

# Rates of Convergence for Distributed Average Consensus with Probabilistic Quantization

Tuncer C. Aysal, Mark J. Coates, Michael G. Rabbat  
Department of Electrical and Computer Engineering  
McGill University, Montréal, Québec, Canada

{tuncer.aysal, mark.coates, michael.rabbat}@mcgill.ca

**Abstract**—*Probabilistically quantized distributed averaging (PQDA) is a fully decentralized algorithm for performing average consensus in a network with finite-rate links. At each iteration, nodes exchange quantized messages with their immediate neighbors. Then each node locally computes a weighted average of the messages it received, quantizes this new value using a randomized quantization scheme, and then the whole process is repeated in the next iteration. In our previous work we introduced PQDA and demonstrated that the algorithm almost surely converges to a consensus (i.e., every node converges to the same value). The present article builds upon this work by characterizing the rate of convergence to a consensus. We illustrate that the rate of PQDA is essentially the same as unquantized distributed averaging when the discrepancy among node values is large. When the network has nearly converged and all nodes' values are at one of two neighboring quantization points, then the rate of convergence slows down. We bound the rate of convergence during this final phase by applying lumpability to compress the state space, and then using stochastic comparison methods.*

## I. INTRODUCTION

A fundamental problem in decentralized networked systems is that of having nodes reach a state of agreement. For example, the nodes in a wireless sensor network must be synchronized in order to communicate using a TDMA scheme or to use time-difference-of-arrival measurements for localization and tracking. Similarly, one would like a host of unmanned aerial vehicles to make coordinated decisions on a surveillance strategy. This paper focuses on a prototypical example of agreement in networked systems, namely, the *average consensus* problem: each node initially has a scalar value,  $y_i$ , and the goal is to compute the average,  $\frac{1}{N} \sum_{i=1}^N y_i$  at every node in the network.

*Distributed averaging* (DA) is a simple iterative distributed algorithm for solving the average consensus problem with many attractive properties. The network state is maintained in a vector  $\mathbf{x}(t) \in \mathbb{R}^N$ , where  $x_i(t)$  is the value at node  $i$  after  $t$  iterations, and there are  $N$  nodes in the network. Network connectivity is represented by a graph  $G = (V, E)$ , with vertex set  $V = \{1, \dots, N\}$  and edge set  $E \subseteq V^2$  such that  $(i, j) \in E$  implies that nodes  $i$  and  $j$  communicate directly with each other. (We assume communication is symmetric.) In the  $t+1$ st DA iteration, node  $i$  receives values  $x_j(t)$  from all the nodes  $j$  in its neighborhood and updates its value by

the weighted linear combination,

$$x_i(t+1) = W_{i,i}x_i(t) + \sum_{j:(i,j) \in E} W_{i,j}x_j(t).$$

For a given initial state,  $\mathbf{x}(0)$ , and reasonable choices of weights  $W_{i,j}$ , it is easy to show that  $\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{N} \sum_{i=1}^N x_i(0) \triangleq \bar{x}$ . The DA algorithm was introduced by Tsitsiklis in [17], and has since been pursued in various forms by many other researchers (e.g., [3, 6, 8, 11, 14, 20]).

Of course, in any practical implementation of this algorithm, communication rates between neighboring nodes will be finite, and thus quantization must be applied to  $x_i(t)$  before it can be transmitted. In applications where heavy quantization must be applied (e.g., when executing multiple consensus computations in parallel, so that each packet transmission carries many values), quantization can actually affect convergence properties of the algorithm. Figure 1 shows the trajectories,  $x_i(t)$ , for all nodes superimposed on one set of axes. In Fig. 1(a), nodes apply deterministic uniform quantization with  $\Delta = 0.1$  spacing between quantization points. Although the algorithm converges in this example, clearly the limit is not a consensus; not all nodes arrive at the same value.

In [1, 2] we introduced *probabilistically quantized distributed averaging* (PQDA). Rather than applying deterministic uniform quantization, nodes independently apply a simple randomized quantization scheme. In this scheme, the random quantized value is equal to the original unquantized value in expectation. Through this use of randomization, we guarantee that PQDA converges almost surely to a consensus. However, since  $\bar{x}$  is not divisible by  $\Delta$  in general, the value we converge to is not precisely the average of the initial values. We presented characteristics of the limiting consensus value, in particular, showing that it is equal to  $\bar{x}$  in expectation.

The main contribution of the present paper is to characterize the rates of convergence for PQDA. We show that, when the values  $x_i(t)$  lie in an interval which is large relative to the quantizer precision  $\Delta$ , then PQDA moves at the same rate as regular unquantized consensus. On the other hand, the transition from when node values are all within  $\Delta$  of each other (i.e., each node is either at  $k\Delta$  or  $(k+1)\Delta$ , for some  $k$ ), is slower than unquantized consensus. We present a scheme for characterizing the time from when PQDA iterates enter

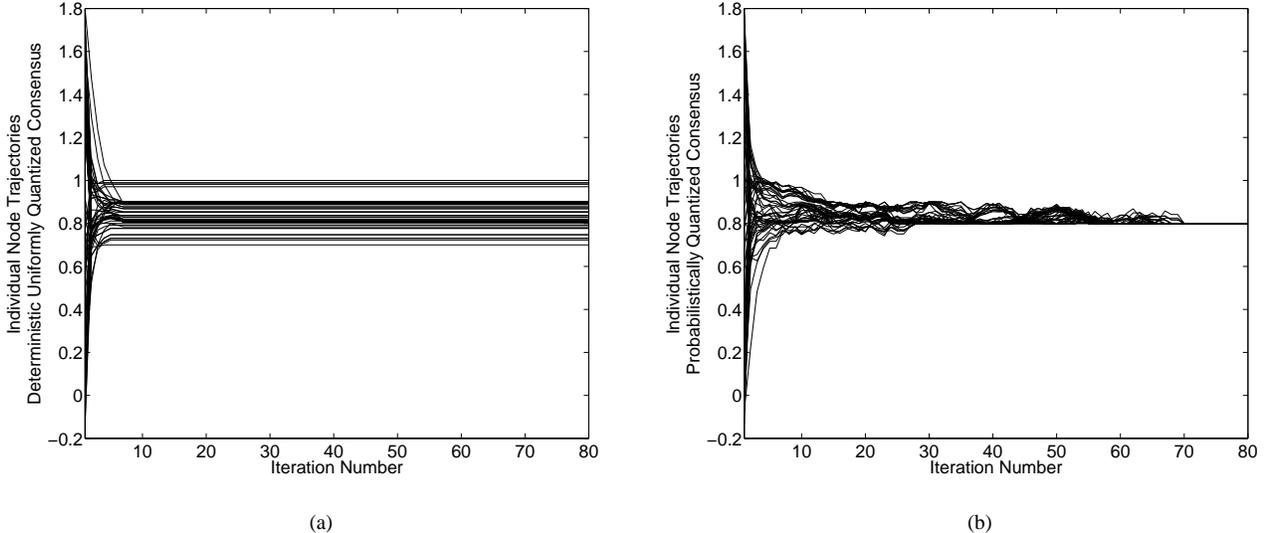


Fig. 1. Individual node trajectories (*i.e.*,  $x_i(t), \forall i$ ) taken by the distributed average consensus using (a:) deterministic uniform quantization and (b:) probabilistic quantization. The number of nodes is  $N = 50$ , the nodes' initial average is  $\bar{x} = 0.85$ , and the quantization resolution is set to  $\Delta = 0.1$ . The consensus value, in this case, is 0.8.

this “final bin” until the algorithm is absorbed at a consensus state.

The remainder of the paper is organized as follows. Section II describes the probabilistic quantization scheme employed by PQDA. Section III formally defines the PQDA algorithm and lists fundamental convergence properties. Section IV explores rates of convergence when the node values are far from a consensus, relative to the quantization precision  $\Delta$ . Section V presents our technique for characterizing convergence rates when all nodes reach the “final bin” and are within  $\Delta$  of each other. We conclude in Section VI. Before proceeding, we briefly review related work.

#### A. Related Work

While there exists a substantial body of work on average consensus protocols with infinite precision and noise-free peer-to-peer communications, little research has been done introducing distortions in the message exchange. In [12], Rabani, Sinclair, and Wanka examine distributed averaging as a mechanism for load balancing in distributed computing systems. They provide a bound on the divergence between quantized consensus trajectories,  $\mathbf{q}(t)$ , and the trajectories which would be taken by an unquantized averaging algorithms. The bound reduces to looking at properties of the averaging matrix  $\mathbf{W}$ .

Recently, Yildiz and Scaglione, in [22], explored the impact of quantization noise through modification of the consensus algorithm proposed by Xiao and Boyd [21]. They note that the noise component in [21] can be considered as the quantization noise and they develop an algorithm for predicting neighbors' unquantized values in order to correct errors introduced by quantization [22]. Simulation studies for small  $N$  indicate that if the increasing correlation among the node states is taken into account, the variance of

the quantization noise diminishes and nodes converge to a consensus.

Kashyap *et al.* examine the effects of quantization in consensus algorithms from a different point of view [7]. They require that the network average,  $\bar{x} = 1/N \sum_{i=1}^N x_i(t)$ , be preserved at every iteration. To do this using quantized transmissions, nodes must carefully account for round-off errors. Suppose we have a network of  $N$  nodes and let  $\Delta$  denote the “quantization resolution” or distance between to quantization lattice points. If  $\bar{x}$  is not a multiple of  $N\Delta$ , then it is not possible for the network to reach a strict consensus (*i.e.*,  $\lim_{t \rightarrow \infty} \max_{i,j} |x_i(t) - x_j(t)| = 0$ ) while also preserving the network average,  $\bar{x}$ , since nodes only ever exchange units of  $\Delta$ . Instead, Kashyap *et. al* define the notion of a “quantized consensus” to be such that all  $x_i(t)$  take on one of two neighboring quantization values while preserving the network average; *i.e.*,  $x_i(t) \in \{k\Delta, (k+1)\Delta\}$  for all  $i$  and some  $k$ , and  $\sum_i x_i(T) = N\bar{x}$ . They show that, under reasonable conditions, their algorithm will converge to a quantized consensus. However, the quantized consensus is clearly not a strict consensus, *i.e.*, all nodes do not have the same value. Even when the algorithm has converged, as many as half the nodes in the network may have different values. If nodes are strategizing and/or performing actions based on these values (*e.g.*, flight formation), then differing values may lead to inconsistent behavior.

Of note is that the related works discussed above all utilize standard deterministic uniform quantization schemes to quantize the data. In contrast to [22], where quantization noise terms are modeled as independent zero-mean random variables, we explicitly introduce randomization in our quantization procedure. Careful analysis of this randomization allows us to provide concrete theoretical rates of convergence in addition to empirical results. Moreover, the algorithm

proposed in this paper converges to a strict consensus, as opposed to the approximate “quantized consensus” achieved in [7] which is clearly not a strict consensus and does not preserve the average, however the network average may not be preserved perfectly by our algorithm. In addition to proving that our algorithm converges, we show that the network average is preserved in expectation, and we characterize the limiting mean squared error between the consensus value and the network average.

## II. PROBABILISTIC QUANTIZATION

Suppose that the scalar value  $x_i \in [-U, U] \subset \mathbb{R}$  lies in a bounded interval. Furthermore, suppose that we wish to obtain a quantized message  $q_i$  with length  $l$  bits, where  $l$  is application dependent. We therefore have  $L = 2^l$  quantization points given by the set  $\tau = \{\tau_1, \tau_2, \dots, \tau_L\}$ . The points are uniformly spaced such that  $\Delta = \tau_{j+1} - \tau_j$  for  $j \in \{1, 2, \dots, L-1\}$ . It follows that  $\Delta = 2U/(2^l - 1)$ . Now suppose  $x_i \in [\tau_j, \tau_{j+1})$  and let  $q_i \triangleq \mathcal{Q}(x_i)$  where  $\mathcal{Q}(\cdot)$  denotes the PQ operation. Then  $x_i$  is quantized in a probabilistic manner:

$$\Pr\{q_i = \tau_{j+1}\} = r, \text{ and, } \Pr\{q_i = \tau_j\} = 1 - r \quad (1)$$

where  $r = (x_i - \tau_j)/\Delta$ . The following lemma, adopted from [19], discusses two important properties of PQ.

*Lemma 1 ([19]):* Suppose  $x_i \in [\tau_j, \tau_{j+1})$  and let  $q_i$  be an  $l$ -bit quantization of  $x_i \in [-U, U]$ . The message  $q_i$  is an unbiased representation of  $x_i$ , i.e.,

$$\mathbb{E}\{q_i\} = x_i, \text{ and, } \mathbb{E}\{(q_i - x_i)^2\} \leq \frac{U^2}{(2^l - 1)^2} \equiv \frac{\Delta^2}{4}. \quad (2)$$

## III. PROBABILISTICALLY QUANTIZED DISTRIBUTED AVERAGING

This section describes the PQDA algorithm introduced in [1, 2]. We assume each node begins with initial condition  $x_i(0) = y_i$ , as before. At iteration  $t$ , nodes independently quantize their values via

$$q_i(t) = \mathcal{Q}(x_i(t)). \quad (3)$$

These quantized values are exchanged among neighbors, and the usual consensus linear update is performed via

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{q}(t). \quad (4)$$

Let  $\mathbf{J} = N^{-1}\mathbf{1}\mathbf{1}^T$ . Following Xiao and Boyd [20], we assume that  $\mathbf{W}$  is a symmetric stochastic matrix which satisfies  $\mathbf{W}\mathbf{1} = \mathbf{1}$ ,  $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$ , and  $\rho(\mathbf{W} - \mathbf{J}) < 1$ , where  $\rho(\cdot)$  denotes the spectral radius. These properties suffice to guarantee that  $\mathbf{x}(t)$  converges to the average consensus when perfect, unquantized transmissions are used [20].

Due to our use of probabilistic quantization,  $\mathbf{x}(t)$  is a random process, and it is natural to ask: *Does  $\mathbf{x}(t)$  converge, and if so, does it converge to a consensus?* In [1, 2], we show that  $\mathbf{x}(t)$  indeed converges almost surely to a consensus.

*Theorem 1 ([1, 2]):* Let  $\mathbf{x}(t)$  be the sequence of iterates generated by the PQDA algorithm defined in (3) and (4). Then

$$\Pr\left(\lim_{t \rightarrow \infty} \mathbf{x}(t) = \tau^* \mathbf{1}\right) = 1,$$

for some quantization point  $\tau^* \in \tau$ .

Proofs of all the theorems stated in this section can be found in [2].

The limiting quantization point,  $\tau^*$ , is not equal to  $\bar{x}$  in general, since  $\bar{x}$  is not necessarily divisible by  $\Delta$ . This compromise is required if we want to guarantee convergence to a consensus in the quantized setting. The theorem above only guarantees to a convergence and does not say anything about how close  $\tau^*$  will be to  $\bar{x}$ . The following results quantify the accuracy of PQDA.

*Theorem 2 ([1, 2]):* For the sequence  $\mathbf{x}(t)$  of iterates generated by PQDA steps (3) and (4),

$$\mathbb{E}\left\{\lim_{t \rightarrow \infty} \mathbf{x}(t)\right\} = \bar{x} \mathbf{1}.$$

*Theorem 3 ([2]):* For the sequence  $\mathbf{x}(t)$  of iterates generated by PQDA steps (3) and (4),

$$\lim_{t \rightarrow \infty} \lim_{N \rightarrow \infty} \mathbb{E}\left\{\frac{1}{N} \|\mathbf{q}(t) - \bar{x} \mathbf{1}\|\right\} \leq \frac{\Delta}{2} \frac{1}{1 - \rho(\mathbf{W} - \mathbf{J})}.$$

Thus, PQDA converges to  $\bar{x}$  in expectation. The upper bound on the standard deviation given in Theorem 3 contains two terms. First, the  $\Delta/2$  worst-case error is due to our use of quantization. The second term,  $(1 - \rho(\mathbf{W} - \mathbf{J}))^{-1}$ , relates to how fast information can diffuse over the network, which is directly a function of the network topology. The more iterations that are required to reach consensus, the more time probabilistic quantization must be applied, and each time we quantize we potentially introduce additional errors.

## IV. CONVERGENCE ANALYSIS: FAR FROM CONSENSUS

To get a handle on the rate of convergence of PQDA, we begin by studying how spread out the individual node values are over the interval  $[-U, U]$ . Let  $I(t) = [\min_i q_i(t), \max_i q_i(t)]$ . It is easy to show (see [2]) that  $I(t+1) \subseteq I(t)$ . This follows since, by our constraints on  $\mathbf{W}$ , each  $x_i(t+1)$  is a convex combination of components of  $\mathbf{q}(t)$ . Therefore, after iterating and quantizing, the minimum value can never decrease and the maximum cannot increase. The analogous result also holds if one defines  $I(t)$  in terms of the components of minimum length quantization range of  $\mathbf{x}(t)$ .

Next, let  $r_{\mathbf{q}}(t) = \max_{i,j} q_i - q_j$  denote the range of quantized network values at time  $t$ . Clearly, since  $\mathbf{x}(t)$  converges to a consensus, eventually  $r_{\mathbf{q}}(t)$  tends to zero. Our first results dealing with rates of convergence for PQDA examine the rate at which  $r_{\mathbf{q}}(t)$  tends to zero.

*Theorem 4 ([2]):* Let  $r_{\mathbf{q}}(t)$  be as above, and let  $r_{\mathbf{x}}(0) = \max_{i,j} x_i(0) - x_j(0)$ . Then

$$\mathbb{E}\{r_{\mathbf{q}}(t)\} \leq \sqrt{\frac{N-1}{N}} \rho^t(\mathbf{W} - \mathbf{J}) r_{\mathbf{x}}(0) + 2\Delta.$$

Ignoring the  $2\Delta$  term for a moment, we see that the range of values decays geometrically at a rate proportional to the spectral gap of  $\mathbf{W}$ ,  $\rho(\mathbf{W} - \mathbf{J})$ . In fact, this is the same rate at which standard, unquantized consensus converges, so we see that when  $r_{\mathbf{q}}(t)$  is large, errors due to quantization do not significantly hamper the rate of convergence. On

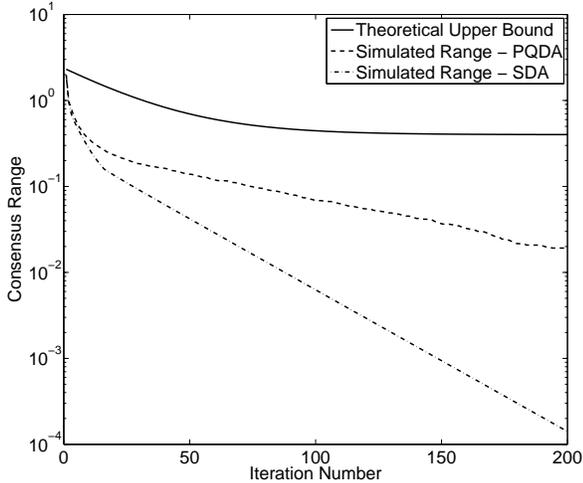


Fig. 2. This figure plots the range,  $r_{\mathbf{q}}(t)$ , as a function of iteration number. The solid curve shows our theoretical upper bound, the dashed curve is generated empirically by averaging over 5000 Monte Carlo simulations. Also plotted for reference is the curve of standard unquantized DA (SDA) that is generated empirically by averaging over 5000 Monte Carlo simulations. For the first 20 iterations, PQDA converges at essentially the same rate as unquantized distributed averaging. However, around 20 iterations, all of the values  $q_i(t)$  are within one or two  $\Delta$ , and convergence slows down considerably.

the other hand, this result is loose in the sense that it only implies  $r_{\mathbf{q}}(t) \leq 2\Delta$ , at best. However, we know that PQDA converges almost surely, so eventually  $r_{\mathbf{q}}(t) = 0$ . The gap is due to worst-case analysis employed to obtain the result. Empirically, we observe that  $r_{\mathbf{q}}(t)$  does indeed tend to zero, but when convergence is nearly obtained (*i.e.*, when  $r_{\mathbf{q}}(t) = \Delta$  or  $2\Delta$ , the rate of decay is significantly slower than that of unquantized consensus. Figure 2 depicts  $r_{\mathbf{q}}(t)$  as a function of iteration. The plot shows our bound, and a simulated curve obtained by averaging over 5000 Monte Carlo trials. In this simulation we use a network of  $N = 50$  nodes, with quantizer precision  $\Delta = 0.2$ . In the figure we see that around the time when  $r_{\mathbf{q}}(t) = 2\Delta$  (roughly the first 20 iterations, also plotted the simulated range of the unquantized standard DA (SDA)), the rate of convergence undergoes a smooth transition and slows down significantly. Next, we look at characterizing the rate of convergence during this final phase, focusing on the final step, without loss of generality, from  $r_{\mathbf{q}}(t) = 1$  to  $r_{\mathbf{q}}(t) = 0$ .

## V. CONVERGENCE ANALYSIS: FINAL BIN

Next, we consider the convergence of the probabilistically quantized consensus algorithm after the time step that all the node state values are in the  $\Delta$  length quantization bin, *i.e.*,  $r_{\mathbf{q}}(t) = \Delta$ . Without loss of generality, we assume that  $\mathbf{x}(t' + t) \in [0, 1]^N$  indicating that  $\mathbf{q}(t) \in \{0, 1\}^N$ , and that  $t' = 0$ .

Let us construct a Markov chain  $\mathbb{Q}$  with  $2^N$  states, where the chain states are given by  $\mathbb{Q} = \{\mathbf{q}^0, \mathbf{q}^1, \dots, \mathbf{q}^{2^N}\}$ , where  $\mathbf{q}^i \in \{0, 1\}^N$  for  $i = 1, 2, \dots, N$ . Hence  $\mathbf{q}(t) \in \mathbb{Q}$  for  $t \geq 1$ . The entries  $P_{ij}$  of the transition probability matrix  $\mathbf{P}$  is given by

$$P_{ij} = \Pr\{\mathbf{q}(t+1) = \mathbf{q}^j | \mathbf{q}(t) = \mathbf{q}^i\}. \quad (5)$$

Using the state recursion and utilizing the conditional independence of  $v_i(t)$  samples [19], we arrive at

$$P_{ij} = \prod_{k=1}^N 1 - (q_k^j - \mathbf{w}^k \mathbf{q}^i) \quad (6)$$

where  $q_k^j$  and  $\mathbf{w}^k$  denote the  $k$ -th element of the state  $j$  and the  $k$ -th row of the weight matrix, respectively. Recall that the constructed Markov chain is an absorbing one. Renumbering the states in  $\mathbf{P}$  so that the transient states comes first yields the canonical form:

$$\mathbf{C} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (7)$$

Here  $\mathbf{I}$  and  $\mathbf{0}$  are identity and zero matrices with appropriate sizes, respectively. The fundamental matrix of a Markov chain is given by,  $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$ . The entry  $F_{ij}$  of  $\mathbf{F}$  gives the expected number of times that the process is in the transient state  $j$  if it is started in the transient state  $i$ . Let  $\eta_i$  be the expected number of steps before the chain is absorbed, given that the chain starts in state  $i$ , and let  $\boldsymbol{\eta}$  be the column vector whose  $i$ -th entry is  $\eta_i$ . Then  $\boldsymbol{\nu} = \mathbf{F}\mathbf{1}$ . Moreover, Let  $B_{ij}$  be the probability that an absorbing chain will be absorbed in the absorbing state  $j$  if it starts in the transient state  $i$ . Let  $\mathbf{B}$  be the matrix with entries  $B_{ij}$ . Then,  $\mathbf{B} = \mathbf{F}\mathbf{R}$ , where  $\mathbf{R}$  is as in the canonical form.

Unfortunately, the exact computation of the expected number of steps to absorption requires inversion of matrix with size  $2^N - 1 \times 2^N - 1$ , which is a challenging task. To overcome this drawback, in the following, we investigate a Markov chain with  $N/2 + 1$  states, requiring the inversion of a matrix of manageable size  $N/2 \times N/2$  with trade-off of yielding only bounds on the expected number of steps to absorption.

We employ the reduction techniques outlined in [16] to analyze the convergence of the Markov chain. We need to introduce a modification, since it is not computationally feasible to compute or process the  $2^N - 1 \times 2^N - 1$  probability transition matrix, as required in [16].

### A. Preliminaries

We first introduce some notation and recall some definitions. The definitions are extracted from [16] and [15].

*Definition 1 (Strong order):* Let  $U$  and  $V$  be two  $\mathbb{R}^n$ -valued random variables.  $U$  is smaller than  $V$  in the sense of *strong order*, if and only if for all nondecreasing real functions  $f$  from  $\mathbb{R}^n$  (in the sense of componentwise ordering on  $\mathbb{R}^n$ ), we have  $\mathbf{E}f(U) \leq \mathbf{E}f(V)$ , provided that the expectations exist.

If this conditions holds, then it is denoted  $U \leq_{st} V$ .  $\leq_{st}$  is a partial order, *i.e.*, reflexive, antisymmetric and transitive. In the case where  $U$  and  $V$  are  $\{1, \dots, m\}$ -valued random variables,  $U \leq_{st} V$  if and only if  $\forall i \in E$ ,  $\sum_{k \geq i} u(i) \leq \sum_{k \geq i} v(i)$ , where  $u$  and  $v$  are the probability distributions of  $U$  and  $V$ . This last relation is also denoted  $u \leq_{st} v$ .

*Definition 2 ( $\leq_{st}$  comparison of stochastic matrices):* Let  $A$  and  $B$  be two  $\eta \times \eta$  stochastic matrices indexed by

elements of  $E$ .

$$A \leq_{st} B \iff \forall i \in E, A(i, \cdot) \leq B(i, \cdot). \quad (8)$$

*Definition 3 ( $\leq_{st}$ -Monotonicity):* Let  $A$  be an  $\eta \times \eta$  stochastic matrix. We say that  $A$  is monotone in the sense of  $\leq_{st}$  if and only if

$$\forall i < \eta, \quad A(i, \cdot) \leq_{st} A(i+1, \cdot). \quad (9)$$

*Definition 4 ( $\leq_{st}$ -monotone optimal matrix):* Let  $A$  be a stochastic matrix.  $M$  is the  $\leq_{st}$ -monotone optimal matrix with respect to  $A$  if it is the lowest (the greatest) with respect to the partial order  $\leq_{(st)}$  such that  $A \leq_{st} M$  ( $M \leq_{st} A$ ).

### B. Truffet's Reduction techniques

Let  $\chi = (\pi, P)$  be a discrete-time homogeneous Markov chain with totally ordered state space  $E = \{1, \dots, \eta\}$  (with respect to a canonical order on  $\mathbb{N}$ ,  $\leq$ ). Here  $\pi$  is the initial distribution and  $P$  is the associated Markov transition matrix. The random variable  $\chi(t)$  represents the state of the system at epoch  $t$ . This chain is used to represent the progression through the quantization states, so the states are the  $\mathbf{q}$ -vectors and  $\eta = 2^N - 1$ . Let  $k(\mathbf{q}) = \min(\sum_{i=1}^N q_i, N - \sum_{i=1}^N q_i)$ . This value captures the minimum Manhattan distance of  $\mathbf{q}$  to an absorbing state. We establish an ordering by requiring that for any two states labelled  $i$  and  $j$ , such that  $i < j$ , the associated  $\mathbf{q}$ -vectors, denoted  $\mathbf{q}^{(i)}$  and  $\mathbf{q}^{(j)}$  satisfy  $k(\mathbf{q}^{(i)}) \geq k(\mathbf{q}^{(j)})$ . If the equality holds, then states are ordered by considering the  $\mathbf{q}$  vectors as binary numbers.

Following [16], consider the surjective mapping  $\epsilon : E \rightarrow \Sigma = \{1, \dots, L\}$ ,  $0 < L < \eta$  such that  $\mathbb{E} = (\epsilon^{-1}(I))_{I=1, \dots, L}$  are lumped states that form a partition of  $E$ . We assume  $\epsilon$  is non-decreasing, and indeed, in our case, we define  $\epsilon(\chi(j)) = N/2 - k(\mathbf{q}^{(j)}) + 1$ , implying  $L = N/2 + 1$ . Denote  $\epsilon(\chi) \triangleq (\epsilon(\chi(t)))_{t \in \mathbb{N}}$ .

The reduction technique presented by Truffet strives to generate stochastic matrices that bound the transition matrix of the lumped Markov chain. The goal is to develop these matrices such that the associated Markov chains, denoted  $\underline{Y}$  and  $\overline{Y}$ , satisfy, for all  $t$ :

$$\begin{aligned} (\underline{Y}(0), \dots, \underline{Y}(t)) &\leq_{st} (\epsilon(\chi(0)), \dots, \epsilon(\chi(t))) \\ &\leq_{st} ((\overline{Y})(0), \dots, \overline{Y}(t)). \end{aligned}$$

Let us denote the cardinality of  $\epsilon^{-1}(I)$  as  $\eta_I$ ,  $I = 1, \dots, L$ . We assume that for  $I = 1, \dots, L$ ,  $E(I) = \epsilon^{-1}(I) = [a_I, b_I]$ , with  $a_{I+1} = b_I + 1$ ,  $I = 1, \dots, L-1$  and  $a_1 = 1$  and  $b_L = \eta$ . Denote by  $\mathcal{A}(E)$  the set of all probability distributions on  $E$ , and equivalently denote by  $\mathcal{A}(\Sigma)$  the set of all probability distributions on  $\Sigma$ .

We recall the result concerning ordinary lumpability, presented in [16], as adapted from [4]. Consider an  $E$ -valued Discrete-time Markov Chain  $\Phi = (\alpha, A)$ .

*Result 1:* If  $\forall I, J \in \Sigma, \forall i \in E(I), o(I, J) = A(i, E(J))$  does not depend on  $i$ , then matrix  $A$  is said to be ordinary lumpable with respect to partition  $\mathbb{E}$  and  $\forall \alpha \in \mathcal{A}(E)$  the process  $\epsilon(\Phi)$  is a  $\Sigma$ -valued discrete-time Markov Chain with  $L \times L$  transition probability matrix  $A_o^\epsilon$  defined by

$$\forall I, J \in \Sigma, \quad A_o^\epsilon(I, J) = o(I, J), \quad (10)$$

and initial condition  $\alpha^\epsilon$ . The Markov chain  $\Phi$  is said to be ordinary lumpable with respect to  $\mathbb{E}$  and matrix  $A_o^\epsilon$  is referred to as an *ordinary lumped* matrix.

The key result from [16] (Lemma 2.1) relies on identifying a  $\leq_{st}$ -monotone optimal matrix  $\overline{M}$  and a discrete Markov chain  $\overline{\chi} = (\overline{\pi}, \overline{P})$ , such that (i)  $P \leq_{st} \overline{M} \leq_{st} \overline{P}$ ; (ii)  $\overline{P}$  is ordinary lumpable with respect to a partition  $\mathbb{E}$ ; and (iii)  $\pi \leq_{st} \overline{\pi}$ . Under these conditions, the  $\Sigma$ -valued Markov chain  $\overline{Y}_o = (\overline{\pi}^\epsilon, \overline{P}_o^\epsilon)$  with  $\overline{P}_o^\epsilon$  defined by (10) (with  $A = \overline{P}$ ) is such that  $\epsilon(\chi) \leq_{st} \overline{Y}_o$ . Note that the result holds if  $\overline{M}$  is merely  $\leq_{st}$ -monotone (not optimal), but the resultant bounds on  $\epsilon(\chi)$  are not as tight.

In our case, we can set  $\overline{\pi} = \pi$  to satisfy the third condition. Lemma 3.1 from [16] provides a recipe for constructing an optimal ordinary lumped matrix  $\overline{P}$  given the  $\leq_{st}$ -monotone matrix  $\overline{M}$ . We first recall the definition of upper optimality in this setting:

*Definition 5:* Let  $M$  be an  $\eta \times \eta$  monotone matrix.  $O$  is an optimal upper  $L \times L$  matrix if and only if  $\forall I, K \in \Sigma$ , the quantity  $\sum_{J=K}^L O(I, J)$  is the smallest quantity such that:

$$\forall k \in E(I), \quad \sum_{J=K}^N \sum_{j \in E(J)} M(k, j) \leq \sum_{J=K}^N O(I, J)$$

*Lemma 2 ([16], Lemma 3.1):* For any arbitrary partition  $\mathbb{E}$ , and  $\leq_{st}$ -monotone matrix  $\overline{M}$ , there exists an optimal ordinary lumped matrix  $\overline{P}_{o, opt}^\epsilon$ , which is defined by:

$$\forall I, J \in \Sigma, \quad \overline{P}_{o, opt}^\epsilon(I, J) = \sum_{j=a_J}^{j=b_J} \overline{M}(b_I, j) \quad (11)$$

Truffet provides an algorithm for identifying  $\overline{M}$ , but it requires calculation and processing of every state in the original  $\eta \times \eta$  Markov chain. The construction of  $\overline{P}_{o, opt}^\epsilon(I, J)$  (and hence  $\overline{Y}$ ) does not require specification of the individual elements of  $\overline{M}(b_I, \cdot)$ ; it suffices to determine the sums over partitions. However, even this task, with  $\eta = 2^N - 1$ , is not computationally feasible in our case. We note that it is not essential to identify  $\overline{P}_{o, opt}^\epsilon$ ; if we can specify an  $L \times L$  matrix  $\overline{P}_o^\epsilon(I, J)$  satisfying  $\overline{P}_{o, opt}^\epsilon \leq \overline{P}_o^\epsilon$ , then we achieve a (potentially looser) bound.

The methodology for constructing a lower bounding Markov chain  $\underline{Y}_o$  such that  $\underline{Y}_o \leq_{st} \epsilon(\chi)$  is directly analogous to the method for  $\overline{Y}_o$ . We now present a method, for our specific problem scenario, for constructing a  $\underline{P}_o^\epsilon(I, J)$  that does not involve direct specification of  $\underline{M}$  or access to all states in the chain  $\chi$ .

### C. Convergence results

We identify a partition  $\Sigma$  defined by the values  $\epsilon(\chi(j)) = N/2 - k(\mathbf{q}^{(j)}) + 1$ . Let us denote, for  $J \in \Sigma$ ,  $k(J) \triangleq (k(\mathbf{q}^{(j)}))$  for any  $j \in J$ . The key observation is the following. For  $i \in I$  and  $I \in \Sigma$ ,

$$\begin{aligned} P(i, J) &\triangleq \sum_{j=a_J}^{j=b_J} P(i, j) \\ &= \begin{cases} B(k(J); W\mathbf{q}^{(i)}) + B(N - k(J); W\mathbf{q}^{(i)}) & k(J) \neq N/2 \\ B(k(J); W\mathbf{q}^{(i)}) & k(J) = N/2 \end{cases} \end{aligned} \quad (12)$$

where  $B(\ell; \mathbf{p})$  denotes the Poisson-Binomial distribution,

$$B(\ell; p) = \left\{ \prod_{j=1}^n (1 - p_j) \right\} \sum_{1 \leq j_1 < \dots < j_\ell \leq n} \prod_{z=1}^{\ell} \frac{p_{j_z}}{1 - p_{j_z}} \quad (13)$$

We can construct the matrix  $\bar{P}_o^\epsilon$  by ensuring that for all  $i \in I$  and  $I, K \in \Sigma$ ,

$$\sum_{J=K}^L \bar{P}_o^\epsilon(I, J) \leq \sum_{J=K}^L \sum_{j=a_J}^{j=b_J} P(i, j)$$

We can achieve this by upper-bounding sums of the expressions in (12).

Let  $\#\text{neigh}(i)$  denote the cardinality of the set of neighbours of the nodes with value 1 in state  $\mathbf{q}^{(i)}$ . Suppose that the matrix  $W$  satisfies the condition that for all  $i, i' \in I$  and  $I, K \in \Sigma$ ,

$$\sum_{J=K}^L \sum_{j=a_J}^{j=b_J} P(i, j) \leq \sum_{J=K}^L \sum_{j=a_J}^{j=b_J} P(i', j)$$

if  $\#\text{neigh}(i) \leq \#\text{neigh}(i')$ .

If we can find the set of states  $I^*$  such that  $\#\text{neigh}(i^*) \leq \#\text{neigh}(i')$  for all  $i^* \in I^*, i' \in I$ , and this set of states is small for each  $I$ , then we can perform a two-step reduction procedure. We first eliminate all rows except those belonging to the sets  $I^*, I \in \Sigma$ . Then we can apply the procedure outlined in [16] to identify a suitable  $\bar{P}_o^\epsilon$ . The reason for pursuing this approach is that very effective heuristics can be developed for searching for the set  $I^*$ , whereas direct identification of  $\bar{P}_o^\epsilon$  is intractable for large  $N$ .

If it is impossible to identify the states belonging to  $I^*$ , then it may be possible to upper bound  $\#\text{neigh}(i^*)$  and lower bound the variance and third-moment of the non-zero component of  $P(i^*, \cdot)$ . Then we can develop bounds for  $P(i^*, \cdot)$  itself, which is a Poisson-Binomial, using an approximation scheme based the Krawtchouk expansion outlined in Section V-D. The upper bound forms a suitable  $\bar{P}_o^\epsilon$ .

#### D. The Krawtchouk Expansion

Let  $S_N = X_1 + X_2 + \dots + X_N$  denote the sum of  $N$  independent Bernoulli random variables,  $X_i$ , each having success probabilities

$$\Pr\{X_j = 1\} = 1 - \Pr\{X_j = 0\} = p_j \in [0, 1] \quad (14)$$

for  $j = 1, 2, \dots, N$ . The distribution  $\Pr\{S_N = n\}$  is commonly referred to as the Poisson-Binomial distribution and is denoted  $B(N, \mathbf{p})$ , where  $\mathbf{p}$  denotes the vector of success probabilities; the exact expression for  $B(N, \mathbf{p})$  is given by (13). Since  $B(N, \mathbf{p})$  has a complicated structure, it is often approximated by other distributions. The most notable are normal approximations and the Edgeworth expansion [10, 18], and Poisson approximations and expansion related to Charlier polynomials [5, 9]. In the following, we consider the approximation of the distribution of  $S_N$  by the Binomial distribution  $\mathcal{B}(N, p)$  with parameter  $N$  and arbitrary success probability  $p$  proposed by Roos [13] due to its ability to provide point metric bounds requires to bound the considered

convergence time. In the following, we define the point metric bound [13].

*Definition 6:* Let  $Q_1$  and  $Q_2$  denote finite signed measures which are concentrated on  $\mathbb{Z}_+ = \{0, 1, \dots\}$  and satisfy that  $Q_1(\mathbb{Z}_+) = Q_2(\mathbb{Z}_+)$ , then, the point metric measure between  $Q_1$  and  $Q_2$  is given by

$$d(Q_1, Q_2) = \sup_{m \in \mathbb{Z}_+} |Q_1(\{m\}) - Q_2(\{m\})|. \quad (15)$$

Before, we present the main result of Roos, let us introduce the following notations to simplify the presentation:

$$\bar{p} = \frac{1}{N} \sum_{j=1}^N p_j, \quad \eta(p) = 2\gamma_2(p) + \gamma_1^2(p), \quad \theta(p) = \frac{\eta(p)}{2Npq} \quad (16)$$

with  $\gamma_k(p) = \sum_{j=1}^N (p - p_j)^k$ . The following theorem, adopted from [13], gives a bound on the point metric measure for approximating the Poisson-Binomial distribution with a Binomial one.

*Theorem 5:* Let  $n \in \{2, 3, \dots\}$  and  $0 < x_1 < x_2 < x_3 < n + 1$  be the zeros of  $K_3(x, n + 1, \bar{p})$ , where  $K_j$  denotes the Krawtchouk polynomial of degree  $j$ . Then,  $d(B(N, \mathbf{p}), \mathcal{B}(N, \bar{p})) = H + R$ , where

$$H = \frac{\gamma_2}{N(N-1)[\bar{p}\bar{q}]^2} \times \max\{|K_2(\lfloor x_i \rfloor, N, \bar{p})| b(\lfloor x_i \rfloor, N, \bar{p}) | i \in \{1, 2, 3\}\} \quad (17)$$

and

$$|R| \leq |\gamma_3| \min \left\{ \frac{2.398}{[N\bar{p}\bar{q}]^2}, 1 \right\} + \min \left\{ \frac{1.627\theta^2(1 - 0.75\sqrt{\theta})}{\sqrt{N\bar{p}\bar{q}}(1 - \sqrt{\theta})^2}, 3.695\gamma_2^2(1 + 2\sqrt{\gamma_2} \exp(4\gamma_2)) \right\} \quad (18)$$

with  $\gamma_k = \gamma_k(\bar{p})$ ,  $\theta = \theta(\bar{p})$  and  $\bar{q} = 1 - \bar{p}$ .

Note that the point metric bound includes the Krawtchouk polynomials given by

$$K_j(x, N, p) = \sum_{k=0}^j \binom{N-x}{j-k} \binom{x}{k} (-p)^{j-k} q^k. \quad (19)$$

Note that we are specifically interested Krawtchouk polynomials of degree two and three; finding the roots of  $K_3$  and evaluating those roots in  $K_2$ . From the theory of orthogonal polynomials, it follows that zeros of the Krawtchouk polynomials are real, simple and lie in the interval  $(0, N)$  [13]. Hence, one can determine the roots of  $K_3$  using numerical techniques. Moreover, closed-form expressions exist for  $K_2$  rendering the evaluation of obtained roots trivial.

#### E. Example

In this section, we include the results of a convergence analysis of the  $W$  matrix used in simulations described in Section IV. For this matrix, we perform a branch-and-bound search procedure to identify the sets  $I^*$ . This procedure is a greedy search that initially selects  $K$  of the sets with  $k(\mathbf{q}^{(j)}) = 1$ , choosing those with the smallest neighbour

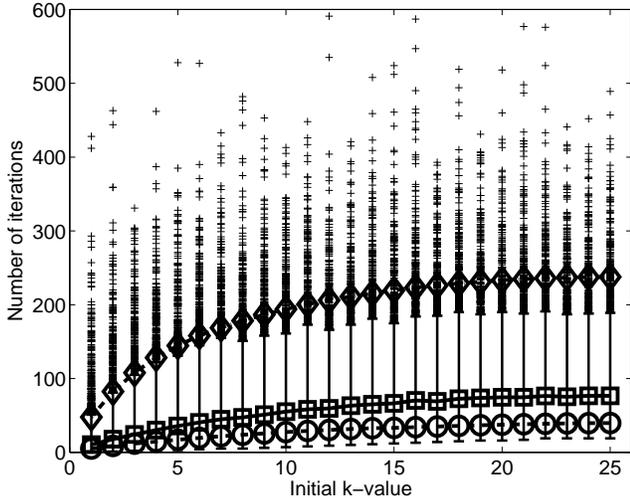


Fig. 3. Bounds on the expected number of iterations to convergence after entry into the final bin. The lines with circle and diamond markers show the lower and upper bounds, as derived from the branch-and-bound search, combined with the lumped-state Markov chain analysis. The empirical mean, derived from 5000 trials for each initial k-value, is depicted by the line with square markers. Error bars depict 5 and 95 percentiles; outliers are shown with the “+” marker.

cardinalities ( $K$  is an algorithm constant defining the degree of branching in the search). Subsequently it attempts to grow each of these sets by adding one neighbour. Through this process it creates  $K'$  sets, but at each step it retains only the  $K$  with smallest neighbour cardinalities. The process ends when the examined sets satisfy  $k(\mathbf{q}^{(j)}) = N/2$ .

We now apply the procedure of [16] to develop stochastic matrices that provide upper- and lower- bounds on the expected time to absorption for the original Markov chain. Figure 3 depicts the results, showing the upper and lower bounds (dashed) and the expected time to convergence, estimated empirically, by running 5000 simulations for each value of  $k(\mathbf{q}^{(i)})$  from random initial states. The significant number of outliers and the substantial variance indicate the value of analytical bounds. The state-space is very large even with only 50 nodes and it is very difficult to reliably evaluate convergence times through empirical studies.

#### F. An Empirical Observation

We investigate the average convergence time of the distributed average consensus using probabilistic quantization for varying  $\Delta \in \{0.05, 0.1, 0.2\}$  against the number of nodes in the network, Fig. 4(a) and (b). We also show the average number of iterations taken to achieve the final quantization bin. Moreover, Fig. 4(c) and (d) plots the average normalized distance to the closest absorbing state at the first time step when all the quantized node state values are in the final quantization bin. The initial state averages are  $\bar{\mathbf{x}}(0) = 0.85$  and  $\bar{\mathbf{x}}(0) = 0.90$ , and the connectivity radius is  $d = \sqrt{4 \log(N)/N}$ . Each data point is an ensemble average of 10000 trials. Note that the convergence time increases with the number of nodes in the network. The plots suggest that the number of iterations taken by the PQDA algorithm to converge to final quantization bin decreases as  $\Delta$  increases. This can be seen by noting that the algorithm has to go

through less “averaging” (multiplication with the weight matrix) before arriving at the final bin. It is hence clear that the algorithm needs to run for a smaller number of iterations to arrive at a larger final bin size.

On the other hand, the expected number of iterations taken to achieve consensus is dominated by the number of iterations taken to converge to an absorbing state after all the node values are in the final bin. Probabilistic quantization is the dominant effect in the final bin. In fact, the time taken to converge to an absorbing state is heavily dependent on the distance to that absorbing state at the first time step when all values enter the final bin. This distance is affected by two factors. First, if more averaging operations occur prior to the instant  $t_\Delta = \min\{t : r_{\mathbf{x}}(t) \leq \Delta\}$ , then there is more uniformity in the values, decreasing the distance from each  $x_i$  to  $\bar{x}$ . Second, if the initial data average  $\bar{x}$  is close to a quantization point, then, on average,  $\mathbf{x}(t_\Delta)$  will be closer to an absorbing state (note that  $\mathbb{E}\{\mathbf{1}^\top \mathbf{q}(t)\} = \mathbf{1}^\top \mathbf{x}(0)$ ). These observations explain the results of Fig. 4. Note that the convergence time order for  $\bar{\mathbf{x}}(0) = 0.85$  and  $\bar{\mathbf{x}}(0) = 0.90$  cases flip for  $\Delta = 0.2$  and  $\Delta = 0.1$ . That is due to the fact that the average distance to an absorbing when, at the first time step, all the node values enter the final bin is smaller for  $\bar{\mathbf{x}}(0) = 0.85$  when  $\Delta = 0.2$  compared to  $\Delta = 0.1$ , and is smaller for  $\bar{\mathbf{x}}(0) = 0.90$  when  $\Delta = 0.1$  compared to  $\Delta = 0.2$ . Moreover, note that  $\Delta = 0.05$  yields the smallest distance to an absorbing state for both initial conditions. Although, it takes more iterations to converge to final bin, in both cases, PQDA algorithm with  $\Delta = 0.05$  yields the smallest average distance to an absorbing state when all the node values enter to the final bin for the first time step, hence, the smallest average number of iterations to achieve the consensus.

## VI. CONCLUSION

This paper presented convergence results for PQDA. We show that when the range of node values is large, PQDA behaves essentially like standard distributed averaging, without quantization. When all nodes have values on two neighboring quantization points, convergence is slowed. We provide reduction techniques and approximations to bound the rate of convergence rate in this final stage of the algorithm.

## REFERENCES

- [1] T. Aysal, M. Coates, and M. Rabbat. Distributed average consensus using probabilistic quantization. In *Proc. IEEE Statistical Signal Processing Workshop*, Madison, WI, Aug. 2007.
- [2] T. Aysal, M. Coates, and M. Rabbat. Probabilistically quantized distributed averaging. submitted to *IEEE Trans. Signal Processing*, Sep. 2007.
- [3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Info. Theory*, 52(6):2508–2530, June 2006.
- [4] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *J. Appl. Probab.*, 31(1):59–75, Jan. 1994.
- [5] P. Deheuvels and D. Pfeifer. A semigroup approach to poisson approximation. *Annals of Probability*, 14:663–676, 1986.
- [6] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. 48(6):988–1001, 2003.
- [7] A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43:1192–1203, July 2007.

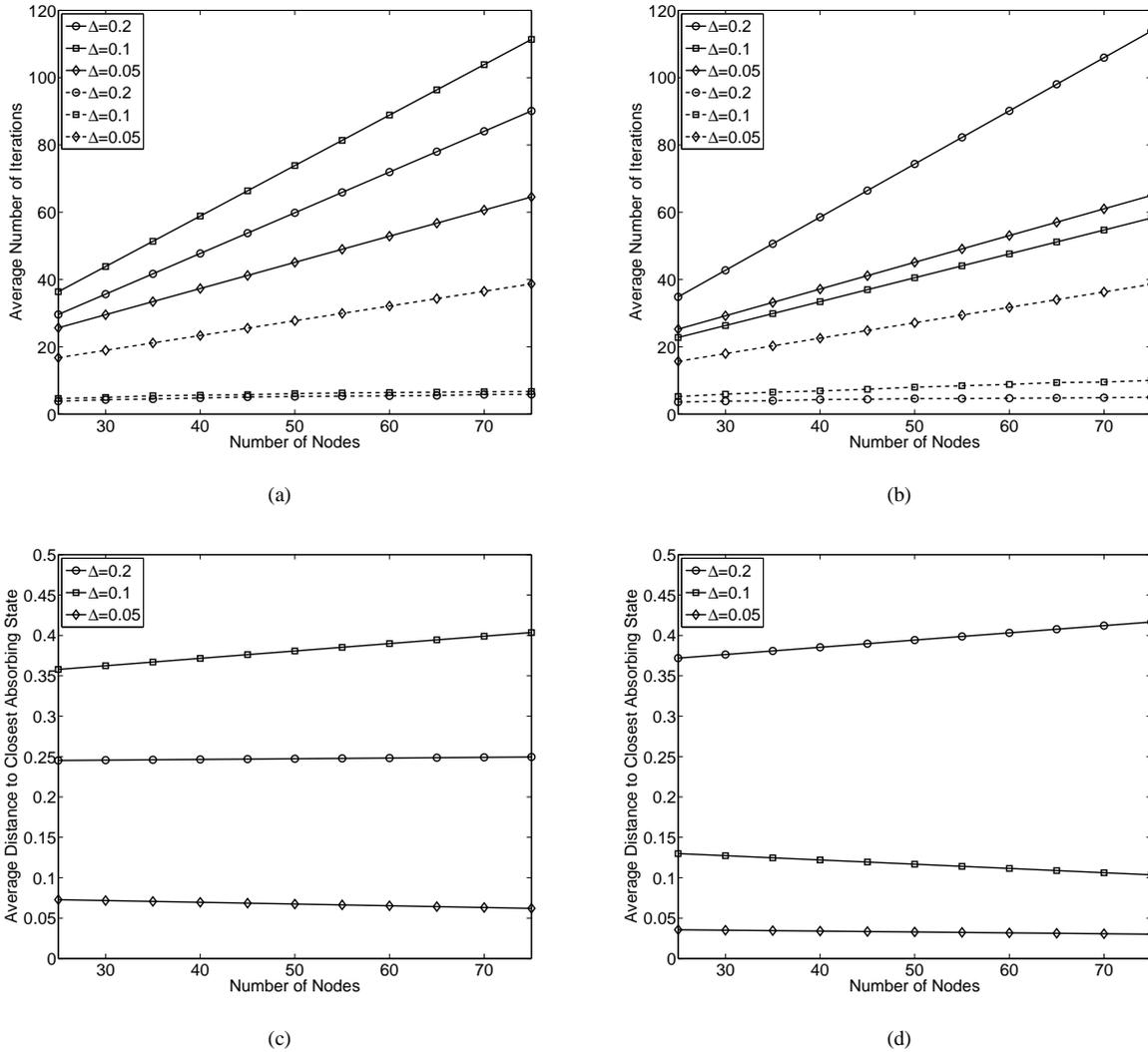


Fig. 4. Average number of iterations taken by the probabilistically quantized distribute average computation to achieve final quantization bin (dashed) and consensus (solid) for  $\Delta \in \{0.05, 0.1, 0.2\}$  and for varying  $N$  with (a):  $\bar{x}(0) = 0.85$  and (b):  $\bar{x}(0) = 0.90$ , along with the corresponding average distance to the closest absorbing state at the first time step when all the quantized node state value are in the final quantization bin for (c):  $\bar{x}(0) = 0.85$  and (d):  $\bar{x}(0) = 0.90$ .

[8] D. Kempe, A. Dobra, and J. Gehrke. Computing aggregate information using gossip. In *Proc. Foundations of Computer Science*, Cambridge, MA, October 2003.

[9] L. LeCam. An approximation theorem for the poisson binomial distribution. *Pacific Journal of Mathematics*, 10:1181–1197, 1960.

[10] H. Makabe. On approximations to some limiting distributions with applications to theory of sampling inspections by attributes. *Kodai Mathematics Seminars Report*, 16:1–17, 1964.

[11] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time delays. 49(9):1520–1533, Sept. 2004.

[12] Y. Rabani, A. Sinclair, and R. Wanka. Local divergence of Markov chains and the analysis of iterative load-balancing schemes. In *Proceedings of the IEEE Symp. on Found. of Comp. Sci.*, Palo Alto, CA, Nov. 1998.

[13] B. Roos. Binomial approximation to the poisson binomial distribution: The krawtchouk expansion. *Theory of Probability and its Applications*, 45:258–272, 2001.

[14] V. Saligrama, M. Alanyali, and O. Savas. Distributed detection in sensor networks with packet loss and finite capacity links. *IEEE Trans. Signal Processing*, 54(11):4118–4132, November 2006.

[15] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. John Wiley, New York, 1976.

[16] L. Truffet. Reduction techniques for discrete-time Markov chains on totally ordered state space using stochastic comparisons. *J. Appl. Probab.*, 37(3):795–806, Mar. 2000.

[17] J. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Boston, MA, 1984.

[18] J. V. Uspensky. *Introduction to mathematical probability*. McGraw-Hill, New York, 1937.

[19] J.-J. Xiao and Z.-Q. Luo. Decentralized estimation in an inhomogeneous sensing environment. 51(10):3564–3575, Oct. 2005.

[20] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[21] 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, Jan. 2007.

[22] M. E. Yildiz and A. Scaglione. Differential nested lattice encoding for consensus problems. In *Proceedings of the Information Processing in Sensor Networks*, Cambridge, MA, Apr. 2007.