Distributed Multiagent Learning with a Broadcast Adaptive Subgradient Method

R. L. G. Cavalcante, A. Rogers, N. R. Jennings
University of Southampton
School of Electronics and Computer Science
{rlgc,acr,nrj}@ecs.soton.ac.uk

I. Yamada
Tokyo Institute of Technology
Dept. of Communications and Integrated Systems
isao@comm.ss.titech.ac.jp

ABSTRACT

Many applications in multiagent learning are essentially convex optimization problems in which agents have only limited communication and partial information about the function being minimized (examples of such applications include, among others, coordinated source localization, distributed adaptive filtering, control, and coordination). Given this observation, we propose a new non-hierarchical decentralized algorithm for the asymptotic minimization of possibly time-varying convex functions. In our method each agent has knowledge of a time-varying local cost function, and the objective is to minimize asymptotically a global cost function defined by the sum of the local functions. At each iteration of our algorithm, agents improve their estimates of a minimizer of the global function by applying a particular version of the adaptive projected subgradient method to their local functions. Then the agents exchange and mix their improved estimates using a probabilistic model based on recent results in weighted average consensus algorithms. The resulting algorithm is provably optimal and reproduces as particular cases many existing algorithms (such as consensus algorithms and recent methods based on the adaptive projected subgradient method). To illustrate one possible application, we show how our algorithm can be applied to coordinated acoustic source localization in sensor networks.

Categories and Subject Descriptors

G.1.6 [Mathematics of Computing]: Optimization—*Convex programming*

General Terms

Algorithms, Theory

Keywords

Decentralized convex optimization, distributed computing, consensus, acoustic source localization

1. INTRODUCTION

Much of the work in multiagent learning has traditionally considered game-theoretic approaches [1], but recently it has also been acknowledged that many important problems would greatly benefit from alternatives [2]. In light of this observation, we address multiagent learning from an engineering point of view where the objective is to minimize a global collective function through local decision making [3]. Examples of applications in multiagent systems that fall within this framework are present in many seemingly distinct areas including, but not limited to, game theory [4], control [5], and signal processing [6–9]. In particular, here we consider problems where agents forming a network must optimize a global convex cost function defined by the sum of local convex functions, each of which is known by only one agent (a problem that occurs in many coordination, control and consensus settings [5–9], and in particular we consider here the problem of coordinated source localization by multiple simple range sensors [7, 8, 10]). In this setting, the main challenges faced by algorithms for convex optimization are that the agents have only partial knowledge of the global function and limited communication capabilities (i.e., not all agents can directly communicate with each other).

Given this background, there has been a great deal of effort devoted to the development of non-hierarchical iterative algorithms to handle the above convex optimization problems [5–9]. Generally speaking, these iterative schemes differ in the way that agents exchange information and improve the estimate of a minimizer of the global function.

Incremental methods where agents are activated sequentially, one at a time, have a long history in the literature [8, 9]. However, one of the major issues of incremental approaches is that many iterations are required to produce an accurate estimate of a minimizer of the global function in every agent (because only one agent is active at each iteration). In addition, acquiring a path visiting all agents in the network is often necessary, and this is challenging in large networks with sparse communication (as in the case with the sensor network we consider here).

More recently there has also been an increasing interest in algorithms where agents work asynchronously and in parallel [5–7]. They are usually faster than incremental methods and do not require complex routing schemes, but they are often analyzed by making extensive use of the assumption of simultaneous information exchange among agents, which may not be possible in every system (e.g., when agents communicate asynchronously). Furthermore, they often do not consider agents with time-varying cost functions, an important class of problems common in systems where data to build those functions arrive sequentially and a good estimate of a minimizer has to be obtained online and in real time [4, 6, 7, 11].

To address these shortcomings, we propose an algorithm that minimizes asymptotically time-varying cost functions without necessarily assuming simultaneous information exchange among

Cite as: Distributed Multiagent Learning with a Broadcast Adaptive Subgradient Method, R. L. G. Cavalcante, A. Rogers, N. R. Jennings and I. Yamada, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. . 1039-1046
Copyright © 2010, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

agents. In this algorithm, each agent first improves the estimate of the minimizer of the global function by applying the adaptive projected subgradient method [11] (which itself is an extension of Polyak's algorithm to handle time-varying cost functions) to its local function. Then the agents locally and asynchronously exchange the improved estimates using a communication model inspired by recent results in gossip consensus algorithms [12]. This model enables us to exploit the fact that in wireless systems data can be broadcast simultaneously to many agents without increasing the complexity of the system. In addition, it does not necessarily require simultaneous information exchange. To exemplify one application of the general method developed here, we derive a new algorithm for coordinated acoustic source localization. We choose this application because it has an intuitive geometrical meaning, illustrates how the limitations of recent incremental algorithms can be overcome without incorporating unnecessary heuristics (by essentially changing the communication model among agents), and shows simple techniques that can be used when many assumptions for the convergence of the proposed method in its most general form do not necessarily hold. In more detail, the main contributions of this study are as follows:

- We extend the communication model of the algorithms in [6, 7] to enable low-complexity subgradient methods to be applied to more general convex optimization problems in multiagent systems. Simultaneous information exchange among agents is not necessarily assumed and the cost functions can be time varying. We name our approach *broadcast adaptive subgradient method* and show that existing algorithms such as those in [6, 7, 12] (and many others) are particular cases of our method.
- We show conditions to guarantee that, with probability one, the agents minimize asymptotically the (time-varying) global function and agree on a minimizer.
- We evaluate our approach by using it to derive a new asynchronous algorithm for coordinated acoustic source localization. This algorithm is called *asynchronous broadcast projection onto convex sets (POCS) algorithm* and outperforms existing algorithms (e.g., the incremental POCS algorithm [8]) in terms of both convergence speed and estimation accuracy in practical scenarios without requiring complex routing schemes.

The structure of the paper is as follows. Sect. 2 outlines basic tools in convex analysis and reviews a class of problems with many applications in multiagent systems. Sect. 3 introduces and analyzes the proposed algorithm, which solves the problem in Sect. 2. In Sect. 4 we specialize the algorithm in Sect. 3 to estimate the position of acoustic sources with sensor networks.

2. PRELIMINARIES

In this section we give some definitions that will be extensively used in the discussion that follows. In particular, we denote the component of the ith row and jth column of a matrix $\boldsymbol{X}$ by $[\boldsymbol{X}]_{ij}$. 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 $\boldsymbol{y}$ of compatible size.

A set C is said to be convex if $\boldsymbol{v} = \nu\boldsymbol{v}_1 + (1-\nu)\boldsymbol{v}_2 \in C$ for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $0 \leq \nu \leq 1$. If $C \subset \mathbb{R}^N$ is a closed convex set, the metric projection $P_C : \mathbb{R}^N \to C$ is a mapping from $\boldsymbol{v} \in \mathbb{R}^N$ to the uniquely existing vector $P_C(\boldsymbol{v}) \in C$ satisfying $\|\boldsymbol{v} - P_C(\boldsymbol{v})\| = \min_{\boldsymbol{y}\in C} \|\boldsymbol{v} - \boldsymbol{y}\| =: d(\boldsymbol{v}, C)$.

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 $\forall \nu \in [0, 1]$, $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$ (in this case Θ is continuous at every point in $\mathbb{R}^N$). The *subdifferential* of a convex function $\Theta : \mathbb{R}^N \to \mathbb{R}$ at $\boldsymbol{y}$ is the nonempty closed convex set of all the *subgradients* of Θ at $\boldsymbol{y}$:

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

In the sequel, $(\Omega, \mathcal{S}, \mathcal{P})$ always denotes probability spaces, where Ω is the sure event, $\mathcal{S}$ is the σ-field of events, and $\mathcal{P}$ is the probability measure. For brevity, we will often omit the underlying probability spaces. Unless otherwise stated, we always use the Greek letter $\omega \in \Omega$ to denote a particular outcome. Thus, by $\boldsymbol{x}_\omega$ ($\boldsymbol{X}_\omega$), we denote an outcome of the random vector $\boldsymbol{x}$ (matrix $\boldsymbol{X}$). We will also often drop the qualifier "almost surely" (or "with probability 1") in equations involving random variables.

We now turn to the problem formulation. We represent a system with N agents by a network with a possibly time-varying directed graph denoted by $\mathcal{G}[i] := (\mathcal{N}, \mathcal{E}[i])$, where $\mathcal{N} = \{1, \ldots, N\}$ is the set of agents and $\mathcal{E}[i] \subseteq \mathcal{N} \times \mathcal{N}$ is the edge set [13]. The edges of the graph indicate possible communication between two agents. More precisely, if agent k can send information to agent l at time i, then $(k, l) \in \mathcal{E}[i]$ (we assume that $(k, k) \in \mathcal{E}[i]$). The inward neighbors of agent k are denoted by $\mathcal{N}_k[i] = \{l \in \mathcal{N} | \ (l, k) \in \mathcal{E}[i]\}$ (i.e., $l \in \mathcal{N}_k[i]$ are agents that can send information to agent k at time i). We assume that each agent k has knowledge of a local convex cost function $\Theta_k[i] : \mathbb{R}^M \to [0, \infty)$ ($i \in \mathbb{N}$). Note that the local cost functions $\Theta_k[i]$ are possibly time-varying and not necessarily differentiable. We define the global cost function $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ of the network by:

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

which is the function that all agents have to minimize. At time i, each agent k also has its own estimate $\boldsymbol{h}_k[i] \in \mathbb{R}^M$ of a minimizer of $\Theta[i]$ and do not know $\Theta_j[i]$ if $j \neq k$. We also require that the agents agree on a minimizer of (2), so an ideal decentralized non-hierarchical algorithm should solve:

$$\begin{aligned} &\text{minimize} \quad \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h}_k[i]) \\ &\text{subject to} \quad \boldsymbol{h}_k[i] = \boldsymbol{h}_l[i], \quad \forall k, l \in \mathcal{N}. \end{aligned} \tag{3}$$

Unfortunately, solving (3) at every time instant i is difficult if the communication among agents is limited because in such a case agents have only partial information of the problem. To solve (asymptotically) the optimization problem in (3) with low computational complexity, we add the following assumption.

ASSUMPTION 1.

At every time index i, the sets of optimizers of the local cost functions have nonempty intersection, i.e.,

$$\mathcal{O}[i] := \bigcap_{k\in\mathcal{N}} \mathcal{O}_k[i] \neq \emptyset, \tag{4}$$

where

$$\mathcal{O}_k[i] := \left\{\boldsymbol{h} \in \mathbb{R}^M \mid \Theta_k[i](\boldsymbol{h}) = \Theta^\star_k[i] := \inf_{\boldsymbol{h}\in\mathbb{R}^M} \Theta_k[i](\boldsymbol{h})\right\} \quad (k \in \mathcal{N}). \tag{5}$$

Assumption 1 is valid in many practical problems [6]. Furthermore, while in some applications Assumption 1 does not necessarily hold because, for example, of the presence of noise in a sensor measurement, we can still use the proposed method because there are simple techniques that can mitigate the effects of noise (c.f. Sect.4).

In light of Assumption 1, any $\boldsymbol{h}^\star[i] \in \mathcal{O}[i]$ is a minimizer of (2), and thus the optimization problem in (3) is solved if every agent agrees on a vector minimizing every local cost function. Based on this fact, we devise an algorithm that, with probability 1, minimizes asymptotically all local cost functions and guarantees that the agents reach asymptotic consensus. Asymptotic minimization of possibly time-varying cost functions is a common requirement in set-theoretic adaptive filtering [11] and is defined below.

DEFINITION 1. *Let $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ be a convex cost function and denote by $\boldsymbol{h}[i] \in \mathbb{R}^M$ an estimate of a minimizer of $\Theta[i]$. Assume that, for every $i \in \mathbb{N}$, there is a time-invariant scalar $\Theta^\star \in [0, \infty)$ such that $\Theta^\star = \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta[i](\boldsymbol{h})$. We say that an algorithm minimizes asymptotically $\Theta[i]$ if the algorithm produces a sequence $\boldsymbol{h}[i]$ satisfying*

$$\lim_{i \to \infty} \Theta[i](\boldsymbol{h}[i]) = \Theta^\star.$$

In turn, asymptotic consensus is mathematically expressed as [6]

$$\lim_{i \to \infty} [(\boldsymbol{I} - \boldsymbol{J})\boldsymbol{\psi}[i]] = \boldsymbol{0}, \tag{6}$$

where $\boldsymbol{\psi}[i] := [\boldsymbol{h}_1[i]^T \ \ldots \ \boldsymbol{h}_N[i]^T]^T$, $\boldsymbol{J} := \boldsymbol{B}\boldsymbol{B}^T \in \mathbb{R}^{MN \times MN}$, $\boldsymbol{B} := [\boldsymbol{b}_1 \ \ldots \ \boldsymbol{b}_M] \in \mathbb{R}^{MN \times M}$, $\boldsymbol{b}_k = (\boldsymbol{1}_N \otimes \boldsymbol{e}_k)/\sqrt{N} \in \mathbb{R}^{MN}$, $\boldsymbol{1}_N \in \mathbb{R}^N$ is the vector of ones, $\boldsymbol{e}_k \in \mathbb{R}^M$ $(k = 1, \ldots, N)$ is the standard basis vector, and $\otimes$ denotes the Kronecker product. Note that $\boldsymbol{J}$ is the orthogonal projection matrix onto the consensus subspace

$$\mathcal{C} := \operatorname{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M\}. \tag{7}$$

(If $\boldsymbol{\psi}[i] \in \mathcal{C}$, then $\boldsymbol{\psi}[i] = \boldsymbol{J}\boldsymbol{\psi}[i]$ and all local estimates $\boldsymbol{h}_k[i]$ $(k \in \mathcal{N})$ are equal, i.e., we have consensus: $\boldsymbol{h}_k[i] = \boldsymbol{h}_j[i]$ for every $k, j \in \mathcal{N}$).

3. THE LEARNING ALGORITHM

To solve (3) asymptotically, as in [6, 7], each agent k first updates $\boldsymbol{h}_k[i]$ by applying the adaptive projected subgradient method [11] to its local function $\Theta_k[i]$:

$$\boldsymbol{h}'_k[i+1] = \boldsymbol{h}_k[i] - \mu_k[i] \frac{(\Theta_k[i](\boldsymbol{h}_k[i]) - \Theta^\star_k[i])}{(\|\Theta'_k[i](\boldsymbol{h}_k[i])\|^2 + \delta_k[i])} \Theta'_k[i](\boldsymbol{h}_k[i]), \tag{8}$$

where $\boldsymbol{h}'_k[i+1]$ is the resulting estimate after the subgradient update; $\Theta'_k[i](\boldsymbol{h}_k[i]) \in \partial\Theta_k[i](\boldsymbol{h}_k[i])$ (see (1)) is a subgradient of $\Theta_k[i]$ at $\boldsymbol{h}_k[i]$; $\mu_k[i] \in [0, 2]$ is a step size; $\Theta^\star_k[i] := \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta_k[i](\boldsymbol{h})$ $(k \in \mathcal{N})$; $\delta_k[i] > 0$ is an arbitrarily small bounded number if $\Theta'_k[i](\boldsymbol{h}_k[i]) = 0$ or $\delta_k[i] = 0$ otherwise; and $\boldsymbol{h}[0]$ is an initial (deterministic) estimate of the parameter of interest.

In the second step of the algorithm, agents exchange information locally. Given a graph $\mathcal{G}[i]$, we consider agents exchanging information according to:

$$\boldsymbol{h}_k[i+1] = \sum_{j \in \mathcal{N}_k[i]} \boldsymbol{W}_{kj}[i]\boldsymbol{h}'_j[i+1], \quad k = 1, \ldots, N, \tag{9}$$

where $\boldsymbol{W}_{kj}[i] : \Omega \to \mathbb{R}^{M \times M}$ is a random weight matrix that agent k assigns to the edge (j, k) at time i ($\boldsymbol{W}_{kj}[i] = \boldsymbol{0}$ if $(j, k) \notin \mathcal{E}[i]$). The information exchange in (9) is decentralized because, as in algorithms for average consensus [12–14], each agent k needs only the estimates $\boldsymbol{h}_j[i]$ of its neighbors $j \in \mathcal{N}_k[i]$ to compute (9). Note that we can rewrite (9) in the equivalent form $[\boldsymbol{h}_1[i+1]^T \ \ldots \ \boldsymbol{h}_N[i+1]^T]^T = \boldsymbol{P}[i][\boldsymbol{h}'_1[i+1]^T \ \ldots \ \boldsymbol{h}'_N[i+1]^T]^T$, where $\boldsymbol{P}[i] : \Omega \to \mathbb{R}^{MN \times MN}$ is a matrix having $\boldsymbol{W}_{kj}[i]$ in (9) as submatrices. For the algorithm to work properly, we require that, periodically (c.f. Theorem 1), $\boldsymbol{P}[i]$ be an ϵ-broadcast consensus matrix conditioned on $[\boldsymbol{h}_1[i]^T \ldots \boldsymbol{h}_N[i]^T]^T$ as defined below.

DEFINITION 2. *For $\epsilon \in (0, 1]$, we define an ϵ-broadcast consensus matrix as a random matrix $\boldsymbol{P} : \Omega \to \mathbb{R}^{MN \times MN}$ satisfying the following properties:*

1. $\|E[\boldsymbol{P}^T(\boldsymbol{I} - \boldsymbol{J})\boldsymbol{P}]\|_2 \leq (1 - \epsilon)$;
2. $\|E[\boldsymbol{P}^T\boldsymbol{P}]\|_2 = 1$;
3. *$\boldsymbol{P}\boldsymbol{v} = \boldsymbol{v}$ for every $\boldsymbol{v} \in \mathcal{C}$ (see (7)).*

If properties 1) and 2) hold when the expectations are replaced by expectations conditioned on a random vector $\boldsymbol{r}$ (i.e., $E[\cdot]$ is replaced by $E[\cdot|\boldsymbol{r}]$), we say that $\boldsymbol{P}$ is an ϵ-broadcast consensus matrix conditioned on $\boldsymbol{r}$.

The above definition raises the question whether the agents can easily construct such matrices without global information about the network topology. Fortunately, the answer is affirmative because in Definition 2 we have used properties also satisfied by consensus matrices of existing broadcast consensus algorithms [12]. In doing so, we can now use the rich literature on consensus algorithms to build (in a decentralized way) ϵ-broadcast consensus matrices, and we show below a particular method. This method will be used later in in Sect. 4 to derive a new algorithm for acoustic sensor localization.

EXAMPLE 1. *(Random geometric graphs [12]):*
For simplicity, consider a time-invariant graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$. Assume that if two agents $k, j \in \mathcal{N}$ are within distance R from each other, then $(j, k), (k, j) \in \mathcal{E}$. Let the resulting graph be strongly connected.[1] *Suppose that only one agent sends data, and let $\boldsymbol{V}_\omega$ be a sample of a random matrix $\boldsymbol{V} : \Omega \to \mathbb{R}^{N \times N}$, where $[\boldsymbol{V}_\omega]_{kj}$ is a scalar weight assigned to the edge (j, k). With probability $1/N$, let agent $k \in \mathcal{N}$ be the agent sending data for this particular realization $\omega \in \Omega$ and define the components of $\boldsymbol{V}_\omega$ by*

$$[\boldsymbol{V}_\omega]_{jl} := \begin{cases} 1, & j \notin \mathcal{N}_k \backslash \{k\} \ \textit{ and } \ j = l \\ \gamma, & j \in \mathcal{N}_k \backslash \{k\} \ \textit{ and } \ j = l, \\ 1 - \gamma, & j \in \mathcal{N}_k \backslash \{k\} \ \textit{ and } \ l = k, \\ 0, & \text{otherwise}, \end{cases}$$

where $\gamma \in (0, 1)$ is a mixing parameter. Then the random matrix $\boldsymbol{V}$ satisfies [12]: (i) $\|E[\boldsymbol{V}(\boldsymbol{I} - 1/N \ \boldsymbol{1}_N\boldsymbol{1}_N^T)\boldsymbol{V}]\|_2 < 1$, (ii) $E[\|\boldsymbol{V}^T\boldsymbol{V}\|_2] = 1$, and (iii) $\boldsymbol{V}\boldsymbol{1}_N = \boldsymbol{1}_N$. With these properties, we can verify that $\boldsymbol{P} := \boldsymbol{V} \otimes \boldsymbol{I}_M$ is an ϵ-broadcast consensus matrix for $0 < \epsilon \leq (1 - \|E[\boldsymbol{V}(\boldsymbol{I} - 1/N \ \boldsymbol{1}_N\boldsymbol{1}_N^T)\boldsymbol{V}]\|_2)$.

We now summarize and analyze the proposed algorithm.

THEOREM 1. *(Broadcast adaptive subgradient method) Consider the problem in Sect. 2 and assume that, for every i, the random matrix $\boldsymbol{P}[i] : \Omega \to \mathbb{R}^{MN \times MN}$ satisfies properties 2) and 3) of Definition 2 with the expectations in those properties replaced*

[1] We refer the reader to [12] and the references therein for the minimum range R to guarantee a strongly connected graph with high probability.

by expectations conditioned on $\psi[i]$ ($\psi[i]$ is defined in (6)). To solve the problem in Sect. 2, we define a sequence given by

$$\boldsymbol{\psi}[i+1] = \boldsymbol{P}[i]\left(\boldsymbol{\psi}[i] - \begin{bmatrix} \mu_1[i]\alpha_1[i]\Theta_1'[i](\boldsymbol{h}_1[i]) \\ \vdots \\ \mu_N[i]\alpha_N[i]\Theta_N'[i](\boldsymbol{h}_N[i]) \end{bmatrix}\right), \tag{10}$$

where

$$\alpha_k[i] = (\Theta_k[i](\boldsymbol{h}_k[i]) - \Theta_k^\star[i])/(\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2 + \delta_k[i]).$$

(NOTE: See also the definitions after (8).) The algorithm in (10) satisfies the following:

(a) (Mean square monotone approximation):

Suppose that Assumption 1 holds, and let the step size be within the interval $\mu_k[i] \in (0,2)$ for every $k \in \mathcal{N}$. Then

$$E[\|\boldsymbol{\psi}[i+1] - \boldsymbol{\psi}^\star[i]\|^2] \le E[\|\boldsymbol{\psi}[i] - \boldsymbol{\psi}^\star[i]\|^2]$$

for every

$$\boldsymbol{\psi}^\star[i] \in C^\star[i] := \{[\boldsymbol{h}^T\ \boldsymbol{h}^T\ \ldots \boldsymbol{h}^T]^T \in \mathbb{R}^{MN} \mid \boldsymbol{h} \in \mathcal{O}[i]\}.$$

(b) (Almost sure asymptotic minimization of the local cost functions):

Assume the following:

1. *The step size is bounded away from zero and two, i.e., there exist $\epsilon_1, \epsilon_2 > 0$ such that $\mu_k[i] \in [\epsilon_1, 2-\epsilon_2] \subset (0,2)$;*
2. *$\Theta_k^\star[i] =: \Theta_k^\star \in \mathbb{R},\ i = 0,1,\ldots$;*
3. *$\mathcal{O} := \bigcap_{i\ge 0} \mathcal{O}[i] \neq \emptyset$;*
4. *$\|\Theta_k'[i](\boldsymbol{h}_k[i])\| < \infty$ for every $k \in \mathcal{N}$ and $i = 0,1,\ldots$*

Then, with probability 1, the local cost functions are asymptotically minimized, i.e.,

$$\mathcal{P}\left(\lim_{i\to\infty} \Theta_k[i](\boldsymbol{h}_k[i]) = \Theta_k^\star\right) = 1.$$

(c) (Asymptotic mean square consensus):

In addition to the assumptions above, for some fixed $\epsilon > 0$, assume the existence of $I \in \mathbb{N}$ such that, for any interval in the form $[i, i+I]$ ($i \in \mathbb{N}$), there is at least one ϵ-broadcast consensus matrix conditioned on $\boldsymbol{\psi}[i]$. Then we have asymptotic mean square consensus, i.e.,

$$\lim_{i\to\infty} E[\|(\boldsymbol{I} - \boldsymbol{J})\boldsymbol{\psi}[i]\|^2] = 0.$$

(d) (Almost sure convergence and asymptotic consensus):

If the assumptions in item (c) hold and $C^\star := \{[\boldsymbol{h}^T \cdots \boldsymbol{h}^T]^T \in \mathbb{R}^{MN} \mid \boldsymbol{h} \in \mathcal{O}\}$ does not lie in a nondegenerate hyperplane, then, with probability 1, $\boldsymbol{\psi}[i]$ converges to a random vector ψ_∞ and the agents reach consensus asymptotically.

PROOF. The proof is omitted due to the space limitation. □

Recall that (under Assumption 1) the problem in (3) is solved when the following properties are satisfied: i) every local function is minimized and ii) the agents are in consensus ($\boldsymbol{h}_1[i] = \ldots = \boldsymbol{h}_N[i]$). These two properties are satisfied asymptotically when we apply the proposed algorithm. More precisely, the local cost functions are asymptotically minimized with probability one (Theorem 1(b)) and agents reach consensus in mean square (Theorem 1(c)). In addition, under the assumptions of Theorem 1(d), agents reach consensus with probability one and their estimates $\boldsymbol{h}_k[i]$ ($k \in \mathcal{N}$) converge. Theorem 1(a) also says that, in every iteration of the algorithm, the Euclidean distance of $[\boldsymbol{h}_1[i]^T\ \ldots\ \boldsymbol{h}_N[i]^T]^T$ to a solution of (3) does not increase (in the mean square sense).

Remark 1. *(On Theorem 1)*

1. *The algorithm in Theorem 1 cannot be analyzed with the deterministic approach in [6] because the mapping $T : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ defined by $T(\boldsymbol{\psi}) = \boldsymbol{P}_\omega[i]\boldsymbol{\psi}$ is not necessarily nonexpansive, i.e.,*

   $$\|T(\boldsymbol{x}) - T(\boldsymbol{y})\| \le \|\boldsymbol{x} - \boldsymbol{y}\|$$

   does not necessarily hold for every $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{MN}$ (see Example 1).

2. *All assumptions in Theorem 1 automatically hold when $\Theta_k[i](\boldsymbol{h}) = 0$, in which case we reproduce conventional consensus algorithms (e.g., those in [12]).*

3. *(Asynchronous updates) Let the assumptions in Theorem 1(c) hold. Suppose that the agents do not have a common clock, so they asynchronously apply subgradient updates (the updates in (8)). In addition, assume that information exchange is also performed asynchronously. Theorem 1 can be used to analyze such an algorithm as follows. Let $i \in \mathbb{N}$ be the time instants where there is at least one subgradient update or information exchange among agents. Denote by $\mathcal{I}_k \subset \mathbb{N}$ an infinite set of time instants where agent k applies a subgradient update. We can consider that the sequence of functions $\Theta_k[i]$ is only defined at time instants $\mathcal{I}_k \subset \mathbb{N}$, and, in (10), agent k is using the extended local function*

   $$\widetilde{\Theta}_k[i](\boldsymbol{h}) = \begin{cases} \Theta_k[i](\boldsymbol{h}), & i \in \mathcal{I}_k \\ \Theta_k^\star & \text{otherwise.} \end{cases}$$

   Similarly, suppose that agents only exchange information at time instants $n \in \mathcal{I}_{\boldsymbol{P}} \subset \mathbb{N}$ using i.i.d. random matrices $\boldsymbol{P}[n]$, where $\mathcal{I}_{\boldsymbol{P}}$ is also an infinite set. We can consider that (10) is using the random matrix

   $$\widetilde{\boldsymbol{P}}[i] = \begin{cases} \boldsymbol{P}[i], & i \in \mathcal{I}_{\boldsymbol{P}} \\ \boldsymbol{I} & \text{otherwise.} \end{cases}$$

   With the above extensions, $\Theta_k[i](\boldsymbol{h}[i])$ is a subsequence of $\widetilde{\Theta}_k[i](\boldsymbol{h}[i])$, and the convergence of $\widetilde{\Theta}_k[i](\boldsymbol{h}[i])$ to $\Theta_k^\star$ (we can use Theorem 1 to reach this conclusion) also implies the convergence of $\Theta_k[i](\boldsymbol{h}[i])$ to $\Theta_k^\star$. Sect. 4 shows a concrete application based on this idea.

4. *(Adding constraints) Constraints can also be easily added by considering time-varying cost functions. For example, with the assumptions in Theorem 1(b), let $\Theta_k : \mathbb{R}^M \to [0,\infty)$ be a (fixed) cost function known by agent k. Suppose that the agent has knowledge of a set C such that $\mathcal{O} \subset C$. Then we can use the following time-varying cost-function instead of the original function $\Theta_k : \mathbb{R}^M \to [0,\infty)$:*

   $$\Theta_k[i](\boldsymbol{h}) = \begin{cases} \Theta_k(\boldsymbol{h}), & i \text{ odd} \\ d(\boldsymbol{h}, C) + \Theta_k^\star & i \text{ even,} \end{cases}$$

4. COORDINATED ACOUSTIC SOURCE LOCALIZATION

We now specialize the method in Theorem 1 to localize acoustic sources with sensor networks. (The new algorithm for source localization is called *asynchronous broadcast POCS algorithm*.) However, note that our method is general and can be applied to many other problems (e.g., coordination, distributed adaptive filtering, etc.).

4.1 Problem description and existing solutions

The objective is to estimate the unknown location $\boldsymbol{r}^\star \in \mathbb{R}^2$ of an acoustic source by using N agents distributed at spatial locations $\boldsymbol{r}_k \in \mathbb{R}^2$ ($k = 1, \ldots, N$). Each agent knows its own position $\boldsymbol{r}_k$, the acoustic source power A, and is equipped with an acoustic sensor that can estimate the range of the acoustic source from the received volume (but not the direction).[2] The acoustic power perceived by agent k can be modeled as [10]

$$y_k = \frac{A}{\|\boldsymbol{r}_k - \boldsymbol{r}^\star\|^2} + n_k, \tag{11}$$

where n_k is noise. For mathematical simplicity, n_k is often modeled as Gaussian noise even though this assumption is unrealistic because y_k is always positive. Nonetheless, algorithms using this unrealistic assumption often gives good performance when deployed in real-world scenarios [10]. By modeling noise as Gaussian, the maximum-likelihood estimate $\boldsymbol{r}_{\rm ML}$ is given by [10]

$$\boldsymbol{r}_{\rm ML} \in \arg\min_{\boldsymbol{h}\in\mathbb{R}^2} \sum_{k=1}^{N} \left[y_k - \frac{A}{\|\boldsymbol{r}_k - \boldsymbol{h}\|^2} \right]^2. \tag{12}$$

Unfortunately, many simple decentralized algorithms used to approximate $\boldsymbol{r}_{\rm ML}$ with low complexity, such as the incremental gradient method, may not provide an estimate close to $\boldsymbol{r}_{\rm ML}$ because the function being minimized is nonconvex. By noticing that each term in the summation in (12) attains its minimum on the circle $C_k := \{\boldsymbol{h} \in \mathbb{R}^2 \mid \|\boldsymbol{h} - \boldsymbol{r}_k\| = \sqrt{A/y_k}\}$, the optimization problem in (12) can be replaced by the alternative convex optimization problem [8]

$$\boldsymbol{r}_{\rm opt} \in \arg\min_{\boldsymbol{h}\in\mathbb{R}^2} \sum_{k=1}^{N} d(\boldsymbol{h}, D_k), \tag{13}$$

where D_k is a convex relaxation of the set C_k: $D_k := \{\boldsymbol{h} \in \mathbb{R}^2 \mid \|\boldsymbol{h} - \boldsymbol{r}_k\| \le \sqrt{A/y_k}\}$. When noise is not present, the solution set to the optimization problem in (13) is $\cap_{k=1}^N D_k \ni \boldsymbol{r}^\star$. If the acoustic source position $\boldsymbol{r}^\star$ lies in the convex hull of the agents' locations, i.e., $\boldsymbol{r}^\star \in H$ where

$$H = \left\{ \boldsymbol{r} \in \mathbb{R}^2 \mid \boldsymbol{r} = \sum_{k=1}^{N} \alpha_k \boldsymbol{r}_k,\ \alpha_k \ge 0,\ \sum_{k=1}^{N} \alpha_k = 1 \right\}, \tag{14}$$

then the unique point in the set $\cap_{k=1}^N D_k$, the solution to the problem in (13), is $\boldsymbol{r}^\star = \boldsymbol{r}_{\rm opt}$ [8]. The incremental POCS algorithm [8] can thus be used to solve (13) in this scenario. This algorithm is a sequential method that can be summarized as follows. In the initialization stage, the algorithm defines a cyclic path visiting all agents in the system. Then agent k in the path becomes active, improves its estimate of the source location by projecting this estimate onto the sphere D_k, and then sends the new estimate to only the next agent in the path, which repeats the process. Unfortunately, the acquisition of a path visiting all agents is a difficult task in large networks or in dynamic scenarios where links can fail, thus the application of the incremental POCS algorithm is limited to small or medium sized networks. In addition, owing to the sequential nature of the algorithm, agents at the end of the path have to wait a long time for an accurate estimate of the acoustic source position.

When noise is present, $\boldsymbol{r}_{\rm opt}$ is not necessarily unique and $\boldsymbol{r}^\star$ may not be a solution to (13). However, if the number of agents is sufficiently large, we can expect that $\boldsymbol{r}_{\rm opt}$ is a good approximation of $\boldsymbol{r}^\star$. The incremental POCS algorithm does not necessarily converge to $\boldsymbol{r}_{\rm opt}$ in such cases [15], but after some iterations the sequence of estimates generated by this algorithm is close to $\boldsymbol{r}^\star$ in this particular application. In addition, many simple heuristics, such as monotonically decreasing the step size after a given number of iterations, can further improve the performance [8]. More generally, in different applications where the POCS algorithm is applied, a simple method to mitigate the detrimental effects of noise consists of using a small (fixed) step size [15].

4.2 Asynchronous broadcast POCS algorithm

Owing to the nature of wireless channels, if agent k broadcasts an estimate $\boldsymbol{h}_k[i]$, all other agents within a certain distance are able to receive this information. However, in the incremental POCS algorithm, even though more than one agent may be able to receive $\boldsymbol{h}_k[i]$, only the next agent in the cycle uses this available information. To avoid this loss of useful data in the system, we derive an algorithm that uses the communication model in Example 1. In doing so, not only do agents not discard useful data, but they also do not need to acquire a path visiting all agents. We start with the following assumption, previously used in Example 1.

ASSUMPTION 2. *The graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is static and strongly connected. In addition, agents within distance R from a given agent k can receive data transmitted by agent k (and vice versa), i.e., $(k, j), (j, k) \in \mathcal{E}$ if $\|\boldsymbol{r}_k - \boldsymbol{r}_j\| \le R$.*

In the proposed algorithm, at iteration i and with probability $1/N$, agent $m \in \mathcal{N}$ is activated and other agents remain idle (in practice this can be easily done with agents having independent clocks ticking according to a Poisson process [12]). Therefore, only agent m is able to apply the iterations in (8), thus we assume that the agents are minimizing asymptotically [3]

$$\Theta_k[i](\boldsymbol{h}) = \begin{cases} 0, & \text{if} \quad k \neq m \\ d(\boldsymbol{h}, D_k[i]), & \text{otherwise}, \end{cases} \tag{15}$$

where

$$D_k[i] := \begin{cases} \mathbb{R}^2, & \text{if } \ y_k \le c_k[i] \\ \left\{ \boldsymbol{h} \in \mathbb{R}^2 \mid \|\boldsymbol{h} - \boldsymbol{r}_k\| \le \sqrt{\dfrac{A}{y_k - c_k[i]}} \right\}, & \text{otherwise}, \end{cases}$$

and $c_k[i] \ge 0$ is a (possibly time-varying) parameter that increases the radius of the sphere D_k used in the optimization problem (13) (note that, by expanding the spheres, we also increase the probability that $\boldsymbol{r}^\star \in D_k[i]$ when noise is present). (Artificial expansion of sets is a common technique to mitigate the effects of noise in set-theoretic filtering [11].) If $\cap_{k\in\mathcal{N}} D_k[i] \neq \emptyset$, any point

[2]We can use the same techniques developed in [8] to extend the proposed algorithm to the case where A is unknown. For brevity, we do not consider such extensions here.

[3]This idea is based on Remark 1.3. Similar time-varying cost functions have also been used in [7], but, as shown in Remark 1.1, the resulting algorithm cannot be analyzed with the theory in [7] because of the communication model.

is this intersection of sets is a minimizer of the global cost function $\Theta[i](\boldsymbol{h}) = \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h})$. Thus, similarly to the incremental POCS algorithm, the main idea of the proposed method is to find a point in $\cap_{k\in\mathcal{N}} D_k[i]$, which ideally should have a small area and include the source location $\boldsymbol{r}^\star$. If we ignore noise for the moment and set $c_k[i]$ to 0, the global function $\Theta[i]$ is guaranteed to be minimized at any time index i only at $\boldsymbol{r}^\star$, where we also have $\Theta_k[i](\boldsymbol{r}^\star) = \Theta_k^\star[i] = 0$ (see the discussion after (13)). Note that a subgradient of $\Theta_k[i](\boldsymbol{h}) = d(\boldsymbol{h}, D_k[i])$ is [11]

$$\partial\Theta_k[i](\boldsymbol{h}) \ni \Theta_k'[i](\boldsymbol{h}) = \begin{cases} \dfrac{\boldsymbol{h} - P_{D_k[i]}(\boldsymbol{h})}{d(\boldsymbol{h}, D_k[i])} & \text{if} \quad \boldsymbol{h} \notin D_k[i], \\ \boldsymbol{0} \quad \text{otherwise}, \end{cases} \tag{16}$$

where [15]

$$P_{D_k[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h}, & \text{if} \quad \boldsymbol{h} \in D_k[i] \\ \boldsymbol{h} + \sqrt{\dfrac{A}{y_k - c_k[i]}} \dfrac{(\boldsymbol{h} - \boldsymbol{r}_k)}{\|\boldsymbol{h} - \boldsymbol{r}_k\|} & \text{otherwise.} \end{cases}$$

After applying the iteration in (8) with the local functions in (15), agent m broadcasts its improved estimate of the source location. Then all agents able to receive this information (i.e., those within distance R from agent k) mix their estimates with the received estimate by using an ϵ-broadcast consensus matrix $\boldsymbol{P}[i]$ constructed with the scheme in Example 1. (NOTE: In the construction of $\boldsymbol{P}[i]$, agents not able to receive $\boldsymbol{h}_k[i]$ can remain idle because they do not mix estimates. The matrices $\boldsymbol{P}[i]$ $(i = 0, 1, \ldots)$ are i.i.d. and independent of $\boldsymbol{\psi}_k[n]$ for every n.) The whole process is repeated with a new active agent.

We summarize below our method for coordinated acoustic source localization, which can be shortly described as the application of the local cost functions in (15) and the ϵ-broadcast matrices in Example 1 to the scheme in Theorem 1 with $\Theta_k^\star[i] = 0$ (i.e., we ignore the presence of noise).

ALGORITHM 1. *(Asynchronous broadcast POCS algorithm):*

1. *Initialize the estimates $\boldsymbol{h}_k[i]$ with an arbitrary $\boldsymbol{h}_k[0] \in \mathbb{R}^2$.*
2. *Only agent $m \in \mathcal{N}$ becomes active (agents have the same probability $1/N$ of becoming active).*
3. $\boldsymbol{h}_m[i+1] = \boldsymbol{h}_m[i] - \mu_m[i]\left(P_{D_m[i]}(\boldsymbol{h}_m[i]) - \boldsymbol{h}_m[i]\right)$, *where $\mu_m[i] \in (0, 2)$ is the step size.*
4. *Agent m broadcasts $\boldsymbol{h}_m[i+1]$*
5. *Agents $j \in \mathcal{N}_m \backslash \{m\}$ (i.e, all agents within distance R to agent m) mix the received estimate $\boldsymbol{h}_m[i+1]$ with their own estimates $\boldsymbol{h}_j[i]$:*

   $$\boldsymbol{h}_j[i+1] = \gamma\boldsymbol{h}_j[i] + (1-\gamma)\boldsymbol{h}_m[i+1], \quad j \in \mathcal{N}_m \backslash \{m\},$$

   where $\gamma \in (0, 1)$ is a mixing parameter common to all agents. (Agents $k \notin \mathcal{N}_m$ do not perform any operations, so we can consider that $\boldsymbol{h}_k[i+1] = \boldsymbol{h}_k[i]$.)
6. *Increment i and go to step 2.*

Note that Algorithm 1 requires neither synchronization nor agents to be aware of their neighbors. Simultaneous information exchange among agents, a common assumption in previous decentralized optimization using subgradient methods [5–7], is also not necessary. In our approach, agents randomly become active, improve their estimates $\boldsymbol{h}_k[i]$, and broadcast $\boldsymbol{h}_k[i]$ to all other agents within range R. We can analyze Algorithm 1 directly with Theorem 1 in the absence of noise for the following reasons:

1. The subgradients are bounded (see (16)).
2. The set $\mathcal{O}$ (as defined in Theorem 1(b)) is nonempty because $\Theta_k[i](\boldsymbol{r}^\star) = 0$ for every $k \in \mathcal{N}$ and $i \in \mathbb{N}$ (i.e., $\boldsymbol{r}^\star \in \cap_{k\in\mathcal{N}} D_k$).
3. At every iteration the algorithm uses samples of an ϵ-broadcast matrices constructed with the method in Example 1 ($\boldsymbol{P}[i]$ and $\boldsymbol{\psi}[i]$ are independent).

Therefore, all conditions of Theorem 1(a)-(c)[4] can be easily satisfied by simply choosing step sizes $\mu_k[i]$ bounded away from 0 and 2.

The assumptions in Theorem 1 do not necessarily hold in the presence of noise, but nevertheless we can apply the same ideas used to mitigate the effects of noise in other POCS-based algorithms or set-theoretic adaptive filters. Here we choose to expand the parameters $c_k[i]$, $k \in \mathcal{N}$. Ideally, these parameters should be small real numbers so that $\boldsymbol{r}^\star \in D_k[i]$ and the area of $\cap_{k\in\mathcal{N}} D_k[i]$ is small. Unfortunately, computing such values is not possible, but we can slowly increase the radius of the sphere $D_k[i]$ every time node k is activated. Intuitively, if the convergence of the algorithm is faster than the increase rate of the sphere $D_k[i]$, we can expect that, once $\cap_{k\in\mathcal{N}} D_k[i]$ is nonempty, all agents soon find a point in $\cap_{k\in\mathcal{N}} D_k[i]$ and stay in this point, which is assumed to be a good approximation of $\boldsymbol{r}^\star$. This approach has also been successfully used by algorithms using the assumption of simultaneous information exchange [7].

4.3 Numerical simulations

We evaluate the performance of the asynchronous broadcast POCS algorithm in settings almost identical to those in which the original incremental POCS algorithm was evaluated [8, Sect. V]. In a $100\text{m} \times 100\text{m}$ field, at each realization of the simulation we randomly distribute 5000 agents and place an acoustic source with $A = 100$ at $\boldsymbol{r}^\star = [50\ 50]^T$. Each agent measures the acoustic power at their own locations according to (11). The noise n_k is modeled as Gaussian with variance $\sigma_k = 1$, and only agents with perceived power greater or equal than 5 (i.e., $y_k \geq 5$) take part in the estimation task. Each agent in the estimation task has a uniquely identifying number from the set $\{1, \ldots, N\}$.

We compare the incremental POCS algorithm with different versions of the proposed broadcast POCS algorithm. We do not show the performance of the maximum likelihood estimator because (12) is a nonconvex optimization problem, and algorithms dealing with nonconvex functions usually have poor performance if they are not initialized with a point close to the unknown acoustic source location [8].

To construct the sequence of agents $[s_0\ \ldots\ s_{N-1}]$ $(s_k \in \mathcal{N})$ for the incremental POCS algorithm, we start with $s_0 = 1$ and set s_{k+1} to be the nearest agent to s_k that has not been previously selected (i.e., $s_{k+1} \neq s_l\ l = 0, \ldots, k$). All agents use the same step size $\mu_k[i] = 0.4$ in the incremental POCS algorithm. Table 1 shows the parameters used by the proposed broadcast POCS algorithm. In this table, $a_k[i]$ is the number of times that agent k has been activated up to time index i.

The performance of interest is the mean square error (MSE) normalized by the number of agents N (because N is a random

[4]The results in Theorem 1 are valid for any sequence of functions obtained in one realization of the algorithm.

Table 1: Parameters used by the different versions of the broadcast POCS algorithm. For each version of the algorithm, all agents use the same values for $\mu_k[i]$, R, and γ.

Name	$\mu_k[i]$	$c_k[i]$	R	γ
Broadcast POCS-a	0.4	0	5	0.5
Broadcast POCS-b	0.4	0	7	0.5
Broadcast POCS-c	1	$0.1\ a_k[i]$	5	0.5
Broadcast POCS-d	1	$0.1\ a_k[i]$	7	0.5

variable in the simulation)

$$\mathrm{MSE}[i] = E\left[\frac{1}{N}\sum_{k=1}^{N}\|\boldsymbol{h}_k[i] - \boldsymbol{r}^\star\|^2\right].$$

We also show the mean square distance to consensus (MSDC) normalized by the number of agents, defined by

$$\mathrm{MSDC}[i] = E\left[\frac{1}{N}\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]\|^2\right].$$

(NOTE: $\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]\|$ is the distance of $\boldsymbol{\psi}[i]$ to the consensus subspace $\mathcal{C}$ defined in (7). When $\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]\|$ is zero, all agents are in consensus.)

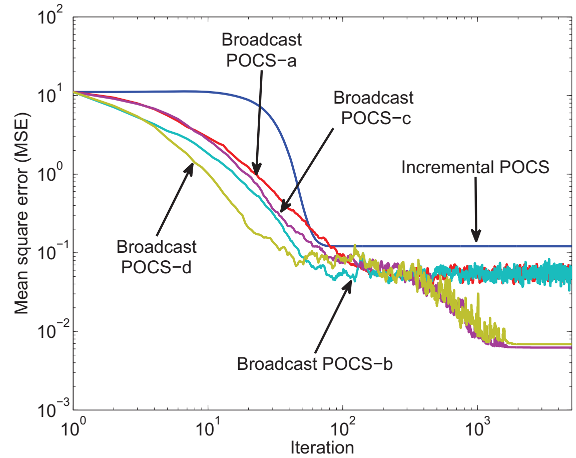
We compute expectations by averaging the results of 100 realizations of the simulation, which, as shown in the figures in this section, is enough for statistical significance (because the curves of the algorithms are smooth enough to draw conclusions on the relative performance of the algorithms).

Fig. 1 shows the results. The convergence speed of the broadcast POCS algorithm increases as a function of R when other parameters are kept constant because fewer iterations are needed to propagate indirectly the information of every agent through the network. We also see that slowly increasing the radius of $D_k[i]$ (by increasing $c_k[i]$) is an efficient method to improve the steady-state performance. The choice of $c_k[i]$ in Table 1 is intuitively appealing because agents with low signal-to-noise rate (SNR), usually those with small y_k, stop their unreliable subgradient updates in few iterations (because the relation $y_k < c_k[i]$ is often satisfied in a short period of time). Subgradient updates last longer in agents with high SNR, which improves the quality of the estimate. In addition, by increasing $c_k[i]$, the probability that the intersection $\cap_{k\in\mathcal{N}} D_k[i]$ is nonempty increases with time, and the assumptions in Theorem 1 are more likely to be satisfied. This fact is observed experimentally by noticing that:

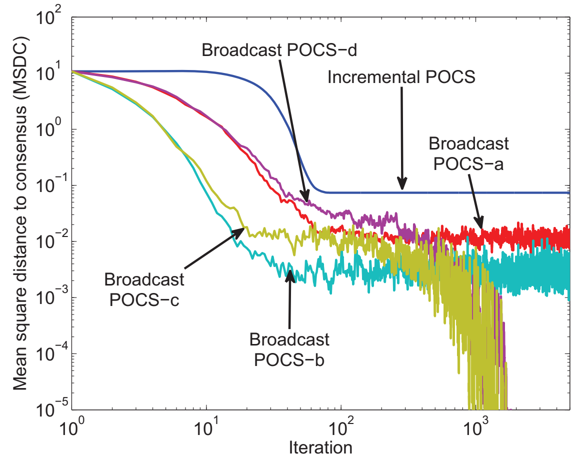
1. In Fig. 1(a), the MSE fluctuations in the curves of versions (c) and (d) of the broadcast POCS algorithm eventually cease when the expectation is computed by averaging the results of only 100 realizations of the simulation. This is an indication that the estimates of all agents are converging in all runs of the simulation for these versions of the algorithm, a fact predicted by Theorem 1(d).

2. The MSDC of versions (c) and (d) of the broadcast POCS algorithm is converging to 0 (see Fig. 1(b)), another indication that the conditions of Theorem 1(d) have been satisfied in all runs of the simulation for these versions of the algorithm.

Care should be taken in the choice of the parameters $c_k[i]$. If the radii of the spheres $D_k[i]$ grow too fast, the subgradient updates cease too soon in every agent, and the steady-state performance



(a)



(b)

Figure 1: Transient performance of the algorithms. (a) Mean square error. (b) Mean square distance to consensus.

decreases. This fact is illustrated in Fig. 2, where we can see that the steady-state performance of the algorithm decreases as $D_k[i]$ increases more quickly. If a typical scenario for the application of the algorithm cannot be defined, in which case a good choice of $c_k[i]$ cannot be obtained by means of simulations, we can fix $c_k[i] = 0$ as done with versions (a) and (b) of the broadcast POCS algorithms. Even though many assumptions of Theorem 1 do not hold in these versions of the algorithm (because of the presence of noise), as shown in Fig. 1, the convergence speed and steady-state performance are still satisfactory. We can also devise schemes where $c_k[i]$ is chosen automatically by each agent, but these approaches will not be investigated here.

The results of the incremental POCS algorithm should be used only as a rough reference of its achievable performance because we have only used a small fixed step size to mitigate the effects of noise. Techniques to improve further the performance of this algorithm in this particular application domain are out of the scope of this study. However, being an incremental method where

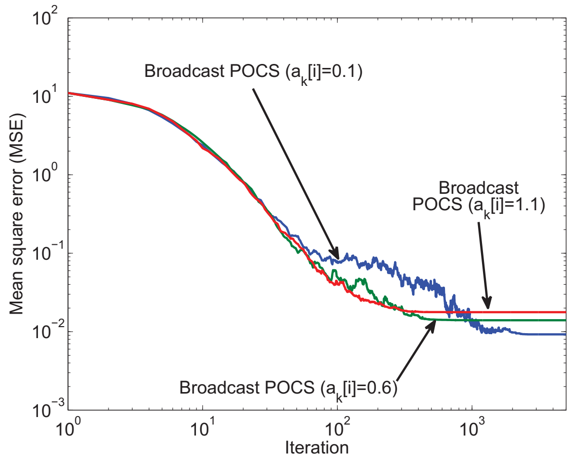


Figure 2: Transient performance of the broadcast POCS algorithm as a function of $a_k[i]$. Except for $a_k[i]$, the parameters of the algorithms are identical to those of the Broadcast POCS-d algorithm in Table 1.

estimates are broadcast to only one neighbor at each iteration, the incremental POCS algorithm always exhibits the slow start observed in Fig. 1 because at least one cycle is necessary to improve the estimate of every agent in the network. In addition, it also requires the definition of a path visiting all agents taking part in the estimation task, thus a complex routing scheme is necessary. Therefore, if the transmission range is long enough, our broadcast POCS algorithm is a good alternative because it is fast, has good steady-state performance, and does not require the acquisition of a path visiting all agents in the estimation task.

5. CONCLUSIONS

We have developed a non-hierarchical algorithm for decentralized optimization of a sum of convex functions. Each term in this sum is a local cost function known by an agent in a network, and we assume that the sets of optimizers of the local functions have nonempty intersection. Unlike existing optimization methods, the local cost functions can be time-varying, a very useful property in online learning, and agents exchange information locally by using recent results in broadcast average consensus algorithms. This mechanism for information exchange enable us to relax the assumption of simultaneous exchange information among agents, a common assumption in the analysis of multiagent algorithms using subgradient methods. We have shown conditions to guarantee almost sure asymptotic minimization of the local cost functions, consensus among agents, and convergence.

As an example of a possible application, we used the proposed method in its most general form to derive a new POCS-based algorithm for acoustic sensor localization. This application shows efficient techniques to apply the general method even when many assumptions do not necessary hold. Our algorithm for source localization and the existing incremental POCS algorithm are similar, but the former usually has better convergence speed because data is simultaneously transmitted to multiple agents in every iteration. In addition, our algorithm also does not require the acquisition of path visiting all agents in the network.

Future work may include the study of mobile agents with heterogeneous sensors, which is possible with the proposed method because it is a very general optimization tool that can handle time-varying cost functions.

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.

6. REFERENCES

[1] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, May 2007.

[2] P. Stone, "Multiagent learning is not the answer. It is the question." *Artificial Intelligence*, vol. 171, no. 7, pp. 402–405, May 2007.

[3] S. Mannor and J. S. Shamma, "Multi-agent learning for engineers," *Artificial Intelligence*, vol. 171, no. 7, pp. 417–422, May 2007.

[4] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. of the 20th International Conference on Machine Learning (ICML)*, 2003.

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

[6] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2762–2774, July 2009.

[7] ——, "Learning in diffusion networks with an adaptive projected subgradient method," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 2853–2856.

[8] 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.

[9] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, Feb. 2007.

[10] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 53, no. 1, pp. 44–53, Jan. 2005.

[11] 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.

[12] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2748–2761, July 2009.

[13] 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.

[14] C.-H. Yu and R. Nagpal, "Sensing-based shape formation on modular multi-robot systems: A theoretical study," in *Proc. of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, May 2008, pp. 71–78.

[15] H. Stark and Y. Yang, *Vector Space Projections – A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. New York: Wiley, 1998.