Fast Decentralized Gradient Descent Method and Applications to In-situ Seismic Tomography

Liang Zhao*‡, Wen-Zhan Song*‡, and Xiaojing Ye†
* Department of Computer Science, Georgia State University, GA USA
† Department of Mathematics & Statistics, Georgia State University, GA USA
‡ School of Electronic and Information Engineering, Shanghai University of Electrical Power, China
Email: lzhao5@student.gsu.edu, wsong@gsu.edu, xye@gsu.edu

***Abstract*—We consider the decentralized consensus optimization problem arising from in-situ seismic tomography in large-scale sensor networks. Unlike traditional seismic imaging performed in a centralized location, each node in this setting privately holds an objective function and partial data. The goal of each node is to obtain the optimal solution of the whole seismic image, while communicating only with its immediate neighbors. We present a fast decentralized gradient descent method and prove that this new method can reach optimal convergence rate of $O(1/k^2)$ where k is the number of communication/iteration rounds. Extensive numerical experiments on synthetic and real-world sensor network seismic data demonstrate that the proposed algorithms significantly outperform existing methods.**

***Keywords*—Big Data, Decentralized Computing, In-network Processing, Seismic Tomography, Sensor Network**

I. Introduction

Seismic tomography is a technique for imaging Earths sub-surface characteristics in an effort to understand deep geologic structure. It involves massive data collection, often manually retrieval, from hundreds to thousands geophones to a central place for post-processing. Real-time subsurface imaging is in great demand today as it is essential to assess the sustainability and potential hazards of geological structures, and reduce the costs and risks of exploration and production activities. Sensor network has been an effective approach for real-time remote environment monitoring. However, collecting massive raw seismic data through a sensor network in real-time is infeasible, due to severe bandwidth and sensor energy constraints. This paper is thus proposing a novel decentralized in-situ computing method for imaging earth subsurface in real-time. It is based on the principle of travel-time seismic tomography [1].

The principle of travel-time seismic tomography is illustrated in the left panel of Figure 1. It uses geophones placed on earth surface to acquire travel time of the compressional waves, known as P-waves, which are then used to derive the internal 3D velocity structure of the earth subsurface. More specifically, the travel-time seismic tomography on sensor networks involves three steps [1]: (i) once an earthquake happens, the sensor nodes (green triangles on the ground) will detect seismic disturbances and determine source location of earthquakes given a prior estimation of geologic structure, (ii) at each sensor node, ray tracing is performed to find the ray paths from the seismic source location to the sensor node, and (iii) the ray paths are used to perform a tomography reconstruction of the velocity structure x of earth subsurface. Here $x \in \mathbb{R}^n$ is the vectorization of the mesh grids with resolution n (n partition blocks of the volcano image on the left of Figure 1), where each component of x represents the slowness (reciprocal of velocity) of the material (e.g. magma, rocks, etc.) at the corresponding location, and hence x can be reconstructed using the travel times and lengths of ray paths obtained in the first two steps. In this work, we focus on the fundamental computational problem in the third step (illustrated by top right tomographic inversion in Figure 1) assuming the first two steps are already done. Our proposed framework is designed to leverage the computational power of all the sensor nodes. It performs in-network processing such that the "big data" stored in the nodes are processed locally. In addition, each node transmits its local estimate of the whole solution instead of the raw sensor data to its direct neighbors. Different from centralized approach, this kind of distributed and decentralized mechanism is much more scalable and fault-tolerant, and is a paradigm-shifting approach to solve a class of Big Data problems arising from distributed systems.

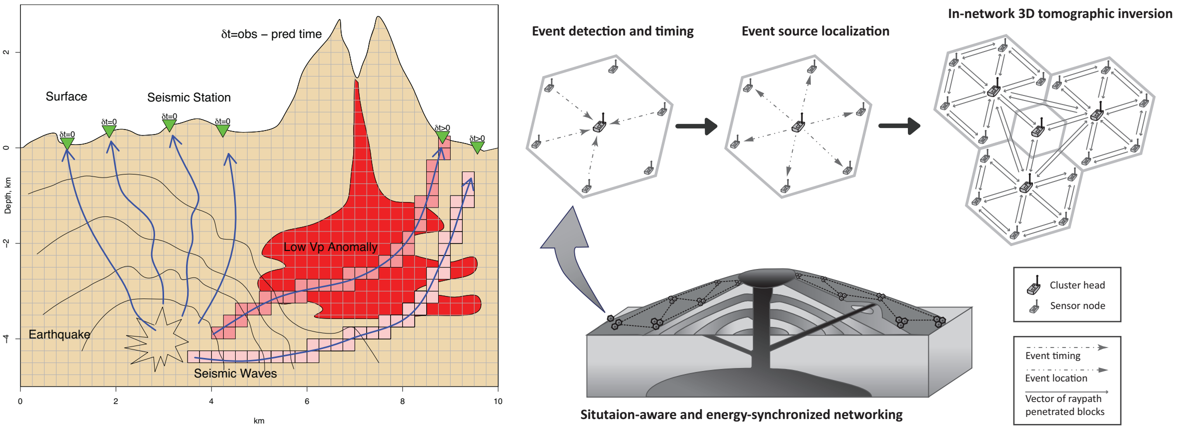
Recent advances in convex optimization provide models and algorithms for decentralized Big Data computing problems, while minimizing the related computation and communication [2]. The problem in this paper has a general form as follows. Consider an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The size of network is $m = |\mathcal{V}|$ and two nodes i, j are called neighbors if $(i, j) \in \mathcal{E}$. Now each node (sensor or agent) i privately holds an objective $F_i : \mathbb{R}^n \to \mathbb{R}$ which describes the data and acquisition process at the node. Here we assume F_i is proper, convex, and continuously differentiable, and its gradient ∇F_i has Lipschitz constant $L_i > 0$. The goal is to find the consensus solution $x \in \mathbb{R}^n$ of the minimization problem

$$\min_{x \in \mathbb{R}^n} \left\{ F(x) := \sum_{i=1}^{m} F_i(x) \right\}, \tag{1}$$

while each node can only communicate with its direct neighbors. The problem is called decentralized since the data is acquired and processed in a distributed network, and the nodes

Our research is supported by NSF-CNS-1066391, NSF-CNS-0914371, NSF-CPS-1135814 and NSF-CDI-1125165.

978-1-4799-9926-2/15/$31.00 ©2015 IEEE



(a)

(b)

Fig. 1: Illustration of seismic tomography and the new decentralized sensor network [1]. Left: Principle of travel-time seismic tomography. Right: Real-time decentralized volcano tomography.

are required to collaboratively solve for a consensus solution.

Our key contribution is twofold. First, we develop a fast gradient-based method for solving the decentralized consensus optimization problem in in-situ seismic tomography. To improve the performance in practice, we also propose a simple backtracking technique for searching local Lipschitz constant. Second, we prove that the proposed algorithms achieve the optimal convergence rate, which is the lower bound of centralized algorithms in solving smooth optimization problems.

The remainder of this paper is organized as follows. In the next section, we survey the related work on the distributed or decentralized consensus optimization problem. In Section III, we derive the proposed fast decentralized gradient descent method with its main convergence results. In Section IV, we conduct extensive numerical tests on synthetic and real world seismic data sets to demonstrate the practical performance of the proposed algorithm. Section V concludes the paper.

II. Related Work

In seismic tomography, the third step: tomography reconstruction (imaging) is the most computationally intensive and time-consuming aspect. Today the centralized approaches for seismic imaging have to be done with clusters of high-performance computers. However, they cannot be directly distributed in wireless sensor network. The tomography imaging problem can be modeled as solving a large-scale linear system of equations. In this paper, we focus on the key research challenge on solving the (regularized) least-squares problem distributedly under the severe constraints of sensor network.

In the literature of sensor networks there are a few studies on consensus-based Distributed Least Mean Square (DLMS) algorithms [3], [4]. These algorithms let each node maintain its own local estimation and, to reach the consensus, exchange information only within its local neighbors. This can also be used for getting least-squares solutions statistically. The problem is that it needs “bridge” sensors as fusion centers for collecting the information within the neighbors and distributing processed information back to the neighbors. This results in huge communication burden in the “bridge” sensors. Also the tomography result will be dramatically affected when these critical “bridge” sensors are malfunctioned. Sayed and Lopes developed a Distributed Recursive Least-Squares (D-RLS) strategy by appealing to collaboration techniques to achieve an exact recursive solution [5]. However, it requires a cyclic path in the network to perform the computation node by node while exchanging a large dense matrix between nodes.

The aforementioned decentralized consensus problem attracts much attention recently, especially in distributed machine learning, multi-agent optimization, etc. For solving the problem in (1), several (sub)gradient-based methods have been proposed [6]–[11]. However, bounded (sub)gradients are usually assumed in analyzing the convergence results in most of the above algorithms. In addition, they cannot converge to an optimal solution of (1) if fixed step size is used [12]. In order to guarantee to converge to an optimal solution, Chen [10] and Jakovetic [11] thus adopt diminishing step sizes in their algorithms. More related algorithmic developments can be found in [13]–[19]. Jakovetic [11] proposed a D-NC algorithm showing an outer-loop convergence rate of $O(1/k^2)$ in terms of objective error, leveraging Nesterov’s acceleration, which is the best theoretical rate known so far. However, significant consensus iterations are required per outer-loop iteration. Without bounded gradient, [20] derives a correction method based on mixing matrix for regular decentralized gradient decent method and obtains $O(1/k)$ convergence rate without diminishing step sizes.

Besides synchronous algorithms discussed in previous, there are several works on asynchronous distributed optimization methods [21], [22]. However, their convergence rates are usually weaker than the ones in the synchronous methods.

[1] Source: http://sensorweb.cs.gsu.edu/?q=VolcanoSRI

III. Algorithm Design

Notation. Let $x \in \mathbb{R}^n$ be a column vector in problem (1), and $x^{(i)} \in \mathbb{R}^n$ be the local copy held privately by node i for every $i \in \mathcal{V}$. Without further remark, vectors are all column vectors. A vector $v = (v_1, \cdots, v_n)^T \in \mathbb{R}^n$ is sometimes written in short as $[v_i]$. Let $\mathbf{W} \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite matrix and $\|v\|_{\mathbf{W}}$ is the (semi-)$\mathbf{W}$-norm of v. If $\mathbf{v} = (v^{(1)}, \cdots, v^{(m)})^T \in \mathbb{R}^{m\times n}$ where each $v^{(i)} \in \mathbb{R}^n$, then $\|\mathbf{v}\|_{\mathbf{W}} = \sqrt{\sum_{i=1}^m \|v^{(i)}\|^2_{\mathbf{W}}}$. Subscript k is outer iteration number, which is also the number of communication.

A. Proposed Algorithms and Interpretation

The decentralized gradient descent (DGD) is a standard approach for solving (1). Recall that if regular gradient descent is applied at each node i to minimize its own objective F_i independent of other nodes, then a solution $x^{(i)} \in \operatorname{argmin}_x F_i(x)$ can be severely biased due to the insufficient information in data at i. Moreover the solutions $x^{(i)}$ will not be all equal, and their average is not the solution to (1) in general. Instead, it is more sophisticated for each node i to request private copies $x^{(j)}$ from its immediate neighbors to gather more information and proceed with its next update of $x^{(i)}$. Motivated by this idea, the DGD iterates

$$x_{k+1}^{(i)} = \sum_{j=1}^{m} w_{ij} x_k^{(j)} - \alpha_k \nabla F_i(x_k^{(i)}) \tag{2}$$

at every node i for $k = 0, 1, 2, \cdots$. Here $x_k^{(i)}$ is the local copy held by node i at iteration k, α_k is the step size that satisfies $\alpha_k \le 1/L$, where

$$L := \max_{1\le i\le m} \{L_i\} \tag{3}$$

and L_i is the Lipschitz constant of ∇F_i. The prescribed symmetric mixing matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{m\times m}$ is nonnegative, $w_{ij} = 0$ if $(i,j) \notin \mathcal{E}$ and $i \neq j$, and $\mathbf{W}v = v$ if and only if $v \in \mathbb{R}^m$ is consensual, i.e., all its components are equal, due to the intuition that the mixing should not make changes if all $x^{(i)}$ are already identical. Therefore, each node i collects $x_k^{(j)}$ sent from its immediate neighbors j, mixes them with its own $x_k^{(i)}$ using weights w_{ij}, and performs a gradient descent at $x_k^{(i)}$ in iteration k.

To simplify notation, we define $\mathbf{x} := (x^{(1)}, \cdots, x^{(m)})^T \in \mathbb{R}^{m\times n}$, $\mathbf{F}(\mathbf{x}) = \sum_{i=1}^m F_i(x^{(i)}) \in \mathbb{R}$, column vector $\nabla F_i(x) \in \mathbb{R}^n$, and $\nabla \mathbf{F}(\mathbf{x}) = (\nabla F_1(x^{(1)}), \cdots, \nabla F_m(x^{(m)}))^T \in \mathbb{R}^{m\times n}$, then the decentralized minimization (1) is equivalent to a consensus optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^{m\times n}} \{\mathbf{F}(\mathbf{x}) : \mathbf{W}\mathbf{x} = \mathbf{x}\}, \tag{4}$$

where the constraint $\mathbf{W}\mathbf{x} = \mathbf{x}$ requires that a solution $\mathbf{x}^* = (x^{(1)}, \cdots, x^{(m)})^T$ needs to be consensual, i.e., $x^{(1)} = x^{(2)} = \cdots = x^{(m)} = x^*$, for some solution $x^* \in \mathbb{R}^n$ to (1), namely x^* satisfies $\sum_{i=1}^m \nabla F_i(x^*) = 0$. Furthermore, the DGD algorithm (2) can be written as

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \alpha_k \nabla \mathbf{F}(\mathbf{x}_k). \tag{5}$$

One can immediately observe that $\alpha_k \to 0$ is a necessary condition for the convergence of $\mathbf{x}_k$ to a solution $\mathbf{x}^*$ using (5), otherwise there will be $\nabla \mathbf{F}(\mathbf{x}^*) = (\nabla F_1(x^*), \cdots, \nabla F_m(x^*))^T = 0$ upon convergence $\mathbf{x}_k \to \mathbf{x}^*$, implying $\nabla F_i(x^*) = 0$ for all i, which is not true in general.

Algorithm 1 Fast Decentralized Gradient Descent (FDGD) with known Lipschitz constant L

```
Initialize $\mathbf{y}_0 = 0$ and arbitrary $\mathbf{x}_0$, set $\mathbf{x}_0^{\mathrm{ag}} = \mathbf{x}_0$.
for $k = 0, 1, \cdots$, do
    $\mathbf{y}_{k+1} = \mathbf{y}_k + (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_k$    (6)
    $\mathbf{x}_k^{\mathrm{md}} = (1-\theta_k)\mathbf{x}_k^{\mathrm{ag}} + \theta_k \widetilde{\mathbf{W}}\mathbf{x}_k$    (7)
    $\mathbf{x}_{k+1} = \widetilde{\mathbf{W}}\mathbf{x}_k - \mathbf{y}_{k+1} - \frac{1}{L\theta_k}\nabla \mathbf{F}(\mathbf{x}_k^{\mathrm{md}})$    (8)
    $\mathbf{x}_{k+1}^{\mathrm{ag}} = (1-\theta_k)\mathbf{x}_k^{\mathrm{ag}} + \theta_k \mathbf{x}_{k+1}$    (9)
end for
Output $\mathbf{x}_{k+1}^{\mathrm{ag}}$
```

In this paper, we develop a fast decentralized gradient descent method which does not require diminishing step size and the method is accelerated to reach an optimal $O(1/k^2)$ convergence rate for general convex differentiable functions F_i. We adopted the idea of Nesterov's optimal gradient method for centralized smooth optimization [23]–[26] and mixing matrix method in network gossip and consensus averaging algorithms [20], [27], [28], and developed the following fast decentralized gradient algorithm (Algorithm 1) to solve the seismic tomography problem (4). In Algorithm 1, k is the (outer) iteration number which also indicates the number of rounds of communications. Every node privately holds its local copies $y^{(i)}, x^{(i),\mathrm{md}}, x^{(i)}, x^{(i),\mathrm{ag}}$ and F_i. At iteration k, each node i sends its current $x_k^{(i)}$ to all its immediate neighbors $\{j : (i,j) \in \mathcal{E}\}$ and receives $x_k^{(j)}$ from them (one round of communication), then performs weighted sums using w_{ij} and $\tilde{w}_{ij}$ (according to multiplications $\mathbf{W}\mathbf{x}_k$ and $\widetilde{\mathbf{W}}\mathbf{x}_k$), and updates its $y^{(i)}, x^{(i),\mathrm{md}}, x^{(i)}$ and $x^{(i),\mathrm{ag}}$. The result $x^{(i),\mathrm{ag}}$ is output as the final reconstruction.

In Algorithm 1, the superscript "ag" stands for "aggregated", and "md" stands for "middle". Matrix $\widetilde{\mathbf{W}} = (\mathbf{I}+\mathbf{W})/2$ is a half-mixing matrix based on $\mathbf{W}$. A few remarks about this algorithm are in place. Firstly, Algorithm 1 is a first-order method since only $\nabla \mathbf{F}$ is required in each iteration, and hence the subproblem has low computation complexity. Secondly, we do not need to use diminishing step sizes which converge to 0 but still can ensure both of convergence and consensus. Thirdly, if $\theta_k = 1$ for all k, then Algorithm 1 reduces to a version very similar to regular decentralized gradient descent (5). However, by the choice of $\theta_k = O(1/k)$ as below, the change from input $\mathbf{x}_k^{\mathrm{md}}$ to output $\mathbf{x}_{k+1}^{\mathrm{ag}}$ is faster than that from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$. This implies that Algorithm 1 will converge faster than regular DGD. The last remark explains intuitively why the multi-step scheme defined in (7), (8), and (9) could potentially

Algorithm 2 Fast Decentralized Gradient Descent with Backtracking (FDGD-BT)

```
Initialize $\mathbf{y}_0 = 0$ and arbitrary $\mathbf{x}_0$, set $\mathbf{x}_0^{\mathrm{ag}} = \mathbf{x}_0$.
for $k = 0, 1, \cdots,$ do
    1)
        $\mathbf{y}_{k+1} = \mathbf{y}_k + (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{x}_k \quad (10)$
    2)
        $\mathbf{x}_k^{\mathrm{md}} = (1-\theta_k)\mathbf{x}_k^{\mathrm{ag}} + \theta_k \widetilde{\mathbf{W}}\mathbf{x}_k \quad (11)$
    3) For each node $i$, find the smallest integer $j_k^{(i)} = 0, 1, 2, \cdots$ such that
        $F_i(x_{k+1}^{(i),\mathrm{ag}}) \leq F_i(x_k^{(i),\mathrm{md}}) + \langle \nabla F_i(x_k^{(i),\mathrm{md}}), x_{k+1}^{(i),\mathrm{ag}} - x_k^{(i),\mathrm{md}} \rangle + \frac{L_k^{(i)} \rho^{j_k^{(i)}}}{2} \| x_{k+1}^{(i),\mathrm{ag}} - x_k^{(i),\mathrm{md}} \|^2 \quad (12)$
       where
        $\mathbf{x}_{k+1} = \widetilde{\mathbf{W}}\mathbf{x}_k - \mathbf{y}_{k+1} - \frac{1}{\theta_k}\mathbf{L}_k^{-1}\nabla\mathbf{F}(\mathbf{x}_k^{\mathrm{md}}) \quad (13)$
        $\mathbf{x}_{k+1}^{\mathrm{ag}} = (1-\theta_k)\mathbf{x}_k^{\mathrm{ag}} + \theta_k \mathbf{x}_{k+1} \quad (14)$
       Here $(x_{k+1}^{(i),\mathrm{md}})^T$ is the $i$-th row of $\mathbf{x}_{k+1}^{\mathrm{md}}$, $(x_{k+1}^{(i),\mathrm{ag}})^T$ is the $i$-th row of $\mathbf{x}_{k+1}^{\mathrm{ag}}$ and $\mathbf{L}_k = \mathrm{diag}(L_k^{(i)} \rho^{j_k^{(i)}})$.
end for
Output $\mathbf{x}_{k+1}^{\mathrm{ag}}$
```

accelerate the convergence of Algorithm 1.

A practical issue with Algorithm 1 is that either the Lipschitz constant $L^{(i)}$ of ∇F_i or the maximum Lipschitz constant L defined in (3) may not be available to the nodes. To overcome this issue, we design a backtracking strategy so that each node can search for its own $L_k^{(i)}$ at iteration k by gradually increasing its previous $L_k^{(i)}$ with multiples of $\rho > 1$ until (12) is satisfied. Note that such searching is guaranteed to finish in finitely many times for each iteration k, and the total number of searches is bounded by $\lceil \log_\rho (L^{(i)}/L_0^{(i)}) \rceil$ at each node i for the entire computation. The resulting algorithm with such backtracking strategy is presented in Algorithm 2.

Remark: From a sensor network point of view, the communication operation in Algorithm 1 & 2 is more costly than the computations within each k round (usually communication is more energy-consuming for sensors). Thus it is preferable to evaluate our algorithm performance in terms of the number of communication rounds to reach desirable results. Notice that in both Algorithm 1& 2, only one communication is needed in one outer k round.

B. Convergence Analysis

In general we have the following convergence result for Algorithm 2 (Theorem 1). Before the analysis of Theorem 1, we would like to first prove the following Lemma.

Lemma 1. *Suppose $\hat{x}$ is a solution of $\min_x \big(\ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2\big)$ with some given y, z and L. Then we have*

$$\ell_{\mathbf{F}}(\hat{x}; y) + \frac{L}{2}\|\hat{x} - z\|^2 \leqslant \ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2 - \frac{L}{2}\|x - \hat{x}\|^2 \tag{15}$$

where $\ell_{\mathbf{F}}(x; y)$ is defined as $\ell_{\mathbf{F}}(x; y) = \mathbf{F}(y) + \nabla\mathbf{F}(y)^T(x - y)$.

Proof. Let $g(x) = \ell_{\mathbf{F}}(x; y) + \frac{L}{2}\|x - z\|^2$, then we see that $g(x)$ is convex and $\nabla g(x) = \nabla\mathbf{F}(y) + L(x - z), \nabla g(y) = \nabla\mathbf{F}(y) + L(y - z)$. Hence $\|\nabla g(x) - \nabla g(y)\| = L\|x - y\|$, which implies that $\nabla g(x)$ has Lipschitz constant L. In consequence, we can have

$$g(\hat{x}) \leqslant g(x) + \nabla g(x)^T(\hat{x} - x) + \frac{L}{2}\|x - \hat{x}\|$$

Since $\hat{x}$ is a minimizer of function $g(x)$, $\nabla g(x) = 0$. We then have $\nabla g(x)^T(\hat{x} - x) = (\nabla g(x)^T - \nabla g(\hat{x})^T)(\hat{x} - x) = -L\|x - \hat{x}\|^2$. Plugging this fact into the previous inequality yields (15). □

Theorem 1. *Suppose $\mathbf{x}^*$ is a solution of* (5)*, and the parameters $\{\theta_k\}$ in Algorithm 1 satisfies*

$$\theta_0 = 1, \quad \theta_k \in (0, 1], \quad \frac{1}{\theta_k^2} \geq \frac{1 - \theta_{k+1}}{\theta_{k+1}^2}, \tag{16}$$

then the iteration $\{\mathbf{x}_k^{\mathrm{ag}}\}$ satisfies

$$\mathbf{F}(\mathbf{x}_k^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}^*) \leq \frac{L\theta_k^2}{2(1 - \theta_k)} \left(\|\mathbf{x}_0\|^2_{\widetilde{\mathbf{W}} - \mathbf{W}} + \|\mathbf{x}^* - \mathbf{x}_0\|^2_{\widetilde{\mathbf{W}}} \right) \tag{17}$$

and

$$\sum_{k=1}^{\infty} \|(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^T)\mathbf{x}_k^{\mathrm{ag}}\|^2 < \infty, \tag{18}$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ is identity matrix and $\mathbf{1} = (1, \cdots, 1)^T \in \mathbb{R}^m$.

Proof. By the backtracking criterion in Algorithm 2 for each node i and definition of $\mathbf{F}$, we have

$$\begin{aligned} \mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) \leq & \mathbf{F}(\mathbf{x}_k^{\mathrm{md}}) + \langle \nabla\mathbf{F}(\mathbf{x}_k^{\mathrm{md}}), \mathbf{x}_{k+1}^{\mathrm{ag}} - \mathbf{x}_k^{\mathrm{md}} \rangle \\ & + \frac{1}{2}\|\mathbf{x}_{k+1}^{\mathrm{ag}} - \mathbf{x}_k^{\mathrm{md}}\|^2 \end{aligned} \tag{19}$$

By the definition of $\ell_{\mathbf{F}}$ we have

$$\begin{aligned}\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_{k+1}) &\le \ell_{\mathbf{F}}(\mathbf{x}^{\mathrm{ag}}_{k+1};\mathbf{x}^{\mathrm{md}}_k) + \frac{1}{2}\|\mathbf{x}^{\mathrm{ag}}_{k+1}-\mathbf{x}^{\mathrm{md}}_k\|^2\\ &= (1-\theta_k)\ell_{\mathbf{F}}(\mathbf{x}^{\mathrm{ag}}_k;\mathbf{x}^{\mathrm{md}}_k) + \theta_k\ell_{\mathbf{F}}(\mathbf{x}_{k+1};\mathbf{x}^{\mathrm{md}}_k)\\ &+ \frac{\theta_k^2}{2}\|\mathbf{x}_{k+1}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\\ &\le \theta_k\left(\ell_{\mathbf{F}}(\mathbf{x}_{k+1};\mathbf{x}^{\mathrm{md}}_k) + \frac{\theta_k}{2}\|\mathbf{x}_{k+1}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\right)\\ &+ (1-\theta_k)\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_k)\end{aligned}\tag{20}$$

Recall the update of $\mathbf{x}_{k+1}$ implies the optimality condition

$$\begin{aligned}\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}}\Big(&\ell_{\mathbf{F}}(\mathbf{x}_{k+1};\mathbf{x}^{\mathrm{md}}_k) + \langle\mathbf{y}_{k+1},\mathbf{x}_{k+1}\rangle\\ &+ \frac{\theta_k}{2}\|\mathbf{x}_{k+1}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\Big)\end{aligned}\tag{21}$$

Hence by Lemma 1 we have

$$\begin{aligned}&\ell_{\mathbf{F}}(\mathbf{x}_{k+1};\mathbf{x}^{\mathrm{md}}_k) + \langle\mathbf{y}_{k+1},\mathbf{x}_{k+1}\rangle + \frac{\theta_k}{2}\|\mathbf{x}_{k+1}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\\ &\le \ell_{\mathbf{F}}(\mathbf{x};\mathbf{x}^{\mathrm{md}}_k) + \langle\mathbf{y}_{k+1},\mathbf{x}\rangle + \frac{\theta_k}{2}\|\mathbf{x}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\\ &- \frac{\theta_k}{2}\|\mathbf{x}-\mathbf{x}_{k+1}\|^2\end{aligned}\tag{22}$$

Substitute this back into

$$\begin{aligned}&\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_{k+1}) \le \theta_k\Big(\ell_{\mathbf{F}}(\mathbf{x};\mathbf{x}^{\mathrm{md}}_k) + \theta_k\langle\mathbf{y}_{k+1},\mathbf{x}-\mathbf{x}_{k+1}\rangle\\ &+ \frac{\theta_k}{2}\|\mathbf{x}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2 - \frac{\theta_k}{2}\|\mathbf{x}-\mathbf{x}_{k+1}\|^2\Big) + (1-\theta_k)\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_k)\end{aligned}\tag{23}$$

Now we set $\mathbf{x}$ to any solution $\mathbf{x}^*$, subtract $\mathbf{F}(\mathbf{x}^*)$ and divide θ_k^2 on both sides to obtain

$$\begin{aligned}&\frac{1}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_{k+1}) - \mathbf{F}(\mathbf{x})\right) \le \frac{1-\theta_k}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_k) - \mathbf{F}(\mathbf{x})\right)\\ &- \frac{1}{\theta_k}\delta_{\mathbf{F}}(\mathbf{x};\mathbf{x}^{\mathrm{md}}_k) + \langle\mathbf{y}_{k+1},\mathbf{x}-\mathbf{x}_{k+1}\rangle + \frac{1}{2}\|\mathbf{x}-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2\\ &- \frac{1}{2}\|\mathbf{x}-\mathbf{x}_{k+1}\|^2\end{aligned}\tag{24}$$

where $\delta_{\mathbf{F}}(\mathbf{x}^*;\mathbf{x}^{\mathrm{md}}_k)$ is defined as $\delta_{\mathbf{F}}(\mathbf{x}^*;\mathbf{x}^{\mathrm{md}}_k) := \mathbf{F}(\mathbf{x}^*) - \ell_{\mathbf{F}}(\mathbf{x}^*;\mathbf{x}^{\mathrm{md}}_k)(\ge 0), \forall\mathbf{x}^*$.

From the update of (6) we can see that

$$\mathbf{y}_{k+1} = (\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k\tag{25}$$

where

$$\mathbf{s}_k = \sum_{t=0}^{k}\mathbf{x}_t\tag{26}$$

for each $k = 1,2,\cdots$. Furthermore, due to the fact that $\mathbf{x}^*$ is consensual and $(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{1} = \mathbf{0}$, we have

$$\begin{aligned}&\langle\mathbf{y}_{k+1},\mathbf{x}^*-\mathbf{x}_{k+1}\rangle = \langle(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k,\mathbf{x}^*-\mathbf{x}_{k+1}\rangle\\ =&-\langle(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k,\mathbf{x}_{k+1}\rangle\\ =&\langle\mathbf{s}_k-\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k\rangle\\ =&\frac{1}{2}\Big(\langle(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k,\mathbf{s}_k\rangle - \langle\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_{k+1}\rangle\\ &+\langle\mathbf{s}_k-\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{s}_k-\mathbf{s}_{k+1})\rangle\Big)\\ =&\frac{1}{2}\Big(\langle\mathbf{s}_k,(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k\rangle - \langle\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_{k+1}\rangle\\ &+\langle\mathbf{x}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{x}_{k+1}\rangle\Big)\end{aligned}\tag{27}$$

Note that here

$$\begin{aligned}&\langle\mathbf{x}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{x}_{k+1}\rangle - \|\mathbf{x}^*-\mathbf{x}_{k+1}\|^2\\ =&\langle\mathbf{x}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{x}_{k+1}\rangle - \langle\mathbf{x}^*-\mathbf{x}_{k+1},\mathbf{x}^*-\mathbf{x}_{k+1}\rangle\\ =&\langle\mathbf{x}^*-\mathbf{x}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{x}^*-\mathbf{x}_{k+1})\rangle\\ &-\langle\mathbf{x}^*-\mathbf{x}_{k+1},\mathbf{x}^*-\mathbf{x}_{k+1}\rangle\\ =&-\langle\mathbf{x}^*-\mathbf{x}_{k+1},(\mathbf{I}+\mathbf{W}-\widetilde{\mathbf{W}})(\mathbf{x}^*-\mathbf{x}_{k+1})\rangle\\ =&-\langle\mathbf{x}^*-\mathbf{x}_{k+1},\widetilde{\mathbf{W}}(\mathbf{x}^*-\mathbf{x}_{k+1})\rangle\end{aligned}\tag{28}$$

Moreover, we know the consensual solution $\mathbf{x}^*$ satisfies $\widetilde{\mathbf{W}}\mathbf{x}^* = \mathbf{x}^*$, hence there is

$$\begin{aligned}&\|\mathbf{x}^*-\widetilde{\mathbf{W}}\mathbf{x}_k\|^2 = \|\widetilde{\mathbf{W}}(\mathbf{x}^*-\mathbf{x}_k)\|^2 = \langle\mathbf{x}^*-\mathbf{x}_k,\widetilde{\mathbf{W}}^2(\mathbf{x}^*-\mathbf{x}_k)\rangle\\ &= \langle\mathbf{x}^*-\mathbf{x}_k,\widetilde{\mathbf{W}}(\mathbf{x}^*-\mathbf{x}_k)\rangle - \langle\mathbf{x}^*-\mathbf{x}_k,(\widetilde{\mathbf{W}}-\widetilde{\mathbf{W}}^2)(\mathbf{x}^*-\mathbf{x}_k)\rangle\end{aligned}\tag{29}$$

Substitute the results we obtained above to, we have

$$\begin{aligned}&\frac{1}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_{k+1}) - \mathbf{F}(\mathbf{x}^*)\right) - \frac{1-\theta_k}{\theta_k^2}\left(\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_k) - \mathbf{F}(\mathbf{x}^*)\right)\\ &+\langle\mathbf{x}^*-\mathbf{x}_k,(\widetilde{\mathbf{W}}-\widetilde{\mathbf{W}}^2)(\mathbf{x}^*-\mathbf{x}_k)\rangle\\ \le&\frac{1}{2}\left(\langle\mathbf{s}_k,(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_k\rangle - \langle\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_{k+1}\rangle\right)\\ &+\frac{1}{2}\Big(\langle\mathbf{x}^*-\mathbf{x}_k,(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{x}^*-\mathbf{x}_k) - \langle\mathbf{x}^*-\mathbf{x}_{k+1},\\ &(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{x}^*-\mathbf{x}_{k+1})\rangle\Big)\end{aligned}\tag{30}$$

Due to the setting of θ_k, we take the sum of $k = 0,1,\cdots,k$ and obtain

$$\begin{aligned}&\frac{1-\theta_{k+1}}{\theta_{k+1}^2}\left(\mathbf{F}(\mathbf{x}^{\mathrm{ag}}_{k+1}) - \mathbf{F}(\mathbf{x}^*)\right) + \sum_{t=0}^{k}\langle\mathbf{x}^*-\mathbf{x}_t,\\ &(\widetilde{\mathbf{W}}-\widetilde{\mathbf{W}}^2)(\mathbf{x}^*-\mathbf{x}_t)\rangle\\ \le&\frac{1}{2}\left(\langle\mathbf{s}_0,(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_0\rangle - \langle\mathbf{s}_{k+1},(\widetilde{\mathbf{W}}-\mathbf{W})\mathbf{s}_{k+1}\rangle\right)\\ &+\frac{1}{2}\Big(\langle\mathbf{x}^*-\mathbf{x}_0,(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{x}^*-\mathbf{x}_0) - \langle\mathbf{x}^*-\mathbf{x}_{k+1},\\ &(\widetilde{\mathbf{W}}-\mathbf{W})(\mathbf{x}^*-\mathbf{x}_{k+1})\rangle\Big)\end{aligned}\tag{31}$$

Note that $\widetilde{\mathbf{W}} \geq \mathbf{W}$ and that $\mathbf{F} > -\infty$, we obtain

$$\begin{aligned}
&-\infty < \frac{1-\theta_{k+1}}{\theta_{k+1}^2}\left(\mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}^*)\right) \\
&\quad + \sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle \qquad (32)\\
&\leq \frac{1}{2}\Big(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0\rangle - \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0)\rangle\Big)
\end{aligned}$$

Since $\widetilde{\mathbf{W}} \geq \widetilde{\mathbf{W}}^2$, this inequality implies

$$\sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle < \infty$$

and that

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}^*) \leq & \frac{\theta_{k+1}^2}{2(1-\theta_k)}\Big(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0\rangle \\
& - \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0)\rangle\Big)
\end{aligned}$$

Since there is

$$\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2 = \widetilde{\mathbf{W}}(\mathbf{I} - \widetilde{\mathbf{W}}) = \frac{\mathbf{I}+\mathbf{W}}{2}\cdot\frac{\mathbf{I}-\mathbf{W}}{2} = \frac{1}{4}(\mathbf{I} - \mathbf{W}^2) \qquad (33)$$

the first inequality implies

$$\begin{aligned}
&\frac{1}{4}\sum_{t=0}^{k}\langle(\mathbf{I} - \mathbf{W}^2)\mathbf{x}_t, \mathbf{x}_t\rangle \\
&= \sum_{t=0}^{k}\langle \mathbf{x}^* - \mathbf{x}_t, (\widetilde{\mathbf{W}} - \widetilde{\mathbf{W}}^2)(\mathbf{x}^* - \mathbf{x}_t)\rangle < \infty \qquad (34)
\end{aligned}$$

We decompose each $\mathbf{x}_t$ into two parts, namely $\mathbf{1}\mathbf{1}^T\mathbf{x}_t$ and $(\mathbf{I} - \mathbf{1}\mathbf{1}^T)\mathbf{x}_t$, then the first part is in $\mathrm{Null}(\mathbf{I} - \mathbf{W})$ and hence the inequality above implies

$$\begin{aligned}
&\frac{1}{4}(1 - \mu(\mathbf{W})^2)\sum_{t=0}^{k}\|(\mathbf{I} - \mathbf{1}\mathbf{1}^T)\mathbf{x}_t\|^2 \\
&\leq \frac{1}{4}\sum_{t=0}^{k}\langle(\mathbf{I} - \mathbf{W}^2)\mathbf{x}_t, \mathbf{x}_t\rangle < \infty \qquad (35)
\end{aligned}$$

where $\mu(\mathbf{W})$ denotes the second largest singular value of matrix $\mathbf{W}$ (since $\mathbf{W}$ is symmetric stochastic matrix, $\mu(\mathbf{W}) < 1$). The above fact means that the nonconsensual part of $\mathbf{x}_t$ is suppressed to 0. Since

$$\mathbf{x}_k^{\mathrm{ag}} = \frac{\sum\limits_{t=1}^{k} t\mathbf{x}_t}{\sum\limits_{t=1}^{k} t}, \quad for\ k = 1, 2, 3, \ldots \qquad (36)$$

we know $\mathbf{x}_k^{\mathrm{ag}}$ tends to be consensual as well.

Due to the setting of θ_k, we can readily show that

$$\theta_k \geq \frac{2}{k+2}, \quad \text{and} \quad \frac{\theta_k^2}{1-\theta_k} \leq \frac{2}{k(k+2)} \qquad (37)$$

for all $k = 1, 2, \cdots$, by induction. This implies that

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_{k+1}^{\mathrm{ag}}) - \mathbf{F}(\mathbf{x}^*) \leq & \frac{1}{k(k+2)}\Big(\langle \mathbf{s}_0, (\widetilde{\mathbf{W}} - \mathbf{W})\mathbf{s}_0\rangle \\
& - \langle \mathbf{x}^* - \mathbf{x}_0, (\widetilde{\mathbf{W}} - \mathbf{W})(\mathbf{x}^* - \mathbf{x}_0)\rangle\Big)
\end{aligned}$$

□

IV. Numerical Tests

A. Experiment Settings

In this section, we perform experiments on the seismic tomography problem using a regularized least squares model: $\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_2^2$, where λ is the regularization parameter. Simulations are performed on three different datasets: 2D synthetic dataset, 3D synthetic dataset and 3D real seismic tomography datasets. All methods are implemented in MATLAB, and experiments are performed on a PC with an Intel i5-3.0G HZ CPU and 8GB memory. In this experiment, we compare

1) Three recent decentralized methods: EXTRA [20], D-NG & D-NC [11] with our proposed FDGD algorithm.
2) FDGD and FDGD with backtracking line search.

We plot the results of average objective value ($\frac{1}{m}\sum_{i=1}^{m} f(x_i^k)$), relative error ($\sqrt{\frac{\sum_{i=1}^{m}\|x_i^k - x^*\|_2^2}{\sum_{i=1}^{m}\|x_i^0 - x^*\|_2^2}}$) and tomography images for all the three data sets, where x^* is a pre-computed centralized solution.

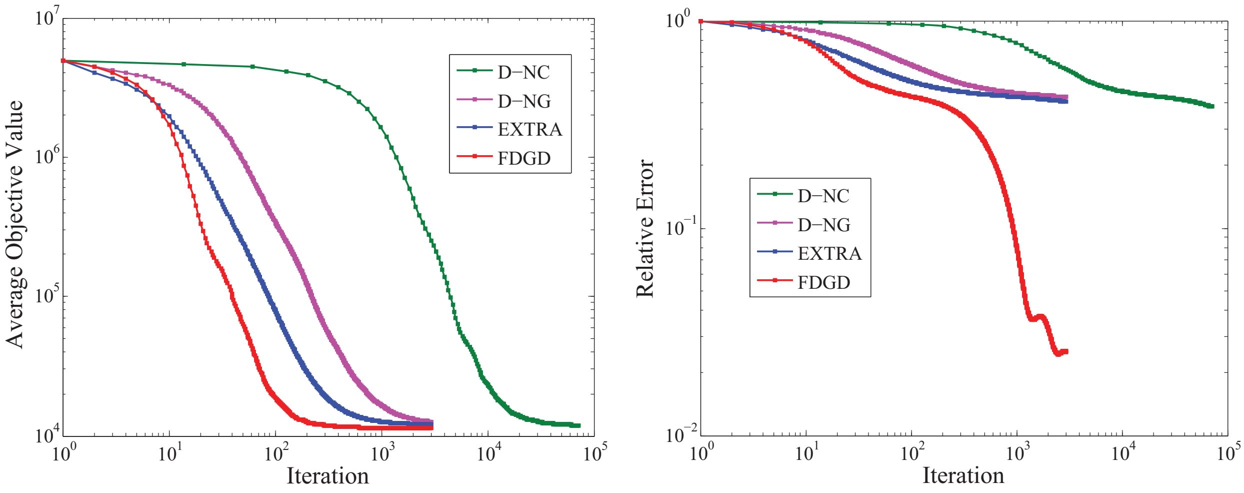
In our simulations, the regularization parameter λ is fixed and the corresponding parameter λ_i for each node i is set to $1/p$, where p is the total number of nodes in the system. In each data set, the centralized solution of optimization problem $\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_2^2$ is obtained by LSQR method. The centralized solution is taken as our benchmark comparing to the decentralized methods tested. The matrix A and vector b is constructed by stacking the sub-matrices of all the nodes. The resolution means the number of blocks along the x, y and z-axis. The communication network is generated randomly with certain number of average node degree. The parameters are described in TABLE I & II.

TABLE I: Summary of data set parameter settings

Data set	Size of A	Resolution	λ
Synthetic 2D	16,384x4,096	64x64	1.0
Synthetic 3D	40,000x32,768	32x32x32	10^{-4}
Real data 3D	18,161x768,000	160x200x24	1.0

TABLE II: Network settings in the data sets

Data set	# of nodes	Avg node degree
Synthetic 2D	32	3
Synthetic 3D	100	3
Real data 3D	11	2



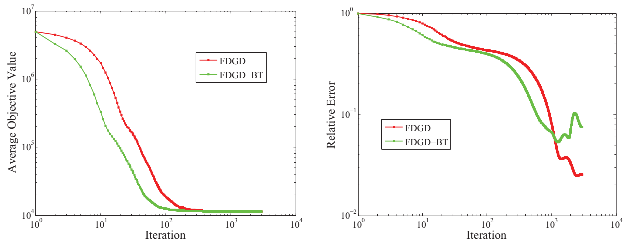
(a)

(b)

Fig. 2: FDGD convergence behavior in 2D synthetic seismic data set. (a) is the plot of average objective value comparing FDGD with EXTRA, D-NG and D-NC methods. (b) is the relative error comparison plot.

B. Synthetic Data (2D Model)

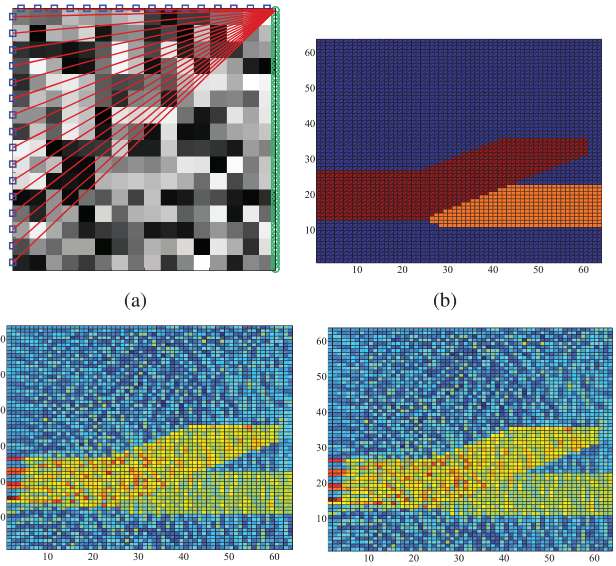
The performance analysis here is based on the data set generated using code in [29]. We create a 2D seismic tomography test problem with a square domain, using sources located on the right boundary (green dots) and receivers (seismographs) scattered along the left and top boundary (blue squares). The rays are transmitted from each source to each receiver (red lines) (see Figure 4(a)). The experiment results are demonstrated in Figure 2-4.



(a)

(b)

Fig. 3: Convergence behavior comparison of FDGD and FDGD-BT in 2D synthetic data set. (a) and (b) depict the FDGD and FDGD with backtracking line search implementation in terms of average objective value and relative error, respectively.



(a)

(b)

(c)

(d)

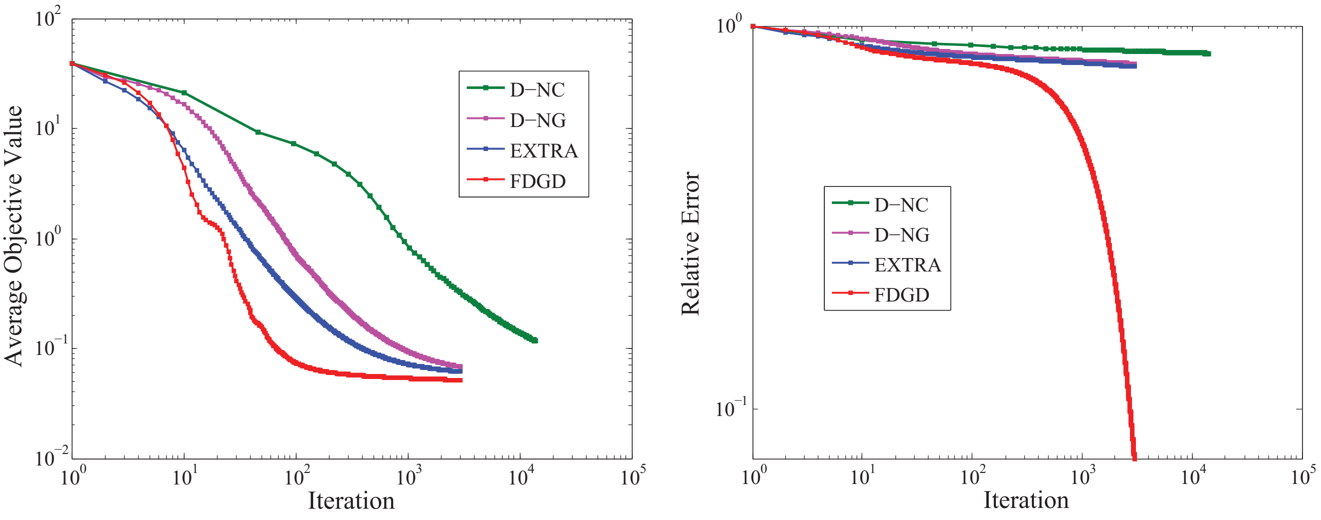
Fig. 4: FDGD tomography results of 2D synthetic data set. (a) describes the 2D seismic model we used. (b) shows the ground truth of original seismic image. (c)-(d) exhibit the tomography results using centralized solution and FDGD, respectively.

C. Synthetic Data (3D Model)

In this section, the evaluation of algorithm is illustrated by simulating seismic data on a synthetic model of resolution 32^3 consisting of a magma chamber (low velocity area) in a 10 km^3 cube. 100 stations are randomly distributed on top of the cube and form a network. To construct the matrix A and vector b, 400 events are generated and we compute the travel times from every event to each node based on the ground truth, and send the event location and travel time to corresponding node with white Gaussian noise. Figure 5-7 illustrate the experiment results in this data set.

D. Real-world Data (3D Model)

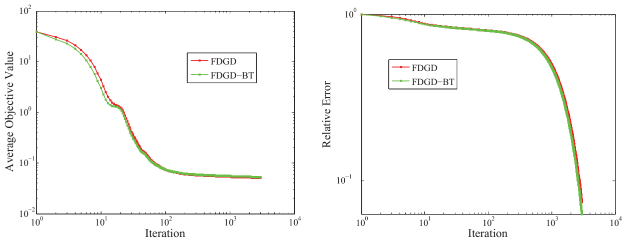
To study the performance of the two proposed algorithms in realistic scenarios, we use ten years (2001-2011) real seismic



(a)

(b)

Fig. 5: Comparison of convergence performance in 3D synthetic data set. (a)-(b) are comparing FDGD with EXTRA, D-NG and D-NC methods.



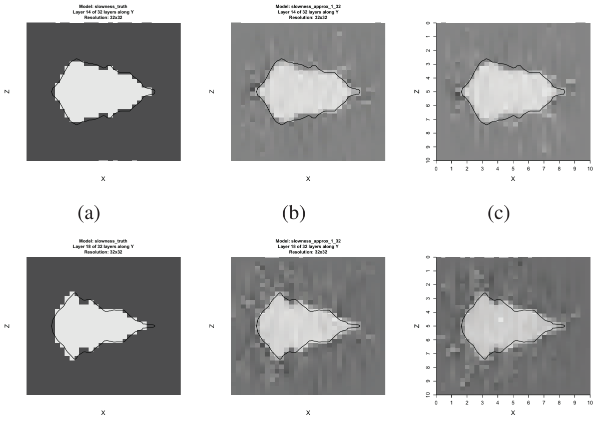
(a)

(b)

Fig. 6: Convergence behavior comparison of FDGD and FDGD-BT in 3D synthetic data set. (a) and (b) depict the FDGD and FDGD with backtracking line search implementation in terms of average objective value and relative error, respectively.

event data of Mount St. Helens in Washington, USA for the experiment. The data were collected from 78 stations and we construct them into 11 clusters and form a network based on the clusters. Notice that unlike synthetic data used in previous section, there is no ground truth in this real data scenario. Hence we focus on the comparison of the proposed methods with centralized processing scheme, which can be seen as a benchmark that *fully* utilize the data available. Results are shown in Figure 8-10.

Remark: Previous simulation results have demonstrated the superiority of proposed FDGD over other existing methods. In all the data sets, FDGD can obtain near "optimal" (the centralized approach) solution with reasonable number of communication rounds even in extremely low-connectivity networks. The performance of FDGD-BT is almost the same as FDGD implying we can still achieve similar results without knowing Lipschitz constant L. Please note that the value of regularization parameter λ also determines the convexity property of the objective function. We do observe linear convergence rate of EXTRA for strongly convex functions as claimed in [20]. However, we found that in the simulated synthetic data sets, smaller λ is better and more suitable for image recovery. In the real data case, since no "ground truth" is available, for simplicity, we also choose $\lambda = 1$ in our experiments. In fact, $\lambda = 1$ is relative small comparing to the data fitting term such that the objective function is not



(a)

(b)

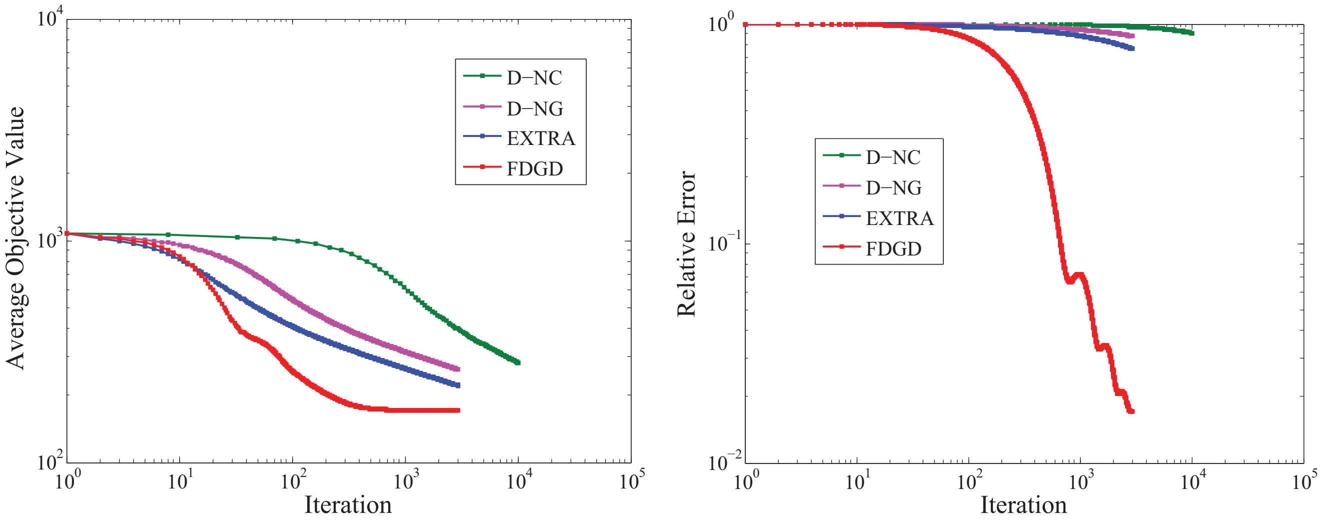
(c)

(d)

(e)

(f)

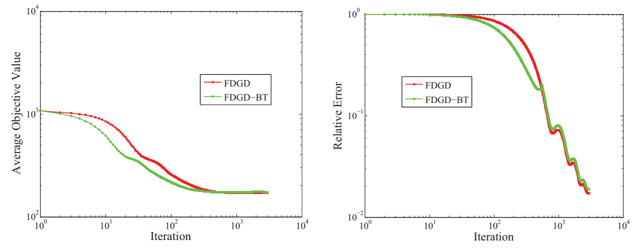
Fig. 7: Vertical slices of 3D synthetic model tomography. Fig. (a)-(c) are results of layer 14 along y-axis and Fig. (d)-(f) are results of layer 18. Left-most column is the ground truth, the middle column shows the centralized solution and the right-most column contains the solution using our proposed FDGD algorithm.



(a)

(b)

Fig. 8: Real data tomography inversion results comparison. (a)-(b) are comparing FDGD with EXTRA, D-NG and D-NC methods. (c) and (d) show peformances of FDGD and FDGD with backtracking line search implementation. (e)-(f) describe the solutions of vertical slices of at depth $0.9\ km$ (left:centralized, right:FDGD). (g)-(h) exhibit the tomography results at depth $4.9\ km$.



(a)

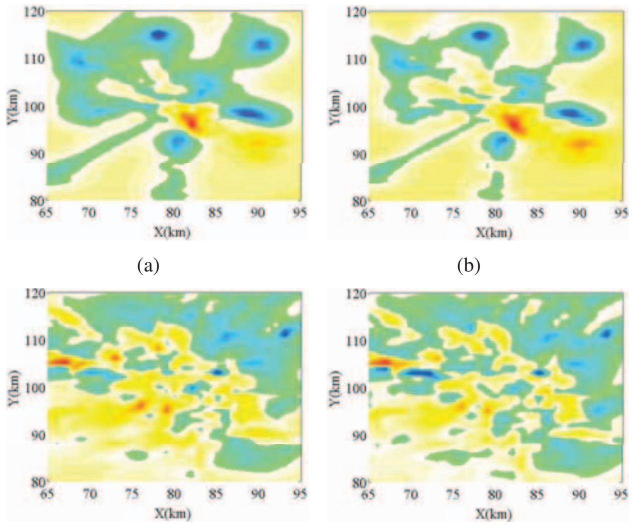
(b)

Fig. 9: Convergence behavior comparison of FDGD and FDGD-BT in 3D real data set. (a) and (b) show performances of FDGD and FDGD with backtracking line search implementation.

quite “strongly convex”. That explains why EXTRA does not show linear convergence in our results. To show an example of this scenario, we also perform the experiment on 2D synthetic data set with $\lambda = 20$ (see Fig. 11).

V. Conclusion

Distributed and decentralized optimization is well suited to Big Data applications, and in particular to analytics in distributed architectures. In this paper we developed a novel fast decentralized gradient descent method whose convergence does not require diminishing step sizes as in regular decentralized gradient descent methods, and prove that this new method can reach optimal convergence rate of $O(1/k^2)$ where k is the communication/iteration number. In the seismic tomography application, we conducted experiments on synthetic and real-world sensor network seismic data. The results exhibit that



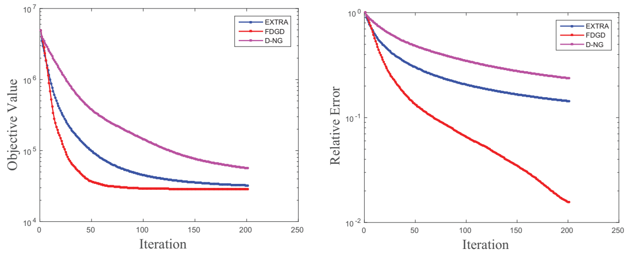
(a)

(b)

(c)

(d)

Fig. 10: Seismic tomography comparison of the 3D real data set. (a)-(b) describe the solutions of vertical slices of at depth $0.9\ km$ (left:centralized, right:FDGD). (c)-(d) exhibit the tomography results at depth $4.9\ km$. The range of x-axis is from 65 to 95 km and the y-axis is from 80 to 120 km. The color in the figure represents the relative velocity perturbation in specific location. More red means larger (negative) value of perturbation. More blue means larger (positive) value of perturbation



(a)

(b)

Fig. 11: Convergence behavior comparison of 2D synthetic data set with regularization parameter $\lambda = 20$.

the proposed algorithms significantly outperform the current state-of-the-arts.

References

[1] R. P. Bording, A. Gersztenkorn, L. R. Lines, J. A. Scales, and S. Treitel, "Applications of seismic travel-time tomography," *Geophysical Journal of the Royal Astronomical Society*, vol. 90, no. 2, pp. 285–303, 1987. [Online]. Available: http://dx.doi.org/10.1111/j.1365-246X.1987.tb00728.x

[2] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 32–43, Sept 2014.

[3] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance Analysis of the Consensus-Based Distributed LMS Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2010.

[4] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[5] A. H. Sayed and C. G. Lopes, "Distributed Recursive Least-Squares Strategies Over Adaptive Networks," in *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, Nov. 2006, pp. 233–237.

[6] I. Matei and J. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 754–771, Aug 2011.

[7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[8] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 6855–6860.

[9] ——, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv:1406.2075*, 2014.

[10] I.-A.Chen, "Fast distributed first-order methods," *PhD thesis, Massachusetts Institute of Technology*, 2012.

[11] J. M. F. M. Dusan Jakovetic, Joao Xavier, "Fast distributed gradient methods," *arXiv:1112.2972v4*, 2014.

[12] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv:1310.7063*, 2013.

[13] M. Zargham, A. Ribeiro, and A. Jadbabaie, "A distributed line search for network optimization," in *American Control Conference (ACC), 2012*, June 2012, pp. 472–477.

[14] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147698

[15] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, Sep 1986.

[16] J.N.Tsitsiklis, "Problems in decentralized decision making and computation," *Technical report, DTIC Document*, 1984.

[17] U. T. H. Terelius and R. Murray, "Decentralized multi-agent optimization via dual decomposition," *IFAC*, 2011.

[18] G. Shi and K. H. Johansson, "Finite-time and asymptotic convergence of distributed averaging and maximizing algorithms," *arXiv:1205.1733*, 2012.

[19] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, April 2004, pp. 20–27.

[20] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *arXiv:1404.6264*, 2014.

[21] E. Wei and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers," *arXiv:1307.8254*, 2013.

[22] P. C. F. Iutzeler, P. Bianchi and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," *arXiv:1303.2837*, 2013.

[23] Y. Nesterov, "Gradient methods for minimizing composite objective function," Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), CORE Discussion Papers 2007076, 2007. [Online]. Available: http://EconPapers.repec.org/RePEc:cor:louvco:2007076

[24] ——, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*, 1st ed. Springer Netherlands. [Online]. Available: http://www.worldcat.org/isbn/1402075537

[25] ——, "A method for unconstrained convex minimization problem with the rate of convergence o(1/k2)." *In Doklady AN SSSR, volume 269, pages 543547*, 1983.

[26] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, 2008.

[27] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2006.874516

[28] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1109/tsp.2009.2016247

[29] P. C. Hansen and M. Saxild-Hansen, "Air tools - a matlab package of algebraic iterative reconstruction methods," *Journal of Computational and Applied Mathematics*, vol. 236, no. 8, pp. 2167 – 2178, 2012, inverse Problems: Computation and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042711005188