LEARNING IN DIFFUSION NETWORKS WITH AN ADAPTIVE PROJECTED SUBGRADIENT METHOD

Renato L. G. Cavalcante †, Isao Yamada ‡, and Bernard Mulgrew †

† The University of Edinburgh,
Digital Communications Research Institute,
Joint Research Institute for Signal
and Image Processing,
Edinburgh EH9 3JL, UK,
email: {r.cavalcante,b.mulgrew}@ed.ac.uk

‡ Tokyo Institute of Technology,
Department of Communications
and Integrated Systems,
Tokyo 152-8550, Japan,
email: isao@comm.ss.titech.ac.jp

ABSTRACT

We present an algorithm that minimizes asymptotically a sequence of non-negative convex functions over diffusion networks. To account for possible node failures, position changes, and/or reachability problems (because of moving obstacles, jammers, etc), the algorithm can cope with dynamic networks and cost functions, a desirable feature for online algorithms where information arrives sequentially. Many projection-based algorithms can be straightforwardly extended to diffusion networks with the proposed scheme. We use the acoustic source localization problem in sensor networks as an example of a possible application.

Index Terms— optimization methods, distributed algorithms, distributed tracking, position measurement

1. INTRODUCTION

Networks consisting of nodes collecting data over a geographic area are envisioned to make a dramatic impact on a number of applications such as, among others, precision agriculture, disaster relief management, radar, and acoustic source localization [1–3]. In these applications, each node has some computational power and is able to send data to a subset of the network nodes. The objective is to improve the estimate of some parameter of interest in every node with this exchange of information [2]. The interaction between the nodes is dictated by network topology and the allowed modes of cooperation. Depending on the mode of cooperation, the network can be classified as an incremental network or a diffusion network.

The incremental mode of cooperation requires the minimum amount of power and is easy to implement in a network with a fairly small number of nodes. However, with the recent advances of wireless communications and electronics, future applications envision nodes densely deployed [1], and diffusion networks are more appropriate in such cases [4]. In addition, diffusion networks can easily deal with node failures, changing topologies, and/or communication problems between the nodes. Therefore, in this study we focus on diffusion networks.

Many existing diffusion algorithms have been devised for a class of problems similar to that of system identification [4], or they have been devised to minimize fixed (convex) cost functions [5]. Here, we consider an algorithm that can be applied to more general problems where both the cost function and/or the network topology is time varying. The algorithm builds on the adaptive projected subgradient method [6], an extension of Polyak's algorithm to the case where the cost function is time varying. The acoustic source localization in diffusion networks is given as an example of a possible application.

2. BASIC TOOLS IN CONVEX ANALYSIS

For every vector $\boldsymbol{v} \in \mathbb{R}^N$, we define the norm of $\boldsymbol{v}$ by $\|\boldsymbol{v}\| := \sqrt{\boldsymbol{v}^T\boldsymbol{v}}$, which is the norm induced by the Euclidean inner product $\langle \boldsymbol{v}, \boldsymbol{y}\rangle := \boldsymbol{v}^T\boldsymbol{y}$ for every $\boldsymbol{v}, \boldsymbol{y} \in \mathbb{R}^N$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{M\times N}$, its spectral norm is $\|\boldsymbol{X}\|_2 := \max\{\sqrt{\lambda} | \ \lambda$ is an eigenvalue of $\boldsymbol{X}^T\boldsymbol{X}\}$, which satisfies $\|\boldsymbol{X}\boldsymbol{y}\| \leq \|\boldsymbol{X}\|_2\|\boldsymbol{y}\|$ for any vector $\boldsymbol{y} \in \mathbb{R}^N$.

A set $C \subseteq \mathbb{R}^N$ is said to be convex if $\boldsymbol{v} = \alpha\boldsymbol{v}_1 + (1-\alpha)\boldsymbol{v}_2 \in C$ for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $0 \leq \alpha \leq 1$ [6]. Let $C \subseteq \mathbb{R}^N$ be a nonempty closed convex set. The metric projection $P_C : \mathbb{R}^N \to C$ assigns $\boldsymbol{v} \in \mathbb{R}^N$ to its (uniquely existing) nearest point in C and satisfies $\|\boldsymbol{v} - P_C(\boldsymbol{v})\| = \min_{\boldsymbol{y}\in C} \|\boldsymbol{v} - \boldsymbol{y}\| =: d(\boldsymbol{v}, C)$.

A function $\Theta : \mathbb{R}^N \to \mathbb{R}$ is said to be *convex* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\forall \nu \in [0, 1]$, $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$ [6]. Let Θ be a convex function. The subdifferential of Θ at $\boldsymbol{y}$ is the set of all *subgradients* of Θ at $\boldsymbol{y}$ [6]:

$$\partial\Theta(\boldsymbol{y}) := \{\boldsymbol{a} \in \mathbb{R}^N | \Theta(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{a}\rangle \leq \Theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^N\} \neq \emptyset. \quad (1)$$

3. PROPOSED ALGORITHM

3.1. Problem formulation

Consider a network with node set $\mathcal{N} := \{1, \ldots, N\}$. At time i, the (dynamic) directed graph of the network is given by $\mathcal{G}[i] := (\mathcal{N}, \mathcal{E}[i])$, where $\mathcal{E}[i] \subseteq \mathcal{N} \times \mathcal{N}$ is the edge set (see also [5, 7]). If node k can send data to node l (at time i), we define the directed link by $(k, l) \in \mathcal{E}[i]$. (We also assume that node k is linked to itself, hence $(k, k) \in \mathcal{E}[i]$.) The set of inward neighbors of node k is $\mathcal{N}_k[i] = \{l \in \mathcal{N} | \ (l, k) \in \mathcal{E}[i]\}$.

Let $\Theta_k[i] : \mathbb{R}^M \to [0, \infty)$ $(\forall i \in \mathbb{N})$ be a sequence of convex cost functions known by the kth node $(k \in \mathcal{N})$. In addition, assume that (at time i) node k has an estimate $\boldsymbol{h}_k[i] \in \mathbb{R}^M$ of a minimizer of $\Theta_k[i]$. The time-varying function $\Theta_k[i]$ is the *local cost function* for node k, and we define the *target cost function* $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ as the sum of the N time-varying local cost functions, i.e.,

$$\Theta[i](\boldsymbol{h}) = \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h}), \quad (2)$$

Note that, unlike most studies, here both the network topology and the cost function are allowed to change at each iteration.

Hereafter we assume that $\Omega[i] := \bigcap_{k\in\mathcal{N}} \Omega_k[i] \neq \emptyset$, where

$$\Omega_k[i] := \left\{ \boldsymbol{h} \in \mathbb{R}^M \mid \Theta_k[i](\boldsymbol{h}) = \Theta_k^\star[i] := \inf_{\boldsymbol{h}\in\mathbb{R}^M} \Theta_k[i](\boldsymbol{h}) \right\} \quad (k \in \mathcal{N}). \quad (3)$$

Therefore, any $\boldsymbol{h}^\star[i] \in \Omega[i]$ is a minimizer of (2). The objective of the proposed algorithm is to minimize (asymptotically) the target cost function in (2) in every node. In addition, the nodes should reach consensus, i.e., $\boldsymbol{h}_k[i] = \boldsymbol{h}_l[i]$ $(k, l \in \mathcal{N})$. Note that each node has only partial knowledge of $\Theta[i]$ and can only communicate with its neighbors.

3.2. Diffusion adaptive projected subgradient method

Before presenting an algorithm that solves asymptotically the problem described in Sect. 3.1, we define and give some properties of *consensus matrices*, which play an important role in the convergence analysis of the algorithm.

Definition 1 *A matrix $\boldsymbol{P} \in \mathbb{R}^{MN\times MN}$ is a consensus matrix for the problem in Sect. 3.1 if $\boldsymbol{P}$ satisfies the following:*

i) $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}$ and $\boldsymbol{P}^T\boldsymbol{x} = \boldsymbol{x}$ for every vector $\boldsymbol{x} \in C := \{[\boldsymbol{a}^T \ \dots \ \boldsymbol{a}^T]^T \in \mathbb{R}^{MN} \mid \boldsymbol{a} \in \mathbb{R}^M\}$;

ii) The M largest singular values of $\boldsymbol{P}$ are equal to one and the remaining $MN - M$ singular values are strictly less than one.

Definition 2 *We say that a matrix is compatible with the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ if $\boldsymbol{\psi}[i+1] = \boldsymbol{P}\boldsymbol{\psi}[i]$ can be equivalently computed as $\boldsymbol{h}_k[i+1] = \sum_{l\in\mathcal{N}_k} \alpha_{k,l}\boldsymbol{h}_l[i]$ for every $k \in \mathcal{N}$, where $\boldsymbol{\psi}[i] := [\boldsymbol{h}_1[i]^T \dots \boldsymbol{h}_N[i]^T]^T$ and $\alpha_{k,l} \in \mathbb{R}$ is the weight associated with the edge (l, k). In other words, if the product $\boldsymbol{P}\boldsymbol{\psi}[i]$ can be computed with the local communication allowed by the network topology, then $\boldsymbol{P}$ is compatible with the graph of the network.*

Lemma 1 (Selected properties of consensus matrices)
Let $\boldsymbol{P} \in \mathbb{R}^{MN\times MN}$ be a consensus matrix, $\boldsymbol{e}_k \in \mathbb{R}^M$ be the vector of zeros, except for its kth entry, which is set to one, and C be the subspace $C := \operatorname{span}\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_M\}$, where $\boldsymbol{b}_k = (\boldsymbol{1}_N \otimes \boldsymbol{e}_k)/\sqrt{N} \in \mathbb{R}^{MN}$, $\boldsymbol{1}_N \in \mathbb{R}^N$ is the vector of ones, and $\otimes$ denotes the Kronecker product. At time i, define $\boldsymbol{\psi}[i] := [\boldsymbol{h}_1[i]^T \ \dots \ \dots \boldsymbol{h}_N[i]^T]^T \in \mathbb{R}^{MN}$. We have the following properties:

1. *Any consensus matrix $\boldsymbol{P}$ can be decomposed into $\boldsymbol{P} = \boldsymbol{B}\boldsymbol{B}^T + \boldsymbol{X}$, where $\boldsymbol{B} := [\boldsymbol{b}_1 \ \dots \ \boldsymbol{b}_M] \in \mathbb{R}^{MN\times M}$ and $\boldsymbol{X} \in \mathbb{R}^{MN\times MN}$ is a matrix satisfying i) $\boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{X} = \boldsymbol{0}$ and ii) $\|\boldsymbol{X}\|_2 < 1$. (Note that $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_M$ by construction.)*
2. *The nodes of the network are in agreement at time i (i.e, $\boldsymbol{h}_1[i] = \dots = \boldsymbol{h}_N[i]$ or, equivalently, $\psi[i] \in C$) if and only if $(\boldsymbol{I}_{MN} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{\psi}[i] = \boldsymbol{0}$.*

Because of the space limitation, we omit the proofs of the previous lemma and the following theorem, which shows the algorithm that can solve asymptotically the problem in Sect. 3.1 (see [8] for details).

Theorem 1 *(Adaptive projected subgradient method for diffusion networks)*
Consider a network with N nodes and dynamic graph given by $\mathcal{G}[i] := (\mathcal{N}, \mathcal{E}[i])$. Let $\boldsymbol{P}[i] \in \mathbb{R}^{MN\times MN}$ $(i = 0, 1, \dots)$ be a sequence of matrices satisfying $\|\boldsymbol{P}[i]\|_2 = 1$ and $\boldsymbol{P}[i]\boldsymbol{x} = \boldsymbol{x}$ for every $\boldsymbol{x} \in C := \operatorname{span}\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_M\}$, where $\boldsymbol{b}_k$ is as defined in Lemma 1 (NOTE: $\boldsymbol{P}[i]$ is not necessarily a consensus matrix). In addition, let $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ $(\forall i \in \mathbb{N})$ be defined as $\Theta[i](\boldsymbol{h}) = \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h})$ $(\boldsymbol{h} \in \mathbb{R}^M)$, where $\Theta_k[i] : \mathbb{R}^M \to [0, \infty)$ $(k \in \mathcal{N})$ is a sequence of convex functions available to the kth node of the network. For $\Omega_k[i] \neq \emptyset$ (see the definition in (3)), we define a sequence given by

$$\boldsymbol{\psi}[i+1] = \boldsymbol{P}[i]\left(\boldsymbol{\psi}[i] - \begin{bmatrix} \mu_1[i]\alpha_1[i]\Theta_1'[i](\boldsymbol{h}_1[i]) \\ \vdots \\ \mu_N[i]\alpha_N[i]\Theta_N'[i](\boldsymbol{h}_N[i]) \end{bmatrix}\right), \quad (4)$$

where $\alpha_k[i] = (\Theta_k[i](\boldsymbol{h}_k[i]) - \Theta_k^\star[i])/(\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2 + \delta_k[i])$, $\mu_k[i] \in [0, 2]$, $\Theta_k'[i](\boldsymbol{h}_k[i]) \in \partial\Theta_k[i](\boldsymbol{h}_k[i])$ (see (1)), $\Theta_k^\star[i] := \inf_{\boldsymbol{h}\in\mathbb{R}^M} \Theta_k[i](\boldsymbol{h})$ $(k \in \mathcal{N})$, $\delta_k[i] > 0$ is an arbitrarily small (bounded) number, $\boldsymbol{\psi}[i] := [\boldsymbol{h}_1[i]^T \ \dots \ \dots \boldsymbol{h}_N[i]^T]^T$, and $\boldsymbol{h}_k[i]$ is the estimate of $\boldsymbol{h}_k^\star[i] \in \Omega_k[i]$ in the kth node. Then we have the following properties:

(a) (Monotone approximation)
Suppose that $\Omega[i] := \bigcap_{k\in\mathcal{N}} \Omega_k[i] \neq \emptyset$. In addition, let $\mu_k[i] \in (0, 2)$ for at least one node such that $\boldsymbol{h}_k[i] \notin \Omega_k[i]$ $(k \in \mathcal{N})$. The iteration in (4) with $\mu_l[i] \in [0, 2]$ $(l \neq k)$ satisfies

$$\|\boldsymbol{\psi}[i+1] - \boldsymbol{\psi}^\star[i]\| < \|\boldsymbol{\psi}[i] - \boldsymbol{\psi}^\star[i]\|$$

for every $\boldsymbol{\psi}^\star[i] \in C^\star[i] := \{\boldsymbol{\psi} = [\boldsymbol{h}^T \ \boldsymbol{h}^T \ \dots \boldsymbol{h}^T]^T \in \mathbb{R}^{MN} \mid \boldsymbol{h} \in \Omega[i]\}$.

(b) (Asymptotic optimality) Suppose that

$$\exists K_0 \in \mathbb{N} \ \textit{s.t.} \begin{cases} \Theta_k^\star[i] =: \Theta_k^\star \in \mathbb{R}, \ \forall i \geq K_0 \ \text{ and} \\ \Omega := \bigcap_{i\geq K_0} \Omega[i] \neq \emptyset \end{cases}$$

and that $\Theta_k'[i](\boldsymbol{h}_k[i])$ is bounded $(k \in \mathcal{N}, \ i = 0, 1, \dots)$. If we use $\mu_k[i] \in [\epsilon_1, 2 - \epsilon_2] \subset (0, 2)$, then

$$\lim_{i\to\infty} \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h}_k[i]) = \Theta^\star. \quad (5)$$

where $\Theta^\star = \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h}^\star)$ for every $\boldsymbol{h}^\star \in \Omega$ and $i \geq K_0$.

(c) (Convergence to consensus) In addition to the assumptions in item (b) above, let $\boldsymbol{P}[i]$ be a possibly time-varying consensus matrix compatible with the graph of the network at time i. We can always decompose $\boldsymbol{P}[i]$ into $\boldsymbol{P}[i] = \boldsymbol{B}\boldsymbol{B}^T + \boldsymbol{X}[i]$, where the matrices $\boldsymbol{B}$ and $\boldsymbol{X}[i]$ and their properties are described in Lemma 1. If $\|\boldsymbol{X}[i]\|_2 \leq 1 - \epsilon_3$ for some $\epsilon_3 > 0$ $(i = 0, 1, 2, \dots)$, then

$$\lim_{i\to\infty} [(\boldsymbol{I}_{MN} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{\psi}[i]] = \boldsymbol{0}, \quad (6)$$

i.e., consensus ($\boldsymbol{h}_1[i] = \dots = \boldsymbol{h}_N[i]$) is asymptotically achieved.

4. APPLICATION AND SIMULATION RESULTS

4.1. Acoustic sensor localization as the intersection of closed convex sets [3]

Consider a sensor network with node set $\mathcal{N}$, where the sensors are distributed at known spatial locations denoted by $\boldsymbol{r}_k \in \mathbb{R}^2$ (extension to $\mathbb{R}^3$ is straightforward), $k \in \mathcal{N}$. If there exists an acoustic source located at the unknown position $\boldsymbol{h}^\star \in \mathbb{R}^2$, a model for the estimated signal strength at node k is

$$y_k = \frac{A}{\|\boldsymbol{r}_k - \boldsymbol{h}^\star\|^2} + v_k \in \mathbb{R}, \quad (7)$$

where A is the energy of the acoustic signal and v_k is a zero-mean white Gaussian noise with variance $\sigma^2_{v_k}$ (see [3] and the references therein for the validity and a detailed description of this model).

Ignoring the presence of noise for the moment, the source location $\boldsymbol{h}^\star$ is on the intersection of disks given by

$$D_k = \left\{\boldsymbol{h} \in \mathbb{R}^2 | \ \|\boldsymbol{h} - \boldsymbol{r}_k\| \le \sqrt{\frac{A}{y_k}}\right\} (k \in \mathcal{N}). \quad (8)$$

Therefore, a reasonable estimate of $\boldsymbol{h}^\star$ is

$$\widetilde{\boldsymbol{h}}^\star \in D = \bigcap_{k\in\mathcal{N}} D_k, \quad (9)$$

which can be found with an incremental distributed version of the projection onto convex sets (POCS) algorithm [3].

For the consistency of the estimate $\widetilde{\boldsymbol{h}}^\star$, the source location $\boldsymbol{h}^\star$ should lie in the convex hull defined by the position of the nodes [3], i.e.,

$$\boldsymbol{h}^\star \in H := \left\{\boldsymbol{h} \in \mathbb{R}^2 | \ \boldsymbol{h} = \sum_{k\in\mathcal{N}} \beta_k \boldsymbol{r}_k, \ \beta_k \ge 0, \ \sum_{k\in\mathcal{N}} \beta_k = 1\right\}$$

For brevity, here we assume that the source power A is known, but the case of unknown A can also be posed as finding a point in the intersection of closed convex sets [3, Sect. IV].

4.2. Proposed algorithm for acoustic source localization

Consider a network with graph given by $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ (for brevity, hereafter the graph is fixed). In diffusion networks, nodes can share data with their neighbors, so we assume that node k knows the position $\boldsymbol{r}_l$ and estimates y_l of its inward neighbors, nodes $l \in \mathcal{N}_k$ ($\boldsymbol{r}_l$ and y_l need to be sent only once).

Taking into account the model in (7), we see that the radius of the disk D_k decreases if $v_k > 0$, thus $\boldsymbol{h}^\star \notin D_k$. To increase the probability that the source location $\boldsymbol{h}^\star$ lies in every disk, we propose possibly time-varying sets given by

$$C_k[i] := \begin{cases} \left\{\boldsymbol{h} \in \mathbb{R}^2 | \ \|\boldsymbol{h} - \boldsymbol{r}_k\| \le \sqrt{\dfrac{A}{y_k - e_k[i]}}\right\} \\ \qquad\qquad \text{if} \quad y_k - e_k[i] > 0 \\ \mathbb{R}^2 \quad \text{otherwise,} \end{cases} \quad (10)$$

where $e_k[i] > 0$ is an inflation parameter for the radius of $C_k[i]$.

To devise a fast algorithm, we use the (possibly time-varying) local cost function given by (see also [6, Example 2])

$$\Theta_k[i](\boldsymbol{h}) := \sum_{l\in\mathcal{N}_k} \omega_{l,k} d^2(\boldsymbol{h}, C_l[i]), \quad (k \in \mathcal{N}) \quad (11)$$

where $\omega_{l,k}$ is the weight that node k assigns to the disk of its neighbor $l \in \mathcal{N}_k$. The weights should be non-negative and satisfy $\sum_{l\in\mathcal{N}_k} \omega_{l,k} = 1$ for every $k \in \mathcal{N}$. Assuming that $\cap_{k\in\mathcal{N}} C_k[i] \neq \emptyset$ (NOTE: this assumption does not necessarily hold in Sect. 4.3), $\widehat{\boldsymbol{h}}^\star \in \bigcap_{k\in\mathcal{N}_k} C_k[i]$ is a minimizer of every local cost function $\Theta_k[i]$ ($k \in \mathcal{N}$), hence $\widehat{\boldsymbol{h}}^\star$ is also a minimizer of the global cost function, i.e., $\Theta[i](\widehat{\boldsymbol{h}}^\star) = \sum_{k\in\mathcal{N}} \Theta_k[i](\widehat{\boldsymbol{h}}^\star) = 0$. Therefore, as in [3], the objective is to find a point in the intersection of closed convex sets, but here the sets are possibly time varying and the network under consideration is a diffusion network. When noise is present, giving larger weights to disks that are likely to have higher signal-to-noise ratio (SNR) is reasonable, so we use $\omega_{l,k} = y_k / \sum_{l\in\mathcal{N}_k} y_l$.

Applying the algorithm in Theorem 1 to $\Theta[i]$, we arrive at:

Algorithm 1
For time $i = 0, 1, \ldots$
  for *k=1:N*
    Choose $\delta_k[i]$ *arbitrarily small and* $\mu_k[i]$ *within the range* $\mu_k[i] \in (0, \mathcal{M}_k[i])$*, where*

$$\mathcal{M}_k[i] = \begin{cases} 1, \quad \text{if} \quad \boldsymbol{h}_k[i] \in \bigcap_{l\in\mathcal{N}_k} C_l[i] \\ \dfrac{\sum_{l\in\mathcal{N}_k} \omega_{l,k} \|P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2}{\left\|\sum_{l\in\mathcal{N}_k} \omega_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\right\|^2 + \delta_k[i]}, \\ \qquad \text{otherwise.} \end{cases}$$

$$\boldsymbol{h}'_k[i] = \boldsymbol{h}_k[i] + \mu_k[i] \left(\sum_{l\in\mathcal{N}_k} \omega_{l,k} P_{C_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\right)$$

  end

$$\begin{bmatrix} \boldsymbol{h}_1[i+1] \\ \vdots \\ \boldsymbol{h}_N[i+1] \end{bmatrix} = \boldsymbol{P}[i] \begin{bmatrix} \boldsymbol{h}'_1[i] \\ \vdots \\ \boldsymbol{h}'_N[i] \end{bmatrix}$$

where $\boldsymbol{P}[i]$ *is a consensus matrix compatible with the graph of the network at time i.*

Remark 1 *If* $y_k - e_k[i] > 0$*, the projection of* $\boldsymbol{h}$ *onto* $C_k[i]$ *is given by*

$$P_{C_k[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in C_k[i] \\ \boldsymbol{r}_k + \dfrac{\epsilon}{\|\boldsymbol{h} - \boldsymbol{r}_k\|}(\boldsymbol{h} - \boldsymbol{r}_k) & \text{otherwise,} \end{cases}$$

where $\epsilon = \sqrt{A/(y_k - e_k[i])}$*. If* $C_k[i] = \mathbb{R}^2$*, then* $P_{C_k[i]}(\boldsymbol{h}) = \boldsymbol{h}$*.*

4.3. Numerical simulations

We consider a sensor network with 5000 nodes randomly distributed in a 100m $\times$ 100m field. A measure of the acoustic source energy is generated following the model in (7), where $A = 100$ and $\sigma^2_{v_k} = 1$. The source is located at $\boldsymbol{h}^\star = [50\ 50]^T$, and, at each realization, only nodes with $y_k \ge 5$ participate in the estimation task. These settings are similar to those in [3]. For brevity, the network is static and the graph is undirected (at each realization). The weight $\alpha_{k,l}$ ($= \alpha_{l,k}$) of the edge (k, l) ((l, k)) is the Metropolis-Hastings weight [4]

$$\alpha_{k,l} = \begin{cases} 1/(\max(g_k, g_l)) & \text{if } k \neq l \text{ and } l \in \mathcal{N}_k \\ 1 - \sum_{l\in\mathcal{N}_k\setminus\{k\}} \alpha_{k,l} & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases}$$

where $g_k = |\mathcal{N}_k|$ is the degree for node k. Hereafter, all diffusion algorithms based on Theorem 1 use the consensus matrix $\boldsymbol{P}[i] = \boldsymbol{A} \otimes \boldsymbol{I}_2 \in \mathbb{R}^{2N\times 2N}$ ($i = 0, 1, \ldots$), where the component of the kth row and lth column of $\boldsymbol{A}$ is given by $\alpha_{k,l}$.

The ultimate goal of every algorithm in this section is to find the source location $\boldsymbol{h}^\star$. Therefore, to compare different algorithms, we use the (average) mean square error (MSE) and the (average) mean square distance to consensus (MSDC) (a measure of how far the network is from achieving consensus), which we define as

$$\text{MSE}[i] = E[(1/N)\|\boldsymbol{\psi}[i] - \boldsymbol{1}_N \otimes \boldsymbol{h}^\star\|^2]$$
$$\text{MSDC}[i] = E[(1/N)\|(\boldsymbol{I}_{2N} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{\psi}[i]\|^2],$$

where $\boldsymbol{B}$ is as defined in Lemma 1 and $\boldsymbol{\psi}[i] = [\boldsymbol{h}_1[i]^T \ \ldots \ \boldsymbol{h}_N[i]^T]^T$. In the following, ensemble average curves are obtained from the average of 20,000 realizations.

We evaluate the performance of two versions of Algorithm 1: Proposed-1 ($e_k[i] = 0$ and $\mu_k[i] = 0.2$) and Proposed-2 ($e_k[i] = i/10$ and $\mu_k[i] = 0.99$). The main idea of Proposed-2 is to expand slightly the sets at each iteration, and, from the moment the intersection of all sets is not empty, the estimates of all nodes should rapidly converge to a point in the intersection (and remain in this point, which is expected to be close to $\boldsymbol{h}^\star$). The nodes are linked as follows. First, we randomly assign consecutive integer numbers to each node, and then we link each node to its 10 closest neighbors. To guarantee that the graph is connected, we also link nodes with consecutive numbers (the last node is linked to the first node).

For comparison purposes, we also plot the results obtained with the incremental POCS algorithm in [3]. The relaxation parameter in [3, Eq. (10)] is set to one in the first 30 iterations, and then it decreases to $1/i$. The order of the sensors in the cycle is selected randomly. Note that Algorithm 1 is not a direct competitor of that in [3]. Here the objective is to develop an algorithm (for networks with densely deployed sensors) robust to link failures, and the estimates of all nodes should converge rapidly to a vicinity of $\boldsymbol{h}^\star$ (hence we use diffusion networks). On the other hand, the idea of the method in [3] is to develop an algorithm with very low communication cost, where an estimate of a node (not necessarily all at the same time) should converge rapidly to a vicinity of $\boldsymbol{h}^\star$ (incremental networks are usually more appropriate for such cases). For all algorithms, the initial estimates are set to $\boldsymbol{h}_k[0] = \boldsymbol{r}_k$ ($k \in \mathcal{N}$). Simulation results are shown in Fig. 1(a)-(b).

Compared to Proposed-1, Proposed-2 is faster because of the larger step size $\mu_k[i]$ and it also better estimate $\boldsymbol{h}^\star$ because the sets are inflated at each iteration, increasing the probability that $\boldsymbol{h}^\star \in \bigcap_{k \in \mathcal{N}_k} C_k[i]$. For Proposed-2, $\bigcap_{k \in \mathcal{N}_k} C_k[i] \neq \emptyset$ for $i \geq K_0$ (K_0 is a positive integer) is guaranteed, so Theorem 1(c) shows that consensus is achieved asymptotically, a fact that can be seen in Fig. 1(b). Proposed-1 does not achieve consensus because the conditions of Theorem 1(b) are not necessarily satisfied due to the presence of noise. However, Proposed-1 shows that, even if the assumptions in Sect. 4.2 do not hold, the algorithm can provide good estimates of the source location in every node with few iterations. The incremental POCS algorithm is the slowest algorithm because only one node is updated at each iteration. Therefore, at least one cycle is necessary to update all nodes with a good estimate of the source location, which is a characteristic of algorithms for incremental networks.

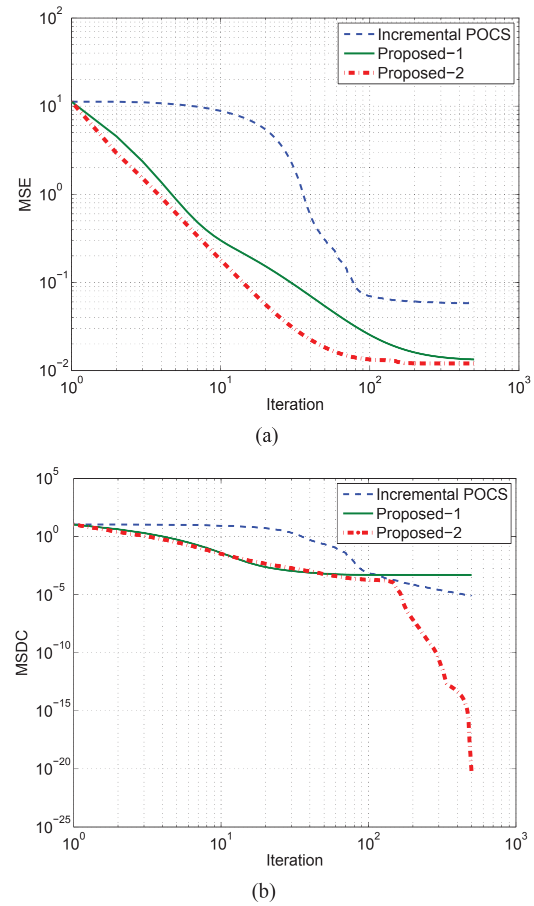

Fig. 1. Transient performance of the algorithms

5. FINAL REMARKS

We have developed an iterative distributed algorithm for the asymptotic minimization of a sequence of convex cost functions. We used the source localization problem over diffusion networks as an example of an application. Numerical simulations showed that, with few iterations, every node provides a good estimate of the source location. The proposed algorithm in its most general form (Theorem 1) can deal with dynamic networks and different cost functions, so we expect that its application to the source localization problem can give better results if we devise more efficient diffusion protocols, sets, etc. Finally, in the future we envision many other applications that can be addressed with the scheme in Theorem 1.

6. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Signal Processing Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Transactions on Fundamentals*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[3] D. Blatt and A. O. Hero III, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Trans. Signal Processing*, vol. 54, no. 9, pp. 3614–3619, Sept. 2006.

[4] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[5] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, to appear.

[6] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numerical Functional Analysis and Optimization*, vol. 25, no. 7/8, pp. 593–617, 2004.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[8] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Processing*, submitted.