Cyber-Physical Systems, 2015
Vol. 1, Nos. 2–4, 91–112, http://dx.doi.org/10.1080/23335777.2015.1062049

Taylor & Francis
Taylor & Francis Group

Decentralised seismic tomography computing in cyber-physical sensor systems

Liang Zhao[a], Wen-Zhan Song[a]*, Lei Shi[a] and Xiaojing Ye[b]

[a]*Department of Computer Science, Georgia State University, Atlanta, GA, USA;* [b]*Department of Mathematics and Statistics, Georgia State University, Atlanta, GA, USA*

(*Received 26 March 2015; accepted 4 June 2015*)

This paper presents an innovative decentralised seismic tomography computing paradigm in cyber-physical sensor systems, where each sensor node computes the tomography based on its *partial* information and through gossip with local neighbours only. The key challenge is the potential high communication overhead due to limited knowledge of each node about the entire network topology and information. The aim of this paper is to develop efficient algorithms for maximising the quality of tomography resolution while minimising the communication cost. We reformulate the conventional seismic tomography problem and exploit the alternating direction method of multipliers method to design two distributed algorithms. One is a synchronous algorithm and the other is asynchronous and more fault-tolerant and scalable. We theoretically prove that both proposed algorithms can reach their convergent solutions in a linear rate in terms of the number of communication rounds. Extensive evaluations on both synthetic and real datasets validate the superior efficiency of the proposed algorithms. They not only achieve near-optimal (compare to centralised solution) high-quality tomography but also retain low communication cost even in sparse networks.

Keywords: cyber-physical system; distributed computing; in-network processing; seismic tomography; sensor network

1. Introduction

Current seismic imaging systems use sensors placed on earth surface to acquire information on the compressional waves generated by underground seismic activities. The acquired data are then used to derive the internal velocity structure of the earth subsurface. However, they are prone to several bottleneck problems. First, for example, in previous volcano monitoring investigations, the number of stations (sensors) is up to 20 due to deployment cost and other issues. The resulting low station coverage inevitably becomes a main constraint on our capability of obtaining high-resolution tomography model.[1] Second, even if thousands of nodes can be incorporated (e.g. petroleum exploration systems), the huge volume of raw seismic data has to be collected into a central place for post-processing. The underlying time-consuming process prohibits its potential for effective disaster warning in which the timescale can be tens of minutes. More introduction and motivation can be seen in [2].

We adopt the travel time-based seismic tomography in the present work to reveal the velocity model inside the volcano.[3] The basic procedure of seismic tomography is illustrated in Figure 1 involving three steps.

* Corresponding author. Email: wsong@gsu.edu

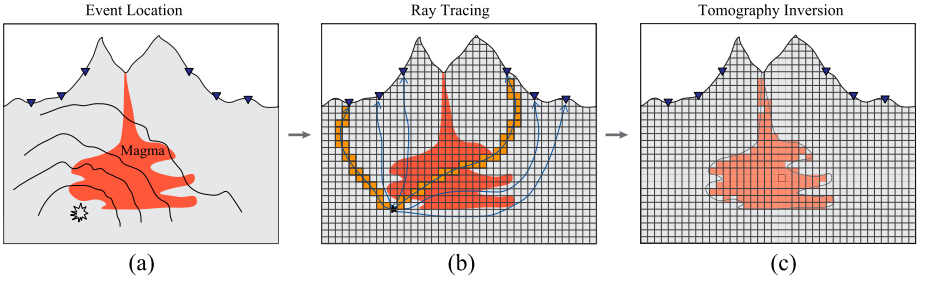


Figure 1. Procedure of seismic tomography.[2]

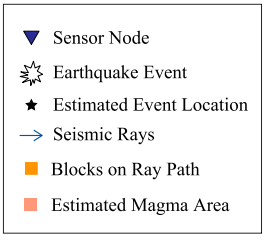


Figure 2. Legend in Figure 1.

The first step is 'Event Location', which means we need estimate the where and when certain seismic event occurs (Figure 1(a)). The second step is named 'Ray tracing'. This process is to estimate the rays coming from the event location to the receivers. The traces of the rays are affected by the velocity structures of the materials they travel along. In other words, the rays contain information of the internal velocity model that we are interested in (see Figure 1(b) as a conceptual view of this process). The final step is 'Tomographic Inversion'. It basically utilises the traced ray paths to image a 3-D tomography of the velocity model within the volcano (Figure 1(c)). Interested readers are referred to [2] for more details.

In this paper, we focus on the tomographic inversion process, which can be formulated as a large linear inversion problem (detailed formulation in next section). However, it is highly demanded to have a distributed computing algorithm to fill the gap between the 'centralised' nature of traditional tomography computing and the 'distributed' feature of sensor networks.

Considering the issues discussed above, we are thus motivated to propose two *in situ* seismic imaging methods on decentralised large-scale cyber-physical sensor systems. They enjoy the nature of distributed and *fully* decentralised computing and gossip between neighbours only. One is synchronous and the other is asynchronous. They exhibit the following merits: (i) simple implementation and no data fusion centre or coordinator required; (ii) close to 'optimal' (centralised algorithm is considered as the benchmark) imaging quality even in sparse networks with severe packet loss, which are scenarios often occur in volcano monitoring; (iii) guaranteed linear rate approaching to consensus optimal solution for every node; (iv) lower communication cost comparing to other potential distributed methods in the literature.

In our hardware design, each station is capable of sensing, communicating and computing. The hardware design of each station adopts the stackable design principles and there are four types of stacks in each station: computing board, communication board, sensor

board and energy board. Detail hardware configuration is described in [4]. We plan to test the proposed decentralised seismic tomography algorithms in the underlying system architecture for real-world deployment.

The rest of the paper is organised as follows. Section 2 reviews the related work. The reformulation of seismic tomography and derivation of Synchronised Distributed Seismic Tomography Algorithm (sDSTA) are described in Section 3. Next in Section 4, we present the derivation of Asynchronous Distributed Seismic Tomography Algorithm (aDSTA). Section 5 focuses on the convergence analysis of sDSTA and aDSTA. In Section 6, the effectiveness of the proposed approaches is evaluated. Section 7 describes potential benefits of proposed methods under different modellings. Section 8 provides the final remarks.

2. Related work

There has been a rich history that people used iterative methods (e.g. conjugate gradient) to solve the seismic tomography inversion problem in (3).[5] However, they are designed or specialised for computing resources in which 'centralised' computing is performed. A parallel splitting method for solving (3) was proposed in [6], where the local solutions are weighted combined to obtain the final solution. The method in [6] can distribute the computational tasks into multiple computers while the communication between the nodes becomes an issue, that is every node needs to communicate with the rest of the nodes in each iteration. Several studies in the context of signal processing applications bring applicable distributed algorithms in sensor networks.[7–9] Sayed and Lopes developed a Distributed Recursive Least Squares (D-RLS) strategy in [7]. The main issue is that a 'cyclic path' exists in the network to execute the computation and a large dense matrix needs to be exchanged between nodes. Schizas proposed the Distributed Least Mean Square (LMS) algorithm [8] in which every node keeps a local copy of the common variable and all the nodes are expected to reach consensus on the value of the decision variable eventually. Although the communication protocol in [8] only transmits the solutions and is more efficient than the one in [6], certain 'bridge' sensors are still required as fusion centres for collecting the information within the neighbours and distributing processed information back to the neighbours. This results in huge communication burden in the 'bridge' sensors. Multi-resolution-based and component-average-based distributed algorithms have been proposed in [2] and [10], respectively. However, their methods are not performed in a decentralised fashion, which means large data aggregations or multi-hop communications are still involved in their methods. In contrast to the aforementioned existing works, we proposed algorithms in the present work which are communication and computational load balanced, multi-hop communication free, robust and privacy preserving.

Recently, several fast distributed algorithms have been proposed for solving general convex and differentiable functions.[11–13] They assume the network has a synchronised clock while this setting might not be always feasible or favourable in distributed sensor networks. In this paper, besides the synchronised algorithm, we develop an asynchronous version, which can alleviate the synchronisation task and further reduce the communication overhead.

3. Synchronised distributed seismic tomography algorithm–sDSTA

3.1. Problem formulation

In this section, we give the formal formulation of decentralised seismic tomography computation problem. Similar or even more detail formulation leading to (3) were described

in [2,10]. We assume the event location and ray tracing steps are completed. Let $\mathbf{t}_i^* = [t_{i1}^*, t_{i2}^*, \ldots, t_{iJ}^*]^T$, where t_{ij}^* is the travel time experienced by node i in the jth event. The travel time of a ray is the sum of the slowness in each block times the length of the ray within that block, i.e. $t_{ij}^* = \mathbf{A}_i[j,h] \cdot \mathbf{s}^*[h]$ where $\mathbf{A}_i[j,h]$ is the length of the ray from the jth event to node i in the hth block and $\mathbf{s}^*[h]$ is the slowness of the hth block. Let $\mathbf{t}_i^0 = [t_{i1}^0, t_{i2}^0, \ldots, t_{iJ}^0]^T$ be the unperturbed travel times where $t_{ij}^0 = \mathbf{A}_i[j,h] \cdot \mathbf{s}^0[h]$. We can thus have the following compact form:

$$\mathbf{A}_i\mathbf{s}^* - \mathbf{A}_i\mathbf{s}^0 = \mathbf{A}_i\mathbf{s} \tag{1}$$

Let $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{iJ}]^T$ denote the travel time residual such that $\mathbf{t}_i = \mathbf{t}_i^* - \mathbf{t}_i^0$, Equation (1) is equivalent to:

$$\mathbf{A}_i\mathbf{s} = \mathbf{t}_i \tag{2}$$

We now have a linear relationship between the travel time residual observations, $\mathbf{t}_i$, and the slowness perturbations, $\mathbf{s}$. Since each ray path intersects with the model only at a small number of blocks compared with n, the design matrix, $\mathbf{A}_i$, is sparse. The seismic tomography inversion problem is to solve a linear system of equations:

$$\mathbf{A}\mathbf{s} = \mathbf{t} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{t} \in \mathbb{R}^m$ and $\mathbf{s} \in \mathbb{R}^n$ is the vector to be estimated. This system is usually organised to be overdetermined (let the number of rays (the number of measurements) m is larger than n) and the inversion aims to find the least squares solution $\mathbf{s}$ such that

$$\mathbf{s} = \arg\min_{\mathbf{s}} \parallel \mathbf{A}\mathbf{s} - \mathbf{t} \parallel^2 \tag{4}$$

Since vector $\mathbf{t}$ is usually noise corrupted, the system is inconsistent. In consequence, one needs to solve a regularised least square problem (the regularisation term can vary, we use Tikhonov regularisation here since it is the most popular one in seismic inversion problem):

$$\min_{\mathbf{s}} \quad \frac{1}{2}\|\mathbf{A}\mathbf{s} - \mathbf{t}\|_2^2 + \lambda^2\|\mathbf{s}\|_2^2 \tag{5}$$

From (2), we can equivalently express (5) as follows.

$$\min_{\mathbf{s}} \quad \sum_{i=1}^{p} c_i(\mathbf{s}) \tag{6}$$

where p sensor nodes are considered in the system, $c_i(\mathbf{s}) = \frac{1}{2}\|\mathbf{A}_i\mathbf{s} - \mathbf{t}_i\|_2^2 + \lambda_i^2\|\mathbf{s}\|_2^2$. The ith sensor node has the knowledge of $\mathbf{A}_i$ and $\mathbf{t}_i$ only. We can see (6) is meant a problem that each sensor node jointly optimises (using their private local functions c_i's) to reach a consensus on the global common interest $\mathbf{s}$ that minimising the total cost. Notice that minimising each local function $\mathbf{c}_i$, respectively, is clearly a suboptimal solution since only partial information is utilised and the result would be inevitably biased.

3.2. Preliminary of ADMM

In this paper, we leverage the ADMM technique to devise our distributed seismic tomography solution for (6). To briefly illustrate the general ADMM algorithm,[14] consider the prototype problem:

$$\begin{aligned} &\text{minimise } f(x) + g(z) \\ &\text{subject to: } Ax + Bz = c \end{aligned} \tag{7}$$

with variables $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^m$, where $A \in \mathbf{R}^{p\times n}$, $B \in \mathbf{R}^{p\times m}$ and $c \in \mathbf{R}^p$. Functions f and g are assumed to be convex. As in the method of multipliers, the augmented Lagrangian can be formed:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\,\|Ax + Bz - c\|_2^2 .$$

ADMM consists of the iterations:

$$x^{k+1} := \arg\min_x \; L_\rho(x, z^k, y^k) \tag{8a}$$

$$z^{k+1} := \arg\min_z \; L_\rho(x^{k+1}, z, y^k) \tag{8b}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \tag{8c}$$

where $\rho > 0$ is the predefined augmented Lagrangian parameter and y is the Lagrangian multiplier (dual variable) of the constraint in (7). The ADMM algorithm is considered to have three steps: an x-minimisation (8a), a z-minimisation step (8b) and a dual variable update (8c).

3.3. *Algorithm design*

To fit the generic two-block ADMM framework, a natural reformulation of (6) can be posed as:

$$\begin{aligned} &\min_{\{\mathbf{s}_i\}} \sum_{i=1}^{p} c_i(\mathbf{s}_i) \\ &\text{s.t. } \mathbf{s}_i = \mathbf{s}_j, \;\; j \in \mathcal{N}_i, \forall i \end{aligned} \tag{9}$$

where $\mathbf{s}_i$ is the local estimate of the global variable $\mathbf{s}$ at ith sensor node and $\mathcal{N}_i$ is the set of neighbours of node i. The constraints in (9) are consensus constraints that lead the neighbours to have agreement on the decision variable $\mathbf{s}$. However, in the constraints of (9), the estimate $\mathbf{s}_i$ is directly coupled with neighbouring nodes, a parallel processing of the optimisation problem is not possible.

To overcome this issue, we introduce auxiliary variables $\mathbf{q}_{ij}$, then the constraints in (9) are decomposed equivalently into the following form (e.g. see [15]):

$$\begin{aligned} &\min_{\{\mathbf{s}_i, \mathbf{q}_{ij}\}} \sum_{i=1}^{p} c_i(\mathbf{s}_i) \\ &\quad \text{s.t. } \mathbf{s}_i = \mathbf{q}_{ij}, \mathbf{s}_j = \mathbf{q}_{ij}, \quad j \in \mathcal{N}_i, \forall i \end{aligned} \tag{10}$$

Let k be the iteration index and the generic ADMM solution of (10) consists of updates in (12a)–(12d). Denote $\mathcal{L}_\rho$ as the augmented Lagrangian of optimisation problem (10) given by:

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{s}_i, \mathbf{q}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}) = \sum_{i=1}^{p} \Bigg\{ c_i(\mathbf{s}_i) + \sum_{j\in\mathcal{N}_i} \{ &\mathbf{y}_{ij}^T(\mathbf{s}_i - \mathbf{q}_{ij}) \\ &+ \mathbf{z}_{ij}^T(\mathbf{s}_j - \mathbf{q}_{ij}) + \frac{\rho}{2}\|\mathbf{s}_i - \mathbf{q}_{ij}\|_2^2 + \frac{\rho}{2}\|\mathbf{s}_j - \mathbf{q}_{ij}\|_2^2 \} \Bigg\} \end{aligned} \tag{11}$$

where $\rho > 0$ is a predefined parameter. Note that $\mathbf{y}_{ij}$ and $\mathbf{z}_{ij}$ are the Lagrangian multipliers for the first and second constraint in (10), respectively.

$$\mathbf{s}_i^{k+1} = \arg\min_{\{\mathbf{s}_i\}} \ \mathcal{L}_\rho(\mathbf{s}_i, \mathbf{q}_{ij}^k, \mathbf{y}_{ij}^k, \mathbf{z}_{ij}^k) \tag{12a}$$

$$\mathbf{q}_{ij}^{k+1} = \arg\min_{\{\mathbf{q}_{ij}\}} \ \mathcal{L}_\rho(\mathbf{s}_i^{k+1}, \mathbf{q}_{ij}, \mathbf{y}_{ij}^k, \mathbf{z}_{ij}^k) \tag{12b}$$

$$\mathbf{y}_{ij}^{k+1} = \mathbf{y}_{ij}^k + \rho(\mathbf{s}_i^{k+1} - \mathbf{q}_{ij}^{k+1}) \tag{12c}$$

$$\mathbf{z}_{ij}^{k+1} = \mathbf{z}_{ij}^k + \rho(\mathbf{s}_j^{k+1} - \mathbf{q}_{ij}^{k+1}) \tag{12d}$$

(12a)–(12d) are over every node i and all its neighbours $j \in \mathcal{N}_i$. Typically, each node i updates its variables in (12a)–(12d) and performs only local communications with its immediate neighbours.

Notice that the generic ADMM in (12a)–(12d) requires sensor node i to update variables $\{\mathbf{s}_i, \mathbf{q}_{ij}, \mathbf{y}_{ij}, \mathbf{z}_{ij}\}$ in each iteration, where each variable is the same size as $\mathbf{s}_i$. It is evidently a huge burden when the size of variable $\mathbf{s}_i$ is large. Fortunately, a simplified efficient version for our problem can be obtained as follows (only needs to transmit $\mathbf{s}_i$ for sensor i).

$$\mathbf{s}_i^{k+1} = \left(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\right)^{-1}\left(\mathbf{A}_i^T\mathbf{t}_i - \mathbf{u}_i^k + \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^k\right)\right) \tag{13a}$$

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \rho'\left(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^{k+1}\right) \tag{13b}$$

where $|\mathcal{N}_i|$ represents the cardinality of set $\mathcal{N}_i$. $\rho' = \rho/2$.

Proposition 1 *For the problem in* (10)*, the updates of* (12a)–(12d) *are equivalent to* (13a)–(13b).

Proof See the detail derivation in Appendix 1. Interested readers are referred to [16] and [17] for similar derivations of the average consensus problem and the sparse linear regression problem, respectively. □

Note that obtaining $\mathbf{s}_i^{k+1}$ in (13a) is solving a linear system of equations. Theoretically, we can use any solver while considering the property of tomography inversion problem, we adopt Bayesian Algebraic Reconstruction Technique (BART) method in this paper.[18]

A closer look into (13a) and (13b) reveals that for node i, the information needed is only the summation of its neighbours' current estimates $\mathbf{s}_j^k$, $j \in \mathcal{N}_i$. Thus a natural implementation is that, in every iteration, all the nodes broadcast their current estimates to all their neighbours. After receiving all the neighbours' estimates, the sensor nodes can perform local updates in parallel. When all the nodes finish their computations, they will broadcast their new estimates again. The idea is summarised in Algorithm 1.

Note that in step 5 of Algorithm 1, the minimisation of $\mathbf{s}_i^k$ is not carried out exactly. In this situation, the proposed ADMM-based algorithm will still converge provided certain mild conditions.[19, p.26]. More importantly, solving the minimisation of $\mathbf{s}_i^k$ to a very high accuracy, especially at the initial iterations, may not be worthwhile. The reason is two-folds: first, at the initial iterations (communications), node i has limited information (even no information) about the whole structure due to its nature of local communication with immediate neighbours. In this situation, the solution $\mathbf{s}_i$ may has a large deviation from the true solution no matter how hard we try for the solution. In addition, solving $\mathbf{s}_i^k$ is an iterative process and might require a large amount of time (iterations) to have a solution with very small relative error.

Algorithm 1 Synchronised Distributed Seismic Tomography Algorithm (sDSTA)

1: Initialization: Input $\mathbf{A}_i, \mathbf{t}_i$ for sensor $i, \forall i$.
Initialize $\mathbf{s}_i^0 = \mathbf{0}, \mathbf{u}_i^0 = \mathbf{0}, \forall i$. Set parameters $\rho' > 0$ and $\lambda_i, \forall i$.
2: At each iteration k, $(k = 0, 1, \ldots)$, every sensor i broadcasts its $\mathbf{s}_i^k$ to its neighbors $j \in \mathcal{N}_i$.
3: Once every head i receives all its neighbors's estimates,
do the following local updating.
4: Compute $\mathbf{u}_i^k$ based on (13b).
5: Update $\mathbf{s}_i^{k+1}$ by running a finite number of BART iterations on (13a), with initial value $\mathbf{s}_i^k$.
6: When all the nodes finish their primal updates $\mathbf{s}_i^{k+1}, \forall i$,
broadcast them to all neighbors, and repeats step 4-6
until certain stopping criterion has been satisfied.

Regarding the communication cost of this proposed scheme, we see that it depends on the network topology. The quantitative relation can be described in Proposition 2.

PROPOSITION 2 *The communication cost of sDSTA for each node is* $o(k)$ *and the total communication cost is at most* $o(kN(N-1))$ *(complete network) and can be as low as* $o(2kN)$ *(ring network) with respective to network size* N *and communication round* k.

In addition, the convergence speed of the proposed iterative method also depends on the communication topology. In general, network with larger average degree is expected to converge faster. We will discuss it in detail in Section 6.

4. Asynchronous distributed seismic tomography algorithm–aDSTA

We can recall that the updating in sDSTA requires each sensor node to wait until receiving the solution of its slowest neighbour. While in some cases, it might be desirable to update with less coordination and in an asynchronous fashion such that every node can *independently* determine its own actions. In this section, we provide a solution methodology to address this problem.

4.1. Component 1: adaptive communication

One major motivation of this adaptive communication scheme is that in sensor network applications such as our distributed seismic tomography, communication is the most energy-consuming component. In each iteration, sensor nodes need to broadcast their local estimates $\mathbf{s}_i$'s, which is a long vector (e.g. in real data of Mount St. Helens, the size could be around 768,000). Thus, it is highly demanded to design a mechanism that can reduce the communication overhead for the proposed distributed algorithm.[20] Specifically, assume a coordinator exists requiring all the nodes to perform m iterations. However, some node may already own a good enough solution in less than m iterations, and its value does not change much for several successive iterations. In this situation, it would be a waste if these nodes continue to communicate and update. To address this problem, an adaptive communication scheme is proposed in the following context.

Each sensor node should be able to determine if it needs to communicate and computing. In each iteration, each node will compute an index which depends on the relative updates. If this index is greater than a threshold, then the node will stop grabbing updates of its neighbours.

Proposed Scheme: For sensor node i, if the following two conditions satisfied:

$$\frac{\|\mathbf{s}_i^k - \mathbf{s}_i^{k-1}\|}{\|\mathbf{s}_i^{k-1}\|} \leq \varphi_i \text{ and } \left\|\mathbf{u}_i^k - \mathbf{u}_i^{k-1}\right\| \leq \mu_i \tag{14}$$

then $ct_i = ct_i + 1$, otherwise $ct_i = 0$, where ct_i is a counter of node i and define E_i as a threshold for node i. When $ct_i > E_i$, sensor node i stops updating, which also means that it stops querying the estimates from its neighbours. When node i stops updating, its neighbours can still ask for the latest value of node i to improve their accuracy. Notice that in this situation, once node j obtains the last update of node i, it will not need more transmissions from i since the estimate of node i will not change. Note that the stopping criterion (14) is different from those of conventional ADMM methods where all the sensors shall stop computations at the same time using a common stopping criteria and common primal and dual tolerances. Stopping criteria (14) allows a sensor i to stop its computation asynchronously and independently from other sensors. However, these criteria themselves are insufficient for asynchronous implementation. Synchronisation is still required for dual and primal variable updates at iteration $k + 1$ due to their dependencies on values of kth iteration.

Algorithm 2 Asynchronous Distributed Seismic Tomography Algorithm (aDSTA): For sensor i

1: Initialization: Input $\mathbf{A}_i, \mathbf{t}_i$.
Initialize $\mathbf{s}_i^0 = \mathbf{0}, \mathbf{u}_i^0 = \mathbf{0}, ct_i = 0$. Set parameters $\rho' > 0,\ \varphi_i,\ \mu_i,\ E_i$ and λ_i.
2: **while** *criterion* $ct_i > E_i$ *is not satisfied* **do**
3: At new iteration $k + 1$, sensor i selects a neighbor j according to the probability distribution in (15).
4: Sensor i contacts j to obtain j's current value $\mathbf{s}_j$.
5: Compute $\mathbf{u}_i^k$ based on (13b) with replacing $\sum_{j\in\mathcal{N}_i} \mathbf{s}_j$ by $|\mathcal{N}_i| \cdot \mathbf{s}_j$.
6: Update $\mathbf{s}_i^{k+1}$ by running a finite number of BART iterations on (A7) with replacing $\sum_{j\in\mathcal{N}_i} \mathbf{s}_j$ by $|\mathcal{N}_i| \cdot \mathbf{s}_j$.
Use $\mathbf{s}_i^k$ as the initial guess.
7: When sensor i finishes its primal updates $\mathbf{s}_i^{k+1}$, go to the following:
8: **if** *criterion in* (14) *is satisfied* **then**
9: $ct_i = ct_i + 1$
10: **else**
11: $ct_i = 0$
12: **end if**
13: **end while**

4.2. Component 2: randomised gossiping

To ensure full asynchronous implementation, the update of each node should be performed in a randomised manner (otherwise there must be some predefined order for updating). To

this end, we use the doubly stochastic matrix, $\mathbf{T} \in \mathbb{R}^{p \cdot p}$ for deciding the communications among sensors, where $\mathbf{T}_{ij}$ is the probability that a sensor i contacts another sensor j at any iteration. In a mathematical form, we can have

$$\mathbf{T}_{ij} = \begin{cases} h(d_{ij}), & j \in \mathcal{N}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

where $d_{ij}, \{j \in S : j \in \mathcal{N}_i\}$ is the distance between sensor i and j. S is the set of sensors. $h(\cdot)$ is a function of d_{ij}. At iteration $k+1$, sensor i may need to ask only one of the neighbours j to send its estimate to i, unless i's stopping criteria are already satisfied, whereas sensor j can be contacted by more than one sensor, even when both of j's stopping criteria are satisfied.

Observation: In seismic tomography, each station has the ray information coming from the events to the station. Thus, the closer the two stations are, the higher the similarity between their obtained ray information is. Based on this observation, we can design the probability $\mathbf{T}_{ij}$ in the following way: at each iteration, for sensor i, the probability of selecting its neighbour sensor j is proportional to their distance d_{ij}. Since we give more weight to the neighbour that contributes more new information to sensor i, our proposed scheme is expected to converge faster than the method selecting its neighbour with uniform distribution (since the unequal weighting might help the sensors reach consensus).

The communication scheme for sensor i can be described as follows. When the conditions in (14) are not satisfied, sensor i selects a neighbor j according to the probability distribution of $\mathbf{T}_{ij}$. Upon contacting j, sensor i pulls the current value of $\mathbf{s}_j$ from j. During the computation, first update dual variable $\mathbf{u}_i$ by replacing $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ with $|\mathcal{N}_i| \mathbf{s}_j$. Second, for updating primal variable $\mathbf{s}_i$, in the right side of (A7), the item $\sum_{j \in \mathcal{N}_i} \mathbf{s}_j$ is replaced by $|\mathcal{N}_i| \mathbf{s}_j$. After updating the primal variable $\mathbf{s}_i$, sensor i will again do the similar randomised communication process for receiving new $\mathbf{s}_j$. Sensor i repeats the above steps until conditions in (14) are already satisfied. The algorithm with this proposed communication scheme is illustrated in Algorithm 2.

Note that the set of neighbours $\mathcal{N}_i$ of sensor i in fact contains the sensors within the communication range of node i. That is, if all the sensors except sensor i in the network are within the communication range of sensor i, then they are all considered as neighbours of sensor i.

5. Convergence analysis of sDSTA and aDSTA

Convergence rate is a critical factor in designing our proposed methods since faster convergence means less communication rounds and more bandwidth saving. In this section, we exploit Markov chain and spectral theory to derive the convergence speed of sDSTA and aDSTA. We find that both methods can achieve linear convergence rate in theory.

THEOREM 1 *The proposed sDSTA and aDSTA methods converge to an optimal solution at a linear rate* $O(\alpha^k)$ *for some* $\alpha < 1$*, in terms of the per node communications* k.

In the following section, we will discuss in detail the proof of Theorem 1 in the convergence analysis. Our analysis tool is similar to that in [16] [III.A] where it considers the problem of average consensus, which is different from our seismic tomography problem defined in (5). The attack plan is to construct a linear system capturing the dynamics of the solution and utilise the properties of Markov chain (described in Lemmas 1 and 2) to obtain the convergence rate.[21]

5.1. Convergence rate analysis

In order to obtain the convergence rate of proposed aDSTA framework, The first step is to describe the linear systems described in aDSTA (we will see that sDSTA is a special case under this framework). To this end, we first focus on the state update for sensor i.

Recall that in aDSTA, the original updating equation for $\mathbf{s}_i^{k+1}$ in (A7) is modified by replacing $\sum_{j\in\mathcal{N}_i}\mathbf{s}_j$ with $|\mathcal{N}_i|\cdot\mathbf{s}_j$, for some random neighbour j. According to the distribution in (15), the expected value of $|\mathcal{N}_i|\cdot\mathbf{s}_j$ is:

$$|\mathcal{N}_i|\cdot\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j \tag{16}$$

At this point, we can rewrite the updating of $\mathbf{s}_i^{k+1}$ in aDSTA as follows.

$$\begin{aligned}&\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)\mathbf{s}_i^{k+1}-\rho'|\mathcal{N}_i|\mathbf{s}_i^k\\&\quad-\rho'|\mathcal{N}_i|\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^k\Big\}+\mathbf{u}_i^k-\mathbf{A}_i^T\mathbf{t}_i=\mathbf{0}\end{aligned} \tag{17}$$

The updating rule for $\mathbf{s}_i^k$ can be obtained accordingly:

$$\begin{aligned}&\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)\mathbf{s}_i^{k}-\rho'|\mathcal{N}_i|\mathbf{s}_i^{k-1}\\&\quad-\rho'|\mathcal{N}_i|\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^{k-1}\Big\}+\mathbf{u}_i^{k-1}-\mathbf{A}_i^T\mathbf{t}_i=\mathbf{0}\end{aligned} \tag{18}$$

Also, (13b) can be rewritten as:

$$\mathbf{u}_i^{k+1}=\mathbf{u}_i^k+\rho'|\mathcal{N}_i|\mathbf{s}_i^{k+1}-\rho'|\mathcal{N}_i\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^{k+1}\Big\} \tag{19}$$

Combining the previous three Equations (17)–(19) yields:

$$\begin{aligned}&\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)\mathbf{s}_i^{k+1}\\&=\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)\mathbf{s}_i^{k}+2\rho'|\mathcal{N}_i|\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^{k}\Big\}\\&\quad-\rho'|\mathcal{N}_i|\mathbf{s}_i^{k-1}-\rho'|\mathcal{N}_i|\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^{k-1}\Big\}\end{aligned} \tag{20}$$

In a clearer form, the state update for sensor i can be expressed as:

$$\begin{aligned}\mathbf{s}_i^{k+1}&=\mathbf{s}_i^k+2\rho'|\mathcal{N}_i|\cdot\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}\\&\quad\cdot\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^k\Big\}-\rho'|\mathcal{N}_i|\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}\mathbf{s}_i^{k-1}\\&\quad-\rho'|\mathcal{N}_i|\cdot\Big(\mathbf{A}_i^T\mathbf{A}_i+(2\lambda_i^2+2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}\Big\{\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{s}_j^{k-1}\Big\}\end{aligned} \tag{21}$$

Now the matrix form for the linear system in aDSTA is:

$$\mathbb{E}\left[\mathbf{v}^{k+1}\right] = \mathbb{E}[\mathbf{M}] \cdot \mathbf{v}^k \tag{22}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{23}$$

where $\mathbf{F}$, $\mathbf{G}$, $\mathbf{I}$ and $\mathbf{0}$ are all $p \times p$ dimensional block matrices, respectively. $\mathbf{v}^{k+1} = \left[\bar{\mathbf{s}}^{k+1}, \bar{\mathbf{s}}^{k}\right]^T$, where $\bar{\mathbf{s}}^k$ stacks all the sensor solutions at iteration k. For notation convenience, we drop $\mathbb{E}[\,\cdot\,]$ in (24) in the later context.

Regarding matrix $\mathbf{F}$, the entry $\mathbf{F}_{ii} = \mathbf{I}$ and entry (i, j) is: $\mathbf{F}_{ij} = 2\mathbf{T}_{ij}\mathbf{P}_i$. With respect to matrix $\mathbf{G}$, we have $\mathbf{G}_{ii} = -\mathbf{P}_i$, $\mathbf{G}_{ij} = -\mathbf{T}_{ij}\mathbf{P}_i$, where $\mathbf{P}_i = \rho'|\mathcal{N}_i|\Big(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}$.

The state update (24) is valid for $k > 0$. Notice that in aDSTA we initialise $\mathbf{s}_i^0 = \mathbf{0}, \mathbf{u}_i^0 = \mathbf{0}, \forall i$. By plugging in these initial conditions into (17), it is shown that $\mathbf{s}_i^1 = \Big(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}\mathbf{A}_i^T\mathbf{t}_i$.

Thus the initial state vector $\mathbf{v}^1$ can be expressed as:

$$\mathbf{v}^1 = \Big[\mathbf{R}_1, \cdots, \mathbf{R}_p, \mathbf{0}, \cdots, \mathbf{0}\Big]^T \tag{24}$$

where $\mathbf{R}_i = \Big(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)^{-1}\mathbf{A}_i^T\mathbf{t}_i$.

Before approaching our final claim, we state two properties of matrix $\mathbf{M}$, which will help derive the convergence rate of proposed algorithms.

LEMMA 1 *Matrix* $\mathbf{M}$ *has an eigenvalue equals* 1.

Proof See details in Appendix 2. □

LEMMA 2 *There exists a selection of value* ρ' *and matrix* $\mathbf{T}$ *such that the second largest eigenvalue of matrix* $\mathbf{M}$ *is smaller than* 1.

Proof See details in Appendix 3. □

Since the state transition matrix $\mathbf{M}$ has the two above properties, based on the classical convergence analysis in [21], the convergence rate can be obtained as:

$$\|\mathbf{v}^{k+1} - \mathbf{v}^{\infty}\|_2 \leqslant C\sigma_2^{k-J+1} \tag{25}$$

where C is a constant, J is the size of the largest Jordan block of $\mathbf{M}$. σ_2 denotes the second largest eigenvalue of matrix $\mathbf{M}$. Theorem 1 is thus verified.

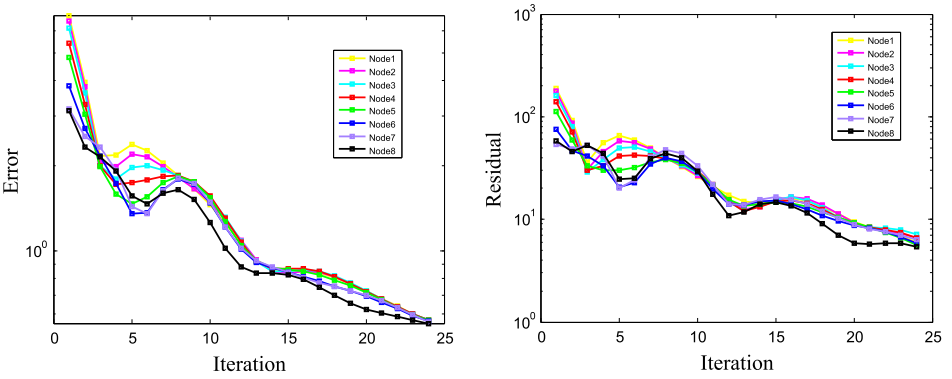
Observation: Recall that in (16), if we assign the probability $\mathbf{T}_{ij}$ in a uniform distributed manner such that:

$$\mathbf{T}_{ij} = \frac{1}{|\mathcal{N}_i|} \tag{26}$$

Then we have,

$$|\mathcal{N}_i| \cdot \sum_{j \in \mathcal{N}_i} \mathbf{T}_{ij}\mathbf{s}_j = \sum_{j \in \mathcal{N}_i} \mathbf{s}_j \tag{27}$$

This implies that in this case, the convergence rate of aDSTA is the same as sDSTA in expectation.



(a) Error (b) Residual

Figure 3. Convergence behaviour of sDSTA.

Considering the convergence rate in (25), we see that the second largest eigenvalue of matrix $\mathbf{M}$ is an important factor, which depends on the value of ρ' and the selection of matrix $\mathbf{T}$. In fact, besides our $\mathbf{T}$ matrix selection rule in III-B (given a certain feasible value ρ'), another strategy is to minimise σ_2 by solving the following optimisation problem (akin to the fastest mixing Markov chain problem [22]):

$$\begin{aligned} &\underset{t,\mathbf{T}}{\text{minimise}} \quad t \\ &\text{subject to} -t\mathbf{I} \preceq \mathbf{M}(\mathbf{T}) - \mathbf{M}^{\infty} \preceq t\mathbf{I} \end{aligned} \tag{28}$$

Here, $\mathbf{M}(\mathbf{T})$ indicates the transition matrix $\mathbf{M}$ is a function of $\mathbf{T}$. $\mathbf{M}^{\infty}$ is a limiting point of $\mathbf{M}$. The symbol $\preceq$ in $\mathbf{A} \preceq \mathbf{B}$ means matrix $\mathbf{B} - \mathbf{A}$ is positive semidefinite.

6. Performance evaluation

In this section, we evaluate the performance and characteristics of sDSTA and aDSTA, respectively. Our experiments are performed through MATLAB and network emulators.[23] In the performance study, both synthetic and real seismic imaging data-sets are tested.

6.1. Synthetic data (2-D model)

The performance analysis here is based on the data-set generated using code in [24]. We create a 2-D seismic tomography test problem with an $N \cdot N$ domain, using n sources located on the right boundary and p receivers (seismographs) scattered along the left and top boundary. The rays are transmitted from each source to each receiver. We know that the regularisation parameter λ depends on the application and data-sets, which means for specific scenario, λ is chosen based on the understanding and knowledge of the application. For simplicity, we fix $\lambda^2 = 1$ and $\lambda_i^2 = 1/p$ in our experiments. The parameter ρ' is set to be 0.5.

6.1.1. Convergence behaviour of sDSTA

We first study the performance of sDSTA with a case that $\{N = 16$ (tomography resolution is 16×16), $n = 64$, $p = 32\}$. The associated communicate network is a complete network

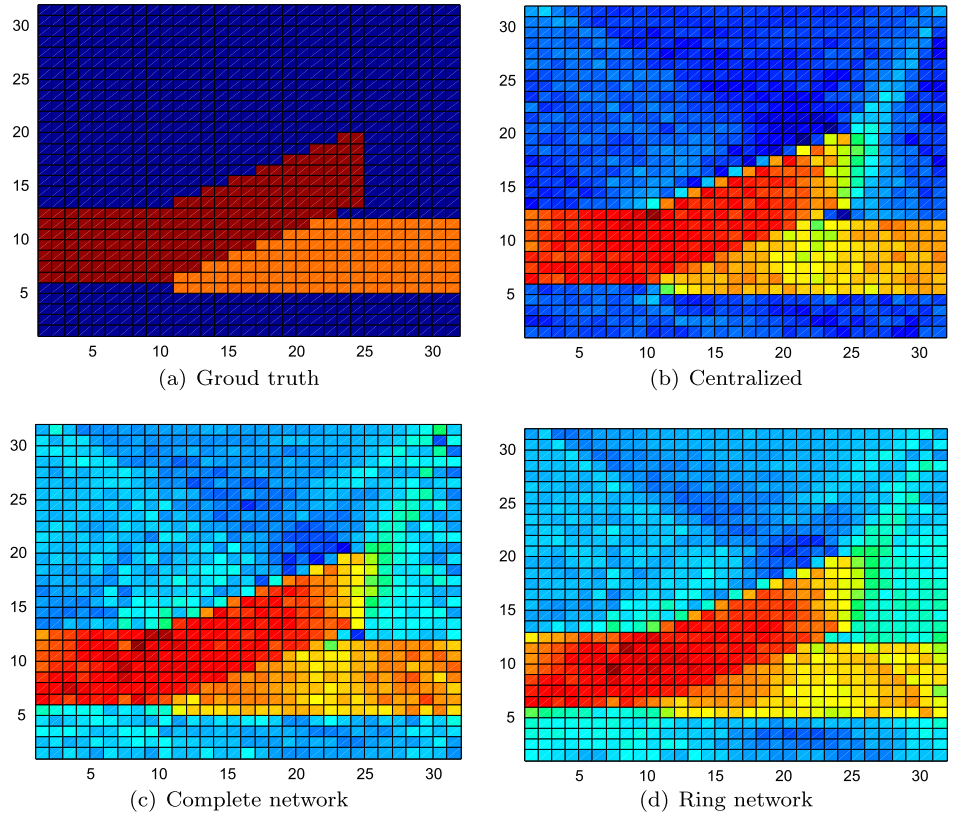


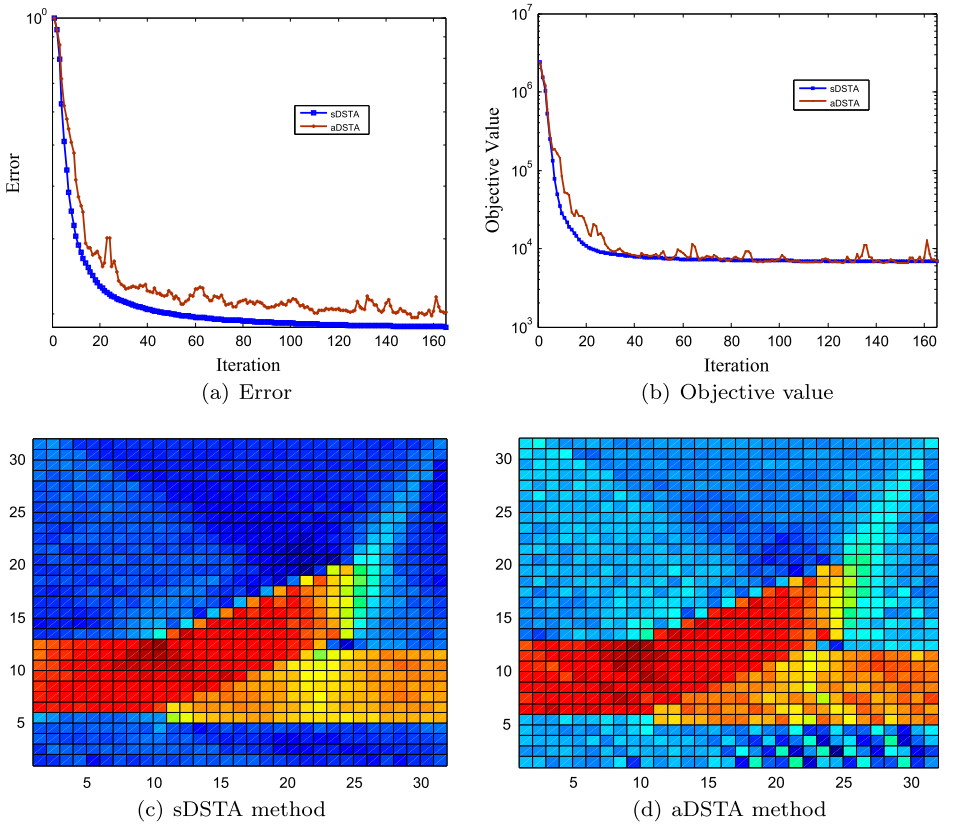
Figure 4. Seismic Tomography Comparison (noisy data). Centralised solution in (b) is obtained by running 50 iterations of BART. (c) is the tomography in Node 1 after 50 message communications assumed in a complete network. (d) is Node 1's tomography after 100 message communications assumed in a ring network.

(every node can communicate with all the other nodes) and there is no noise in the data. Note that in this scenario, the size of matrix $\mathbf{A}$ is 2048×256 and the size of each submatrix $\mathbf{A}_i$ is 64×256.

In Figure 3, we plot the error (measured by $\|\mathbf{s} - \mathbf{s}_t\|_2$, $\mathbf{s}_t$ refers to the ground truth) and residual (measured by $\|\mathbf{As} - \mathbf{t}\|_2$) for sensor 1 - sensor 8 (first eight nodes) in the system. It shows that these 8 nodes finally reach consensus after around 25 iterations. It is worth noting that one iteration here means one message communication for every node. For the seismic tomography application investigated in this paper, our key concern is to find a communication efficient algorithm since in this problem communication for each node is much more expensive than computation within each node.

6.1.2. Tomography results of sDSTA

Now we consider a larger data size model with $\{N = 32$ (tomography resolution is $32 \cdot 32)$, $n = 128$, $p = 64\}$. A 5% Gaussian noise is added into data vector $\mathbf{t}$. The tomography results are depicted in Figure 4. Since noise is included, both centralised and sDSTA results are less accurate comparing to the noiseless case. Nevertheless, the outline of the fault zone (the brown part in (a)) is almost recovered. Another interesting point is that when



(a) Error (b) Objective value

(c) sDSTA method (d) aDSTA method

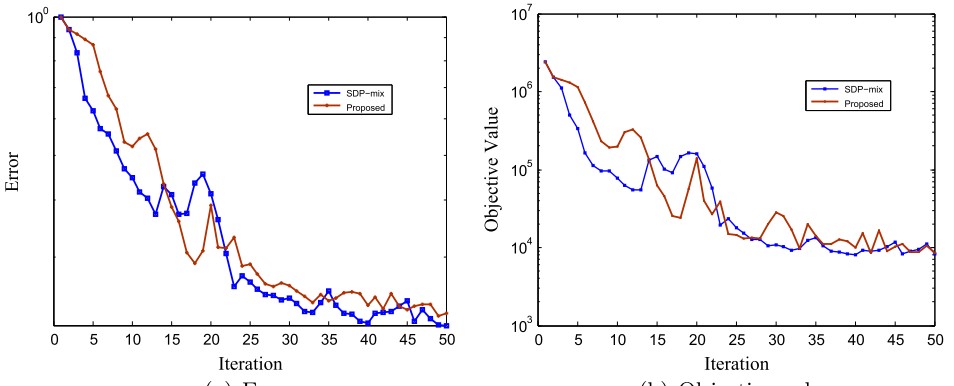
Figure 5. sDSTA vs. aDSTA. (a) is the relative error plot. (b) shows the objective value curves. (c) and (d) are tomography results.

connectivity ratio of the communication network is low (as the ring network in (d)), more communications among the neighbours are required in order to achieve similar results as the high connectivity ratio ones (e.g. (c)). This validates our claim in the last paragraph of Section 3.3.

6.1.3. Performance comparison: sDSTA vs. aDSTA

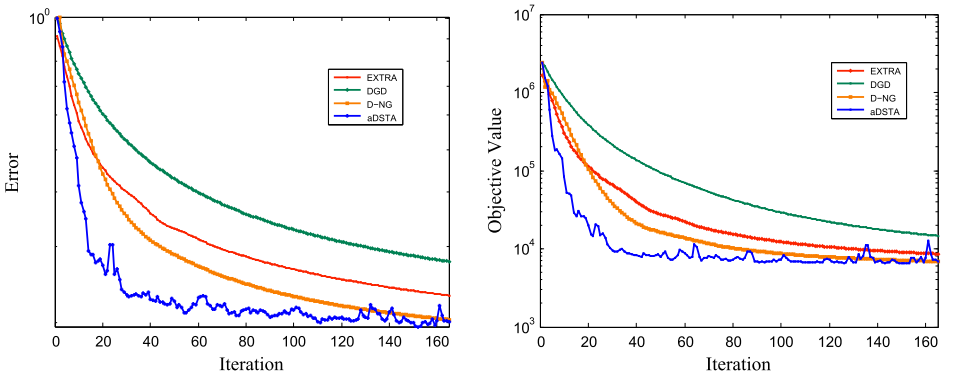
We study a case with $\{N = 32, n = 256, p = 128\}$. Noise level is the same as the previous example. A random communication network is created, and on average each sensor node has only three neighbours. This setting is to emulate the situation in real that the sensor network is widely spread in the field to cover a large area and for each sensor node, there are very few number of nodes are within its communication range.

In Figure 5, we compare the performance of two proposed algorithms sDSTA and aDSTA. In particular, though the tomography is a bit worse, the convergence speed of aDSTA is shown to be close to sDSTA (in (a) and (b)). This observation conforms to the theoretical analysis in Section 4.2. In fact, if the application has a base station or a coordinator, and is not very real-time sensitive, synchronised-based sDSTA will meet the requirement. In other situations, aDSTA might be more suitable.



(a) Error (b) Objective value

Figure 6. Performance of aDSTA with different strategies of choosing probability matrix **T**.



(a) Error (b) Objective value

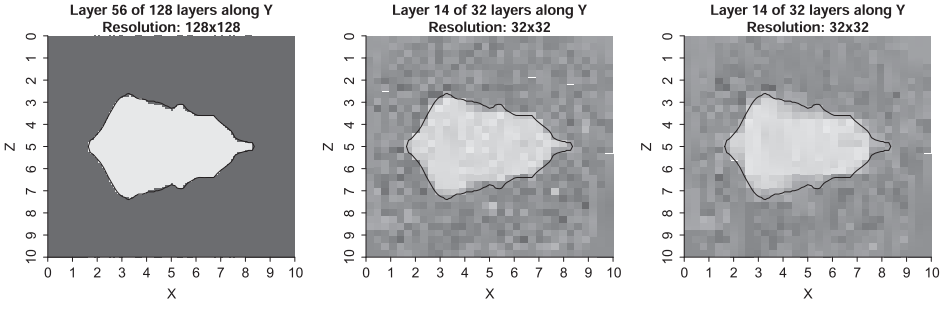
Figure 7. Convergence performance comparison: aDSTA, DGD, EXTRA, D-NG.

Remark 1 We conclude that aDSTA can provide comparable tomography solution with sDSTA and has less communication and coordination.

In the later context, we will focus on the performance evaluations of aDSTA.

6.1.4. Impact of probability matrix **T** *on aDSTA*

It is also interesting to study the effect of using different selection rules for probability matrix **T**. In Figure 6, 'Proposed mix' refers to the rule suggested in Section 4.2 (Equation (15)) and 'SDP-based mix' is the method described in (28). For each method, we run five realisations of aDSTA and average the solutions. We find that 'SDP-based mix' is slightly better than 'Proposed mix' in terms of convergence rate. Nevertheless, the distinction between them is not that obvious. In addition, solving (28) might need optimisation solver installed in the nodes. In short, if solving (28) is not available or wanted in the application, the simple 'Proposed mix' can be a good alternative.



(a) Ground truth (b) Centralized (c) aDSTA

Figure 8. Vertical slices of tomography model. The ground truth in (a) is generated by simulating seismic data on resolution 128^3 and 650 events are used. Centralised and aDSTA methods are simulated with resolution 32^3 and 400 events.

6.1.5. Compare aDSTA with other methods in the literature

We compare aDSTA with three recently developed general distributed optimisation algorithms: Decentralised Gradient Descent (DGD) [11], EXTRA [12] and D-NG method [13]. All of these three methods are tested using hand-tuned optimal parameters, respectively. Figure 7 demonstrates that aDSTA (blue curve) significantly outperforms the other methods. Recall that in Figure 5(a) and (b), sDSTA is very close to aDSTA, which insinuates that sDSTA is also faster than DGD, EXTRA and D-NG. This comparison conveys a message that the proposed sDSTA and aDSTA algorithms might be more suitable for our communication-sensitive distributed seismic tomography.

6.2. Synthetic data (3-D model)

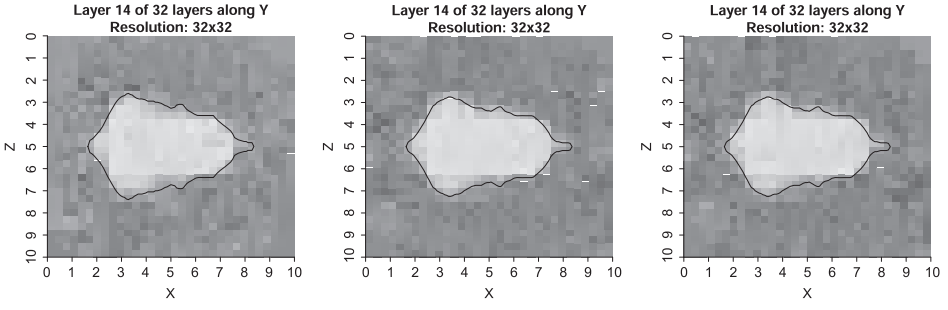
We follow the routine in [2] to generate a similar synthetic 3-D model data-set and use the same code in [2] for visualising our solutions in Figures 8 and 9. The synthetic model consists of a magma chamber (low velocity area) in a 10 km^3 cube. One hundred stations are randomly distributed on top of the cube and form a network. To construct the matrix $\mathbf{A}$ and vector $\mathbf{b}$, 650 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.

6.2.1. Comparison of tomography results

Figure 8 demonstrates that aDSTA is close to centralised method in tomography quality. Another interesting point is that aDSTA almost reconstructs the surface of magma even in a much lower resolution comparing to the ground truth.

6.2.2. Data loss tolerance of aDSTA

We evaluate the robustness of aDSTA in Figure 9. Two packet loss ratios 10% and 30% are tested in the emulator. We can find that even for 30% packet loss ratio, the distinction between the result without packet loss is relatively small. This validates the robustness of aDSTA in dealing with severe packet loss situations.



(a) No packet loss (b) 10% packet loss (c) 30% packet loss

Figure 9. Effect of packet loss in aDSTA.

6.3. Real data (3-D model)

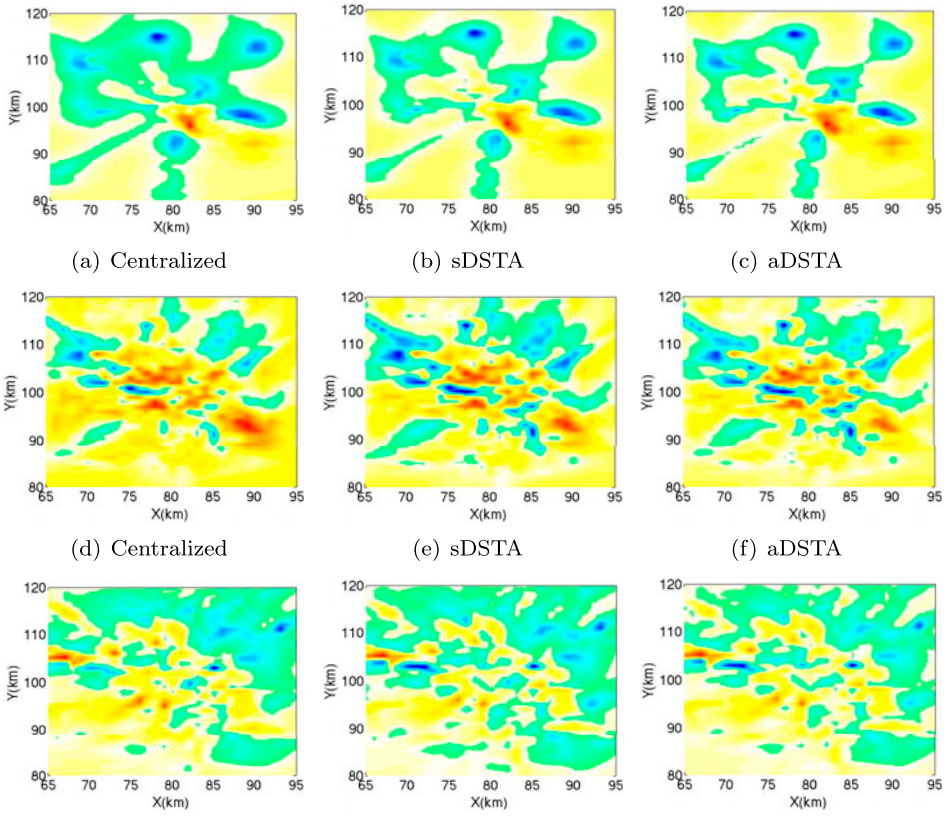
To study the performance of the two proposed algorithms in realistic scenarios, we use ten years (2001–2011) real seismic event data of Mount St. Helens in Washington, USA for the experiment. The data were collected from 78 stations and a 3-D velocity model is used, which assumes the velocity in the volcano changes along x-axis, y-axis and depth. Notice that unlike synthetic data used in previous section, there is no ground truth in this real data scenario. In other words, the true velocity structure of volcano Mount St. Helens is currently unknown. Hence we focus on the comparison of the proposed methods with centralised processing scheme, which can be seen as a benchmark that *fully* utilise the data available.

6.3.1. Comparison of tomography results in real data

Figure 10 illustrates vertical slices of tomography model with various depths. The range of x-axis is from 65 to 95 km, and the range of y-axis is from 80 to 120 km. The underlying resolution is $160 \times 200 \times 24$. The color in Figure 10 represents the relative velocity perturbation in specific location. More red means larger (negative) value of perturbation. More blue means larger (positive) value of perturbation. It is shown in Figure 10 that both sDSTA and aDSTA (at the solution of 300 iterations) can effectively invert the tomography model close to the benchmark (centralised algorithm) using real data.

6.3.2. Communication cost evaluation in real data

Figure 11 visualises the communication cost characteristics of aDSTA. Figure 11(a) shows that sDSTA and aDSTA need much less amount of communication than the centralised method and aDSTA is more efficient than sDSTA. Figure 11(b) illustrates the communication distribution of aDSTA on each node with a heat map. We can see that, unlike synchronised sDSTA, in aDSTA some nodes stop communication earlier than others, which saves the bandwidth. Second, the cost disparities between several high cost nodes and the lowest one are around 18%, which implies that the communication load is still balanced over the network.

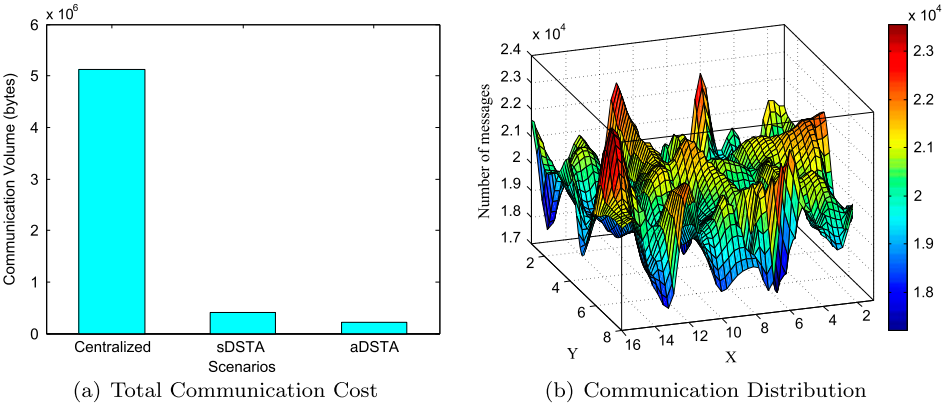


(a) Centralized (b) sDSTA (c) aDSTA

(d) Centralized (e) sDSTA (f) aDSTA

(g) Centralized (h) sDSTA (i) aDSTA

Figure 10. Real data tomography inversion results comparison. (a)–(c) are results of layer depth 0.9 km. (d)–(f) are results of layer depth 2.9 km. (g)–(i) are results of layer depth 4.9 km.



(a) Total Communication Cost (b) Communication Distribution

Figure 11. Communication cost comparison.

7. Discussion

There are two potential benefits of using our proposed algorithms for seismic tomographic inversion problem. First, both sDSTA and aDSTA can be easily modified to accommodate problems with constraints on solution **s**. For example, we can add constraint that the value of velocity perturbation is within some range (need knowledge from geophysics community). The solution in this case might be better in revealing the real situation within the volcano. Second, they can deal with different regularisations (as long as it is convex). For instance, an efficient distributed algorithm can still be derived for the ℓ_1-norm regularised inversion problem in [25].

8. Conclusion

This work opens a relatively underexploited area where seismic tomography inversion is done over a sensor network in a distributed, decentralised fashion. This paper presents novel designs for in-network travel time seismic imaging in cyber-physical sensor systems under synchronous and asynchronous communication scenarios. The performance and features are verified with experiments in both synthetic and real data-sets through network emulators. The merits of the proposed methods elucidate that they are promising solutions for real-time in situ seismic tomography in the near future.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Our research is partially supported by NSF-1066391, NSF-1442630 and NSF-1125165.

References

[1] Lees JM. Seismic tomography of magmatic systems. J. Volcanol. Geoth. Res. 2007;167:37–56.
[2] Shi L, Song WZ, Xu M, Xiao Q, Lees J, Xing G. Imaging seismic tomography in sensor network. In: 2013 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON); 2013 Jun; New Orleans, LA. p. 327–335.
[3] Bording RP, Gersztenkorn A, Lines LR, Scales JA, Treitel S. Applications of seismic travel-time tomography. Geophys. J. R. Astron. Soc. 1987;90:285–303. doi:10.1111/j.1365-246X.1987.tb00728.x.
[4] Song W. Real-time *in-situ* seismic imaging: overview and case study (submitted to Society of Exploration Geophysicists International Exposition and 85th Annual Meeting; 2015).
[5] Lees JM, Crosson RS. Bayesian Art versus conjugate gradient methods in tomographic seismic imaging: an application at Mount St. Helens. In: Possolo A, editor. Spatial statistics and imaging. Vol. 20, Lecture Notes-Monograph Series. Hayward (CA): Institute of Mathematical Statistics; 1991. p. 186–208.
[6] Renaut RA. A parallel multisplitting solution of the least squares problem. Numer. Linear Algebra Appl. 1998;5:11–31.
[7] Sayed AH, Lopes CG. Distributed recursive least-squares strategies over adaptive networks. In: Fortieth Asilomar Conference on Signals, Systems and Computers, 2006. ACSSC'06; 2006 Nov; Pacific Grove, CA. p. 233–237.
[8] Schizas ID, Mateos G, Giannakis GB. Distributed LMS for consensus-based in-network adaptive processing. IEEE Trans. Signal Process. 2009;57:2365–2382.

[9] Mateos G, Schizas ID, Giannakis GB. Performance analysis of the consensus-based distributed LMS algorithm. EURASIP J. Adv. Signal Process. 2009;2009:68.
[10] Kamath G, Shi L, Song WZ. Component-average based distributed seismic tomography in sensor networks. In: 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS); 2013 May; Cambridge, MA. p. 88–95.
[11] Yuan K, Ling Q, Yin W. On the convergence of decentralized gradient descent; 2013. arXiv:1310.7063.
[12] Shi W, Ling Q, Wu G, Yin W. Extra: an exact first-order algorithm for decentralized consensus optimization; 2014. arXiv:1404.6264.
[13] Jakovetic D, Xavier J, Moura JMF. Fast distributed gradient methods; 2014. arXiv:1112.2972v4.
[14] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 2011;3:1–122. doi:10.1561/2200000016.
[15] Bertsekas DP, Tsitsiklis JN. Parallel and distributed computation: numerical methods. Upper Saddle River (NJ): Prentice-Hall; 1989.
[16] Erseghe T, Zennaro D, Dall'Anese E, Vangelista L. Fast consensus by the alternating direction multipliers method. IEEE Trans. Signal Process. 2011;59:5523–5537.
[17] Mateos G, Bazerque J, Giannakis G. Distributed sparse linear regression. IEEE Trans. Signal Process. 2010;58:5262–5276.
[18] Lees JM, Crosson RS. Spatial statistics and imaging. In: Possolo A, editor. Spatial statistics and imaging. Vol. 20, Lecture Notes–Monograph Series. Hayward (CA): Institute of Mathematical Statistics; 1991. p. 186–208. doi:10.1214/lnms/1215460502.
[19] Eckstein J, Bertsekas D. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math Program. 1992;55:293–318. doi:10.1007/BF01581204.
[20] Tsianos K, Lawlor S, Yu JY, Rabbat M. Networked optimization with adaptive communication. In: 2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP); 2013 Dec; Austin, TX. p. 579–582.
[21] Rosenthal JS. Convergence rates of Markov chains. SIAM Rev. 1995;37:387–405.
[22] Boyd S, Diaconis P, Xiao L. Fastest mixing markov chain on a graph. SIAM Rev. 2004;46:667–689. doi:10.1137/S0036144503423264.
[23] Ahrenholz J. Comparison of CORE network emulation platforms. In: Military Communications Conference, 2010 – MILCOM 2010; 2010 Oct; San Jose, CA. p. 166–171.
[24] Hansen PC, Saxild-Hansen M. Air tools – a MATLAB package of algebraic iterative reconstruction methods. J. Comput. Appl. Math. 2012;236:2167–2178. Available from: http://www.sciencedirect.com/science/article/pii/S0377042711005188, Inverse Problems: Computation and Applications.
[25] Loris I, Nolet G, Daubechies I, Dahlen FA. Tomographic inversion using l1-norm regularization of wavelet coefficients. Geophys. J. Int. 2007;170:359–370.
[26] Xiao L, Boyd S. Fast linear iterations for distributed averaging. Syst. Control Lett. 2003;53:65–78.

Appendix 1. Derivation of (13a)–(13b)

The goal here is to reduce the original ADMM steps (12a)–(12d) to (13a)–(13b). To this end, we first focus on (12b). A closer look reveals that a closed-form solution of (12b) can be obtained as follows:

$$\mathbf{q}_{ij}^{k+1} = \frac{1}{2\rho}\left\{\rho(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}) + (\mathbf{y}_{ij}^{k} + \mathbf{z}_{ij}^{k})\right\} \tag{A1}$$

Adding both sides of (12c) and (12d) yields:

$$\mathbf{y}_{ij}^{k+1} + \mathbf{z}_{ij}^{k+1} = (\mathbf{y}_{ij}^{k} + \mathbf{z}_{ij}^{k}) + \rho(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}) - 2\rho \cdot \mathbf{q}_{ij}^{k+1} \tag{A2}$$

After plugging (A1) into the right side of (A2), we obtain the following result:

$$\mathbf{y}_{ij}^{k+1} + \mathbf{z}_{ij}^{k+1} = \mathbf{0} \tag{A3}$$

Consequently, by plugging (A3) back into (A1), we have

$$\mathbf{q}_{ij}^{k+1} = \frac{1}{2}\left(\mathbf{s}_i^{k+1} + \mathbf{s}_j^{k+1}\right) \tag{A4}$$

We now re-express (12a) removing all the independent items with respect to variable $\mathbf{s}_i$:

$$\begin{aligned}\mathbf{s}_i^{k+1} &= \arg\min_{\{\mathbf{s}_i\}} \left\{ c_i(\mathbf{s}_i) + \sum_{j\in\mathcal{N}_i} (\mathbf{y}_{ij}^k)^T \mathbf{s}_i + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \|\mathbf{s}_i - \mathbf{q}_{ij}^k\|_2^2 \right\} \\ &= \arg\min_{\{\mathbf{s}_i\}} \left\{ c_i(\mathbf{s}_i) + \sum_{j\in\mathcal{N}_i} (\mathbf{y}_{ij}^k)^T \mathbf{s}_i + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \|\mathbf{s}_i - \frac{1}{2}(\mathbf{s}_i^k + \mathbf{s}_j^k)\|_2^2 \right\}\end{aligned} \tag{A5}$$

Let $\mathbf{u}_i^k = \sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)$. Since function c_i is differentiable with respect to $\mathbf{s}_i$, (A5) is equivalent to solving the following equation:

$$\nabla c_i(\mathbf{s}_i^{k+1}) + \mathbf{u}_i^k + 2\rho|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \rho'\Big(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j\in\mathcal{N}_i} \mathbf{s}_j^k\Big) = 0 \tag{A6}$$

where ∇ denotes the gradient of the function. Plugging $c_i(\mathbf{s}_i^{k+1}) = \frac{1}{2}\|\mathbf{A}_i\mathbf{s}_i^{k+1} - \mathbf{t}_i\|_2^2 + \lambda_i^2\|\mathbf{s}_i^{k+1}\|_2^2$ into (A6) yields the following equation, which becomes (13a) immediately.

$$\Big(\mathbf{A}_i^T\mathbf{A}_i + (2\lambda_i^2 + 2\rho'|\mathcal{N}_i|)\mathbf{I}\Big)\mathbf{s}_i^{k+1} = \mathbf{A}_i^T\mathbf{t}_i - \mathbf{u}_i^k + \rho'\Big(|\mathcal{N}_i|\mathbf{s}_i^k + \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^k\Big) \tag{A7}$$

Now we need to derive the updating equation for $\mathbf{u}_i$. Substituting (A4) into (12c), we can have (consider $\mathbf{u}_i^{k+1}$):

$$\begin{aligned}\mathbf{u}_i^{k+1} &= \sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^{k+1}) \\ &= \left\{\sum_{j\in\mathcal{N}_i}(\mathbf{y}_{ij}^k)\right\} + \frac{\rho}{2}\Big(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^{k+1}\Big) \\ &= \mathbf{u}_i^k + \frac{\rho}{2}\Big(|\mathcal{N}_i|\mathbf{s}_i^{k+1} - \sum_{j\in\mathcal{N}_i}\mathbf{s}_j^{k+1}\Big)\end{aligned} \tag{A8}$$

Define $\rho' = \rho/2$, then (13b) follows.

Appendix 2. Proof of Lemma 1

First, in the second row of matrix $\mathbf{M}$, the summation is $\mathbf{I}$, thus in the lower part, the summation of every row is 1. Now we look at the upper part, considering the facts in about matrices $\mathbf{F}$ and $\mathbf{G}$, the summation of ith row can be expressed as:

$$\begin{aligned}&\mathbf{F}_{ii} + \mathbf{G}_{ii} + \sum_{j\in\mathcal{N}_i}\mathbf{F}_{ij} + \sum_{j\in\mathcal{N}_i}\mathbf{G}_{ij} \\ =&\mathbf{I} - \mathbf{P}_i + \sum_{j\in\mathcal{N}_i} 2\mathbf{T}_{ij}\mathbf{P}_i - \sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\cdot\mathbf{P}_i \\ =&\mathbf{I} - \mathbf{P}_i + \sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij}\mathbf{P}_i\end{aligned} \tag{B1}$$

Recall that matrix $\mathbf{T}$ is defined as a doubly stochastic matrix yielding $\sum_{j\in\mathcal{N}_i}\mathbf{T}_{ij} = 1$. Hence the above equation equals $\mathbf{I}$. Consequently, both the upper and lower part of matrix $\mathbf{M}$ satisfies the

condition that the summation of each row is 1. Based on this fact, it is straightforward to show Lemma 1 (the corresponding eigenvector contains scaling factor for each element in a row).

Appendix 3. Proof of Lemma 2

Based on Lemma 1, we can construct the corresponding left and right eigenvectors denoted by l_v^1 and r_v^1, respectively (also scale the eigenvectors such that $l_v^1 r_v^1$=1). Next, we need to find a limiting point of the state transition matrix. From the definition, we can have $l_v^1 \mathbf{M} = l_v^1$, $\mathbf{M} r_v^1 = r_v^1$. Consequently,

$$\lim_{k \to \infty} \mathbf{M}^k = r_v^1 l_v^1$$

The above fact implies that $r_v^1 l_v^1 = \mathbf{M}^\infty$ is a limiting point.[21] Akin to the techniques used in [26], it can be shown that once the following condition is satisfied, all the other eigenvalues of matrix $\mathbf{M}$ is less than 1. Thus Lemma 2 is verified.

$$\|\mathbf{M} - \mathbf{M}^\infty\|_2 < 1 \tag{C1}$$

Here $\|\cdot\|_2$ is the spectral norm of a matrix.