta^{2} \sigma_{\max}(A) \epsilon^{-1}) \), and \( \sum_{k=1}^{N_{FAL}^{\lambda}(\epsilon)} N^{(k)} = O (\Theta \sigma_{\max}(A) \epsilon^{-1}) \), where \( \Theta = \frac{\sigma_{\max}(A)}{\min_{i \in N} \sigma_{\min}(A)} \).

Proof. We assume that \( \sigma_{\max}(A) \geq \sqrt{\max_{i \in N} d_{i} + 1} \) and \( \sigma_{\min}(A) = \sqrt{d_{i} + 1} \) for all \( i \in N \), where \( d_{i} \) denotes the degree of \( i \in N \). As discussed in the proof of Theorem 1, this is a valid assumption for distributed optimization problem in (1.3). Let \( \theta^{*} \) denote an optimal dual solution to (2.2). Note that from the first-order optimality conditions for (2.2), we have \( 0 \in \nabla \gamma_{i}(x_{i}^{*}) + A_{i}^{T} \theta^{*} + \partial \rho_{i}(x_{i})|_{x_{i} = x_{i}^{*}} \); hence, \( \|A_{i}^{T} \theta^{*}\|_{2} \leq B_{i} + G_{i} \). Therefore, \( \|\theta^{*}\|_{2} \leq \min_{i \in N} \frac{B_{i} + G_{i}}{\sigma_{\min}(A)} \).

Given \( 0 < \lambda^{(1)} \leq \sigma_{\max}(A)/\bar{L} \), choose \( \alpha^{(1)}, \xi^{(1)} > 0 \) such that \( \alpha^{(1)} = \frac{1}{4N} \left( \lambda^{(1)} \bar{\tau} \right)^{2} \), and \( \xi^{(1)} = \frac{1}{2} \lambda^{(1)} \bar{\tau} \). Then Lemma 3 and \( \sigma_{\max}(A) \geq 1 \) together imply that for all \( k \geq 1 \)

\[
\|\theta^{(k)}\|_{2} \leq \min_{i \in N} \left\{ \frac{\bar{\tau} \sigma_{\max}(A) / \sqrt{N} + B_{i} + G_{i}}{\sigma_{\min}(A)} \right\} := B_{\theta}.
\]

Hence, note that \( \|\theta^{*}\|_{2} \leq B_{\theta} \).

To simplify notation, suppose that \( \lambda^{(1)} = \min \left\{ 1, \sigma_{\max}(A)/\bar{L} \right\} = 1 \). (2.9) implies that for all \( k \geq 1 \)

\[
\|x^{(k)} - x^{*}\|_{2} \leq \frac{2}{\bar{\tau}} \left[ \bar{F}^{*} + \bar{\tau} \|x^{*}\|_{2} + \frac{1}{2} \left( B_{\theta}^{2} + \bar{\tau}^{2} \right) \right] := B_{x}.
\]

Note that (2.13) implies that \( \frac{\xi^{(1)}}{(\lambda^{(1)})^{2}} B_{x} = \frac{1}{\lambda^{(1)}} \frac{\bar{\tau}}{\sqrt{N}} B_{x} \geq \frac{1}{2} B_{\theta}^{2} + \frac{\sigma_{\max}(A)^{2}}{8N} \geq \frac{\sigma_{\max}(A)^{2}}{8N} \geq \frac{\sigma_{\max}(A)^{2}}{8N} \geq \frac{\sigma_{\max}(A)^{2}}{8N} \). Note that the last inequality follows from our assumption on \( A \) stated at the beginning of the proof, i.e. \( \sigma_{\max}(A) \geq \sqrt{\max_{i \in N} d_{i} + 1} \) and \( \sigma_{\min}(A_{i}) = d_{i} \) for
all \( i \in \mathcal{N} \). Hence, Theorem\textsuperscript{2} \( \lambda^{(1)} = 1 \), and \( \|\theta^{*}\|_2 \leq B_{\theta} \) imply that

\[
N_{\text{FAL}}^f(\epsilon) \leq \log_\frac{1}{c} \left( \frac{2B_{\theta}}{\epsilon} \right) = \log_\frac{1}{c} \left( 2 \min_{i \in \mathcal{N}} \left\{ \frac{\bar{\tau} \sigma_{\max}(A) / \sqrt{N} + B_i + G_i}{\sigma_{\min}(A_i) \epsilon} \right\} \right) := N_f, \tag{2.14}
\]

\[
N_{\text{FAL}}^o(\epsilon) \leq \log_\frac{1}{c} \left( \frac{1}{\epsilon} \max \left\{ \frac{1}{2} \left( \|\theta^{*}\|_2 + B_{\theta} \right)^2, B_{\theta}^2 + \bar{F}^{*} + \bar{\tau} \|x^{*}\|_2 + \frac{\bar{\tau}^2}{8N} \right\} \right),
= \log_\frac{1}{c} \left( \frac{2B_{\theta}^2 + \bar{F}^{*} + \bar{\tau} \|x^{*}\|_2 + \frac{\bar{\tau}^2}{8N}}{\epsilon} \right) := N_o. \tag{2.15}
\]

Since \( \alpha^{(1)} = \frac{1}{4N} (\lambda^{(1)} \bar{\tau})^2 \), we have \( \sqrt{\alpha^{(k)}} = \frac{1}{\sqrt{4N}} \epsilon^k \). Hence, Lemma\textsuperscript{5} implies that

\[
N^{(k)} \leq 2B_x \sqrt{\frac{2(\lambda^{(k)} \bar{L} + \sigma_{\max}^2(A))}{\alpha^{(k)}}} \leq \frac{8B_x \sqrt{N}}{\bar{\tau}} \sigma_{\max}(A) \epsilon^{-k}. \tag{2.16}
\]

Hence, (2.14) and (2.16) imply that the total number of MS-APG iterations to compute an \( \epsilon \)-feasible solution can be bounded above:

\[
\sum_{k=1}^{N_{\text{FAL}}^f(\epsilon)} N^{(k)} \leq \frac{8B_x \sqrt{N}}{\bar{\tau}} \sigma_{\max}(A) \sum_{k=1}^{\tilde{N}_f} c^{-k} \leq \frac{8B_x \sqrt{N}}{c(1 - c) \bar{\tau}} \sigma_{\max}(A) \left( \frac{1}{c} \right)^{\tilde{N}_f},
\]

\[
\leq \frac{16B_x \sqrt{N}}{c(1 - c) \bar{\tau}} \min_{i \in \mathcal{N}} \left\{ \frac{\bar{\tau} \sigma_{\max}(A) / \sqrt{N} + B_i + G_i}{\sigma_{\min}(A_i) \epsilon} \right\} \sigma_{\max}(A) \left( \frac{1}{c} \right)^{\tilde{N}_f} = \mathcal{O} \left( \frac{\sigma_{\max}^3(A)}{\min_{i \in \mathcal{N}} \sigma_{\min}^2(A_i) \epsilon} \right). \tag{2.17}
\]

Similarly, (2.15) and (2.16) imply that the total number of MS-APG iterations to compute an \( \epsilon \)-optimal solution can be bounded above:

\[
\sum_{k=1}^{N_{\text{FAL}}^o(\epsilon)} N^{(k)} \leq \frac{8B_x \sqrt{N}}{c(1 - c) \bar{\tau}} \sigma_{\max}(A) \left( \frac{1}{c} \right)^{\tilde{N}_o} = \mathcal{O} \left( \frac{\sigma_{\max}^3(A)}{\min_{i \in \mathcal{N}} \sigma_{\min}^2(A_i) \epsilon} \right). \tag{2.17}
\]

2.2.3 Synchronous Algorithm for distributed optimization

In this section, we show that the decentralized optimization problem (1.3) is a special case of (2.2); therefore, Theorem\textsuperscript{3} establishes the main result in the introduction. We also show that the steps in FAL can be further simplified in this context.

Construct a directed graph by introducing an arc \((i, j)\) where \( i < j \) for every edge \((i, j)\) in the undirected graph \( G = (\mathcal{N}, \mathcal{E}) \). Then the constraints \( x_i - x_j = 0 \) for all \((i, j) \in \mathcal{E}\) in the distributed optimization problem (1.3) can be reformulated as \( C x = 0 \), where \( C \in \mathbb{R}^{n|\mathcal{E}| \times nN} \) is a block matrix.
such that the block $C_{(i,j),l} \in \mathbb{R}^{n \times n}$ corresponding to the edge $(i, j) \in \mathcal{E}$ and node $l \in \mathcal{N}$, i.e. $C_{(i,j),l}$ is equal to $I_n$ if $l = i$, $-I_n$ if $l = j$, and $0_n$ otherwise, where $I_n$ and $0_n$ denote $n \times n$ identity and zero matrices, respectively.

Let $\Omega \in \mathbb{R}^{N \times N}$ be the Laplacian of $G$, i.e. for all $i \in \mathcal{N}$, $\Omega_{ii} = d_i$, and for all $(i, j) \in \mathcal{N} \times \mathcal{N}$ such that $i \neq j$, $\Omega_{ij} = -1$ if either $(i, j) \in \mathcal{E}$ or $(j, i) \in \mathcal{E}$, where $d_i$ denotes the degree of $i \in \mathcal{N}$. Then it follows that

$$
\Psi := C^TC = 
\begin{bmatrix}
\Omega_{11}I_n & \Omega_{12}I_n & \cdots & \Omega_{1N}I_n \\
\Omega_{21}I_n & \Omega_{22}I_n & \cdots & \Omega_{2N}I_n \\
\vdots & \vdots & \ddots & \vdots \\
\Omega_{N1}I_n & \Omega_{N2}I_n & \cdots & \Omega_{NN}I_n
\end{bmatrix}.
$$

Let $\psi_1 \geq \psi_2 \geq \ldots \geq \psi_N$ be the eigenvalues of $\Omega$. Since $G$ is connected, $\text{rank}(\Omega) = N - 1$, i.e. $\psi_{N-1} > 0$ and $\psi_N = 0$. From the structure of $\Psi$ it follows that that $\{\psi_i\}_{i=1}^N$ are also the eigenvalues of $\Psi$, each with algebraic multiplicity $n$. Hence, $\text{rank}(C) = n(N-1)$. Let $C = U \Sigma V^T$ denote the reduced singular value decomposition (SVD) of $C$, where $U \in \mathbb{R}^{n|\mathcal{E}| \times n(N-1)}$, $\Sigma = \text{diag}(\sigma)$, and $\sigma \in \mathbb{R}^{n(N-1)}_+$, and $V \in \mathbb{R}^{nN \times n(N-1)}$. Note that $\sigma_{\max}^2(C) = \psi_1$, and $\sigma_{\min}^2(C) = \psi_{N-1}$.

Let $A := \Sigma V^T$. Clearly, $A \in \mathbb{R}^{n(N-1) \times nN}$ has linearly independent rows; more importantly, $A^T A = C^TC = \Psi$; hence, $\sigma_{\max}^2(A) = \psi_1$, and $\sigma_{\min}^2(A) = \psi_{N-1}$. Moreover, we have $\{x \in \mathbb{R}^{nN} : Ax = 0\} = \{x \in \mathbb{R}^{nN} : Cx = 0\}$. Hence, the general problem in (2.2) with $A := \Sigma V^T$ and $b = 0 \in \mathbb{R}^{n(N-1)}$ is equivalent to (1.3), i.e. if $x^*$ is a minimizer of (2.2), then it is also a minimizer of (1.3). Moreover, since $\nabla \tilde{\gamma}(x^*) \in \partial \rho(x^*)$, we have $\|\nabla x_i \tilde{\gamma}(x^*)\|_2 \leq B_i$ for all $i \in \mathcal{N}$.

Let $A := \Sigma V^T \in \mathbb{R}^{n(N-1) \times nN}$. For all $i \in \mathcal{N}$, let $A_i \in \mathbb{R}^{n(N-1) \times nN}$ and $C_i \in \mathbb{R}^{n|\mathcal{E}| \times nN}$ be the submatrices of $A$ and $C$, respectively, corresponding to $x_i$, i.e. $A = [A_1, A_2, \ldots, A_N]$, and $C = [C_1, C_2, \ldots, C_N]$. Clearly, it follows from the definition of $C$ that $\sigma_{\max}(C_i) = \sigma_{\min}(C_i) = \sqrt{d_i}$ for all $i \in \mathcal{N}$. Using the property of SVD, it can also be shown for $A = \Sigma V^T$ that $\sigma_{\max}(A_i) = \sigma_{\min}(A_i) = \sqrt{d_i}$ for all $i \in \mathcal{N}$.

We now show that we do not have to compute the SVD of $C$, or $A$, or even the dual multipliers $\theta^{(k)}$ when FAL is used to solve (1.3). In FAL the matrix $A$ is used in Step 1 (i.e. within the oracle MS-APG) to compute $\nabla f^{(k)}$, and in Step 2 to compute $\theta^{(k+1)}$. Since $\theta^{(1)} = 0$, Step 2 in FAL implies that $\theta^{(k+1)} = -\sum_{t=1}^{k} \frac{A x^{(t)}}{\lambda^{(t)}}$. Therefore, $\nabla f^{(k)}(x) = \lambda^{(k)} \nabla \tilde{\gamma}(x) + A^T(Ax - \lambda^{(k)} \theta^{(k)}) = \lambda^{(k)} \nabla \tilde{\gamma}(x) + \Psi \left( x + \lambda^{(k)} \sum_{t=1}^{k-1} \frac{1}{\lambda^{(t)}} x^{(t)} \right)$. Moreover, from the definition of $\Psi$, it follows that

$$
\nabla x_i f^{(k)}(x) = \lambda^{(k)} \nabla \psi_i(x_i) + d_i \left( x_i + \bar{x}_i^{(k)} \right) - \sum_{j \in \mathcal{O}(i)} \left( x_j + \bar{x}_j^{(k)} \right), \quad \bar{x}_i^{(k)} := \sum_{t=1}^{k-1} \frac{\lambda^{(k)}}{\lambda^{(t)}} x^{(t)},
$$

where $\mathcal{O}(i)$ denotes the set of nodes adjacent to $i \in \mathcal{N}$. Thus, it follows that Step 1 of MS-APG can
be computed in a distributed manner by only communicating with the adjacent nodes without explicitly computing $\theta^{(k)}$ in Step 2 of FAL. Indeed, for the $k$-th FAL iteration, each node $i \in \mathcal{N}$ stores $\bar{x}_i^{(k)}$ and $\{\bar{y}_j^{(k)}\}_{j \in \mathcal{O}(i)}$, which can be easily computed locally if $\{x_j^{(t)}\}_{j \in \mathcal{O}(i)}$ is transmitted to $i$ at the end of Step 1 in the previous FAL iterations $1 \leq t < k - 1$. Hence, during the $t$-th iteration of MS-APG $(\lambda^{(k)}_{\bar{\rho}}, f^{(k)}, x^{(k-1)})$ call, each node $i \in \mathcal{N}$ can compute $\nabla_{y_i} f^{(k)}(y_i^f)$ locally if $\{\bar{y}_j^{(t)}\}_{j \in \mathcal{O}(i)}$ is transmitted to $i$ at the end of Step 3 in MS-APG. It is important to note that every node can independently check (2.4)(b), i.e., $\exists g_i^{(k)} \in \partial \rho_i(x_i) | x_i = y_i^{(k)} + \nabla x_i f^{(k)}(x^{(k)})$ for all $i \in \mathcal{N}$ s.t. $\max_{i \in \mathcal{N}} \|g_i^{(k)}\|_2 \leq \frac{\xi_i^{(k)}}{\sqrt{N}}$. Hence, nodes can reach a consensus to move to the next FAL iteration without communicating their private information.

### 2.2.4 Asynchronous implementation

Nesterov [16] proposed randomized block coordinate descent (RBCD) for solving $\min_{y \in \mathbb{R}^n} f(y)$, where $f$ is a convex function with block Lipschitz continuous gradient, i.e., $\nabla_{y_i} f(y_i; y_{-i})$ is Lipschitz continuous in $y_i$ with constant $L_i$ for all $i$. Later, Richtárik et al. [17] extended the convergence rate results to $\min_{y \in \mathbb{R}^n} \Phi(y) := \sum_{i=1}^{N} \rho_i(y_i) + f(y)$, such that $\text{prox}_{\rho_i}$ can be computed efficiently for all $t > 0$ and $i \in \mathcal{N}$, and established that for all $\alpha > 0$ and $0 < p < 1$, the iterate sequence $\{y^{(\ell)}\}$ computed by RBCD displayed in Fig. 3 satisfies

$$\mathbb{P}(\Phi(y^{(\ell)}) - \Phi^* \leq \alpha) \geq 1 - p, \text{ for } \ell \geq \frac{2NC}{\alpha} \left(1 + \log \frac{1}{p}\right),$$

where $C = \max\{\mathcal{R}_L^2(y^{(0)}), \Phi(y^{(0)}) - \Phi^*\}, \mathcal{R}_L^2(y^{(0)}) := \max_{y, y^*} \{\sum_{i=1}^{N} L_i \|y_i - y_i^*\|_2^2 : \Phi(y) \leq \Phi(y^*), y^* \in \mathcal{Y}^*\}$, and $\mathcal{Y}^*$ denotes the set of optimal solutions. RBCD is significantly faster in practice for very large scale problems, particularly when the partial gradient $\nabla_{y_i} f(y)$ can be computed more efficiently as compared to the full gradient $\nabla f(y)$.

**Algorithm RBCD ( $\bar{\rho}, f, y^{(0)}$ )**

Step $\ell$: ($\ell \geq 0$)

1. $i \in \mathcal{N}$ is realized with probability $\frac{1}{N}$
2. $y_{i}^{(\ell+1)} = \text{prox}_{\rho_i / L_i} \left( y_i^{(\ell)} - \nabla_{y_i} f(y^{(\ell)}) / L_i \right)$ and $y_{\overline{i}}^{(\ell+1)} = y_{\overline{i}}^{(\ell)}$

Figure 3: Randomized block coordinate descent algorithm

The RBCD algorithm can be implemented for the distributed minimization problem when the nodes in $\mathcal{G}$ work asynchronously. Assume that for any $y = (y_i)_{i \in \mathcal{N}} \in \mathbb{R}^{nN}$, each node $i$ is equally likely to complete computing $\text{prox}_{\rho_i / L_i} (y_i - \nabla_{y_i} f(y) / L_i)$. Suppose node $k \in \mathcal{N}$ is the first node to complete Step 2 of RBCD. Then, instead of waiting for the other nodes to finish, node $k$ sends a message to its neighbors $j \in \mathcal{O}(k)$ to terminate their computations, and shares $y_i^{(\ell+1)}$ with them.
Note that RBCD can be easily incorporated into FAL as an oracle to solve subproblems in (2.3) by replacing (2.4)(a) with

$$\mathbb{P}\left(P^{(k)}(x^{(k)}) - P^{(k)}(x^*_k) \leq \alpha_k\right) \geq (1 - p)^{\frac{1}{N(e)}},$$

(2.19)

where $N(e) = \mathcal{O}\left(\log(e^{-1})\right)$ is defined in Corollary 1. Since $(1 - p)^{\frac{1}{N(e)}} \approx 1 - \frac{p}{N(e)}$ when $p$ is close to zero, the total number of RBCD iterations for the k-th subproblem is bounded by $N(k) = \mathcal{O}\left(\frac{1}{(\lambda(k))^2} \log \left(\frac{N(e)}{p}\right)\right) = \mathcal{O}\left(\frac{1}{(\lambda(k))^2} (\log \left(\frac{1}{p}\right) + \log \log \left(\frac{1}{e}\right))\right)$. Hence, Corollary 1 and (2.18) imply that asynchronous FAL, i.e. (2.4)(a) replaced with (2.19), can compute an $\epsilon$-optimal and $\epsilon$-feasible solution to (1.3) with probability $1 - p$ within $\mathcal{O}\left(\frac{1}{e^2} \log \left(\frac{1}{p}\right)\right)$ RBCD iterations.

3 Simulation

In this section, we test the performance of FAL on (1.3) with $\{\rho_i, \gamma_i\}_{i \in \mathcal{N}}$ defined as follows.

$$\gamma_i(x) := \sum_{j=1}^{m_i} h_\delta (a_i^T(j)x - b_i), \rho_i(x) := \beta_1 \|x\|_1 + \beta_2 \|x\|_{G_i},$$

where $h_\delta(x) = \max\{t | x| - t^2/2 : 0 \leq t \leq \delta\}$ is the Huber loss function, $A_i \in \mathbb{R}^{m_i \times n}$, $b_i \in \mathbb{R}^{m_i}$, $a_i^T(j)$ denotes the j-th row of $A_i$, $\beta_1, \beta_2 > 0$, and $\|x\|_{G_i} := \sum_{k=1}^{K} \|x_{g_i(k)}\|_2$ denotes the group norm with respect to the partition $G_i$ of $[1, n] := \{1, \ldots, n\}$ for all $i \in \mathcal{N}$, i.e. $G_i = \{g_i(k)\}_{k=1}^{K}$ such that $\bigcup_{k=1}^{K} g_i(k) = [1, n]$, and $g_i(j) \cap g_i(k) = \emptyset$ for all $j \neq k$. Thus, the overall problem is the group-sparse LASSO problem with a Huber penalty, i.e. $\min_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^{N} \beta_1 \|x\|_1 + \beta_2 \|x\|_{G_i} + \sum_{j=1}^{m_i} h_\delta (a_i^T(j)x - b_i) \right\}$.

The following lemma shows that each node $i \in \mathcal{N}$ can check (2.4)(b) very efficiently. Given $t \in \mathbb{R}$, define $\text{sgn} : \mathbb{R} \rightarrow \{-1, 0, 1\}$ such that $\text{sgn}(t)$ is equal to -1, 0 and 1 when $t < 0$, $t = 0$, and $t > 0$, respectively. Moreover, for $x \in \mathbb{R}^n$, $\text{sgn}(x) = [\text{sgn}(x_1), \text{sgn}(x_2), \ldots, \text{sgn}(x_n)]^T$.

**Lemma 6.** Let $G = \{g(k)\}_{k=1}^{K}$ be a partition of $[1, n]$ and $\|x\|_G := \sum_{k=1}^{K} \|x_{g(k)}\|_2$. Given $\beta_1, \beta_2 > 0$, define $\rho : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\rho(x) := \beta_1 \|x\|_1 + \beta_2 \|x\|_G$. Given $\lambda > 0$ and a differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, define $P : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $P = \lambda \rho + f$. For all $\bar{x} \in \mathbb{R}^n$ and $\xi > 0$, there exists $\nu \in \partial P(x)|_{x=\bar{x}}$ such that $\|\nu\|_2 \leq \xi$ if and only if $\|\pi^* + \omega* + \nabla f(\bar{x})\|_2 \leq \xi$ for $\pi^*, \omega*$:

$$\omega^*_{g(k)} = \begin{cases} -\left(\pi^*_{g(k)} + \nabla x_{g(k)} f(\bar{x})\right) \min \left\{1, \frac{\lambda \beta_2}{\|x_{g(k)}\|_2}\right\}, & \text{if } \bar{x}_{g(k)} = 0; \\ \frac{x_{g(k)}}{\|x_{g(k)}\|_2}, & \text{if } \bar{x}_{g(k)} \neq 0; \end{cases}$$

(3.1)

$$\pi^*_{g(k)} = \begin{cases} \eta_{g(k)}, & \text{if } \bar{x}_{g(k)} = 0; \\ \lambda \beta_1 \text{sgn}(\bar{x}_{g(k)}) + (1 - \text{sgn}(|\bar{x}_{g(k)}|)) \odot \eta_{g(k)}, & \text{if } \bar{x}_{g(k)} \neq 0; \end{cases}$$

(3.2)

$$\eta_{g(k)} = -\text{sgn}\left(\nabla x_{g(k)} f(\bar{x})\right) \odot \min \left\{ |\nabla x_{g(k)} f(\bar{x})|, \lambda \beta_1 \right\},$$

(3.3)

where $\odot$ denotes componentwise multiplication.
Proof. Given any convex function \( \rho : \mathbb{R}^n \to \mathbb{R} \) and \( \bar{x} \in \mathbb{R}^n \), in order to simplify the notation throughout the proof, \( \partial \rho(x)|_{x=\bar{x}} \subset \mathbb{R}^n \), the subdifferential of \( \rho \) at \( \bar{x} \), will be written as \( \partial \rho(\bar{x}) \).

Given \( \bar{x} \in \mathbb{R}^n \), there exists \( \nu \in \partial P(\bar{x}) \) such that \( \|\nu\|_2 \leq \xi \), if and only if \( \|\nu^*\| \leq \xi \), where \( \nu^* = \arg\min \{\|\nu\|_2 : \nu \in \partial P(\bar{x})\} \). Note that \( \partial P(\bar{x}) = \lambda \partial \rho(\bar{x}) + \nabla f(\bar{x}) \), and

\[
\partial \rho(\bar{x}) = \beta_1 \prod_{k=1}^K \partial \|\bar{x}_{g(k)}\|_1 + \beta_2 \prod_{k=1}^K \partial \|\bar{x}_{g(k)}\|_2,
\]

where \( \prod \) denotes the Cartesian product.

Since the groups \( \{g(k)\}_{k=1}^K \) are not overlapping with each other, the minimization problem is separable in groups. Hence, for all \( k \in [1, K] \), we have \( \nu^*_{g(k)} = \pi^*_{g(k)} + \omega^*_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x}) \) such that

\[
(\pi^*_{g(k)}, \omega^*_{g(k)}) = \arg\min \|\pi_{g(k)} + \omega_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2^2,
\]

s.t. \( \pi_{g(k)} \in \lambda \beta_1 \partial \|\bar{x}_{g(k)}\|_1 \), \( \omega_{g(k)} \in \lambda \beta_2 \partial \|\bar{x}_{g(k)}\|_2 \).

Fix \( k \in [1, K] \). We will consider the solution to above problem in two cases. Suppose that \( \bar{x}_{g(k)} = 0 \). Since \( \partial \|0\|_1 \) is the unit \( \ell_\infty \)-ball, and \( \partial \|0\|_2 \) is the unit \( \ell_2 \)-ball, \( (3.5) \) can be equivalently written as

\[
(\pi^*_{g(k)}, \omega^*_{g(k)}) = \arg\min \|\pi_{g(k)} + \omega_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2^2,
\]

s.t. \( \|\pi_{g(k)}\|_\infty \leq \lambda \beta_1 \), \( \|\omega_{g(k)}\|_2 \leq \lambda \beta_2 \).

Clearly, it follows from Euclidean projection on to \( \ell_2 \)-ball that

\[
\omega^*_{g(k)} = -\left(\pi^*_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\right) \min \left\{1, \frac{\lambda \beta_2}{\|\pi^*_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2}\right\}.
\]

Hence, \( \|\pi^*_{g(k)} + \omega^*_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2 = \max\{0, \|\pi^*_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2 - \lambda \beta_2\} \). Therefore,

\[
\pi^*_{g(k)} = \arg\min \{\|\pi_{g(k)} + \nabla_{x_{g(k)}} f(\bar{x})\|_2 : \|\pi_{g(k)}\|_\infty \leq \lambda \beta_1\},
\]

\[
= -\text{sgn}(\nabla_{x_{g(k)}} f(\bar{x})) \odot \min \{\|\nabla_{x_{g(k)}} f(\bar{x})\|, \lambda \beta_1\}.
\]

Now, suppose that \( \bar{x}_{g(k)} \neq 0 \). This implies that \( \partial \|\bar{x}_{g(k)}\|_2 = \{\bar{x}_{g(k)}/\|\bar{x}_{g(k)}\|_2\} \). Hence, when \( \bar{x}_{g(k)} \neq 0 \), we have \( \omega^*_{g(k)} = \lambda \beta_2 \bar{x}_{g(k)}/\|\bar{x}_{g(k)}\|_2 \), and the structure of \( \partial \| \cdot \|_1 \) implies that \( \pi^*_j = \lambda \beta_1 \text{sgn}(\bar{x}_j) \) for all \( j \in g(k) \) such that \( |\bar{x}_j| > 0 \); and it follows from \( (3.5) \) that for all \( j \in g(k) \) such that \( \bar{x}_j = 0 \), we have

\[
\pi^*_j = \arg\min \left\{(\pi_j + \frac{\partial f(\bar{x})}{\partial \bar{x}_j})^2 : \|\pi_j\| \leq \lambda \beta_1\} = -\text{sgn}\left(\frac{\partial f(\bar{x})}{\partial \bar{x}_j}\right) \min \left\{\left|\frac{\partial f(\bar{x})}{\partial \bar{x}_j}\right|, \lambda \beta_1\right\}.
\]

\( \square \)
In our experiments, the network was either a star tree or a clique with either 5 or 10 nodes. The remaining problem parameters defining \( \{ \rho_i, \gamma_i \}_{i \in \mathcal{N}} \) were set as follows. We set \( \beta_1 = \beta_2 = \frac{1}{K}, \) \( \delta = 1, \) and \( K = 10. \) Let \( n = Kn_g \) for \( n_g \in \{100, 300\} \), i.e. \( n \in \{1000, 3000\}. \) We generated partitions \( \{ G_i \}_{i \in \mathcal{N}} \) in two different ways. For test problems in \textbf{CASE 1}, we created a single partition \( G = \{ g(k) \}_{k=1}^K \) by generating \( K \) groups uniformly at random such that \( |g(k)| = n_g \) for all \( k; \) and set \( G_i = G \) for all \( i \in \mathcal{N}, \) i.e. \( \rho_i(x) = \rho(x) := \beta_1 \|x\|_1 + \beta_2 \|x\|_G \) for all \( i \in \mathcal{N}. \) For the test problems in \textbf{CASE 2}, we created a different partition \( G_i \) for each node \( i, \) in the same manner as in \textbf{CASE 1}. For all \( i \in \mathcal{N}, \) \( m_i = \frac{n}{4N}, \) and the elements of \( A_i \in \mathbb{R}^{m_i \times n} \) are i.i.d. with standard Gaussian, and we set \( b_i = A_i \bar{x} \) for \( \bar{x}_j = (-1)^j e^{-(j-1)/n_g} \) for \( j \in [1, n]. \)

We solved the distributed optimization problem \((\ref{eq:1.3})\) using \textbf{FAL} and SDPT3 for both cases, on both star trees, and cliques, and for \( N \in \{5, 10\} \) and \( n_g \in \{100, 300\}. \) For benchmarking, we solved the centralized problem \((\ref{eq:1.2})\) using SDPT3 for both cases. Note that for the first case \( \sum_{i \in \mathcal{N}} \rho_i(x) = \|x\|_1 + \|x\|_G \) and its prox mapping can be computed efficiently, while for the second case \( \sum_{i \in \mathcal{N}} \rho_i(x) \) does not assume a simple prox map. Therefore, for the first case we were also able to use APG, described in Section 2.1, to solve \((\ref{eq:1.2})\) with the help of following lemma.

**Lemma 7.** Let \( G = \{ g(k) \}_{k=1}^K \) denote a partition of \([1, n], \) and \( \rho(x) = \beta_1 \|x\|_1 + \beta_2 \|x\|_G, \) where \( \|x\|_G := \sum_{k=1}^K \|x_{g(k)}\|_2. \) Then for all \( t > 0 \) and \( \bar{x} \in \mathbb{R}^n, \) \( x^p = \text{prox}_{t \rho}(\bar{x}) \) can be written in closed form \( x^p_{g(k)} = \eta'_{g(k)}(k) \max \left\{ 1 - \frac{t \beta_2}{\|x_{g(k)}\|_2^2} \cdot 0 \right\}, \) for \( 1 \leq k \leq K, \) where \( \eta' = \text{sgn}(\bar{x}) \max \{ |\bar{x}| - t \beta_1, 0 \}. \)

**Proof.** Since the groups are not overlapping with each other, the proximal problem becomes separable in groups. Let \( n_k := |g(k)| \) for all \( k. \) Thus, it suffices to show that \( \min_{x_{g(k)} \in \mathbb{R}^{n_k}} \{ \beta_1 \|x\|_1 + \beta_2 \|x_{g(k)}\|_2 + \frac{1}{2t} \|x_{g(k)} - \bar{x}_{g(k)}\|_2^2 \} \) has a closed form solution as shown in the statement for some fixed \( k. \) By the definition of dual norm, we have

\[
\min_{x_{g(k)} \in \mathbb{R}^{n_k}} \beta_1 \|x_{g(k)}\|_1 + \beta_2 \|x_{g(k)}\|_2 + \frac{1}{2t} \|x_{g(k)} - \bar{x}_{g(k)}\|_2^2, \tag{3.9}
\]

\[
= \min_{x_{g(k)} \in \mathbb{R}^{n_k}} \max_{\|u_1\|_\infty \leq \beta_1} u_1^T x_{g(k)} + \max_{\|u_2\|_2 \leq \beta_2} u_2^T x_{g(k)} + \frac{1}{2t} \|x_{g(k)} - \bar{x}_{g(k)}\|_2^2, \tag{3.10}
\]

\[
= \max_{\|u_1\|_\infty \leq \beta_1} \min_{\|u_2\|_2 \leq \beta_2} (u_1 + u_2)^T x_{g(k)} + \frac{1}{2t} \|x_{g(k)} - \bar{x}_{g(k)}\|_2^2, \tag{3.11}
\]

Let \( (u_1^*, u_2^*) \) be the optimal solution of \((3.11).\) Since \( x^p_{g(k)} \) is the optimal solution to \((3.9),\) it follows from \((3.10)\) that

\[
x^p_{g(k)} = \bar{x}_{g(k)} - t(u_1^* + u_2^*). \tag{3.12}
\]

Note that \((3.11)\) can be equivalently written as \( \min \{ \|u_1 + u_2 - \frac{1}{t} \bar{x}_{g(k)}\|_2^2 : \|u_1\|_\infty \leq \beta_1, \|u_2\|_2 \leq \)
\( \beta_2 \}. \) Minimizing over \( u_2 \), we have

\[
\begin{align*}
  u^*_2(u_1) &= \left( \frac{1}{t} \bar{x}_g(k) - u_1 \right) \min \left\{ \frac{\beta_2}{\| \frac{1}{t} \bar{x}_g(k) - u_1 \|_2}, 1 \right\}.
\end{align*}
\]

Hence, we have

\[
\begin{align*}
  u^*_1 &= \arg\min_{\|u_1\|_\infty \leq \beta_1} \left\| \left( u_1 - \frac{1}{t} \bar{x}_g(k) \right) \max \left\{ 1 - \frac{\beta_2}{\| u_1 - \frac{1}{t} \bar{x}_g(k) \|_2}, 0 \right\} \right\|_2, \\
  &= \arg\min_{\|u_1\|_\infty \leq \beta_1} \max \left\{ \| u_1 - \frac{1}{t} \bar{x}_g(k) \|_2 - \beta_2, 0 \right\}.
\end{align*}
\]

Clearly, \( u^*_1 = \arg\min_{\|u_1\|_\infty \leq \beta_1} \| (u_1 - \frac{1}{t} \bar{x}_g(k)) \|_2 = \text{sgn}(\bar{x}_g(k)) \min \{ \frac{1}{t} |\bar{x}_g(k)|, \beta_1 \} \). The final result follows from combining (3.12) and (3.13). \( \square \)

In Table 1, 'C.xxx' stands for “algorithm xxx is used to solve the centralized problem”. Similarly, 'D.xxx' for the decentralized one. For the results separated by comma, the left and right ones are for the star tree and clique, resp. Table 1 displays the results for one random realization for each case. On the other hand, Table 2 displays the results for 5 random replications for each case, which shows that FAL is indeed stable.

The number of iterations in each case clearly illustrates the topology of the network plays an important role in the convergence speed of FAL, which coincides to our analysis in Section 2.2.2. In contrast to FAL, the results are not in favor of AFAL, which performs as expected with \( O(1/\epsilon^2) \) convergence rate. However, there is still hope to achieve \( O(1/\epsilon) \) rate by designing accelerated RBCD methods for convex composite minimization, which we will investigate in our future work.
Table 1: Comparison of FAL, AFAL (Asynchronous FAL), APG, and SDPT3

|                | \(F^*\) | \(\max_{(i,j)\in E} \|x_i - x_j\|_2^2\) | CPU Time (sec.) | Iterations |
|----------------|---------|--------------------------------------|-----------------|------------|
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |

(a) \(n_g = 100, N = 5\)

|                | \(F^*\) | \(\max_{(i,j)\in E} \|x_i - x_j\|_2^2\) | CPU Time (sec.) | Iterations |
|----------------|---------|--------------------------------------|-----------------|------------|
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |

(b) \(n_g = 100, N = 10\)

|                | \(F^*\) | \(\max_{(i,j)\in E} \|x_i - x_j\|_2^2\) | CPU Time (sec.) | Iterations |
|----------------|---------|--------------------------------------|-----------------|------------|
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |

(c) \(n_g = 300, N = 5\)

|                | \(F^*\) | \(\max_{(i,j)\in E} \|x_i - x_j\|_2^2\) | CPU Time (sec.) | Iterations |
|----------------|---------|--------------------------------------|-----------------|------------|
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |
|                | Case 1  | Case 2                              | Case 1          | Case 2     |

(d) \(n_g = 300, N = 10\)
| Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| max(\|x_i - x_j\|) | CPU Time (sec.) | Iterations |
| C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL |
| 101.57 | 102.26 | 0 | 0 | 29 | 84 | 24 | 22 |
| 289.235 | 289.235 | 0 | 0 | 28 | 96 | 25 | 22 |
| 101.14 | 100.34 | 0 | 0 | 28.64 | 93 | 23 | 23 |
| 283.46 | 282.09 | 0 | 0 | 21.11 | 18.10 | 3536 | 1746 |
| 100.12 | 99.75 | 0 | 0 | 26.94 | 90 | 24 | 22 |
| 289.74 | 99.37 | 0 | 0 | 21.11 | 18.10 | 3536 | 1746 |
| 101.30 | 101.10 | 0 | 0 | 27.99 | 83 | 24 | 23 |
| 287.23 | 286.73 | 0 | 0 | 27.99 | 83 | 24 | 23 |
| 101.14 | 100.97 | 0 | 0 | 28.64 | 89 | 25 | 22 |
| 287.23 | 286.73 | 0 | 0 | 27.99 | 83 | 24 | 23 |
| 101.14 | 101.37 | 0 | 0 | 29.67 | 89 | 25 | 22 |
| 99.73 | 99.73 | 3E-7, 1E-8 | 1E-6, 8E-6 | 24 | 22 |
| 101.14 | 100.44 | 0 | 0 | 28.64 | 89 | 25 | 22 |
| 100.99 | 100.17 | 0 | 0 | 21.18 | 18.10 | 3536 | 1746 |
| 100.12 | 100.10 | 0 | 0 | 26 | 92 | 24 | 22 |
| 99.73 | 99.73 | 3E-7, 1E-8 | 1E-6, 8E-6 | 24 | 22 |
| 101.30 | 100.68 | 0 | 0 | 27.99 | 83 | 24 | 22 |
| 101.09 | 100.95 | 0 | 0 | 27.99 | 83 | 24 | 22 |
| 100.57 | 99.86 | 0 | 0 | 95.16 | 18.19 | 3292 | 4450 |
| 100.33 | 100.30 | 99.69 | 99.53 | 3E-7, 6E-8 | 2E-6, 2E-6 | 25 | 17 |
| (a) n_g = 100, N = 5 |

| Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| max(\|x_i - x_j\|) | CPU Time (sec.) | Iterations |
| C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL |
| 288.86 | 287.28 | 0 | 0 | 801 | 1634 | 26 | 29 |
| 283.46 | 282.09 | 0 | 0 | 820 | 1650 | 26 | 30 |
| 287.23 | 286.02 | 0 | 0 | 794 | 1683 | 27 | 30 |
| 286.96 | 287.19 | 85.68, 25.72 | 3E-7, 2E-7 | 8E-7, 6E-7 | 26 | 29 |
| 291.37 | 291.53 | 0 | 0 | 803 | 1678 | 26 | 29 |
| 289.23 | 289.07 | 0 | 0 | 810 | 1620 | 26 | 29 |
| 288.88 | 288.89 | 289.74 | 289.58 | 3E-7, 2E-7 | 8E-7, 6E-7 | 95 | 136 | 60 | 53 | 3292 | 4450 | 2073 | 1977 |
| (c) n_g = 300, N = 5 |

| Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 | Case 1 | Case 2 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| max(\|x_i - x_j\|) | CPU Time (sec.) | Iterations |
| C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL | C_SDPT3 | D_FAL |
| 288.86 | 287.28 | 0 | 0 | 801 | 1625 | 26 | 29 |
| 283.46 | 282.70 | 0 | 0 | 820 | 1615 | 26 | 30 |
| 287.23 | 286.73 | 0 | 0 | 794 | 1663 | 27 | 30 |
| 289.62 | 287.20 | 288.00, 286.58 | 1E-7, 3E-8 | 2E-6, 6E-6 | 27 | 30 |
| 291.37 | 291.21 | 0 | 0 | 803 | 1643 | 26 | 29 |
| 289.23 | 289.09 | 0 | 0 | 810 | 1660 | 26 | 29 |
| (d) n_g = 300, N = 10 |
References

[1] Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B Giannakis. Distributed sparse linear regression. *Signal Processing, IEEE Transactions on*, 58(10):5262–5276, 2010.

[2] Ion Necoara and Johan AK Suykens. Application of a smoothing technique to decomposition in convex optimization. *Automatic Control, IEEE Transactions on*, 53(11):2674–2679, 2008.

[3] Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464. Association for Computational Linguistics, 2010.

[4] Victor Lesser, Charles L Ortiz Jr, and Milind Tambe. *Distributed sensor networks: A multiagent perspective*, volume 9. Springer, 2003.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Automat. Contr.*, 57(3):592–606, 2012.

[6] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

[7] Ermin Wei and Asuman Ozdaglar. On the o (1/k) convergence of asynchronous distributed alternating direction method of multipliers. *arXiv preprint arXiv:1307.8254*, 2013.

[8] Dusan Jakovetic, Joao Xavier, and J Moura. Fast distributed gradient methods. 2011.

[9] Annie I Chen and Asuman Ozdaglar. A fast distributed proximal-gradient method. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 601–608. IEEE, 2012.

[10] N. S. Aybat and G. Iyengar. A first-order penalty method for compressed sensing. *SIAM J. Optimization*, 21:287–313, 2011.

[11] D. Blatt, A. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

[12] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009.
[13] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[14] R. Grone and R. Merris. The laplacian spectrum of a graph. ii. *SIAM J. Discrete Math.*, 7(2):221–229, 1994.

[15] N. S. Aybat and G. Iyengar. A first-order augmented lagrangian method for compressed sensing. *SIAM Journal on Optimization*, 22(2):429–459, 2012.

[16] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[17] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *forthcoming, Mathematical Programming, Series A*, 2012. DOI: 10.1007/s10107-012-0614-z.