Distributed averaging on digital noisy networks

Ruggero Carli [a], Giacomo Como [b], Paolo Frasca [c], Federica Garin [d]

[a] *University of California at Santa Barbara, Santa Barbara, CA 93106*

[b] *Massachusetts Institute of Technology, Cambridge, MA 02139 USA*

[c] *DIIMA, Università di Salerno, and Istituto per le Applicazioni del Calcolo (CNR), Roma, Italy*

[d] *DEI, Università di Padova, Italy*

Abstract

Iterative distributed algorithms are studied for computing arithmetic averages over networks of agents connected through digital noisy broadcast channels. These algorithms do not require the agents to have global knowledge of the network structure or size. Almost sure convergence to state agreement is proved, and the communication and computational complexities of the algorithms are analyzed. Both the number of transmissions and computations performed by each agent of the network are shown to grow poly-logarithmically in the desired precision. The impact of the graph topology on the algorithms' performance is analyzed as well. Moreover, it is shown how, in the presence of noiseless communication feedback, one can modify the algorithms, significantly improving their performance vs complexity tradeoff. Finally, simulations are presented confirming the theoretical results and suggesting that the presented algorithms may outperform algorithms based on decreasing gains recently proposed in the literature.

Key words: Distributed computation; distributed control; digital communications; control with communication constraints; average consensus; graphs.

1 Introduction

As large-scale networks have emerged –characterized by the lack of centralized access to information, and possibly time-varying topologies–, the last few years have witnessed an increasing research interest in problems of distributed computation. In many scenarios of current applicative interest, large collections of identical anonymous agents – each having access to some partial information– aim at computing an application-specific function of the global information [31,26,18]. As the network has to be reconfigurable and scalable, the computation must be completely distributed, i.e. each agent can rely only on local observations, while iteratively processing the available information and communicating with the other agents. The main challenge in the design of such distributed computation systems is posed by the scarce energetic autonomy of the agents, which severely constrains both their computational and communication capabilities. A special instance, which has been the object of recent extensive work, is the average consensus problem, in which a large number of agents aims at computing –in a distributed fashion– a value coinciding with the arithmetic average of some initial scalar measurements. While most of the literature on consensus algorithms has modeled communication constraints in the average consensus algorithm by a communication graph in which a link between two nodes is assumed to support the noise-free transmission of a real value, there is a clear demand for more realistic communication models. In fact, some recent work has addressed the cases of quantized data or computations [22,27,5], quantized communication [3,20,15], packet losses [14], or transmission affected by additive noise [19,28,21]. However, to the best of our knowledge, there is no contribution yet toward the design of consensus

⋆ A preliminary version of this work has appeared as [8]. Corresponding author P. Frasca.

Email addresses: carlirug@engineering.ucsb.edu (Ruggero Carli), giacomo@mit.edu (Giacomo Como), paolo.frasca@gmail.com (Paolo Frasca), garin@dei.unipd.it (Federica Garin).

algorithms on networks in which the communication links are modeled as digital noisy channels. The latter models of communication are particularly significant as in practice bandwidth limitations imply that the channels have finite capacity. For such digital noisy networks, information-theoretic bounds on the performance of distributed computation algorithms have been established in [2,11]. More general problems of distributed computation have been considered, for instance, in [16,34,1,18].

In the present paper, we study iterative distributed averaging algorithms for networks whose nodes can communicate through digital noisy broadcast channels. In order to compare the performance of different algorithms, we define suitable complexity measures, which account for the number of channel transmissions (*communication complexity*), and, respectively, of in-node computations (*computational complexity*) required to achieve a desired precision. [1] These performance measures are particularly relevant, as they allow to directly estimate the energy consumption of such distributed computation systems, as well as their time-complexity.

The algorithms proposed in this paper combine the classical iterative linear consensus algorithm with coding schemes for the reliable transmission of real numbers on noisy channels, recently proposed in [12]. They involve a sequence of transmission phases, of increasing duration, in which the agents attempt to broadcast their state, i.e. their current estimate of the global average, to their neighbors, alternated to averaging steps, in which the agents' states are updated. These algorithms are fully distributed, and they do not require the agents to have any global knowledge of the network structure or size. Our main result –stated in Theorem 4– shows that such algorithms drive the agents to state agreement –or consensus– which can be made arbitrarily close to the true average. The number of channel transmissions and in-node computations is shown to grow at most poly-logarithmically in the desired precision.

We also show how communication feedback, when available, allows one to modify the algorithms, achieving asymptotic average consensus (i.e. state agreement on the average of the initial observations), and reducing the computational and communication complexities –see Theorem 5. Scaling properties as the number of agents grows are investigated as well, and the computational and communication complexities are related to spectral properties of the underlying communication graph. Moreover, simulations are presented, suggesting that the performance of our algorithms may in practice be better than that guaranteed by their theoretical analysis.

General results on coding for interacting communication [30,29] may suggest logarithmic communication complexity to possibly be achievable by embedding an efficient quantized consensus algorithm [7,24] in a global error correcting coding scheme. However, as it has also been argued in [17], the tree-code constructions proposed in [30,29] suffer from high computational complexity which likely prevents their practical implementation. Moreover, their global design requires each agent to have knowledge of the whole network topology, an assumption which contrasts the reconfigurability requirements. Instead, the algorithms we shall present in this paper do not require the agents to have any global knowledge of the network topology, and both their communication and computational complexity can be kept tractable.

Finally, it is worth comparing the performance of our algorithms with that of two other approaches, which apply different ideas from the literature. Indeed, depending on the application, one might prefer not to increase the duration of the communication phases: it is argued in Sect. 4.3 that this would lead to accumulation of errors and poor performance, unless communication feedback is available. For this reason, one might try to keep the phases' lengths fixed, and to compensate the accumulation of errors by applying the decreasing gain strategy studied in [19,28,21] for networks whose links support the transmission of a real number, affected by additive noise. Indeed the effect of errors in digital communications can be thought of as noise. Compared to ours, the results in [19,28,21] show almost sure convergence to average consensus, with mean square error decreasing as the inverse of time, under slightly more stringent assumptions on the noise (mainly, independence of the additive noise). This guarantees communication and computation complexities growing polynomially in the desired precision, as opposed to the polylogarithmic dependence of our algorithms. These remarks are confirmed by simulation results for these families of algorithms, presented in Sect. 6.

The remainder of this paper is organized as follows. In Sect. 2 we formally state the problem, introduce the relevant performance measures, and revise some results on the transmission of continuous information through digital noisy channels. In Sect. 3 we present our algorithms and present the main convergence results, as well as bounds on their performance. In Sect. 4, we discuss how to efficiently modify our algorithms in the presence of communication feedback. Sect. 5 analyzes the scaling properties of our algorithms in the network size, for some network families of interest. Simulations are presented in Sect. 6, comparing our algorithm with the fixed phase length strategies.

[1] Related measures to evaluate distributed algorithms have been proposed in various settings: see, for instance, [23,25,13].

Before proceeding, let us establish some notational conventions to be used throughout the paper. We shall denote by $\mathbb{N}$, $\mathbb{Z}^+$, and $\mathbb{R}$, respectively, the sets of naturals, nonnegative integers, and real numbers. The set of the smallest t naturals will be denoted by $[t] := \{1, 2, \dots, t\}$. The transposes of a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a matrix $M \in \mathbb{R}^{n\times n}$, will be denoted by $\boldsymbol{v}^*$ and M^*, respectively. Given two matrices M, M', we denote by $M \odot M'$ their entrywise (Hadamard) product. With the symbol $\mathbf{1}$ we denote the n-dimensional vector all of whose entries equal 1. Given a set $\mathcal{V}$ of finite cardinality, $|\mathcal{V}| = n$, we denote by $\mathbb{R}^{\mathcal{V}}$ and $\mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ the vector spaces, isomorphic to $\mathbb{R}^n$ and $\mathbb{R}^{n\times n}$ respectively, where entries of the vectors and matrices are labeled by elements of $\mathcal{V}$ instead of numbers $1, \dots, n$. A (directed) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the pair of a finite vertex set $\mathcal{V}$ and of a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of directed edges. For a vertex $v \in \mathcal{V}$, we shall denote by $\mathcal{N}_v^+ := \{w \in \mathcal{V} : (v, w) \in \mathcal{E}\}$, and $\mathcal{N}_v^- := \{w \in \mathcal{V} : (w, v) \in \mathcal{E}\}$, respectively, the sets of its out- and in-neighbors. Given a matrix $M \in \mathbb{R}^{n\times n}$, we define the induced graph $\mathcal{G}_M$ by taking a set $\mathcal{V}$ of n nodes, identifying $\mathbb{R}^{n\times n}$ with $\mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ and putting an edge (j, i) in $\mathcal{E}$ if $i \neq j$ and $M_{ij} > 0$. A matrix M is adapted to a graph $\mathcal{G}$ if $\mathcal{G}_M$ is a subgraph of $\mathcal{G}$.

2 Problem setting

In this section, we present a formal statement of the problem, introduce the main performance measures, and gather some recently appeared results on the reliable point-to-point transmission of continuous information through digital noisy channels.

2.1 Problem statement

We shall consider a finite set $\mathcal{V}$ of $|\mathcal{V}| = n$ agents, and assume that each agent $v \in \mathcal{V}$ has access to some partial information consisting in the observation of a scalar value θ_v. The full vector of observations will be indicated by $\boldsymbol{\theta} = (\theta_v)_{v\in\mathcal{V}}$. We shall consider the case when all θ_v's take values in some bounded interval $\Theta \subseteq \mathbb{R}$. With no loss of generality, we will assume $\Theta = [0, 1]$. For the network, the goal is to compute the arithmetic average of such values,

$$y := f(\boldsymbol{\theta}) = n^{-1} \sum_{v\in\mathcal{V}} \theta_v$$

through repeated exchanges of information among the agents and without a centralized computing or scheduling system.

Communication among the agents takes place as follows. At each time instant $t = 1, 2, \dots$, every agent v broadcasts a binary signal $a_v(t) \in \{0, 1\}$ to its out-neighbourhood $\mathcal{N}_v^+$. Every agent $w \in \mathcal{N}_v^+$ receives a possibly erased version $b_{v\to w}(t) \in \{0, 1, ?\}$ of $a_v(t)$. Here, the symbol ? represents a lost binary signal. We shall denote by $b_v(t) = (b_{w\to v}(t))_{w\in\mathcal{N}_v^-}$, and $b'_v(t) = (b_{v\to w}(t))_{w\in\mathcal{N}_v^+}$ the vector of signals received by agent v at time t, and, respectively, the vector of signals received from agent v by its out-neighbours. At time t, each agent $v \in \mathcal{V}$ makes an estimate $\hat{y}_v(t)$ of y. The compact notation $\boldsymbol{a}(t) = (a_v(t))_{v\in\mathcal{V}}$, $\boldsymbol{b}(t) = (b_v(t))_{v\in\mathcal{V}}$, and $\hat{\boldsymbol{y}}(t) = (\hat{y}_v(t))_{v\in\mathcal{V}}$, will be used for the full vectors of transmitted signals, received signals, and estimates at time t, respectively.

We shall assume the communication network to be memoryless, i.e., that $\boldsymbol{b}(t)$ is conditionally independent from the initial observations $\boldsymbol{\theta}$ and the previous transmissions $\{\boldsymbol{a}(s), \boldsymbol{b}(s) : 1 \leq s < t\}$, given the currently broadcasted signals $\boldsymbol{a}(t)$. Further, we shall assume that, given $\boldsymbol{a}(t)$, for every $v \in \mathcal{V}$ and $w \in \mathcal{N}_v^+$,

$$b_{v\to w}(t) = \begin{cases} ? & \text{w.p. } \varepsilon \\ a_v(t) & \text{w.p. } 1 - \varepsilon\,. \end{cases}$$

Here ε is some erasure probability which –for the sake of simplicity– is assumed to remain constant in t, v and w. [2]

Distributedness of the computation algorithm is then modeled by constraining the transmitted signal $a_v(t)$ to be a function of the local information available to agent v at the end of the $(t-1)$-th round of communication, and the estimate $\hat{y}_v(t)$ to be a function of the information available to agent v at the end of the t-th round of

[2] It is not necessary, for the validity of our results, to assume mutual independence of the received signals $\{b_{v\to w}(t)\}_w$ given $a_v(t)$. On the other hand, the assumption that the channel is memoryless remains crucial.

communication. We shall consider two different local information structures, corresponding to the cases when there is no communication feedback, and when there is causal communication feedback, respectively. When there is no communication feedback, the local information available to agent v at the end of the t-th round of communication, consists of its initial observation, as well as of the signals received by v up to time t:

$$i_v(t) := \{\theta_v, b_v(s) : 1 \leq s \leq t\} .$$

On the other hand, when there is causal communication feedback, the local information available to agent v at the end of the t-th round of communication includes also all the signals received insofar from v by its out-neighbours:

$$i'_v(t) := \{\theta_v, b_v(s), b'_v(s) : 1 \leq s \leq t\} .$$

The communication setting outlined above can be conveniently described by a directed graph $\mathcal{G}_\varepsilon = (\mathcal{V}, \mathcal{E})$ (the communication graph), whose vertices are the agents, and such that an ordered pair (v, w) with $v \neq w$ belongs to $\mathcal{E}$ if and only if $w \in \mathcal{N}_v^+$ (or, equivalently, if $v \in \mathcal{N}_w^-$), i.e., if v transmits to w with erasure probability $\varepsilon < 1$. Throughout the paper, we shall assume that the graph $\mathcal{G}_\varepsilon$ is strongly connected, i.e. that there exists a directed path connecting any pair of its vertices.

A distributed computation algorithm on the communication graph $\mathcal{G}_\varepsilon = (\mathcal{V}, \mathcal{E})$ is specified by a pair $\mathcal{A} = (\Phi, \Psi)$ of double-indexed families of maps $\Phi = \{\phi_v^{(t)} : v \in \mathcal{V}, t \in \mathbb{N}\}$, and $\Psi = \{\psi_v^{(t)} : v \in \mathcal{V}, t \in \mathbb{N}\}$. When there is no communication feedback one has

$$\phi_v^{(t)} : \Theta \times \{0, 1, ?\}^{\mathcal{N}_v^- \times [t-1]} \to \{0, 1\} , \qquad \psi_v^{(t)} : \Theta \times \{0, 1, ?\}^{\mathcal{N}_v^- \times [t]} \to \Theta ,$$

and $a_v(t) = \phi_v^{(t)}(i_v(t-1))$, $\hat{y}_v(t) = \psi_v^{(t)}(i_v(t))$. On the other hand, in the case when causal communication feedback is available, one has

$$\phi_v^{(t)} : \Theta \times \{0, 1, ?\}^{(\mathcal{N}_v^- \cup \mathcal{N}_v^+) \times [t-1]} \to \{0, 1\} , \qquad \psi_v^{(t)} : \Theta \times \{0, 1, ?\}^{(\mathcal{N}_v^- \cup \mathcal{N}_v^+) \times [t]} \to \Theta ,$$

and $a_v(t) = \phi_v^{(t)}(i'_v(t-1))$, $\hat{y}_v(t) = \psi_v^{(t)}(i'_v(t))$. We shall see that the averaging algorithms we propose and study in this paper can be framed in the above general setting.

2.2 Performance measures

In order to analyze the performance of different distributed computation algorithms, we shall study the distance of the estimates $\hat{y}_v(t)$ from the average of the initial values y:

$$\boldsymbol{e}(t) = \hat{\boldsymbol{y}}(t) - y\mathbf{1} .$$

Namely, we define two complexity figures, the *communication complexity* and the *computational complexity*. The communication complexity of a distributed algorithm $\mathcal{A}$ on a communication graph $\mathcal{G}_\varepsilon$ will be measured in terms of the function

$$\tau(\delta) := \inf \left\{ t \in \mathbb{N} : n^{-1}\mathbb{E}\left[||\boldsymbol{e}(s)||^2\right] \leq \delta, \forall s \geq t \right\} , \qquad \delta \in]0, 1] .$$

In other words, for $\delta \geq 0$, $\tau(\delta)$ denotes the minimum number of binary transmissions each agent has to perform in order to guarantee that the average mean squared estimation error does not exceed δ.

Instead, the computational complexity of an algorithm $\mathcal{A}$ on a communication graph $\mathcal{G}_\varepsilon$ will be measured as follows. For every $t \in \mathbb{N}$, and $v \in \mathcal{V}$, we shall denote by $\kappa_v(t)$ the minimum number of binary operations required by agent v to evaluate the functions $\phi_v^{(t)}(\cdot)$ and $\psi_v^{(t)}(\cdot)$. Then, we shall define

$$\kappa(\delta) := \max \left\{ \sum_{1 \leq t \leq \tau(\delta)} \kappa_v(t) : v \in \mathcal{V} \right\} , \qquad \delta \in]0, 1] .$$

Hence, for any $\delta \geq 0$, $\kappa(\delta)$ denotes the maximum, over all agents $v \in \mathcal{V}$, of the total number of binary operations required to be performed, in order to achieve an average mean squared estimation error not exceeding δ.

2.3 Reliable transmission of continuous information through digital noisy channels

When the communication graph is complete, with all the agents connected through binary erasure broadcast channels, the problem reduces to that of reliable transmission of continuous information through digital noisy channels, which has been recently addressed in [12]. While referring to [12] for general information-theoretical limits and complexity vs performance tradeoffs, we shall revise here some results which will be used in the sequel.

Let θ be a random variable taking values in a bounded interval $\Theta \subseteq \mathbb{R}$, according to some a-priori probability law. Consider a memoryless binary erasure channel with erasure probability $\varepsilon \in (0,1)$ (briefly BEC(ε)). At each time $t \in \mathbb{N}$, the channel has input $a_t \in \{0,1\}$, output $b_t \in \{0,1,?\}$, with b_t conditionally independent from x, $\{a_s, b_s : 1 \leq s \leq t-1\}$, given a_t, and such that $b_t = a_t$ with probability $1-\varepsilon$, and $b_t =?$ with probability ε. The goal is to design a sequence of encoders $\Upsilon = (\Upsilon_t : \Theta \to \{0,1\})_{t\in\mathbb{N}}$, and of decoders $\Lambda = (\Lambda_t : \{0,1,?\}^t \to \Theta)_{t\in\mathbb{N}}$, such that, if $a_t = \Upsilon_t(x)$, b_t is the corresponding channel output, and $\hat{\theta}_t := \Lambda_t(b_1, \ldots, b_t)$ the current estimate, the mean squared error $\mathbb{E}[(\theta - \hat{\theta}_t)^2]$ is minimized. The computational complexity of the sequential coding scheme (Υ, Λ) is measured, for every time horizon $\ell \in \mathbb{N}$, in terms of the total number k_ℓ of binary operations required to compute $\Upsilon_t(x)$ and $\Lambda_t(b_1, \ldots, b_t)$ for all $1 \leq t \leq \ell$.

Here, in particular, we shall consider two specific classes of sequential transmission schemes described and analyzed in [12]. The first class is that of random linear tree codes, referred to by the superscript L. These codes have exponential convergence rates with respect to the number of channel uses, and computational complexity proportional to the the cube of the number of channel uses. The second class is that of irregular repetition codes (superscript R). Such codes have linear computational complexity, but subexponential converge rates. The performance of these two classes of codes is summarized in the following lemmas.

Lemma 1 ([12], Coroll. 6.2) *There exist a sequence of linear encoders Υ^L, and a sequence of decoders Λ^L, such that, if $\hat{\theta}_\ell = \Lambda^L_\ell(b_1, \ldots, b_\ell)$, then, for all $\ell \geq 0$,*

$$\mathbb{E}\left[(\theta - \hat{\theta}_\ell)^2\right] \leq \beta_L^{2\ell}\,, \qquad k_\ell^L \leq B\ell^3\,, \tag{1}$$

where $\beta_L \in (0,1)$, and $B > 0$ are constants depending on the erasure probability ε only.

Lemma 2 ([12], Prop. 5.1) *There exist a sequence of linear encoders Υ^R, and a sequence of decoders Λ^R, such that, if $\hat{\theta}_\ell = \Lambda^R_\ell(b_1, \ldots, b_\ell)$, then, for all $l \geq 0$,*

$$\mathbb{E}\left[(\theta - \hat{\theta}_\ell)^2\right] \leq \beta_R^{2\sqrt{\ell}}\,, \qquad k_\ell^R \leq 2\ell\,, \tag{2}$$

where $\beta_R \in (0,1)$ is a constant depending on the erasure probability ε only.

In the next sections we shall use the results above as building blocks for our iterative distributed averaging algorithms.

3 Distributed averaging without communication feedback

In this section, we present some iterative distributed averaging algorithms, working on a strongly connected communication graph $\mathcal{G}_\varepsilon$, without explicit communication feedback. Our algorithms involve a sequence of communication phases, alternated to averaging steps. During each communication phase, agents broadcast an encoded version of their current state –a scalar value coinciding with their current estimate of the average y– to their out-neighbors. At the end of each phase, each agent estimates the states of its in-neighbors from the signals received from them, and it updates its state to a convex combination of these estimates and its own current state; the process is then iterated. In the absence of communication feedback it is not clear how to guarantee that these algorithms preserve the average of the initial observations. To cope with this, we allow the length of each communication phase to increase with the phase index. For suitable choices of the length of the subsequent phases, we are able to prove that state agreement is asymptotically achieved with probability one, as well as to estimate the communication and computational complexities of the algorithms. Sect. 3.1 introduces the algorithms in further detail, while the main results on their performance are stated in Sect. 3.2.

3.1 Description of the algorithms

We shall present two iterative distributed averaging algorithms on a communication graph $\mathcal{G}_\varepsilon$. Both algorithms are based on a sequence of transmission phases, indexed by $j \geq 1$, alternated to averaging steps. Each agent $v \in \mathcal{V}$ maintains a scalar state $x_v(j)$, $j \geq 0$, which is initialized to equal the original observation θ_v. The state $x_v(j)$ has to be thought as v's estimate of y at the beginning of the $(j+1)$-th phase. During the j-th transmission phase, each agent broadcasts ℓ_j binary signals to its out-neighbors. These binary signals represent an encoding of the state $x_v(j-1)$. At the end of the j-th phase, each agent estimates each of its in-neighbors' states from the signals received from it, and it updates its state to a convex combination of these estimates and its own current state.

More specifically, let P be a doubly-stochastic, irreducible matrix adapted to $\mathcal{G}_\varepsilon$, with non-zero diagonal entries. Let $(\ell_j)_{j\in\mathbb{N}}$ be a sequence of positive integers, each ℓ_j representing the length of the j-th transmission phase, and define $h_j := \sum_{i\leq j} \ell_i$, for all $j \in \mathbb{N}$ and $h_0 = 0$. Further, let Υ and Λ be sequences of encoders and decoders as introduced in Section 2.3. Then, the proposed distributed algorithms consist of the following steps. First of all, each agent initializes its state setting:

$$x_v(0) = \theta_v\,, \qquad \forall v \in \mathcal{V}\,. \tag{3}$$

Then, for all $j \in \mathbb{N}$ and $v \in \mathcal{V}$:

Communication phase: v broadcasts an encoded version of its state $x_v(j-1)$ to its out-neighbours, namely, $\forall\, h_{j-1} < t \leq h_j$, it transmits the binary signal

$$a_t = \Upsilon_k\left(x_v(j-1)\right)\,, \qquad k = t - h_{j-1}\,, \tag{4}$$

State update: at the end of the j-th communication phase, v estimates the state of all its in-neighbours, based on the received signals $\{\boldsymbol{b}_v(t)\}_{t=h_{j-1}+1}^{h_j}$; let $\hat{x}_w^{(v)}(j-1)$ be the estimate of $x_w(j-1)$ built by agent v, then

$$\hat{x}_w^{(v)}(j-1) = \Lambda_{\ell_j}\left(b_{w\to v}(h_{j-1}+1), \ldots, b_{w\to v}(h_j)\right)\,, \qquad \forall w \in \mathcal{N}_v^-. \tag{5}$$

Then, v updates its own state according to the following *consensus-like step*:

$$x_v(j) = \sum_{w\in\mathcal{N}_v^-} P_{vw}\hat{x}_w^{(v)}(j-1) + P_{vv}x_v(j-1)\,. \tag{6}$$

Observe that the above-described algorithms can be framed in the general setting described in Sect. 2.1. Indeed, for all $j \geq 1$, one has

$$\begin{aligned}
&\phi^{(v)}_{h_{j-1}+k}(i_v(h_{j-1}+k)) = \Upsilon_i\left(x_v(j-1)\right) && 0 < k \leq \ell_j\,,\\
&\psi^{(v)}_{h_{j-1}+k}(i_v(h_{j-1}+k)) = x_v(j-1) && 0 \leq k < \ell_j\,.
\end{aligned}$$

Notice that state $x_v(j-1)$ represents the estimate that agent v has of y along all j-th phase, i.e.,

$$\hat{y}_v(t) = x_v(j-1)\,, \qquad \forall h_{j-1} \leq t < h_j\,; \tag{7}$$

In what follows, we shall consider two implementations of the algorithm. In the first implementation, referred to as algorithm $\mathcal{A}_L$, we use linear tree codes $\Upsilon = \Upsilon^L$, $\Lambda = \Lambda^L$, and phase-lengths $l_j^L = S_L j$ for some $S_L \in \mathbb{N}$. In the second implementation, referred to as algorithm $\mathcal{A}_R$, we use repetition codes $\Upsilon = \Upsilon^R$, $\Lambda = \Lambda^R$, and phase-lengths $l_j^R = S_R j^2$, for some $S_R \in \mathbb{N}$. Observe that, thanks to (1), one has, for the algorithm $\mathcal{A}_L$,

$$\mathbb{E}\left[\left(\hat{x}_w^{(v)}(j-1) - x_w(j-1)\right)^2\right] \leq \alpha_L^{2j}\,, \tag{8}$$

for every $j \in \mathbb{N}$, $v \in \mathcal{V}$, and $w \in \mathcal{N}_v^-$, where $\alpha_L := \beta_L^{S_L}$. Similarly, for the algorithms $\mathcal{A}_R$, Equation (2) guarantees that

$$\mathbb{E}\left[\left(\hat{x}_w^{(v)}(j-1) - x_w(j-1)\right)^2\right] \leq \alpha_R^{2j}\,, \tag{9}$$

for every $j \in \mathbb{N}$, $v \in \mathcal{V}$, and $w \in \mathcal{N}_v^-$, where $\alpha_R := \beta_R^{\sqrt{S_R}}$.

It should be mentioned that many other choices could have been made for the communication phase lengths, as well as for the coding schemes used during each of them. For instance, block codes of different lengths could have been used during each phase. Our choice of using the same anytime transmission scheme for every agent during each communication phase, has the advantage of fewer memory requirements (only one transmission scheme has to be memorized by each agent), anonymity (each agent uses the same transmission scheme, and the state updating rules only depend on its position in the graph), and adaptiveness with respect to the erasure probability ε. In fact, it is not required to know the actual value of ε in order to design Υ and Λ, see Remarks 3 and 5 in [12].

3.2 Performance analysis

We shall now present results characterizing the performance of the algorithms $\mathcal{A}_L$, $\mathcal{A}_R$ introduced in Sect. 3.1. Throughout, we shall assume that $\mathcal{G}_\varepsilon$ is a strongly connected communication directed graph, and P is a doubly stochastic, irreducible matrix which is adapted to $\mathcal{G}_\varepsilon$, and has positive diagonal entries. Notice that this implies that P^*P is doubly-stochastic and irreducible. It then follows from Perron-Frobenius theorem that P^*P has the eigenvalue 1 with multiplicity one and corresponding eigenvector $\mathbf{1}$, and all its other eigenvalues have modulus strictly smaller than 1. Hence, P has largest singular value equal to 1 and all other singular values strictly smaller than 1. We shall denote by $\rho := \rho(P) < 1$ the second largest singular value of P, and assume that $\rho \geq \underline{\rho}$, where $\underline{\rho} > 0$ is some a priori constant. [3]

Observe that the vector of the estimation errors on y made by the different agents, $\boldsymbol{e}(t) = \hat{\boldsymbol{y}}(t) - y\mathbf{1}$, is constant during each transmission phase, i.e., for any $j \geq 0$ we have that

$$\boldsymbol{e}(t) = \boldsymbol{e}(h_j)\,, \qquad \forall\, h_j \leq t < h_{j+1}\,. \tag{10}$$

To analyze the performance of our algorithms, it is useful to introduce a suitable decomposition of $\boldsymbol{e}$; for all $j \geq 0$, we can write that

$$\boldsymbol{e}(h_j) = \boldsymbol{z}(j) + \zeta(j)\mathbf{1}\,, \tag{11}$$

where

$$\boldsymbol{z}(j) = \boldsymbol{x}(j) - \left(n^{-1}\mathbf{1}^*\boldsymbol{x}(j)\right)\mathbf{1} \tag{12}$$

represents the difference between the current estimates and the average of the current states, whereas

$$\zeta(j) = n^{-1}\mathbf{1}^*\boldsymbol{x}(j) - y = n^{-1}\mathbf{1}^*\left(\boldsymbol{x}(j) - \boldsymbol{x}(0)\right) \tag{13}$$

accounts for the distance between the current average of the estimates and the average of the initial conditions.

Now, observe that the initialization (3) and the state dynamics (6) may be rewritten in the following compact form

$$\begin{cases} \boldsymbol{x}(0) = \boldsymbol{\theta} \\ \boldsymbol{x}(j+1) = P\boldsymbol{x}(j) + \left(P \odot \Delta(j+1)\right)\mathbf{1}\,, \qquad j \geq 0\,, \end{cases} \tag{14}$$

where $\Delta(j) = (\Delta_{vw}(j))_{v,w\in\mathcal{V}}$ is defined, for all $j \in \mathbb{N}$, by

$$\Delta_{vw}(j) := \begin{cases} \hat{x}_w^{(v)}(j-1) - x_w(j-1) & \text{if} \quad w \in \mathcal{N}_v^- \\ 0 & \text{if} \quad w \notin \mathcal{N}_v^-\,. \end{cases} \tag{15}$$

Notice that, in general, $\Delta_{vw}(j)$ has non-zero mean, and it is not independent from $x_w(j)$, and therefore from the errors introduced by the previous transmission phases $\{\Delta(i) : 1 \leq i < j\}$. We have the following result.

[3] This may be guaranteed without any global knowledge, for instance by assuming $P_{vv} \geq (1+\underline{\rho})/2$. See also the Metropolis weights definition in Sect. 6.
Note that this assumption is for analysis' purpose only, and the agents do not need to know $\underline{\rho}$ to run the algorithms. The assumption entails a minimal loss of generality in that it rules out the case $\rho = 0$: however, a version of Proposition 3 and of subsequent results can be stated, which cover this case as well [8].

Proposition 3 *Consider the stochastic system (14), driven by a noise process* $\{\Delta(j) : j \geq 1\}$ *satisfying*

$$\mathbb{E}[\Delta_{vw}(j)^2] \leq \alpha^{2j}\,, \qquad j \geq 1\,. \tag{16}$$

for some $0 < \alpha < \rho$*. Then, for all* $j \geq 0$*,*

$$\mathbb{E}[\zeta^2(j)] \leq \frac{\alpha^2}{(1-\alpha)^2}\,, \tag{17}$$

$$n^{-1}\mathbb{E}[\|\boldsymbol{z}(j)\|^2] \leq \rho^{2j}\left(1-\frac{\alpha}{\rho}\right)^{-2}\,. \tag{18}$$

Proof: See Section A.1. ■

The following result characterizes the performance of both algorithms $\mathcal{A}_L$ and $\mathcal{A}_R$.

Theorem 4 (Performance without communication feedback, increasing phase lengths) *For any choice of the initial phase's length* S_L *(respectively,* S_R*), there exists a real-valued random variable* $\hat{y}$ *such that*

$$\mathbb{E}\left[(y-\hat{y})^2\right] \leq \frac{\alpha^2}{(1-\alpha)^2}\,, \tag{19}$$

where $\alpha = \beta_L^{S_L}$ *(respectively,* $\alpha = \beta_R^{\sqrt{S_R}}$*) and that the estimates of algorithm* $\mathcal{A}_L$ *(respectively,* $\mathcal{A}_R$*) satisfy, with probability one,*

$$\lim_{t\to\infty} \hat{y}_v(t) = \hat{y}\,, \qquad \forall v \in \mathcal{V}\,. \tag{20}$$

Moreover, it is possible to choose the initial phase length S_L *(respectively,* S_R*) in such a way that the algorithm* $\mathcal{A}_L$ *(respectively,* $\mathcal{A}_R$*) has communication and computational complexities satisfying*

$$\tau_L(\delta) \leq C_1 + C_2\frac{\log^3\delta^{-1}}{\log^2\rho^{-1}}\,, \qquad \kappa_L(\delta) \leq C_3 + C_4\frac{\log^7\delta^{-1}}{\log^4\rho^{-1}}\,, \tag{21}$$

and, respectively,

$$\tau_R(\delta) \leq C_5 + C_6\frac{\log^5\delta^{-1}}{\log^3\rho^{-1}}\,, \qquad \kappa_R(\delta) \leq C_7 + C_8\frac{\log^5\delta^{-1}}{\log^3\rho^{-1}}\,, \tag{22}$$

for all $\delta \in]0,1]$*, where* $\{C_i : i = 1,\ldots,8\}$ *are positive constants depending on* ε *only.*

Proof: See Section A.2. ■

It is worth observing that, by (19), the mean squared distance between the asymptotic estimate $\hat{y}$ and the actual value y, is upper bounded by a constant which, quite remarkably, is independent of either the size of the network or the consensus matrix P, and depends only on the length of the first transmission phase. Hence, the asymptotic estimation error in both the algorithms $\mathcal{A}_L$ and $\mathcal{A}_R$ is independent of the size and topology of the network.

Theorem 4 shows that both the algorithms $\mathcal{A}_L$ and $\mathcal{A}_R$ have communication and computational complexities growing at most poly-logarithmically in the desired precision. It is interesting to observe that the bounds on the communication complexities suggest that the agents have to use fewer channel transmissions in order to achieve a desired precision when running the algorithm $\mathcal{A}_L$ than when running $\mathcal{A}_R$. On the other hand, the bounds on the computational complexities suggest that algorithm $\mathcal{A}_R$ requires the agents to perform fewer binary computations than $\mathcal{A}_L$ in order to achieve the same estimate precision. Furthermore, both complexities are shown to grow not faster than some negative power of ρ, the second largest singular value of the matrix P. As the matrix P is adapted to the communication graph $\mathcal{G}_\varepsilon$, the dependence of the bounds on ρ captures the effect of the network topology. As ρ approaches 1 –as it occurs for the families of graphs of increasing size described in Sect. 5– the bounds of Theorem 4 again suggest that algorithm $\mathcal{A}_L$ has lower communication complexity but higher computational complexity than the algorithm $\mathcal{A}_R$.

4 Distributed averaging with communication feedback

In this section we discuss how to efficiently modify the algorithms of Sect. 3 when there is communication feedback. The key point is that, in the presence of noiseless communication feedback, it is possible to modify the algorithms $\mathcal{A}_L$ and $\mathcal{A}_R$ and make them average-preserving: this is explained in Sect. 4.1. In Sect. 4.2, the modified algorithms will be shown to converge to average consensus with probability one, and to have lower communication and computational complexities than their feedbackless counterpart.

4.1 Average-preserving state update

We shall consider distributed averaging algorithms with the same iterative structure described in Sect. 3. We shall use the same communication phase rule (4) of Sect. 3.1, and modify the state update step as follows. Observe that, at the end of the j-th communication phase, not only can agent v estimate the state of all its in-neighbours as in (5), but it can as well use its knowledge of the signals $\{b_{v\to w}(t)\}_{t=h_{j-1}+1}^{h_j}$ received by its out-neighbors $w \in \mathcal{N}_v^+$ in order to compute their estimates $\hat{x}_v^{(w)}(j-1)$ of its own current state. Then, in the presence of communication feedback, the state update step (6) can be replaced by the following one

$$x_v(j+1) = x_v(j) - \sum_{w\in\mathcal{N}_v^+} P_{wv}\hat{x}_v^{(w)}(j) + \sum_{w\in\mathcal{N}_v^-} P_{vw}\hat{x}_w^{(v)}(j)\,. \tag{23}$$

Clearly, such algorithms can be framed in the general setting described in Sect. 2.1. Indeed, for all $j \geq 1$, one has

$$\begin{aligned} \phi^{(v)}_{h_{j-1}+k}(i'_v(h_{j-1}+k)) &= \Upsilon_k\left(x_v(j-1)\right) && 0 < k \leq \ell_j \\ \psi^{(v)}_{h_{j-1}+k}(i'_v(h_{j-1}+k)) &= x_v(j-1) && 0 \leq k < \ell_j\,. \end{aligned}$$

Notice that the state update (23) requires every agent v to know not only the entries of the v-th row of the matrix P, but also those of the v-th column of P. However, no global knowledge of the communication graph topology is required.

The state update equation (23) may be written in the compact form

$$\boldsymbol{x}(j+1) = P\boldsymbol{x}(j) + \left[(P\odot\boldsymbol{\Delta}(j+1)) - (P\odot\boldsymbol{\Delta}(j+1))^*\right]\mathbf{1}. \tag{24}$$

Observe that $\mathbf{1}^*\left[(P\odot\boldsymbol{\Delta}(j)) - (P\odot\boldsymbol{\Delta}(j))^*\right]\mathbf{1} = 0$, so that, since P is a doubly-stochastic matrix, one has $\mathbf{1}^*\boldsymbol{x}(j+1) = \mathbf{1}^*P\boldsymbol{x}(j) = \mathbf{1}^*\boldsymbol{x}(j)$. It follows that $n^{-1}\mathbf{1}^*\boldsymbol{x}(j) = n^{-1}\mathbf{1}^*\boldsymbol{x}(0) = y$ for any j. Hence,

$$\zeta(j) = 0\,, \qquad \boldsymbol{e}(h_j) = \boldsymbol{z}(j)\,, \qquad \forall j \geq 0\,. \tag{25}$$

4.2 Feedback algorithms with increasing phase lengths

We shall now consider two implementations of the above described algorithms. Such implementations have increasing communication phase lengths, analogously to those introduced in Sect. 3. In the first implementation, referred to as algorithm $\mathcal{A}'_L$, linear codes are used in the communication phase, and the length of the j-th phase is $l_j = S_L j^2$ for some $S_L \in \mathbb{N}$. The second implementation, named $\mathcal{A}'_R$, uses repetition codes in the communication phase, and the length of the j-th phase is $l_j = S_R j^3$ for some $S_R \in \mathbb{N}$.

The following result characterizes the performance of the algorithms $\mathcal{A}'_L$ and $\mathcal{A}'_R$, showing that with probability one, the estimates of all the agents converge to the actual value y, and estimating the communication and computational complexities.

Theorem 5 (Performance with communication feedback, increasing phase lengths) *For any choice of the initial phases length S_L (respectively, S_R), the estimates of the algorithm $\mathcal{A}'_L$ and $\mathcal{A}'_R$ satisfy, with probability one,*

$$\lim_{t\to\infty} \hat{y}_v(t) = y\,, \qquad \forall v \in \mathcal{V}\,.$$

Moreover, it is possible to choose the initial phase length S_L (S_R, respectively) in such a way that the algorithm $\mathcal{A}'_L$ (respectively, $\mathcal{A}'_R$) has communication and computational complexities satisfying

$$\tau'_L(\delta) \leq C'_1 + C'_2 \frac{\log^2 \delta^{-1}}{\log^2 \rho^{-1}}, \qquad \kappa'_L(\delta) \leq C'_3 + C'_4 \frac{\log^4 \delta^{-1}}{\log^4 \rho^{-1}}, \tag{26}$$

and, respectively,

$$\tau'_R(\delta) \leq C'_5 + C'_6 \frac{\log^3 \delta^{-1}}{\log^3 \rho^{-1}}, \qquad \kappa'_R(\delta) \leq C'_7 + C'_8 \frac{\log^3 \delta^{-1}}{\log^3 \rho^{-1}}, \tag{27}$$

for all $\delta \in]0,1]$, where $\{C'_i : i = 1, \ldots, 8\}$ are positive constants depending on ε only.

Proof: See Section A.3. ■

Theorem 5 shows that the communication and computational complexities of the algorithms $\mathcal{A}'_L$ and $\mathcal{A}'_R$ have a better dependence on the desired precision δ with respect to their feedbackless analogous $\mathcal{A}_L$ and $\mathcal{A}_R$. On the other hand, the dependence on ρ is the same. The reason for such an improvement in performance relies on the average-preserving property which can be guaranteed when communication feedback is available. Thanks to this property, as shown by Theorem 5, it is not necessary to determine the initial phase's length as a function of final desired precision, since the estimates produced by both $\mathcal{A}'_L$ and $\mathcal{A}'_R$ converge to y with probability one. In contrast, when communication feedback is not available, it is not clear how to guarantee that the average of the agents' estimates is preserved. This is the reason why, in Sect. 3, we had to adjust the initial phase's length as a function of the desired precision δ, inducing a worse dependence on δ of the bounds on communication and computational complexities of the algorithms $\mathcal{A}_L$ and $\mathcal{A}_R$ shown in Theorem 4.

4.3 Feedback algorithms with constant phase lengths

While the algorithms we proposed require increasing phase lengths, in many practical circumstances it proves convenient to use distributed averaging algorithms with constant phase length. For instance, this is a desirable property in cases when the algorithm is expected to operate in (slowly) varying environment, and it is important that the timescale of the algorithm remains faster than the timescale on which the environment is changing, for otherwise it would never adapt. In the absence of communication feedback, it may be expected that algorithms with constant phase lengths would not work very well, for the noise keeps accumulating on the linear space generated by the vector $\mathbf{1}$, and the error $\zeta(j)$ tends to increase. This intuition is confirmed by simulations in Sect. 6; see also the related arguments developed in [32]. On the other hand, the average-preserving state update (24) prevents this to happen when communication feedback is available. One may therefore consider two versions of the algorithm, $\mathcal{A}''_L$ and $\mathcal{A}''_R$, with state update (24), and communication phases using linear and repetition codes, respectively, and constant lengths $l_j = S_L$, and $l_j = S_R$. Using such algorithms, the agents' estimates do not converge to consensus, but their estimation error remains bounded by a constant which depends on the phase length, as shown in the following:

Theorem 6 (Performance with communication feedback, constant phase lengths) *For any choice of the initial phase length S_L (resp. S_R), the estimation error in the algorithm $\mathcal{A}''_L$ (resp. $\mathcal{A}''_R$) satisfies*

$$n^{-1}\mathbb{E}[||\boldsymbol{e}(t)||^2] \leq \rho^{2j} + (1-\rho)^{-2}\alpha^2,$$

where $\alpha = \beta_L^{S_L}$ (resp. $\alpha = \beta_R^{\sqrt{S_R}}$).

5 Complexity scaling with network size

In this section we gather some considerations on how the complexity of our algorithms may vary with respect to the size of the network. As the matrix P is adapted to the communication graph $\mathcal{G}_\varepsilon$, the second largest singular value ρ of P depends on the network topology: hence, so do the complexities of our algorithms. A precise characterization of the these tradeoffs can be led for some interesting families of graphs. In what follows we start analyzing two simple examples: the ring and the 2-dimensional torus, and then we describe the more general family of grids on toruses and grids on cubes of any dimension.

In all the examples we will introduce here, P is a normal matrix; in this case, the second largest singular value ρ is simply the second largest in modulus of the eigenvalues of P.

Example 7 (Ring) Consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = [n]$ and where $\mathcal{E}$ is described by the arcs $(1, n)$ and $(i+1, i)$ for $1 \leq i \leq n-1$. Assume that P is such that $P_{i,i} = P_{i,i+1} = 1/2$. Such a matrix P is a circulant matrix and in particular is normal. In this case, it is easy to explicitly compute the eigenvalues of P and see that

$$\rho = 1 - \tfrac{C}{n^2} + O\left(\tfrac{1}{n^4}\right) \qquad n \to \infty,$$

for some positive constant C. Observe that, when $n \to \infty$, then $\rho \to 1$, and this implies that the convergence time of the standard linear consensus algorithm diverges as n diverges. The consequence of such result on the performance of our algorithms is the following: when $n \to \infty$, $\tau_L(\delta) = O(n^4)$, and $\kappa_L(\delta) = O(n^8)$, whereas $\tau_R(\delta)$ and $\kappa_R(\delta)$ are both $O(n^6)$.

Example 8 (2-dimensional torus) Consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = [m] \times [m]$, so that $n = m^2$, and where $\mathcal{E}$ contains the edges $(i, j) \leftarrow (i, j+1)$, $(i, j) \leftarrow (i+1, j)$ and all the self-loops $((i, j), (i, j))$; all the sums are meant component-wise $\bmod m$. Assume that the matrix P is such that $P_{(i,j),(i,j+1)} = P_{(i,j),(i+1,j)} = P_{(i,j),(i,j)} = 1/3$. Such a matrix P is a block-circulant matrix and is normal; in this case we have

$$\rho = 1 - \tfrac{C}{n} + O\left(\tfrac{1}{n^2}\right) \qquad n \to \infty$$

This implies that, when $n \to \infty$, $\tau_L(\delta) = O(n^2)$, and $\kappa_L(\delta) = O(n^4)$, whereas $\tau_R(\delta)$ and $\kappa_R(\delta)$ are both $O(n^3)$.

A natural generalization of the two examples above is a grid with $n = m^d$ nodes on a d-dimensional torus and with edges to possibly several neighbors direction. This class of graphs is known as Cayley graphs on the Abelian group $\mathbb{Z}_m^d$ (i.e., the set $\{0, 1, \ldots, m-1\}^d$, where the sum is componentwise $\bmod m$). Cayley matrices are matrices P adapted to such graphs and with labels which respect the graph symmetry, in the sense that the coefficient associated with an edge (u, v) depends only on the difference (in $\mathbb{Z}_n^d$) $u - v$. Such matrices are normal and it is possible to write explicitly their eigenvalues and eigenvectors, exploiting a notion of Fourier transform over Abelian groups (see e.g. [4]). When there is a family of Cayley graphs of increasing size with bounded neighborhoods (each node has a number of neighbors which does not grow with n) and where the coefficients used for weighting neighbors do not vary with n either, then the asymptotic behavior of the second largest singular value is:

$$\rho = 1 - \tfrac{C}{n^{2/d}} + O\left(\tfrac{1}{n^{4/d}}\right) \qquad n \to \infty$$

where C is a positive constant, depending only on d and on the coefficients assigned to neighbors. This implies that the complexities of our algorithms, for $n \to \infty$, satisfy: $\tau_L(\delta) = O(n^{4/d})$, and $\kappa_L(\delta) = O(n^{8/d})$, whereas $\tau_R(\delta)$ and $\kappa_R(\delta)$ are both $O(n^{6/d})$.

A related family of graphs consists in grids on a cube in $\mathbb{R}^d$ instead of on a torus (e.g., for $d = 1$, a line of points, instead of a circle). A technique introduced in [6] allows to find the eigenvalues of matrices associated with d-dimensional grids on cubes, provided that coefficients of P satisfy some assumptions of symmetry, and that the coefficients on the self-loops at the borders are suitably arranged. The important result is that, asymptotically with n, ρ has the same behavior for a d-dimensional grid on a torus and on the cube, and thus also the performance bounds for our algorithms are the same as in the Cayley case. The interested reader can find in [10] more formal definitions and a summary of the relevant properties concerning the eigenvalues of Cayley matrices and grids.

6 Simulation results and comparison with decreasing gains strategy

This section is devoted to some examples illustrating the averaging algorithms proposed in this paper, their implementation, and their comparison with other sensible algorithms, inspired by the consensus literature.

Example 9 ($\mathcal{A}_R$ algorithms) We start by providing a practical implementation of the algorithms $\mathcal{A}_R$ described in Section 3.1, and commenting on its performance. To do so, we need to specify an encoding/decoding scheme and a consensus matrix P. For the encoding/decoding scheme, we implement an instance of the low-complexity repetition transmission schemes described in [12, Section 5.1], whose performance are characterized in Lemma 2. For the sake

of the clarity we briefly describe them. Let $x \in [0,1]$ be a quantity to be transmitted and let $\sum_{i\geq 1} c_i 2^{-i}$, $c_i \in \{0,1\}$ be its diadic expansion. The key idea underlying the scheme we adopt is based on the following observation: since the different bits of the binary expansion of x require different levels of protection, it is sensible that they be repeated with a frequency monotonically decreasing in their significance. Informally the sequence of transmitted symbols is described as $(c_1, c_1, c_2, c_1, c_2, c_3, c_1, c_2, c_3, c_4, \ldots)$. The estimate $\hat{x} = \sum_{i\geq 1} \hat{c}_i 2^{-i}$ is built as follows: $\hat{c}_i = c_i$ if at least one of the repeated occurrences of c_i in the transmitted word has been received un-erased, otherwise $\hat{c}_i$ is set to 0 or 1 uniformly at random.

We describe now the communication graph and the consensus matrix P adopted. We consider $n = 30$ agents, and a communication graph which is a strongly connected realization of a two-dimensional random geometric graph, where vertices are 30 points uniformly distributed in the unit square, and there is a pair of edges (u, v) and (v, u) whenever points u, v have a distance smaller than 0.4. The communication graph is bidirectional, in the sense that $\mathcal{N}_v^- = \mathcal{N}_v^+$ for all $v \in \mathcal{V}$. The consensus matrix P is built according to the Metropolis weights, illustrated for instance in [33]. Such matrix can be constructed distributedly, using only information on neighbors, as follows:

$$P_{uv} = \begin{cases} \frac{1}{1+\max\{\deg(u),\deg(v)\}} & \text{if } (u,v) \in \mathcal{E} \\ 1 - \sum_{w \in \mathcal{N}_u^-} P_{uw} & \text{if } u = v \\ 0 & \text{otherwise} \end{cases}$$

where $\deg(v)$ is the number of neighbors of node v.

In all our simulations we assume that the erasure probability ε is equal to 0.5. The initial condition $\boldsymbol{\theta}$ of each experiment is randomly sampled from a uniform distribution on $[0,1]^n$.

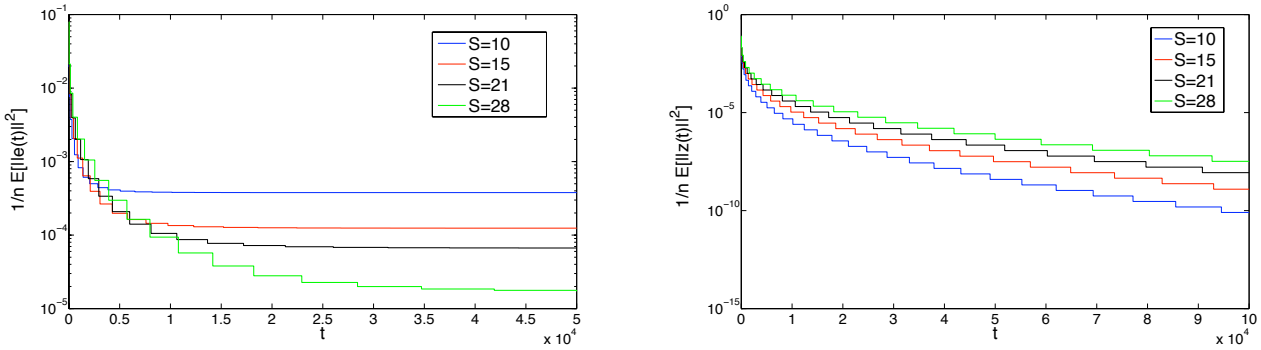

Fig. 1. Behavior of variable $n^{-1}\mathbb{E}\left[\|\boldsymbol{e}(t)\|^2\right]$ ***(left plot)*** and of variable $n^{-1}\mathbb{E}\left[\|\boldsymbol{z}(t)\|^2\right]$ ***(right plot)*** for different values of S_R when no communication feedback is available.

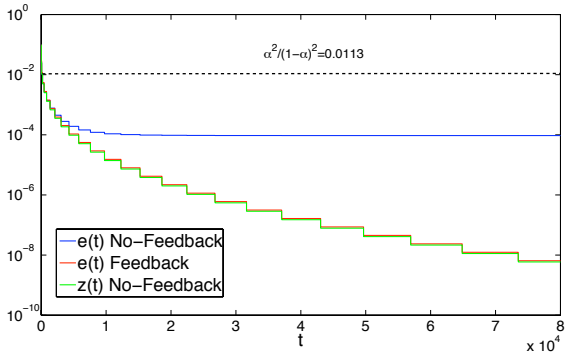

Fig. 2. Behavior of $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$ when communication feedback is available.

The simulation results obtained are plotted in Figures 1 and 2: they are averaged over 1000 trials (a different random geometric graph and a different initial condition are generated for each trial). In Figure 1 we depict the behaviors

of $n^{-1}\mathbb{E}\left[\|\boldsymbol{e}(t)\|^2\right]$ and $n^{-1}\mathbb{E}\left[\|\boldsymbol{z}(t)\|^2\right]$ for different values of S_R (we recall that in algorithms of type $\mathcal{A}_R$ the length of the j-th generic phase is equal to $S_R j^2$) under the assumption that no communication feedback is available. Observe that the larger is the value of S_R, the better is the attainable performance in terms of the variable $\boldsymbol{e}$. On the other hand, larger values of S_R also imply a slower convergence to 0 of the variable $\boldsymbol{z}$. In Figure 2 we provide a comparison between the algorithm without communication feedback and the algorithm with communication feedback. For these simulations we assume that $S_R = 21$. As predicted in Theorem 5, the advantage of having feedback in the communication is evident: the variable $\boldsymbol{e}(t)$ in the presence of communication feedback converges to zero similarly to the variable $\boldsymbol{z}(t)$ when no communication feedback is available (the two lines are almost indistinguishable in the plot). In Figure 2 we also plot, in dashed black line, the value of the bound in (19), which is $\frac{\alpha^2}{(1-\alpha)^2}$: if we compute α using [12, Proposition 5.1], it turns out that $\frac{\alpha^2}{(1-\alpha)^2} = 0.0113$. Note that the achieved precision is two orders of magnitude better than predicted by the bound in (19).

Example 10 (Decreasing Gains) In this example we want to compare our algorithm with a different strategy which was used in previous literature to compute approximate averages running a consensus algorithm in the presence of noisy communications. We will refer to such family of algorithms as to 'decreasing gain' algorithms, because the key idea is to have time-varying gains, which give decreasing weight to information coming from neighbors, so as to avoid accumulating an amount of error growing to infinity with time. Algorithms exploiting this idea were presented independently by various authors (see [19] , [28] and [21]). Such algorithms, after initialization $\boldsymbol{x}(0) = \boldsymbol{\theta}$, consist of consensus-like iterations with time-varying weights:

$$x_v(j+1) = \big(1 - \mu(j) + \mu(j)P_{vv}\big)x_v(j) + \mu(j) \sum_{w \in \mathcal{N}_v^-} P_{vw}\tilde{x}_w^{(v)}(j)$$

where $\tilde{x}_w^{(v)}(j) = x_w(j) + \eta_{w\to v}(j)$ is the version of $x_w(j)$ received by v, affected by additive noise $\eta_{w\to v}(j)$, and the *gains* $\mu(j) \in (0,1)$ are chosen to satisfy

$$\textstyle\sum_{k\geq 0} \mu(j) = \infty \text{ and } \sum_{j\geq 0} \mu^2(j) < \infty\,.$$

Such algorithms were designed for (analog) channels, where real numbers can be transmitted and are affected by an additive noise with zero-mean, bounded variance and independent from past history as well as from other channels in the network. Under such assumptions, [19], [28] and [21] use techniques of stochastic approximation theory to prove convergence to consensus. However, we can apply them also to the digital noisy networks considered in this paper, if we replace $\tilde{x}_w^{(v)}(j)$ with the value $\hat{x}_w^{(v)}(j)$ obtained after the process of encoding – transmitting over the channel from v to w – decoding, by some suitable coding scheme. What we want to compare is the strategy of increasing transmission lengths versus that of decreasing gains, where we plug into the decreasing gain algorithm an encoding/encoding of fixed length $\bar{\ell}$, not varying with j. Thus, we obtain the following algorithm. After initialization $x_v(0) = \theta_v$ for all $v \in \mathcal{V}$, the algorithm has time phases of constant length $\bar{\ell}$, with $\mu(j) \in (0,1)$ satisfying $\sum_{j\geq 0}\mu(j) = \infty$ and $\sum_{j\geq 0}\mu^2(j) < \infty$. In the j-th time phase, each agent v broadcasts an encoded version of its state $x_v(j-1)$ to its out-neighbours, namely it transmits the binary signal $a_t = \Upsilon_i\left(x_v(j-1)\right)$, where $i = t - \bar{\ell}(j-1)$. At the end of the transmission phase, v computes the state estimate of all its in-neighbours, based on the received signals, putting $\hat{x}_w^{(v)}(j-1) = \Lambda_{\bar{\ell}j}\left(b_{w\to v}(\bar{\ell}(j-1)+1), \ldots, b_{w\to v}(\bar{\ell}j)\right)$ for all $w \in \mathcal{N}_v^-$. Then, it updates its own state, with a weighted consensus-like iteration:

$$x_v(j) = (1 - \mu(j-1) + \mu(j-1)P_{vv})x_v(j-1) + \mu(j-1) \sum_{w\in\mathcal{N}_v^-} P_{vw}\hat{x}_w^{(v)}(j-1)\,.$$

State $x_v(j-1)$ represents the estimate that agent v has of y along all j-th phase, i.e., $\hat{y}_v(t) = x_v(j-1)$ for $\bar{\ell}(j-1) \leq t < \bar{\ell}j$. Also the above-described scheme can be cast in the general setting described in Section 2.1 by letting

$$\phi^{(v)}_{\bar{\ell}(k-1)+i}(\theta_v, b_v(1), \ldots, b_v(\bar{\ell}(k-1)+i-1)) = \Upsilon_i\left(x_v(k-1)\right)\,, \qquad \forall 1 \leq i \leq \bar{\ell};$$

and

$$\psi_t^{(v)}(\theta_v, b_v(1), \ldots, b_v(t)) = x_v(k-1)\,, \qquad \forall \bar{\ell}(k-1) \leq t < \bar{\ell}k\,.$$

Clearly, the fact that in the 'decreasing gain' algorithms the length of the phases is fixed has the advantage of having an asymptotically shorter time within successive consensus-like updates, but prevents the variances of the

errors associated to the estimates $\hat{x}(k)$ to decrease to zero. The decreasing gains allow to overcome this problem, by giving decreasing weight to the accumulating errors; however this comes at the price of a slower convergence of the consensus-like steps. Moreover there is no guarantee of convergence to (or near to) the correct average in the case of digital noise. It is not clear a priori which strategy can provide the best performance, and thus it is interesting to compare the two of them by simulations. We consider the same encoding/decoding scheme introduced in the previous example where we set $\bar{\ell} = 75$ and $S_R = 21$ and where again $\epsilon = 0.5$. Moreover, as far as the decreasing gains algorithms are concerned, we assume that $\mu(j) = 1/j$.

The results obtained are plotted in Figures 3 and 4: again they are generated based on 1000 trials where for each trial the consensus matrix P is built as in Example 9. Both figures show that the proposed $\mathcal{A}_R$ strategy is more effective than the decreasing gains one. From Figure 4 we are confirmed about the effectiveness of the exploitation of communication feedback, which is based on the average preservation.

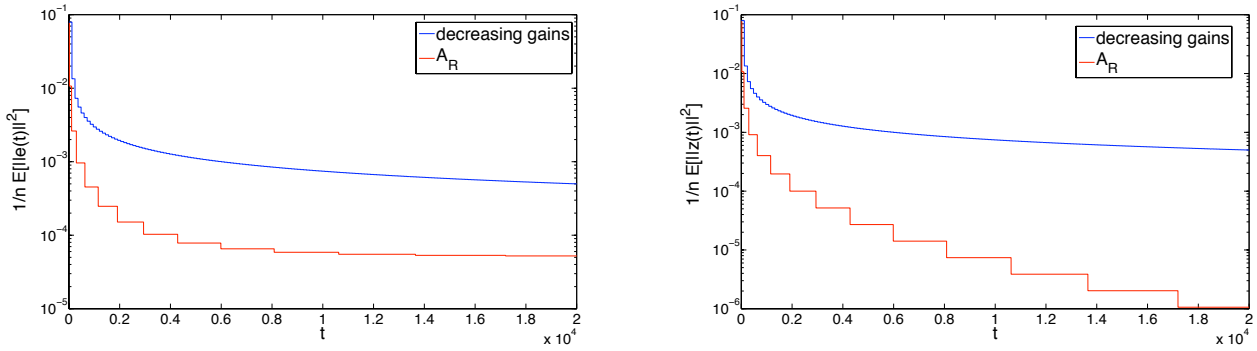

Fig. 3. Behavior of variable $n^{-1}\mathbb{E}\left[||\boldsymbol{e}(t)||^2\right]$ ***(left plot)*** and of variable $n^{-1}\mathbb{E}\left[||\boldsymbol{z}(t)||^2\right]$ ***(right plot)*** when no communication feedback is available.

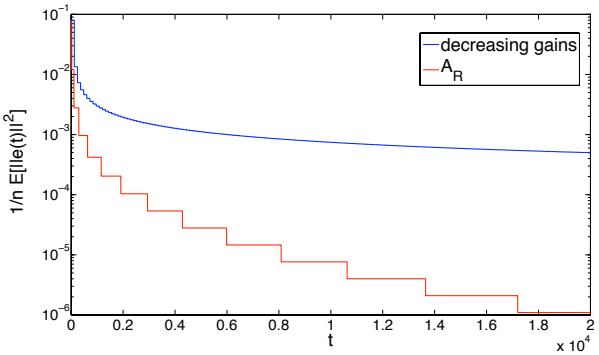

Fig. 4. Behavior of $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$ when feedback-communication is available.

Example 11 (Fixed gains and phase lengths) In this example we consider an algorithm which uses a linear repetition code for communication, and keeps fixed along all the iterations both the length of the communication phases and the consensus gains. As argued in Sect. 4.3, such an algorithm is important for the applications, since keeping both phase lengths and consensus gains fixed is useful when operating in slowly varying environment, where it is important that the timescale of the algorithm remains faster than the timescale at which the environment is changing. Precisely, all the communication phases lengths are assumed to be equal to a certain value $\bar{\ell}$, as in the decreasing gains algorithm, and the update rule is given by (6), as in $\mathcal{A}_R$. Our simulations results are reported in Figures 5 (left) and 6; they are generated based on 1000 trials adopting the encoding/decoding scheme used in the previous Example with $\bar{\ell} = 75$ and $\epsilon = 0.5$ and where, for each trial, the consensus matrix P is constructed as explained in Example 9. It is clear that in this fixed gains-fixed phases algorithm there is no compensation to smooth out the communication errors. This implies, for instance, that the average is not preserved: this results (Figure 5, left) in a poor performance with respect to the variable $\boldsymbol{e}$. In particular, the variable $\boldsymbol{e}$ drifts, and there is no convergence; a similar phenomenon was observed in [32,9] in the case of communications affected by white noise.

On the other hand, when communication feedback is available, one can consider the average-preserving algorithm $\mathcal{A}''_R$, introduced in Sect. 4.3. In this case, reported in Figures 5 (right) and 7, the mean square error $n^{-1}\mathbb{E}[||\boldsymbol{e}(t)^2||]$

decreases exponentially fast, and then remains bounded, though oscillating: this is consistent with the statement of Theorem 6. Figures 6 and 7 display the comparison between the $\mathcal{A}_R$ algorithm proposed in this paper and the fixed gains-fixed phases algorithms (with and without feedback). It is worth to notice that the latter can be effective in the transient, while their asymptotic behaviors are qualitatively different from the $\mathcal{A}_R$ algorithm.

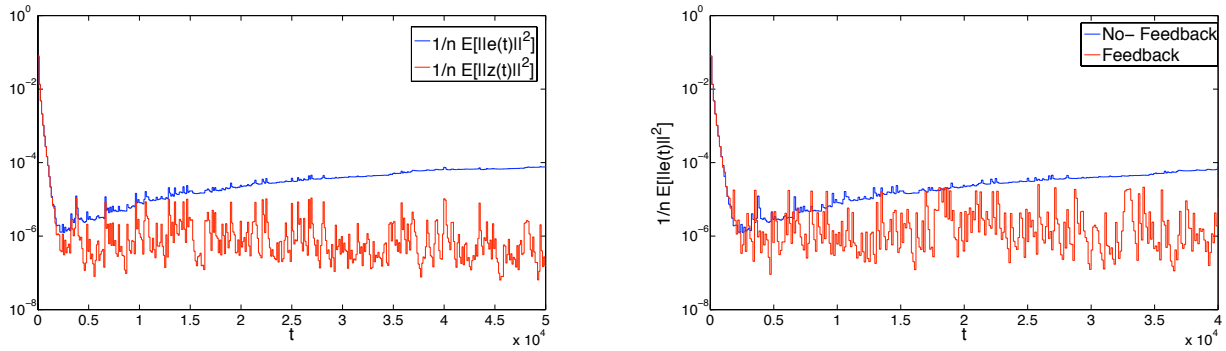

Fig. 5. Behavior of $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$ and $n^{-1}\mathbb{E}[\|\boldsymbol{z}(t)\|^2]$ when both length of communication phases and gains are fixed and when no communication feedback is available ***(left)***. Comparison, in terms of the variable $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$, between the no communication feedback case and the communication feedback case, when both length of communication phases and gains are fixed ***(right)***.

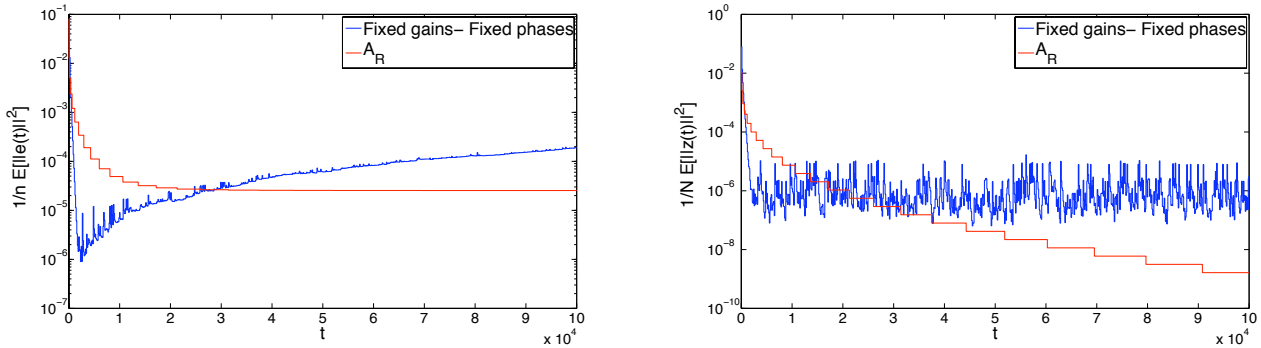

Fig. 6. Comparison in terms of $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$ ***(left)*** and $n^{-1}\mathbb{E}[\|\boldsymbol{z}(t)\|^2]$ ***(right)*** between and $\mathcal{A}_R$ and the fixed length communication phases and fixed gains when no communication feedback is available.

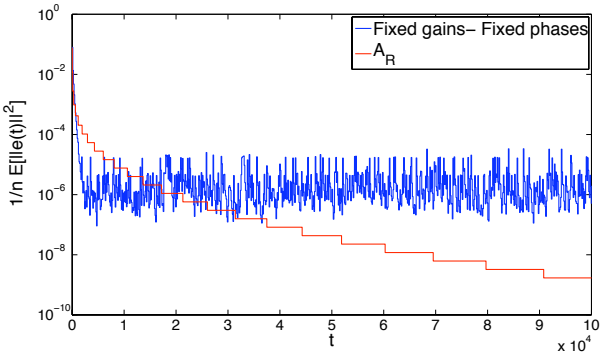

Fig. 7. Comparison in terms of $n^{-1}\mathbb{E}[\|\boldsymbol{e}(t)\|^2]$ between and $\mathcal{A}_R$ and the fixed length communication phases and fixed gains when communication feedback is available.

7 Conclusion

In this paper, for the first time we have considered the averaging problem on networks of digital links, and established suitable performance figures to evaluate its algorithmic solutions, in terms of communication and computation

complexities. On this ground, the main contribution of the paper has consisted in proposing a family of average consensus algorithms designed for digital networks, based on encoding/decoding schemes with precision increasing with time. Estimates of the communication and computation complexities to achieve a prescribed precision have been given: we showed that both the number of transmissions and of the number of computations are poly-logarithmic in the prescribed precision. Also, we have investigated how to make use of communication feedback, when available, in order to make the algorithms average-preserving, and improve their performance. The question is open whether a logarithmic algorithm can be designed for average consensus on digital networks, and how much global information it would require to be run by the agents.

Acknowledgements

The authors wish to thank Fabio Fagnani and Sandro Zampieri for many discussions on the topics of this paper. The research of Federica Garin has received funding from the European Community Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

References

[1] O. Ayaso, D. Shah, and M.A. Dahleh. Counting bits for distributed function computation. In *IEEE International Symposium on Information Theory (ISIT)*, pages 652–656, July 2008.

[2] O. Ayaso, D. Shah, and M.A. Dahleh. Information theoretic bounds for distributed computation. *IEEE Transactions on Information Theory*, 2008. Submitted.

[3] T. C. Aysal, M. J. Coates, and M. G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Transactions on Signal Processing*, 56(10):4905–4918, 2008.

[4] E. Behrends. *Introduction to Markov Chains (with Special Emphasis on Rapid Mixing)*. Vieweg Verlag, 1999.

[5] F. Benezit, P. Thiran, and M. Vetterli. Interval consensus: From quantized gossip to voting. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3661–3664, April 2009.

[6] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao. Symmetry analysis of reversible markov chains. *Internet Mathematics*, 2(1):31–71, 2005.

[7] R. Carli, F. Bullo, and S. Zampieri. Quantized average consensus via dynamic coding/decoding schemes. *International Journal of Robust and Nonlinear Control*, 2009. (Submitted May 2008) to appear.

[8] R. Carli, G. Como, P. Frasca, and F. Garin. Average consensus on digital noisy networks. In *IFAC Workshop on Estimation and Control of Networked Systems*, Venice, Italy, September 2009.

[9] R. Carli, F. Fagnani, P. Frasca, T. Taylor, and S. Zampieri. Average consensus on networks with transmission noise or quantization. In *European Control Conference*, pages 1852–1857, Kos, Greece, June 2007.

[10] R. Carli, F. Garin, and S. Zampieri. Quadratic indices for the analysis of consensus algorithms. In *4th Information Theory and Applications Workshop*, San Diego, CA, February 2009.

[11] G. Como and M. Dahleh. Lower bounds on the estimation error in problems of distributed computation. In *Information Theory and Applications Workshop*, San Diego, CA, February 2009.

[12] G. Como, F. Fagnani, and S. Zampieri. Anytime reliable transmission of real-valued information through digital noisy channels. *SIAM Journal on Control and Optimization*, December 2008. Submitted.

[13] P. Denantes, F. Benezit, P. Thiran, and M. Vetterli. Which distributed averaging algorithm should I choose for my sensor network? In *IEEE Conference on Computer Communications (INFOCOM)*, pages 986–994, April 2008.

[14] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48(1):102–133, 2009.

[15] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control*, 19(16):1787–1816, 2009.

[16] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):755–764, 2005.

[17] A. Giridhar and P. R. Kumar. Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98–107, 2006.

[18] J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Distributed anonymous function computation in information fusion and multiagent systems. arXiv:0907.2949, July 2009.

[19] M. Huang and J. H. Manton. Coordination and consensus of networked agents with noisy measurements: Stochastic algorithms and asymptotic behavior. *SIAM Journal on Control and Optimization*, 48(1):134–161, 2009.

[20] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor networks: Quantized data. December 2007. Submitted.

[21] S. Kar, J. M. F. Moura, and K. Ramanan. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. 2008. Submitted.

[22] A. Kashyap, T. Başar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

[23] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[24] T. Li, M. Fu, L. Xie, and J.-F. Zhang. Distributed consensus with limited communication data rate. *IEEE Transactions on Automatic Control*, 2009. Submitted.

[25] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli. On synchronous robotic networks – Part I: Models, tasks and complexity. *IEEE Transactions on Automatic Control*, 52(12):2199–2213, 2007.

[26] D. Mosk-Aoyama and D. Shah. Fast distributed algorithms for computing separable functions. *IEEE Transactions on Information Theory*, 54(7):2997–3007, 2008.

[27] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. Technical Report 2778, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, November 2007.

[28] R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. ArXiv:0805.0438, May 2008.

[29] S. Rajagopalan and L. J. Schulman. A coding theorem for distributed computation. In *Annual ACM Symposium on Theory of Computing*, pages 790–799, Montréal, Québec, Canada, May 1994.

[30] L. J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

[31] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[32] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[33] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Symposium on Information Processing of Sensor Networks (IPSN)*, pages 63–70, Los Angeles, CA, April 2005.

[34] L. Ying, R. Srikant, and G.E. Dullerud. Distributed symmetric function computation in noisy wireless sensor networks. *IEEE Transactions on Automatic Control*, 53(12):4826–4833, 2007.

A Proofs

A.1 Proof of Proposition 3

Let us first consider $\zeta(j)$ defined in (13). It is straightforward to verify that the recursion

$$\zeta(0) = 0; \qquad \zeta(j+1) = \zeta(j) + \xi(j+1)\,, \quad j \geq 0\,,$$

is satisfied, with $\xi(j) := n^{-1}\mathbf{1}^*(P \odot \Delta(j))\mathbf{1}$. For every $j \geq 0$, $\xi(j)$ is a random variable whose second moment can be upper bounded using the Cauchy-Schwarz inequality:

$$\begin{aligned}
\mathbb{E}\left[\xi(j)^2\right] &= n^{-2}\mathbb{E}\left[\left(\textstyle\sum_{v,w} P_{vw}\Delta_{vw}(j)\right)^2\right] \\
&= n^{-2}\sum_{v',w'}\sum_{v,w}\Delta_{v'w'}\Delta_{vw}\mathbb{E}\left[\Delta_{v'w'}(j)\Delta_{vw}(j)\right] \\
&\leq n^{-2}\sum_{v',w'}\sum_{v,w}P_{v'w'}P_{vw}\sqrt{\mathbb{E}\big[\Delta_{v'w'}(j)^2\big]\mathbb{E}\big[\Delta_{vw}(j)^2\big]} \\
&\leq n^{-2}\left(\textstyle\sum_{v,w}P_{vw}\alpha^j\right)^2 \\
&= \alpha^{2j}\,.
\end{aligned} \tag{A.1}$$

It follows again from the Cauchy-Schwarz inequality that, for all $1 \leq s, r \leq j$,

$$\mathbb{E}\left[\xi(s)\xi(r)\right] \leq \mathbb{E}\left[\xi(s)^2\right]^{1/2}\mathbb{E}\left[\xi(r)^2\right]^{1/2} \leq \alpha^{s+r}\,.$$

Therefore,

$$\mathbb{E}[\zeta^2(j)] = \sum_{1\leq s,r\leq j}\mathbb{E}\left[\xi(s)\xi(r)\right] \leq \sum_{1\leq s,r\leq j}\alpha^{r+s} = \Big(\sum_{1\leq s\leq j}\alpha^s\Big)^2 \leq \frac{\alpha^2}{(1-\alpha)^2}\,.$$

Now, let us focus on $\boldsymbol{z}(j)$ defined in (12). Observe that the recursion

$$\boldsymbol{z}(0) = \boldsymbol{u}(0)\,, \qquad \boldsymbol{z}(j+1) = P\boldsymbol{z}(j) + \boldsymbol{u}(j+1)\,, \quad j \geq 0\,,$$

is satisfied with

$$\boldsymbol{u}(0) := \boldsymbol{x}(0) - n^{-1}\mathbf{1}^T\boldsymbol{x}(0)\mathbf{1}, \qquad \boldsymbol{u}(j) := (P \odot \Delta(j))\mathbf{1} - \xi(j)\mathbf{1}\,.$$

Notice that $||P\boldsymbol{x}|| \leq \rho||\boldsymbol{x}||$ for all $\boldsymbol{x} \in \mathbb{R}^{\mathcal{V}}$ such that $\mathbf{1}^*\boldsymbol{x} = 0$. Since $\mathbf{1}^*\boldsymbol{u}(j) = 0$, we have

$$\|P\boldsymbol{u}(j)\| \leq \rho\|\boldsymbol{u}(j)\|\,. \tag{A.2}$$

On the other hand, again from the Cauchy-Schwarz inequality, for all $v, w \in \mathcal{V}$, we have that

$$\mathbb{E}\left[\Delta_{vw}(j)\Delta_{vw'}(j)\right] \leq \mathbb{E}\left[\Delta_{vw}(j)^2\right]^{1/2}\mathbb{E}\left[\Delta_{vw'}(j)^2\right]^{1/2} \leq \alpha^{2j}\,,$$

so that the random vector $\boldsymbol{u}(j)$, for $j \geq 1$, satisfies the following bound

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{u}(j)\|^2\right] &= \mathbb{E}\left[\|(P \odot \Delta(j))\mathbf{1}\|^2\right] - n\mathbb{E}\left[\xi(j)^2\right] \\
&\leq \sum_v \mathbb{E}\left[\left(\textstyle\sum_w P_{vw}\Delta_{vw}(j)\right)^2\right] \\
&= \sum_v \sum_{w,w'} P_{vw}P_{vw'}\mathbb{E}\left[\Delta_{vw}(j)\Delta_{vw'}(j)\right] \\
&\leq n\alpha^{2j}\,.
\end{aligned} \tag{A.3}$$

Moreover, recall that $\theta_v \in \Theta$ for any $v \in \mathcal{V}$, where Θ is an interval of unitary length. Consequently, $|\theta_v - y| \leq 1$ for any $v \in \mathcal{V}$, so that

$$\mathbb{E}\left[\|\boldsymbol{u}(0)\|^2\right] = \mathbb{E}\left[\|\boldsymbol{z}(0)\|^2\right] = \sum_v \mathbb{E}[(\theta_v - y)^2] \leq n\,. \tag{A.4}$$

Consider now $\mathbb{E}[\|\boldsymbol{z}(j)\|^2] = \mathbb{E}\left[\|\sum_{0\leq s\leq j} P^{j-s}\boldsymbol{u}(s)\|^2\right]$. By successively applying the Cauchy-Schwarz inequality, (A.2), (A.3) and (A.4), we get

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{z}(j)\|^2] &\leq \sum_{0\leq s,r\leq j} \sqrt{\mathbb{E}\left[\|P^{j-s}\boldsymbol{u}(s)\|^2\right]\mathbb{E}\left[\|P^{j-r}\boldsymbol{u}(r)\|^2\right]} \\
&\leq \left(\textstyle\sum_{0\leq s\leq j} \rho^{(j-s)}\sqrt{\mathbb{E}\left[\|\boldsymbol{u}(s)\|^2\right]}\right)^2 \\
&\leq n\left(\textstyle\sum_{0\leq s\leq j} \rho^{(j-s)}\alpha^s\right)^2 \\
&\leq n\left(\rho^j\textstyle\sum_{s\geq 0}\left(\frac{\alpha}{\rho}\right)^s\right)^2 \\
&= n\rho^{2j}\left(1 - \tfrac{\alpha}{\rho}\right)^{-2}
\end{aligned}$$

which completes the proof. ■

A.2 Proof of Theorem 4

We begin by estimating the difference $\boldsymbol{x}(j+1) - \boldsymbol{x}(j)$, for $j \geq 0$. Toward this goal, let $\xi(j) := n^{-1}\mathbf{1}^*(P \odot \Delta(j))\mathbf{1}$. Then, we may rewrite

$$\begin{aligned}
\boldsymbol{x}(j+1) - \boldsymbol{x}(j) &= \boldsymbol{x}(j+1) - n^{-1}\mathbf{1}^*\boldsymbol{x}(j+1)\mathbf{1} + n^{-1}\mathbf{1}^*\boldsymbol{x}(j+1)\mathbf{1} - \boldsymbol{x}(j) \\
&= \boldsymbol{x}(j+1) - n^{-1}\mathbf{1}^*\boldsymbol{x}(j+1)\mathbf{1} + n^{-1}\mathbf{1}^*\boldsymbol{x}(j)\mathbf{1} + \xi(j)\mathbf{1} - \boldsymbol{x}(j) \\
&= \boldsymbol{z}(j+1) - \boldsymbol{z}(j) + \xi(j)\mathbf{1}\,.
\end{aligned}$$

By successively applying the triangle inequality, Proposition 3, and (A.1), we get

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}(j+1)-\boldsymbol{x}(j)\|^2\right] &\leq \mathbb{E}\left[\left(\|\boldsymbol{z}(j+1)\|+\|\boldsymbol{z}(j)\|+\|\xi(j)\mathbf{1}\|\right)^2\right]\\ &\leq 3\left(\mathbb{E}\left[\|\boldsymbol{z}(j+1)\|^2\right]+\mathbb{E}\left[\|\boldsymbol{z}(j)\|^2\right]+n\mathbb{E}\left[\xi(j)^2\right]\right)\\ &\leq 3n\left(\rho^{2(j+1)}+\rho^{2j}\right)\left(1-\tfrac{\alpha}{\rho}\right)^{-2}+3n\alpha^{2j}\\ &\leq 9n\left(1-\tfrac{\alpha}{\rho}\right)^{-2}\left(\max\{\rho,\alpha\}\right)^{2j}\\ &= 9n\left(1-\tfrac{\alpha}{\rho}\right)^{-2}\rho^{2j}\,.\end{aligned}$$

Hence, the probability of the event $E_j := \left\{\|\boldsymbol{x}(j+1)-\boldsymbol{x}(j)\|^2 \geq \rho^{2j}\right\}$ can be estimated by Markov's inequality, obtaining

$$\mathbb{P}\left(E_j\right) \leq 9n\left(1-\frac{\alpha}{\rho}\right)^{-2}\rho^{2j}\,.$$

Therefore, $\sum_{j\geq 0}\mathbb{P}\left(E_j\right)$ is finite, and the Borel-Cantelli lemma implies that, with probability one, the event E_j occurs for finitely many $j\in\mathbb{Z}^+$. This, in turn, implies that, with probability one, the sequence $\{\boldsymbol{x}(j)\}$ is Cauchy, and henceforth convergent. Therefore, there exists a $\mathbb{R}^{\mathcal{V}}$-valued random variable $\boldsymbol{x}_\infty$ such that $\lim_j \boldsymbol{x}(j)=\boldsymbol{x}_\infty$ with probability one.

On the other hand, define $g:\mathbb{R}^{\mathcal{V}}\to\mathbb{R}^{\mathcal{V}}$ by $g(\boldsymbol{x})=x-n^{-1}\mathbf{1}^*\boldsymbol{x}\mathbf{1}$. Then, it can be deduced from (18), again using Markov's inequality and the Borel-Cantelli lemma in an analogous fashion, that $g(\boldsymbol{x}(j))=\boldsymbol{z}(j)$ converges to $\mathbf{0}$ with probability one. Then, from the continuity of g, it immediately follows that $g(\boldsymbol{x}_\infty)=0$, i.e. $\boldsymbol{x}_\infty=\hat{y}\mathbf{1}$ for some scalar random variable $\hat{y}$. In order to verify that (19) holds, it is sufficient to observe that $\zeta(j)=n^{-1}\mathbf{1}^*\boldsymbol{x}(j)-y$ is bounded and convergent to $\hat{y}-y$ with probability one. Hence, in particular, $\lim_j \mathbb{E}[\zeta(j)^2]=\mathbb{E}\left[(\hat{y}-y)^2\right]$. Then, it follows from (18) that

$$\frac{\alpha^2}{(1-\alpha)^2} \geq \lim_j \mathbb{E}[\zeta(j)^2] = \mathbb{E}\left[(\hat{y}-y)^2\right]\,.$$

Therefore, (20) follows by simply recalling that $\hat{y}_v(t)=x_v(j)$, for $h_j<t\leq h_{j+1}$.

In order to prove the second part of the claim, first recall that $\alpha_L=\beta_L^{S_L}$, with β_L depending on ε only. Hence, $\alpha_L\leq\rho/2$ for all $S_L\geq\frac{\log\rho^{-1}+\log 2}{\log\beta_L^{-1}}$. Then, for $\delta\in]0,1]$, let $u:=\sqrt{\delta/2}$. It follows from (10), (8), and Proposition 3, that, for

$$n^{-1}||\boldsymbol{e}(t)||^2\leq\delta\,,\qquad \forall t\geq h_j \tag{A.5}$$

to hold, it is sufficient that

$$\frac{\alpha_L}{(1-\alpha_L)}\leq u\,,\qquad \rho^j\leq u/2\,. \tag{A.6}$$

The left inequality in (A.6) is satisfied if $S_L\geq\frac{\log(2u^{-1})}{\log(\beta_L^{-1})}$. On the other hand, the right inequality in (A.6) is satisfied if $j\geq\frac{\log(2u^{-1})}{\log(\rho^{-1})}$. Now, recall that $h_j=\sum_{1\leq i\leq j}l_i=S_L\sum_{1\leq i\leq j}i\leq S_Lj^2$. It follows that

$$h_j\geq S_L\frac{\log^2 u}{\log^2\rho}\geq\frac{1}{\log\beta_L}\frac{\log^3 u}{\log^2\rho}\,,$$

implies (A.5). Then, the upper bound on $\tau_L(\delta)$ displayed in (21) easily follows. In order to prove the bound on $\kappa_L(\delta)$, observe that (1) implies that, for every $v\in\mathcal{V}$,

$$\sum_{1\leq t\leq h_j}\kappa_v(t)\leq\sum_{1\leq i\leq j}Bl_i^3=BS_L^3\sum_{1\leq i\leq j}i^3\leq BS_L^3j^4\,.$$

Finally, (22) follows from analogous arguments. ■

A.3 Proof of Theorem 5

From (25) one has $\zeta(j) = 0$ for all j. On the other hand, in the same way as in the proof of Theorem 4, one can argue that $\lim_j \boldsymbol{z}(j) = \mathbf{0}$ with probability one. Hence $\lim_t \boldsymbol{e}(t) = \mathbf{0}$ with probability one, which is equivalent to the first part of the claim.

Now, let us consider algorithm $\mathcal{A}'_L$. Recall that one has $\alpha_L \leq \rho/2$ for all $S_L \geq \frac{\log \rho^{-1} + \log 2}{\log \beta_L^{-1}}$. Once fixed any such S_L, (25) and (18) imply that

$$n^{-1}\mathbb{E}[||\boldsymbol{e}(t)||] = n^{-1}\mathbb{E}[||\boldsymbol{z}(j)||] \leq \rho^{2j}\,(1-\alpha/\rho)^{-2} \leq \delta$$

for all $t \geq h_j$, if $j \geq \frac{\log \delta^{-1} + \log 4}{2 \log \rho^{-1}}$. The upper bound on $\tau'_L(\delta)$ displayed in (26) then follows upon recalling that $l_j \leq S_L j^2$. The upper bound on $\kappa'_L(\delta)$ displayed in (26) follows using (1). The bounds displayed in (27) follow from similar arguments. ■

A.4 Proof of Theorem 6

From the fact that $\zeta(j) = 0$, and following the same arguments as in the proof of Prop. 3, one gets

$$\mathbb{E}[||\boldsymbol{e}(h_j)||^2] = \mathbb{E}[||\boldsymbol{z}(j)||^2] \leq n \left(\rho^j + \sum_{0 \leq s < j} \rho^{j-s}\alpha \right)^2 \leq n \left(\rho^{2j} + \alpha^2 (1-\rho)^{-2} \right) .$$

■