A Lightweight Alternating Direction Method of Multipliers for Decentralized Event Detection

Qing Ling, Chun Shi, Anhong He
Department of Automation, University of Science and Technology of China, Hefei, Anhui, 230027
E-mail: qingling@mail.ustc.edu.cn, cshi@mail.ustc.edu.cn, heanhong@mail.ustc.edu.cn

Abstract: In this paper we address the problem of decentralized event detection in a large-scale wireless sensor network (WSN). Comparing with existing centralized solutions, decentralized algorithms are superior in energy efficiency and network scalability, and thus fit for the distributed nature of a WSN. We formulate the event detection problem as a linear program, and solve it with the alternating direction method of multipliers (ADMM). Under mild conditions, this iterative algorithm is shown to be fully decentralized. Further, in view of the fact that the communication burden per iteration directly decides the energy consumption of sensor nodes, we simplify the classic ADMM to a lightweight one, which requires much lower communication burden while keeps the global convergence of the classic ADMM. Effectiveness of the proposed algorithm is validated with simulation results.

Key Words: Wireless sensor networks, Decentralized event detection, Alternating direction method of multipliers

1 INTRODUCTION

Event detection is one of the most important applications of wireless sensor networks (WSNs). Typical event detection tasks include detecting nuclear radioactive sources [1], monitoring structural health conditions [2], discovering the presence of contaminants [3], etc. Generally speaking, these scenarios have the following common characteristics: i) multiple events may occur in the sensing area; ii) one event only influences sensory measurements of its neighboring region; and iii) measurement of one sensor node is the linear superposition of the influences of multiple events. The objective of event detection is to locate the events and estimate their amplitudes from the sensory measurements.

Different from the traditional centralized event detection approaches which process collected sensory measurements at a fusion center, this paper focuses on decentralized event detection algorithms in large-scale WSNs. By decentralized event detection we mean that, without any fusion center, sensor nodes autonomously exchange information with their neighbors and collaboratively detect the events. The main advantage of this decentralized scheme, over the centralized one, is that the distributed sensor nodes do not need to transmit data to a faraway fusion center in a large-scale WSN. Therefore, the communication burden is limited and the energy consumption is scalable with the network size.

In a nutshell, a decentralized event detection task can be formulated as an optimization problem and solved with decentralized optimization algorithms, which have attracted much attention for implementing large-scale WSNs [4, 5, 6]. Among these decentralized algorithms, the alternating direction method of multipliers (ADMM) [7] is a promising one due to its ability of efficiently handling a large class of separable constrained convex programs. Successful applications can be found in, for example, decentralized least-mean-square estimation [8], spectrum sensing [9], and structural health monitoring [10], etc.

This paper formulates the event detection task as a sparse signal recovery problem. Under some mild conditions it has the form of a separable linear program and can be solved with the classic ADMM. Considering that in the algorithm the communication burden per iteration directly decides the energy consumption of sensor nodes, we further simplify the classic ADMM to a lightweight one, which requires much lower amount of exchanged data while keeps the global convergence of the classic ADMM. Simulation results demonstrate the effectiveness of the proposed algorithm.

This work is supported in part by National Nature Science Foundation under Grant 61004137 and in part by the Fundamental Research Funds for the Central Universities.

2 PROBLEM FORMULATION

Let us consider a large-scale WSN in which sensor nodes are uniformly randomly deployed in a two-dimensional sensing area. The WSN has a set of L sensor nodes, denoted as $\mathcal{L}$. Sensor nodes have a common communication range r_C. Each sensor node can only communicate with its neighbors within the communication range. The network is supposed to be connected under the chosen communication range r_C. The event detection task is performed periodically. Within each sampling period, multiple events may occur in the sensing area. We make the following assumptions in this paper:

(A1): *Events only occur at some sensor node positions. When the position of one event coincides with the position of sensor node i, we denote the amplitude of the event by a nonnegative scalar c_i.*

(A2): *Each event only influences its neighboring region. The influence of a unit-amplitude event at position j on position i is $f_{ij} = f(d_{ij})$, where d_{ij} is the distance between sensor nodes i and j and f is a nonincreasing function. We suppose that $f_{ij} = 0$ if $d_{ij} \geq r_C$, $f_{ij} = 1$ if $d_{ij} = 0$ and $f_{ij} = f_{ji}$.*

(A3): *The measurement of one sensor node is the superposition of the influences of all events plus random noise. Since $f_{ij} = 0$ for $d_{ij} \geq r_C$, the measurement b_i of sensor node i can be written as $b_i = \sum_{j \in \mathcal{I}_i} f_{ij} c_j + e_i$, where e_i is the random noise, and $\mathcal{I}_i$ denotes the set of sensor node i along with its neighbors.*

2.1 Justification of the Assumptions

The assumption **(A1)** selects positions of the sensor nodes as candidate positions of the sensing area and confines the events on these candidate positions [11]. This way, the oth-

erwise highly nonlinear and nonconvex event detection problem turns to be tractable. The resolution of event detection is now directly decided by the density of sensor nodes. Based on **(A1)**, we can formulate the event detection problem as recovering the vector $\mathbf{c} = [c_1, ..., c_L]^T$ from the sensory measurements. If c_i is nonzero, then there is an event detected at the position of sensor node i.

The assumption **(A2)** describes the phenomenon of *limited influence* of a single event on the whole sensing area [12]. For example, in nuclear radioactive detection, the influence of a radioactive source varies polynomially with distance. Similar distance-dependent influence can be observed from events in many practical scenarios such as fire sources and structural damages. Therefore, by carefully setting the communication range r_C, it is possible to make $f_{ij} \simeq 0$ when $d_{ij} \geq r_C$. In the ensuring paper we will show that the *limited influence* phenomena greatly facilitates the implementation of energy efficient and scalable decentralized algorithms.

2.2 A Linear Program Model

We now introduce a key observation that the optimization vector $\mathbf{c}$ is sparse, namely, the number of nonzero elements in $\mathbf{c}$ is much smaller than the vector size L. This prior knowledge holds since the WSN is large-scale while events generally occur sparsely. Nevertheless, we do allow the influence of these sparse events to span over the large sensing area. Hence, recovery of $\mathbf{c}$ boils down to minimizing a sparsity-imposing metric $||\mathbf{c}||_1$ that is the ℓ_1 norm of $\mathbf{c}$ [13], subject to measurement constraints. Note that **(A1)** assumes $\mathbf{c} \geq 0$, thus $||\mathbf{c}||_1 = \sum_{i=1}^{L} |c_i| = \sum_{i=1}^{L} c_i$. Further, we consider the case that the measurement error is within $[-\theta, \theta]$ where θ is a known positive constant. From **(A3)**, a linear program formulation for the sparse signal recovery problem arises:

$$\begin{array}{ll} \min & \sum_{i=1}^{L} c_i, \\ s.t. & \sum_{i\in\mathcal{I}_j} f_{ji} c_i \geq b_j - \theta, \quad j = 1, ..., L, \\ & \sum_{i\in\mathcal{I}_j} f_{ji} c_i \leq b_j + \theta, \quad j = 1, ..., L, \\ & c_j \geq 0, \quad j = 1, ..., L. \end{array} \tag{1}$$

Written in a matrix form, (1) turns to:

$$\begin{array}{ll} \min & \mathbf{1}^T\mathbf{c}, \\ s.t. & \mathbf{F}\mathbf{c} \geq \mathbf{b} - \theta\mathbf{1}, \\ & \mathbf{F}\mathbf{c} \leq \mathbf{b} + \theta\mathbf{1}, \\ & \mathbf{c} \geq \mathbf{0}, \end{array} \tag{2}$$

where $\mathbf{1}$ and $\mathbf{0}$ are two $L \times 1$ vectors with all ones and zeros, $\mathbf{b} = [b_1, ..., b_L]^T$ is the $L \times 1$ measurement vector, and $\mathbf{F}$ is the $L \times L$ measurement matrix with its entry on i-th column and j-th row as f_{ji}. In the following paper we will focus on solving this linear program in a decentralized way.

2.3 Extensions of the Linear Program Model

The linear program model (2) can be extended to many other formulations according to practical settings.

First, for the case that the total energy of the measurement error is bounded with a known positive constant δ, the mathematical model can be:

$$\begin{array}{ll} \min & \mathbf{1}^T\mathbf{c}, \\ s.t. & ||\mathbf{F}\mathbf{c} - \mathbf{b}||_2^2 \leq \delta, \\ & \mathbf{c} \geq \mathbf{0}, \end{array} \tag{3}$$

which is a second-order cone program and called as the basis pursuit denoising (BPDN) [14]. The BPDN has two equivalent formulations, the least absolute shrinkage and selection operator (LASSO) [15] and the ℓ_1 regularized least squares (ℓ_1-LS) [16]. Intuitively, (2) handles uniform noise while (3) handles Gaussian noise; though both of them indeed exploit the signal sparsity. This paper focuses on (2) which takes the advantage of a separable linear program and can be solved by the ADMM efficiently.

It should be noted that there has already been a linear program formulation available for this kind of sparse signal recovery problem, known as the Dantzig selector (DS) [17]:

$$\begin{array}{ll} \min & \mathbf{1}^T\mathbf{c}, \\ s.t. & \mathbf{F}^T\mathbf{F}\mathbf{c} \geq \mathbf{F}^T\mathbf{b} - \epsilon\mathbf{1}, \\ & \mathbf{F}^T\mathbf{F}\mathbf{c} \leq \mathbf{F}^T\mathbf{b} + \epsilon\mathbf{1}, \\ & \mathbf{c} \geq \mathbf{0}, \end{array} \tag{4}$$

where ϵ is a known positive constant. Despite the similarity between (2) and (4), the latter cannot be solved easily with a decentralized algorithm.

Second, though **(A1)** confine the amplitudes of the events to be nonnegative, this assumption can be relaxed easily. For an arbitrary optimization vector $\mathbf{c}$, we can define $\mathbf{c} = \mathbf{c}^+ - \mathbf{c}^-$, $\mathbf{c}^+ \geq \mathbf{0}$ and $\mathbf{c}^- \geq \mathbf{0}$. This way,

$$\begin{array}{ll} \min & ||\mathbf{c}||_1, \\ s.t. & \mathbf{F}\mathbf{c} \geq \mathbf{b} - \theta\mathbf{1}, \\ & \mathbf{F}\mathbf{c} \leq \mathbf{b} + \theta\mathbf{1}, \end{array} \tag{5}$$

is equivalent to

$$\begin{array}{ll} \min & \mathbf{1}^T\mathbf{c}^+ + \mathbf{1}^T\mathbf{c}^-, \\ s.t. & \mathbf{F}\mathbf{c}^+ - \mathbf{F}\mathbf{c}^- \geq \mathbf{b} - \theta\mathbf{1}, \\ & \mathbf{F}\mathbf{c}^+ - \mathbf{F}\mathbf{c}^- \leq \mathbf{b} + \theta\mathbf{1}, \\ & \mathbf{c}^+ \geq \mathbf{0}, \mathbf{c}^- \geq \mathbf{0}, \end{array} \tag{6}$$

which has the same form as (2) [18].

Third, **(A1)** chooses the positions of sensor nodes as candidate positions for the events in the sensing area. An alternative way is to draw a grid in the sensing field and choose the K grid points as candidate positions [9, 19, 20]. Supposing that the number of events is much smaller than the number of grid points, a similar sparse signal recovery problem appears. Therein a $K \times 1$ sparse signal, say $\tilde{\mathbf{c}}$, represents the positions and magnitudes of the events. By reformulating it as a consensus optimization problem which is solvable by the ADMM, [9] considers the decentralized optimization of this formulation, where each sensor node optimizes the local copy of $\tilde{\mathbf{c}}$. This approach can handle the case that the events are with *global influences*; however, the side effect is that each sensor node needs to recover the whole sensing field, namely the whole optimization vector $\tilde{\mathbf{c}}$, and the resulting communication burden is high. Contrarily, in our proposed algorithms each sensor only needs to recover the corresponding element of $\mathbf{c}$ with much lower communication burden.

3 THE CLASSIC ADMM

The classic ADMM is able to solve a large class of separable constrained convex programs. In this paper we apply it in the separable linear program (2) and obtain a neat solution.

Consider a linear program [7] (see pp. 249–253):

$$\begin{array}{ll} \min & \mathbf{h}^T\mathbf{x}, \\ s.t. & \mathbf{A}\mathbf{x}=\mathbf{y}, \\ & x_i \in \mathcal{X}_i, \quad i=1,...,M, \end{array} \tag{7}$$

where $\mathbf{h}$ is an $M\times 1$ coefficient vector, $\mathbf{x}$ is an $M\times 1$ optimization vector, $\mathbf{A}$ is an $N\times M$ matrix, $\mathbf{y}$ is an $N\times 1$ vector, and $y_i \in \mathcal{X}_i$ denotes a polyhedral constraint such as $y_i \geq 0$. The entry on i-th column and j-th row of $\mathbf{A}$ is a_{ji}. Let $\mathcal{J}_j$ be the set $\{i|a_{ji}\neq 0\}, j=1,...,N$. Reformulate (7) as:

$$\begin{array}{ll} \min & \sum_{i=1}^M h_i x_i, \\ s.t. & a_{ji}x_i = z_{ji}, \quad j=1,...,N, i\in\mathcal{J}_j, \\ & \sum_{i\in\mathcal{J}_j} z_{ji} = y_i, \quad j=1,...,N, \\ & x_i\in\mathcal{X}_i, \quad i=1,...,M. \end{array} \tag{8}$$

For each $j=1,...,N$, we consider Lagrange multipliers p_{ji} for the equality constraints $a_{ji}x_i = z_{ji}, i\in\mathcal{J}_j$. According to the method of multipliers, at time t, p_{ji} is updated with:

$$\begin{array}{c} p_{ji}(t+1) = p_{ji}(t) + d\left(a_{ji}x_i(t+1) - z_{ji}(t+1)\right), \\ j=1,...,N, i\in\mathcal{J}_j, \end{array} \tag{9}$$

where the parameter d is a positive constant. The optimization variables $x_i(t+1)$ and the auxiliary variables $z_{ji}(t+1)$ minimize the augmented Lagrangian function:

$$\begin{array}{c} \sum_{i=1}^M h_i x_i + \sum_{j=1}^N \sum_{i\in\mathcal{J}_j} p_{ji}(t)(a_{ji}x_i - z_{ji}) \\ +\frac{d}{2}\sum_{j=1}^N \sum_{i\in\mathcal{J}_j}(a_{ji}x_i - z_{ji})^2 \end{array} \tag{10}$$

under the constraints $\sum_{i\in\mathcal{J}_j} z_{ji} = y_i, j=1,...,N$ and $x_i \in \mathcal{X}_i, i=1,...,M$.

In the ADMM, the augmented Lagrangian function in (10) is optimized in an alternating direction manner, first for the optimization variables x_i and second for the auxiliary variables z_{ji}. The closed-form solutions are:

$$\begin{array}{c} x_i(t+1) = \left[\frac{\sum_{j|i\in\mathcal{J}_j} da_{ji}z_{ji}(t) - \sum_{j|i\in\mathcal{J}_j} a_{ji}p_{ji}(t) - h_i}{\sum_{j|i\in\mathcal{J}_j} da_{ji}^2}\right]^{\mathcal{X}_i}, \\ i=1,...,M, \end{array} \tag{11}$$

and

$$\begin{array}{c} z_{ji}(t+1) = a_{ji}x_i(t+1) + \frac{1}{d}p_{ji}(t) + \frac{y_j}{|\mathcal{J}_j|} \\ -\sum_{k\in\mathcal{J}_j}\frac{a_{jk}x_k(t+1)}{|\mathcal{J}_j|} - \sum_{k\in\mathcal{J}_j}\frac{p_{jk}(t)}{d|\mathcal{J}_j|}, \\ j=1,...,N, i\in\mathcal{J}_j. \end{array} \tag{12}$$

Here $[\cdot]^{\mathcal{X}_i}$ denotes the mapping to the polyhedron $\mathcal{X}_i$.

The classic ADMM iteratively updates the optimization variables x_i, auxiliary variables z_{ji}, and Lagrange multipliers p_{ji} with (11), (12), and (9), respectively. It has been proved to converge to the globally optimal solution of (7) [7].

3.1 The Classic ADMM for Event Detection

To apply the classic ADMM in (2), we now rewrite (2) to its equivalent form:

$$\begin{array}{ll} \min & \mathbf{1}^T\mathbf{c}, \\ s.t. & \mathbf{F}\mathbf{c} - \mathbf{s}_1 = \mathbf{b} - \theta\mathbf{1}, \\ & \mathbf{F}\mathbf{c} + \mathbf{s}_2 = \mathbf{b} + \theta\mathbf{1}, \\ & \mathbf{c}\geq\mathbf{0}, \mathbf{s}_1\geq\mathbf{0}, \mathbf{s}_2\geq\mathbf{0}, \end{array} \tag{13}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are two $L\times 1$ slack vectors. Comparing (7) and (17), we can readily obtain $\mathbf{h}=[\mathbf{1};\mathbf{0};\mathbf{0}]$, $\mathbf{x}=[\mathbf{c};\mathbf{s}_1;\mathbf{s}_2]$, $\mathbf{y}=[\mathbf{b}-\theta\mathbf{1};\mathbf{b}+\theta\mathbf{1}]$, $\mathcal{X}_i=\{x_i\geq 0\}$, and

$$\mathbf{A} = \begin{pmatrix} \mathbf{F} & -\mathbf{I}_L & \mathbf{0}_L \\ \mathbf{F} & \mathbf{0}_L & \mathbf{I}_L \end{pmatrix}.$$

Here $\mathbf{I}_L$ and $\mathbf{0}_L$ are $L\times L$ identity matrix and zero matrix, respectively.

Hence at time t, the classic ADMM procedures for the event detection problem are:

$$\begin{array}{c} c_i(t+1) = \left[\frac{\sum_{j\in\mathcal{I}_i} f_{ji}(dz_{1ji}(t)+dz_{2ji}(t)-p_{1ji}(t)-p_{2ji}(t))-1}{\sum_{j\in\mathcal{I}_i} 2df_{ji}^2}\right]^+, \\ i=1,...,L, \end{array} \tag{14}$$

$$\begin{array}{c} s_{1i}(t+1) = \left[\frac{1}{d}\lambda_{1i}(t) - \xi_{1i}(t)\right]^+, \\ s_{2i}(t+1) = \left[\xi_{2i}(t) - \frac{1}{d}\lambda_{2i}(t)\right]^+, \\ i=1,...,L, \end{array} \tag{15}$$

$$\begin{array}{c} \tau_{1j}(t+1) = \sum_{k\in\mathcal{I}_j}\frac{df_{jk}c_k(t+1)+p_{1jk}(t)}{|\mathcal{I}_j|+1} \\ +\frac{\lambda_{1j}(t)-ds_{1j}(t+1)-d(b_j-\theta)}{|\mathcal{I}_j|+1}, \\ \tau_{2j}(t+1) = \sum_{k\in\mathcal{I}_j}\frac{df_{jk}c_k(t+1)+p_{2jk}(t)}{|\mathcal{I}_j|+1} \\ +\frac{\lambda_{2j}(t)+ds_{2j}(t+1)-d(b_j+\theta)}{|\mathcal{I}_j|+1}, \\ j=1,...,L, \end{array} \tag{16}$$

$$\begin{array}{c} \xi_{1i}(t+1) = -s_{1i}(t+1) + \frac{\lambda_{1i}(t)}{d} - \frac{\tau_{1i}(t+1)}{d}, \\ \xi_{2i}(t+1) = s_{2i}(t+1) + \frac{\lambda_{2i}(t)}{d} - \frac{\tau_{2i}(t+1)}{d}, \\ i=1,...,L, \end{array} \tag{17}$$

$$\begin{array}{c} z_{1ji}(t+1) = f_{ji}c_i(t+1) + \frac{1}{d}p_{1ji}(t) - \frac{\tau_{1j}(t+1)}{d}, \\ z_{2ji}(t+1) = f_{ji}c_i(t+1) + \frac{1}{d}p_{2ji}(t) - \frac{\tau_{2j}(t+1)}{d}, \\ j=1,...,L, i\in\mathcal{I}_j, \end{array} \tag{18}$$

$$\begin{array}{c} \lambda_{1i}(t+1) = \lambda_{1i}(t) + d(-s_{1i}(t+1) - \xi_{1i}(t+1)), \\ \lambda_{2i}(t+1) = \lambda_{2i}(t) + d(s_{2i}(t+1) - \xi_{2i}(t+1)), \\ i=1,...,L, \end{array} \tag{19}$$

$$\begin{array}{c} p_{1ji}(t+1) = p_{1ji}(t) + d(f_{ji}c_i(t+1) - z_{1ji}(t+1)), \\ p_{2ji}(t+1) = p_{2ji}(t) + d(f_{ji}c_i(t+1) - z_{2ji}(t+1)), \\ j=1,...,L, i\in\mathcal{I}_i. \end{array} \tag{20}$$

Here $[\cdot]^+$ denotes the nonnegative mapping, ξ_{1i}, ξ_{2i}, z_{1ji} and z_{2ji} are auxiliary variables, τ_{1j} and τ_{2j} are intermediate variables to calculate the auxiliary variables, and λ_{1i}, λ_{2i}, p_{1ji}, and p_{1ji} are Lagrange multipliers.

3.2 The Decentralized Algorithm

From the derivation above, the decentralized algorithm with the classic ADMM is outlined as follows.

We first indicate that the algorithm has a nice decentralized implementation since for sensor node i, it only needs to communicate with its neighbors in $\mathcal{I}_i$. The enabling factor is that we have assumed $f_{ij}=0$ if $d_{ij}\geq r_C$ in **(A1)**. Therefore, $f_{ij}\neq 0$ only when $j\in\mathcal{I}_i$.

Second, the algorithm is robust to varying network topologies. Network reconfiguration is done in Step 1 upon having new sensory measurements. Further, the algorithm is robust to the network asynchronization. Though we update the variables here in a Gauss-Seidel manner, namely using the most up-to-date information, it is also workable to update the variables in a Jaccobi manner, namely using out-dated information. Readers of interest are referred to [7].

Decentralized Algorithm with the Classic ADMM

Step 0: Setting. Settings of the communication range r_C, the distance-dependent influence function $f(\cdot)$, and the measurement error range θ are known in advance to all sensor nodes.

Step 1: Initializing. In each event detection period, sensor node i initializes after having obtained the sensory measurement b_i, $i = 1, ..., L$. Sensor node i broadcasts a pilot message to all of its neighbors. Sensor node j, which is one of the neighbors of i, answers upon receiving the pilot message. Then i estimates the distance d_{ji} with the TOA or RSSI technologies [21] and calculates f_{ji} as well as $\mathcal{I}_i$ which contains itself and its neighbors.

Step 2: Updating Optimization and Slack Variables. At time t, sensor node i calculates $c_i(t+1)$, $s_{1i}(t+1)$, and $s_{2i}(t+1)$ with (14) and (15) by using $z_{1ji}(t)$, $z_{2ji}(t)$, $p_{1ji}(t)$, $p_{2ji}(t)$, $\lambda_{1i}(t)$, $\lambda_{2i}(t)$, $\xi_{1i}(t)$, and $\xi_{2i}(t)$ stored in itself, for $j \in \mathcal{I}_i$, $i = 1, ..., L$.

Step 3: Updating Intermediate Variables. At time t, sensor node i requests $c_j(t+1)$, $p_{1ij}(t)$, and $p_{2ij}(t)$ from $j \in \mathcal{I}_i$, then calculates $\tau_{1i}(t+1)$ and $\tau_{2i}(t+1)$ with (16), $i = 1, ..., L$.

Step 4: Updating Auxiliary Variables and Lagrange Multipliers. At time t, sensor node i requests $\tau_{1j}(t+1)$ and $\tau_{2j}(t+1)$ from $j \in \mathcal{I}_i$, and calculates $\xi_{1i}(t+1)$, $\xi_{2i}(t+1)$, $z_{1ji}(t+1)$, $z_{2ji}(t+1)$, $\lambda_{1i}(t+1)$, $\lambda_{2i}(t+1)$, $p_{1ji}(t+1)$, and $p_{2ji}(t+1)$ with (17), (18), (19), and (20), $i = 1, ..., L$.

Step 5: Iteratively Optimizing. Repeat from Step 2 to Step 4 iteratively until satisfying the stopping criterion.

Step 6: Alarming. Sensor node i alarms when the optimized c_i exceeds a certain threshold.

Third, we discuss the energy efficiency of the decentralized algorithm. Since the main source of energy consumption in a WSN is communication, we investigate the communication burden alternatively. Supposing that the iteration number is T, in each iteration, sensor node i needs to broadcast c_i, τ_{1i}, and τ_{2i}, and transmit p_{1ji} and p_{2ji} to its neighbor j. Hence the overall communication burden per sensor node is $T + 2T|\mathcal{I}_i|$, which is irrelevant to the network size L and hence scalable. As a comparison, in centralized algorithms, sensors nodes need to transmit all sensory measurements to a fusion center, and the average communication burden can be as high as $\sim L^{\frac{1}{2}}$. Further, considering the method of using grid points as candidate positions, the decentralized algorithm in [9] requires an average communication burden per iteration as $\sim K$. Recalling that K is the number of grid points, this method is also unscalable when the sensing field is large andor the expected spatial resolution is high.

From the analysis above, we can find that the communication burden of the decentralized algorithm is proportional to the amount of exchanged data per iteration. This fact motivates us to improve the energy efficiency by reducing the amount of exchanged data, which leads to a lightweight ADMM.

4 THE LIGHTWEIGHT ADMM

Revisiting the classic ADMM and substituting (17) to (19) and (18) to (20), we have $\lambda_{1i}(t) = \tau_{1i}(t)$, $\lambda_{2i}(t) = \tau_{2i}(t)$, $i = 1, ..., L$, as well as $p_{1ji}(t) = \tau_{1j}(t)$, $p_{2ji}(t) = \tau_{2j}(t)$, $j = 1, ..., L, i \in \mathcal{I}_j$. It means that in the classic ADMM, the intermediate variables τ_{1j} and τ_{2j} actually serve as Lagrange multipliers. Applying this conclusion in (14)-(18), introducing new intermediate variables w_{1j} and w_{2j}, and eliminating ξ_{1i}, ξ_{2i}, z_{1ji}, and z_{2ji}, the ADMM procedures turn to:

$$c_i(t+1) = \left[\frac{\sum_{j\in\mathcal{I}_i} f_{ji}(2df_{ji}(t)c_i(t) - dw_{1j}(t) - dw_{2j}(t))}{\sum_{j\in\mathcal{I}_i} 2df_{ji}^2} - \frac{\sum_{j\in\mathcal{I}_i} f_{ji}(\tau_{1j}(t)+\tau_{2j}(t))+1}{\sum_{j\in\mathcal{I}_i} 2df_{ji}^2}\right]^+, \quad i = 1, ..., L, \tag{21}$$

$$\begin{aligned} s_{1i}(t+1) &= \left[\tfrac{1}{d}\tau_{1i}(t) + s_{1i}(t) + w_{1i}(t)\right]^+, \\ s_{2i}(t+1) &= \left[s_{2i}(t) - w_{2i}(t) - \tfrac{1}{d}\tau_{2i}(t)\right]^+, \\ & i = 1, ..., L, \end{aligned} \tag{22}$$

$$\begin{aligned} w_{1j}(t+1) &= \frac{\sum_{k\in\mathcal{I}_j} f_{jk}c_k(t+1) - s_{1j}(t+1) - (b_j - \theta)}{|\mathcal{I}_j|+1}, \\ w_{2j}(t+1) &= \frac{\sum_{k\in\mathcal{I}_j} f_{jk}c_k(t+1) + s_{2j}(t+1) - (b_j + \theta)}{|\mathcal{I}_j|+1}, \\ & j = 1, ..., L, \end{aligned} \tag{23}$$

$$\begin{aligned} \tau_{1j}(t+1) &= \tau_{1j}(t) + dw_{1j}(t+1), \\ \tau_{2j}(t+1) &= \tau_{2j}(t) + dw_{2j}(t+1), \\ & j = 1, ..., L, \end{aligned} \tag{24}$$

Note that we are just simplifying the classic ADMM; thus the global convergence still keeps. The lightweight ADMM is outlined as follows.

In the lightweight ADMM, in each iteration, sensor node i needs to broadcast c_i, τ_{1i}, and τ_{2i} to its neighbors. When the iteration number is T, the overall communication burden is $3T$. Bring to mind the fact that the average node degree in a connected random graph is much larger than 1, obviously $|\mathcal{I}_i| - 1 \gg 1$, $i = 1, ..., L$. Therefore, for sensor node i, the communication burden of the lightweight ADMM, namely $3T$, is significantly reduced from that of the classic ADMM, namely $T + 2T|\mathcal{I}_i|$.

Decentralized Algorithm with the Lightweight ADMM

Step 0: Setting. Same to that in the classic ADMM.

Step 1: Initializing. Same to that in the classic ADMM.

Step 2: Updating Optimization and Slack Variables. At time t, sensor node i calculates $c_i(t+1)$, $s_{1i}(t+1)$, and $s_{2i}(t+1)$ with (21) and (22).

Step 3: Updating Intermediate Variables. At time t, sensor node i requests $c_j(t+1)$ from $j \in \mathcal{I}_i$, then calculates $w_{1i}(t+1)$ and $w_{2i}(t+1)$ with (23), $i = 1, ..., L$.

Step 4: Updating Lagrange Multipliers. At time t, sensor node i calculates $\tau_{1i}(t+1)$, $\tau_{2i}(t+1)$ with (24), and broadcasts them to its neighbors, $i = 1, ..., L$.

Step 5: Iteratively Optimizing. Repeat from Step 2 to Step 4 iteratively until satisfying the stopping criterion.

Step 6: Alarming. Sensor node i alarms when the optimized c_i exceeds a certain threshold.

5 SIMULATION RESULTS

In the simulation we uniformly randomly deploy 100 sensor nodes in a 100×100 sensing area. The communication range r_C varies in different experiments, with a minimal value of 20 which guarantees network connectivity. At the snapshot of interest, two events occur, one with magnitude 0.7142 at sensor node $\#001$ and another with magnitude $\#0.1413$ at sensor node $\#066$. The influence function $f(d_{ij})$ is set as $\exp(-d_{ij}^2/\sigma^2)$ with $\sigma = 10$, which makes $f(d_{ij}) \simeq 0$ when $d_{ij} \geq r_C$, and thus satisfies the assumption **(A2)**. Sensory measurement error is uniformly randomly distributed within $[-\theta, \theta]$ and $\theta = 0.01$. The network topology with $r_C = 20$ and the events occurring are depicted in Fig. 1.

The parameter d of the ADMM is chosen as 2. In the first experiment, we set $r_C = 20$ and compare the performance of the classic ADMM and the lightweight ADMM. Fig. 2 shows the estimated magnitudes by the sensor nodes. Sensor nodes $\#001$ and $\#066$ provides nearly accurate estimates which are close to the true values, while all of other sensor nodes report zero after a number of iterations. Squared error between the estimates and the true values and the squared error between the estimates and the optimal solution to (2) are shown in Fig. 3. The simulation experiment validates the effectiveness of the linear program formulation (2) as well as the algorithms using the ADMM.

Unsurprisingly we can find that the classic ADMM and the lightweight ADMM have the same convergence property from Fig. 2 and Fig. 3. Indeed, as we have indicated, the lightweight ADMM is simplified from the classic ADMM. Nevertheless, the two algorithms do imply totally different communication burden. Both of them nearly converge for $T = 80$; however, considering that the average node degree is ~ 10.7 in this case, the the average communication burden is $22.4T$ for the classic ADMM, which is much higher than that of $3T$ for the lightweight ADMM.

Note that in practical applications, it is not necessary to wait for optimization variables converging to the optimal values since in the event detection task, existence and position of an event are of rather more importance than its magnitude.

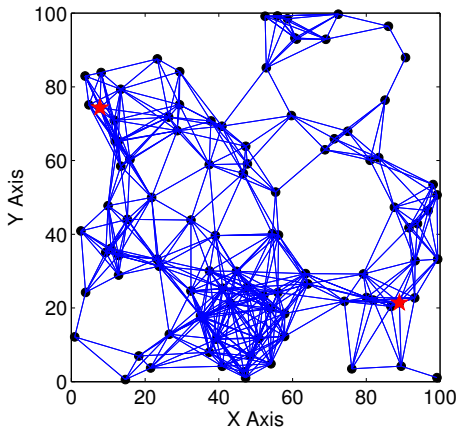

Fig. 1: Network topology with $r_C = 20$ is depicted; herein black dots denote sensor nodes and blue lines denote edges. Two events denoted as red pentagrams occur at sensor nodes $\#001$ (northwest) and $\#066$ (southeast), respectively.

In the second experiment, we vary the communication range r_C as 20, 30, and 40. As shown in Fig. 4, the convergence rates are similar and the energy consumptions are hence also similar for the lightweight ADMM. However, the average node degrees for these communication ranges are ~ 10.7, 21.3, and 34.3, respectively, and thus the energy consumptions for the classic ADMM are quite different; the larger the communication range, the worse the energy efficiency.

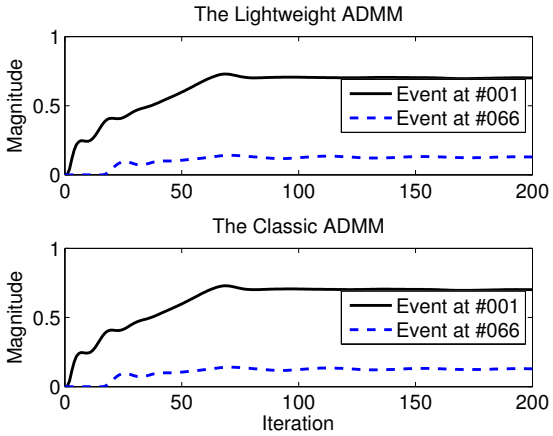

Fig. 2: Estimated event magnitudes.

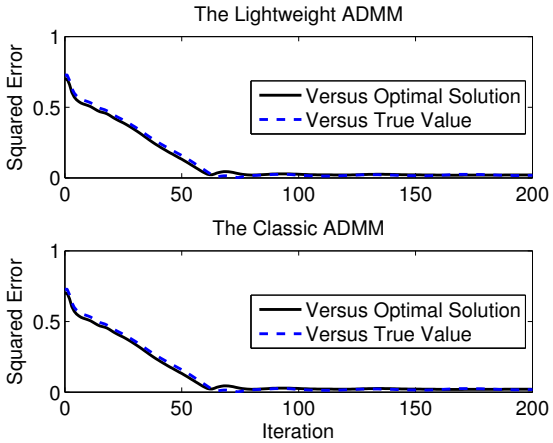

Fig. 3: Squared errors.

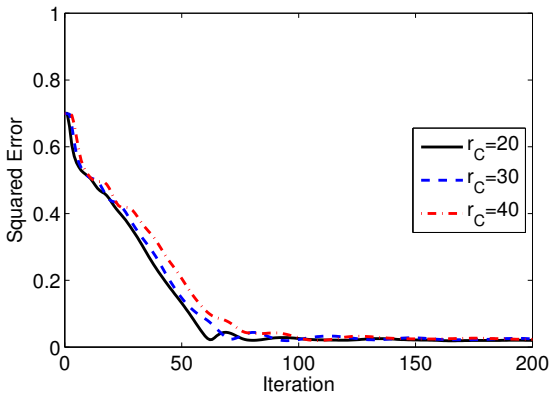

Fig. 4: Squared errors of the lightweight ADMM for different settings of the communication range r_C.

6 CONCLUSIONS

In this paper we address the problem of decentralized event detection in a large-scale WSN. In the existing centralized scheme, sensory measurements are all collected by a fusion center which processes the information thereafter. This centralized scheme is inefficient with regard to energy consumption and unscalable to network size. Contrarily, we propose a fully decentralized information processing scheme, where each sensor node makes decision by itself via only limited communication with its neighbors. In consequence, energy efficiency and network scalability are greatly strengthened.

To enable decentralized event detection, we formulate the event detection task as a linear program which can be solved by the classic ADMM. Under mild conditions, this iterative algorithm is fully decentralized. Considering that the energy consumption is proportional to communication burden, we further simplify the classic ADMM to a lightweight ADMM, which requires much fewer amount of exchanged data per iteration without affecting the convergence property. Extensive simulation results validate effectiveness of the proposed lightweight ADMM.

The discussions in this paper also imply that the energy consumption of a decentralized algorithm is proportional to the iteration number. Therefore, how to accelerate the algorithm will be an important future research direction.

REFERENCES

[1] A. Sundaresan, P. Varshney, and N. Rao, "Distributed detection of a nuclear radioactive source using fusion of correlated decisions," In: Proceedings of FUSION, 2007.

[2] J. Lynch, "An overview of wireless structural health monitoring for civil strucutres," Philosophical Transactions of the Royal Society A, vol. 365, pp. 345–372, 2007.

[3] X. Sun and E. Coyle, "Low-complexity algorithms for event detection in wireless sensor networks," IEEE Journal on Selected Areas in Communications, vol. 28, pp. 1138–1148, 2010.

[4] M. Cetin, L. Chen, J. Fisher III, A. Ihler, R. Moss, M. Wainwright, and A. Willsky, "Distributed fusion in sensor networks," IEEE Signal Processing Magazine, vol. 23, pp. 42–55, 2006.

[5] J. Predd, S. Kulkarni, and H. Poor, "A collaborative training algorithm for distributed learning," IEEE Transactions on Information Theory, vol. 55, pp. 1856–1871, 2009.

[6] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," In: Proceedings of IPSN, 2004.

[7] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[8] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," IEEE Transactions on Signal Processing, vol. 57, pp. 2365–2381, 2009.

[9] J. Bazerque and G. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," IEEE Transactions on Signal Processing, vol. 58, pp. 1847–1862, 2010.

[10] Q. Ling, Z. Tian, Y. Yin, and Y. Li, "Localized structural health monitoring using energy-efficient wireless sensor networks," IEEE Sensors Journal, vol.9, pp. 1596–1604, 2009.

[11] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," IEEE Transactions on Signal Processing, vol. 58, pp. 3816–3827, 2010.

[12] M. Lucchi and M. Chiani, "Distributed detection of local phenomena with wireless sensor networks," In: Proceedings of ICC, 2010.

[13] D. Donoho,"Compressed sensing," IEEE Transactions on Information Theory, vol. 52, pp. 1289–1306, 2006.

[14] S. Chen, D. Donoho, and M. Saunders, "Atomic decompostion by basis pursuit," SIAM Journal on Scientific Computing, vol. 20, pp. 33–61, 1998.

[15] R. Tibshirani, "Regression shrinkage and selection via the Lasso," Journal of Royal Statistical Society B, vol. 58, pp. 267–288, 1996.

[16] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," IEEE Journal of Selected Topics in Signal Processing, vol. 1, pp. 586–597, 2007.

[17] E. Candes and T. Tao, "The Dantzig selector: statistical estimation when p is much larger than n," Annals of Statistics, vol. 35, pp. 2313–2351, 2007.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[19] V. Cevher, M. Duarte, and R. Baraniuk, "Distributed target localization via spartial sparsity," In: Proceedings of EUSIPCO, 2008.

[20] A. Schmidt and J. Moura, "Field inversion by consensus and compressed sensing," In: Proceedings of ICASSP, 2009.

[21] K. Ssu, C. Ou, and H. Jiau, "Localization with mobile anchor points in wireless sensor networks," IEEE Transactions on Vehicular Technology, vol. 54, pp. 1187–1197, 2005.