On Distributed Averaging Algorithms and Quantization Effects

On Distributed Averaging Algorithms and Quantization Effects

Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, *Member, IEEE*, and John N. Tsitsiklis, *Fellow, IEEE*

***Abstract*—We consider distributed iterative algorithms for the averaging problem over time-varying topologies. Our focus is on the convergence time of such algorithms when complete (unquantized) information is available, and on the degradation of performance when only quantized information is available. We study a large and natural class of averaging algorithms, which includes the vast majority of algorithms proposed to date, and provide tight polynomial bounds on their convergence time. We also describe an algorithm within this class whose convergence time is the best among currently available averaging algorithms for time-varying topologies. We then propose and analyze distributed averaging algorithms under the additional constraint that agents can only store and communicate quantized information, so that they can only converge to the average of the initial values of the agents within some error. We establish bounds on the error and tight bounds on the convergence time, as a function of the number of quantization levels.**

***Index Terms*—Decentralized and distributed control, multi-agent systems.**

I. Introduction

THERE has been much recent interest in distributed control and coordination of networks consisting of multiple, potentially mobile, agents. This is motivated mainly by the emergence of large scale networks, characterized by the lack of centralized access to information and time-varying connectivity. Control and optimization algorithms deployed in such networks should be completely distributed, relying only on local observations and information, and robust against unexpected changes in topology such as link or node failures.

A canonical problem in distributed control is the *consensus problem*. The objective in the consensus problem is to develop distributed algorithms that can be used by a group of agents in order to reach agreement (consensus) on a common decision (represented by a scalar or a vector value). The agents start with some different initial decisions and communicate them locally under some constraints on connectivity and inter-agent information exchange. The consensus problem arises in a number of applications including coordination of UAVs (e.g., aligning the agents' directions of motion), information processing in sensor networks, and distributed optimization (e.g., agreeing on the estimates of some unknown parameters). The *averaging problem* is a special case in which the goal is to compute the exact average of the initial values of the agents. A natural and widely studied consensus algorithm, proposed and analyzed in [18] and [19], involves, at each time step, every agent taking a weighted average of its own value with values received from some of the other agents. Similar algorithms have been studied in the load-balancing literature (see for example [8]). Motivated by observed group behavior in biological and dynamical systems, the recent literature in cooperative control has studied similar algorithms and proved convergence results under various assumptions on agent connectivity and information exchange (see [13]–[15], [17], [20]).

In this paper, our goal is to provide tight bounds on the convergence time (defined as the number of iterations required to reduce a suitable Lyapunov function by a constant factor) of a general class of consensus algorithms, as a function of the number n of agents. We focus on algorithms that are designed to solve the averaging problem. We consider both problems where agents have access to exact values and problems where agents only have access to quantized values of the other agents. Our contributions can be summarized as follows.

In the first part of the paper, we consider the case where agents can exchange and store continuous values, which is a widely adopted assumption in the previous literature. We consider a large class of averaging algorithms defined by the condition that the weight matrix is a possibly nonsymmetric, doubly stochastic matrix. For this class of algorithms, we prove that the convergence time is $O(n^2/\eta)$, where n is the number of agents and η is a lower bound on the nonzero weights used in the algorithm. To the best of our knowledge, *this is the best polynomial-time bound on the convergence time of such algorithms* . We also show that this bound is tight. Since all previously studied linear schemes force η to be of the order of $1/n$, this result implies an $O(n^3)$ bound on convergence time. In Section IV, we present a distributed algorithm that selects the weights dynamically, using three-hop neighborhood information. Under the assumption that the underlying connectivity graph at each iteration is undirected, we establish an improved $O(n^2)$ upper bound on convergence time. This matches the best currently available convergence time guarantee for the much simpler case of static connectivity graphs [16].

In the second part of the paper, we impose the additional constraint that agents can only store and transmit quantized values.

Manuscript received November 30, 2007; revised July 19, 2008 and January 14, 2009. First published October 13, 2009; current version published November 04, 2009. A preliminary version of this paper has appeared in the Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008. This work was supported in part by the National Science Foundation under Grants ECCS-0701623, CMMI 07-42538, and DMI-0545910 and by the DARPA ITMANET Program. Recommended by Associate Editor D. Henrion.

A. Nedić is with the Industrial and Enterprise Systems Engineering Department, University of Illinois at Urbana-Champaign, Urbana IL 61801 USA (e-mail: angelia@uiuc.edu).

A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis are with the Laboratory for Information and Decision Systems, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge MA 02139 USA (e-mail: alex_o@mit.edu; asuman@mit.edu; jnt@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2009.2031203

This model provides a good approximation for communication networks that are subject to communication bandwidth or storage constraints. We focus on a particular quantization rule, which rounds down the values to the nearest quantization level. We propose a distributed algorithm that uses quantized values and, using a slightly different Lyapunov function, we show that the algorithm guarantees the convergence of the values of the agents to a common value. In particular, we prove that all agents have the same value after $O((n^2/\eta)\log(nQ))$ time steps, where Q is the number of quantization levels per unit value. Due to the rounding-down feature of the quantizer, this algorithm does not preserve the average of the values at each iteration. However, we provide bounds on the error between the final consensus value and the initial average, as a function of the number Q of available quantization levels. In particular, we show that the error goes to 0 at a rate of $(\log Q)/Q$, as the number Q of quantization levels increases to infinity.

Other than the papers cited above, our work is also related to [11] and [5], [6], which studied the effects of quantization on the performance of averaging algorithms. In [11], Kashyap *et al.* proposed randomized *gossip-type* quantized averaging algorithms under the assumption that each agent value is an integer. They showed that these algorithms preserve the average of the values at each iteration and converge to approximate consensus. They also provided bounds on the convergence time of these algorithms for specific static topologies (fully connected and linear networks). In the recent work [5], Carli *et al.* proposed a distributed algorithm that uses quantized values and preserves the average at each iteration. They showed favorable convergence properties using simulations on some static topologies, and provided performance bounds for the limit points of the generated iterates. Our results on quantized averaging algorithms differ from these works in that *we study a more general case of time-varying topologies, and provide tight polynomial bounds on both the convergence time and the discrepancy from the initial average, in terms of the number of quantization levels*.

The paper is organized as follows. In Section II, we introduce a general class of averaging algorithms, and present our assumptions on the algorithm parameters and on the information exchange among the agents. In Section III, we present our main result on the convergence time of the averaging algorithms under consideration. In Section IV, we present a distributed averaging algorithm for the case of undirected graphs, which picks the weights dynamically, resulting in an improved bound on the convergence time. In Section V, we propose and analyze a quantized version of the averaging algorithm. In particular, we establish bounds on the convergence time of the iterates, and on the error between the final value and the average of the initial values of the agents. Finally, we give our concluding remarks in Section VI.

II. A Class of Averaging Algorithms

We consider a set $N = \{1, 2, \ldots, n\}$ of agents, which will henceforth be referred to as "nodes." Each node i starts with a scalar value $x_i(0)$. At each nonnegative integer time k, node i receives from some of the other nodes j a message with the value of $x_j(k)$, and updates its value according to

$$x_i(k+1) = \sum_{j=1}^{n} a_{ij}(k)x_j(k) \tag{1}$$

where the $a_{ij}(k)$ are nonnegative weights with the property that $a_{ij}(k) > 0$ only if node i receives information from node j at time k. We use the notation $A(k)$ to denote the *weight matrix* $[a_{ij}(k)]_{i,j=1,\ldots,n}$, so that our update equation is

$$x(k+1) = A(k)x(k).$$

Given a matrix A, we use $\mathcal{E}(A)$ to denote the set of directed edges (j,i), including self-edges (i,i), such that $a_{ij} > 0$. At each time k, the nodes' connectivity can be represented by the directed graph $G(k) = (N, \mathcal{E}(A(k)))$.

Our goal is to study the convergence of the iterates $x_i(k)$ to the average of the initial values, $(1/n)\sum_{i=1}^{n} x_i(0)$, as k approaches infinity. In order to establish such convergence, we impose some assumptions on the weights $a_{ij}(k)$ and the graph sequence $G(k)$.

Assumption 1: For each k, the weight matrix $A(k)$ is a doubly stochastic matrix[1] with positive diagonal entries. Additionally, there exists a constant $\eta > 0$ such that if $a_{ij}(k) > 0$, then $a_{ij}(k) \geq \eta$.

The double stochasticity assumption on the weight matrix guarantees that the average of the node values remains the same at each iteration (cf. the proof of Lemma 4 below). The second part of this assumption states that each node gives significant weight to its values and to the values of its neighbors at each time k.

Our next assumption ensures that the graph sequence $G(k)$ is sufficiently connected for the nodes to repeatedly influence each other's values.

Assumption 2: There exists an integer $B \geq 1$ such that the directed graph

$$\Big(N, \mathcal{E}(A(kB)) \bigcup \mathcal{E}(A(kB+1)) \bigcup \cdots \bigcup \mathcal{E}(A((k+1)B-1))\Big)$$

is strongly connected for all nonnegative integers k.

Any algorithm of the form given in (1) with the sequence of weights $a_{ij}(k)$ satisfying Assumptions 1 and 2 solves the averaging problem. This is formalized in the following proposition.

Proposition 3: Let Assumptions 1 and 2 hold. Let $\{x(k)\}$ be generated by the algorithm (1). Then, for all i, we have

$$\lim_{k \to \infty} x_i(k) = \frac{1}{n}\sum_{j=1}^{n} x_j(0).$$

This fact is a minor modification of known results in [3], [10], [18], [19], where the convergence of each $x_i(k)$ to the same value is established under weaker versions of Assumptions 1 and 2. The fact that the limit is the average of the entries of the vector $x(0)$ follows from the fact that multiplication of a

[1] A matrix is called doubly stochastic if it is nonnegative and all of its rows and columns sum to 1.

vector by a doubly stochastic matrix preserves the average of the vector's components.

Recent research has focused on methods of choosing weights $a_{ij}(k)$ that satisfy Assumptions 1 and 2, and minimize the convergence time of the resulting averaging algorithm (see [21] for the case of static graphs, see [15] and [2] for the case of symmetric weights, i.e., weights satisfying $a_{ij}(k) = a_{ji}(k)$, and also see [4], [7]). For static graphs, some recent results on optimal time-invariant algorithms may be found in [16].

III. Convergence Time

In this section, we give an analysis of the convergence time of averaging algorithms of the form (1). Our goal is to obtain tight estimates of the convergence time, under Assumptions 1 and 2.

As a convergence measure, we use the "sample variance" of a vector $x \in \Re^n$, defined as

$$V(x) = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

where $\bar{x}$ is the average of the entries of x

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Let $x(k)$ denote the vector of node values at time k [i.e., the vector of iterates generated by algorithm (1) at time k]. We are interested in providing an upper bound on the number of iterations it takes for the "sample variance" $V(x(k))$ to decrease to a small fraction of its initial value $V(x(0))$. We first establish some technical preliminaries that will be key in the subsequent analysis. In particular, in the next subsection, we explore several implications of the double stochasticity assumption on the weight matrix $A(k)$.

A. Preliminaries on Doubly Stochastic Matrices

We begin by analyzing how the sample variance $V(x)$ changes when the vector x is multiplied by a doubly stochastic matrix A. The next lemma shows that $V(Ax) \leq V(x)$. Thus, under Assumption 1, the sample variance $V(x(k))$ is nonincreasing in k, and $V(x(k))$ can be used as a Lyapunov function.

Lemma 4: Let A be a doubly stochastic matrix. Then,[2] for all $x \in \Re^n$

$$V(Ax) = V(x) - \sum_{i<j} w_{ij}(x_i - x_j)^2$$

where w_{ij} is the (i,j)th entry of the matrix A^TA.

Proof: Let $\mathbf{1}$ denote the vector in $\Re^n$ with all entries equal to 1. The double stochasticity of A implies

$$A\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^T A = \mathbf{1}^T.$$

[2]In the sequel, the notation $\sum_{i<j}$ will be used to denote the double sum $\sum_{j=1}^{n} \sum_{i=1}^{j-1}$.

Note that multiplication by a doubly stochastic matrix A preserves the average of the entries of a vector, i.e., for any $x \in \Re$, there holds

$$\overline{Ax} = \frac{1}{n} \mathbf{1}^T Ax = \frac{1}{n} \mathbf{1}^T x = \bar{x}.$$

We now write the quadratic form $V(x) - V(Ax)$ explicitly, as follows:

$$\begin{aligned} V(x) - V(Ax) &= (x - \bar{x}\mathbf{1})^T (x - \bar{x}\mathbf{1}) \\ &\quad - (Ax - \overline{Ax}\mathbf{1})^T (Ax - \overline{Ax}\mathbf{1}) \\ &= (x - \bar{x}\mathbf{1})^T (x - \bar{x}\mathbf{1}) \\ &\quad - (Ax - \bar{x}A\mathbf{1})^T (Ax - \bar{x}A\mathbf{1}) \\ &= (x - \bar{x}\mathbf{1})^T (I - A^TA)(x - \bar{x}\mathbf{1}). \end{aligned} \quad (2)$$

Let w_{ij} be the (i,j)th entry of A^TA. Note that A^TA is symmetric and stochastic, so that $w_{ij} = w_{ji}$ and $w_{ii} = 1 - \sum_{j \neq i} w_{ij}$. Then, it can be verified that

$$A^TA = I - \sum_{i<j} w_{ij}(e_i - e_j)(e_i - e_j)^T \quad (3)$$

where e_i is a unit vector with the ith entry equal to 1, and all other entries equal to 0 (see also [22] where a similar decomposition was used).

By combining (2) and (3), we obtain

$$\begin{aligned} &V(x) - V(Ax) \\ &\quad = (x - \bar{x}\mathbf{1})^T \left(\sum_{i<j} w_{ij}(e_i - e_j)(e_i - e_j)^T \right) \\ &\qquad \times (x - \bar{x}\mathbf{1}) \\ &\quad = \sum_{i<j} w_{ij}(x_i - x_j)^2. \end{aligned}$$

■

Note that the entries $w_{ij}(k)$ of $A(k)^TA(k)$ are nonnegative, because the weight matrix $A(k)$ has nonnegative entries. In view of this, Lemma 4 implies that

$$V(x(k+1)) \leq V(x(k)) \quad \text{for all } k.$$

Moreover, the amount of variance decrease is given by

$$V(x(k)) - V(x(k+1)) = \sum_{i<j} w_{ij}(k)(x_i(k) - x_j(k))^2.$$

We will use this result to provide a lower bound on the amount of decrease of the sample variance $V(x(k))$ in between iterations.

Since every positive entry of $A(k)$ is at least η, it follows that every positive entry of $A(k)^TA(k)$ is at least η^2. Therefore, it is immediate that

$$\text{if} \quad w_{ij}(k) > 0, \quad \text{then} \quad w_{ij}(k) \geq \eta^2.$$

In our next lemma, we establish a stronger lower bound. In particular, we find it useful to focus not on an individual w_{ij}, but

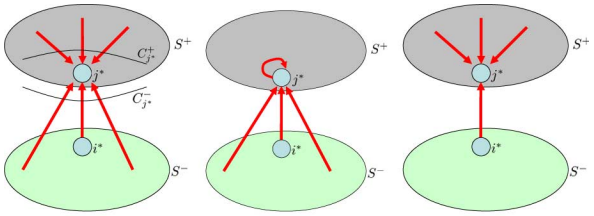

Fig. 1. (a) Intuitively, $C^+_{j^*}$ measures how much weight j^* assigns to nodes in S^+ (including itself), and $C^-_{j^*}$ measures how much weight j^* assigns to nodes in S^-. Note that the edge (j^*, j^*) is also present, but not shown. (b) For the case where $C^-_{j^*} \geq 1/2$, we only focus on two-hop paths between j^* and elements $i \in S^-$ obtained by taking (i, j^*) as the first step and the self-edge (j^*, j^*) as the second step. (c) For the case where $C^+_{j^*} \geq 1/2$, we only focus on two-hop paths between i^* and elements $j \in S^+$ obtained by taking (i^*, j^*) as the first step in $\mathcal{E}(A)$ and (j^*, j) as the second step in $\mathcal{E}(A^T)$.

rather on all w_{ij} associated with edges (i, j) that cross a particular cut in the graph $(N, \mathcal{E}(A^T A))$. For such groups of w_{ij}, we prove a lower bound which is linear in η, as seen in the following.

Lemma 5: Let A be a row-stochastic matrix with positive diagonal entries, and assume that the smallest positive entry in A is at least η. Also, let (S^-, S^+) be a partition of the set $N = \{1, \ldots, n\}$ into two disjoint sets. If

$$\sum_{i \in S^-,\ j \in S^+} w_{ij} > 0$$

then

$$\sum_{i \in S^-,\ j \in S^+} w_{ij} \geq \frac{\eta}{2}.$$

Proof: Let $\sum_{i \in S^-,\ j \in S^+} w_{ij} > 0$. From the definition of the weights w_{ij}, we have $w_{ij} = \sum_k a_{ki} a_{kj}$, which shows that there exist $i \in S^-$, $j \in S^+$, and some k such that $a_{ki} > 0$ and $a_{kj} > 0$. For either case where k belongs to S^- or S^+, we see that there exists an edge in the set $\mathcal{E}(A)$ that crosses the cut (S^-, S^+). Let (i^*, j^*) be such an edge. Without loss of generality, we assume that $i^* \in S^-$ and $j^* \in S^+$.

We define

$$C^+_{j^*} = \sum_{i \in S^+} a_{j^* i},$$
$$C^-_{j^*} = \sum_{i \in S^-} a_{j^* i}.$$

See Fig. 1(a) for an illustration. Since A is a row-stochastic matrix, we have

$$C^-_{j^*} + C^+_{j^*} = 1$$

implying that at least one of the following is true:

$$\text{Case (a)}: \quad C^-_{j^*} \geq \frac{1}{2},$$
$$\text{Case (b)}: \quad C^+_{j^*} \geq \frac{1}{2}.$$

We consider these two cases separately. In both cases, we focus on a subset of the edges and we use the fact that the elements w_{ij} correspond to paths of length 2, with one step in $\mathcal{E}(A)$ and another in $\mathcal{E}(A^T)$.

Case (a): $C^-_{j^*} \geq 1/2$: We focus on those w_{ij} with $i \in S^-$ and $j = j^*$. Indeed, since all w_{ij} are nonnegative, we have

$$\sum_{i \in S^-,\ j \in S^+} w_{ij} \geq \sum_{i \in S^-} w_{ij^*}. \tag{4}$$

For each element in the sum on the right-hand side, we have

$$w_{ij^*} = \sum_{k=1}^{n} a_{ki}\, a_{kj^*} \geq a_{j^* i}\, a_{j^* j^*} \geq a_{j^* i}\, \eta$$

where the inequalities follow from the facts that A has nonnegative entries, its diagonal entries are positive, and its positive entries are at least η. Consequently,

$$\sum_{i \in S^-} w_{ij^*} \geq \eta \sum_{i \in S^-} a_{j^* i} = \eta C^-_{j^*}. \tag{5}$$

Combining (4) and (5), and recalling the assumption $C^-_{j^*} \geq 1/2$, the result follows. An illustration of this argument can be found in Fig. 1(b).

Case (b): $C^+_{j^*} \geq 1/2$: We focus on those w_{ij} with $i = i^*$ and $j \in S^+$. We have

$$\sum_{i \in S_-,\ j \in S^+} w_{ij} \geq \sum_{j \in S^+} w_{i^* j}, \tag{6}$$

since all w_{ij} are nonnegative. For each element in the sum on the right-hand side, we have

$$w_{i^* j} = \sum_{k=1}^{n} a_{ki^*}\, a_{kj} \geq a_{j^* i^*}\, a_{j^* j} \geq \eta\, a_{j^* j}$$

where the inequalities follow because all entries of A are nonnegative, and because the choice $(i^*, j^*) \in \mathcal{E}(A)$ implies that $a_{j^* i^*} \geq \eta$. Consequently

$$\sum_{j \in S^+} w_{i^* j} \geq \eta \sum_{j \in S^+} a_{j^* j} = \eta\, C^+_{j^*}. \tag{7}$$

Combining (6) and (7), and recalling the assumption $C^+_{j^*} \geq 1/2$, the result follows. An illustration of this argument can be found in Fig. 1(c). ■

B. A Bound on Convergence Time

With the preliminaries on doubly stochastic matrices in place, we can now proceed to derive bounds on the decrease of $V(x(k))$ in between iterations. We will first somewhat relax our connectivity assumptions. In particular, we consider the following relaxation of Assumption 2.

Assumption 6: Given an integer $k \geq 0$, suppose that the components of $x(kB)$ have been reordered so that they are in nonincreasing order. We assume that for every $d \in \{1, \ldots, n-1\}$, we either have $x_d(kB) = x_{d+1}(kB)$, or there exist some time $t \in \{kB, \ldots, (k+1)B - 1\}$ and some $i \in \{1, \ldots, d\}$, $j \in \{d+1, \ldots, n\}$ such that (i, j) or (j, i) belongs to $\mathcal{E}(A(t))$.

Lemma 7: Assumption 2 implies Assumption 6, with the same value of B.

Proof: If Assumption 6 does not hold, then there must exist an index d [for which $x_d(kB) \neq x_{d+1}(kB)$ holds] such that

there are no edges between nodes $1,2,\ldots,d$ and nodes $d+1,\ldots,n$ during times $t=kB,\ldots,(k+1)B-1$. But this implies that the graph

$$\Big(N,\mathcal{E}(A(kB))\bigcup\mathcal{E}(A(kB+1))\bigcup\cdots\bigcup\mathcal{E}(A((k+1)B-1))\Big)$$

is disconnected, which violates Assumption 2. ■

For our convergence time results, we will use the weaker Assumption 6, rather than the stronger Assumption 2. Later on, in Section IV, we will exploit the sufficiency of Assumption 6 to design a decentralized algorithm for selecting the weights $a_{ij}(k)$, which satisfies Assumption 6, but not Assumption 2.

We now proceed to bound the decrease of our Lyapunov function $V(x(k))$ during the interval $[kB,(k+1)B-1]$. In what follows, we denote by $V(k)$ the sample variance $V(x(k))$ at time k.

Lemma 8: Let Assumptions 1 and 6 hold. Let $\{x(k)\}$ be generated by the update rule (1). Suppose that the components $x_i(kB)$ of the vector $x(kB)$ have been ordered from largest to smallest, with ties broken arbitrarily. Then, we have

$$V(kB)-V((k+1)B)\geq\frac{\eta}{2}\sum_{i=1}^{n-1}(x_i(kB)-x_{i+1}(kB))^2.$$

Proof: By Lemma 4, we have for all t

$$V(t)-V(t+1)=\sum_{i<j}w_{ij}(t)(x_i(t)-x_j(t))^2 \qquad (8)$$

where $w_{ij}(t)$ is the (i,j)-th entry of $A(t)^TA(t)$. Summing up the variance differences $V(t)-V(t+1)$ over different values of t, we obtain

$$V(kB)-V((k+1)B)=\sum_{t=kB}^{(k+1)B-1}\sum_{i<j}w_{ij}(t)(x_i(t)-x_j(t))^2. \qquad (9)$$

We next introduce some notation.

a) For all $d\in\{1,\ldots,n-1\}$, let t_d be the first time larger than or equal to kB (if it exists) at which there is a communication between two nodes belonging to the two sets $\{1,\ldots,d\}$ and $\{d+1,\ldots,n\}$, to be referred to as a communication across the cut d.

b) For all $t\in\{kB,\ldots,(k+1)B-1\}$, let $D(t)=\{d\mid t_d=t\}$, i.e., $D(t)$ consists of "cuts" $d\in\{1,\ldots,n-1\}$ such that time t is the first communication time larger than or equal to kB between nodes in the sets $\{1,\ldots,d\}$ and $\{d+1,\ldots,n\}$. Because of Assumption 6, the union of the sets $D(t)$ includes all indices $1,\ldots,n-1$, except possibly for indices for which $x_d(kB)=x_{d+1}(kB)$.

c) For all $d\in\{1,\ldots,n-1\}$, let $C_d=\{(i,j),\ (j,i)\mid i\leq d,\ d+1\leq j\}$.

d) For all $t\in\{kB,\ldots,(k+1)B-1\}$, let $F_{ij}(t)=\{d\in D(t)\mid (i,j)\text{ or }(j,i)\in C_d\}$, i.e., $F_{ij}(t)$ consists of all cuts d such that the edge (i,j) or (j,i) at time t is the first communication across the cut at a time larger than or equal to kB.

e) To simplify notation, let $y_i=x_i(kB)$. By assumption, we have $y_1\geq\cdots\geq y_n$.

We make two observations, as follows:

1) Suppose that $d\in D(t)$. Then, for some $(i,j)\in C_d$, we have either $a_{ij}(t)>0$ or $a_{ji}(t)>0$. Because $A(t)$ is nonnegative with positive diagonal entries, we have

$$w_{ij}(t)=\sum_{k=1}^{n}a_{ki}a_{kj}\geq a_{ii}(t)a_{ij}(t)+a_{ji}(t)a_{jj}(t)>0,$$

and by Lemma 5, we obtain

$$\sum_{(i,j)\in C_d}w_{ij}(t)\geq\frac{\eta}{2}. \qquad (10)$$

2) Fix some (i,j), with $i<j$, and time $t\in\{kB,\ldots,(k+1)B-1\}$, and suppose that $F_{ij}(t)$ is nonempty. Let $F_{ij}(t)=\{d_1,\ldots,d_k\}$, where the d_j are arranged in increasing order. Since $d_1\in F_{ij}(t)$, we have $d_1\in D(t)$ and therefore $t_{d_1}=t$. By the definition of t_{d_1}, this implies that there has been no communication between a node in $\{1,\ldots,d_1\}$ and a node in $\{d_1+1,\ldots,n\}$ during the time interval $[kB,t-1]$. It follows that $x_i(t)\geq y_{d_1}$. By a symmetrical argument, we also have

$$x_j(t)\leq y_{d_k+1}. \qquad (11)$$

These relations imply that

$$x_i(t)-x_j(t)\geq y_{d_1}-y_{d_k+1}\geq\sum_{d\in F_{ij}(t)}(y_d-y_{d+1}).$$

Since the components of y are sorted in nonincreasing order, we have $y_d-y_{d+1}\geq 0$, for every $d\in F_{ij}(t)$. For any nonnegative numbers z_i, we have

$$(z_1+\cdots+z_k)^2\geq z_1^2+\cdots+z_k^2$$

which implies that

$$(x_i(t)-x_j(t))^2\geq\sum_{d\in F_{ij}(t)}(y_d-y_{d+1})^2. \qquad (12)$$

We now use these two observations to provide a lower bound on the expression on the right-hand side of (8) at time t. We use (12) and then (10), to obtain

$$\begin{aligned}\sum_{i<j}w_{ij}(t)(x_i(t)-x_j(t))^2&\geq\sum_{i<j}w_{ij}(t)\sum_{d\in F_{ij}(t)}(y_d-y_{d+1})^2\\&=\sum_{d\in D(t)}\sum_{(i,j)\in C_d}w_{ij}(t)(y_d-y_{d+1})^2\\&\geq\frac{\eta}{2}\sum_{d\in D(t)}(y_d-y_{d+1})^2.\end{aligned}$$

We now sum both sides of the above inequality for different values of t, and use (9), to obtain

$$\begin{aligned} V(kB) - V((k+1)B) &= \sum_{t=kB}^{(k+1)B-1} \sum_{i<j} w_{ij}(t)(x_i(t) - x_j(t))^2 \\ &\geq \frac{\eta}{2} \sum_{t=kB}^{(k+1)B-1} \sum_{d \in D(t)} (y_d - y_{d+1})^2 \\ &= \frac{\eta}{2} \sum_{d=1}^{n-1} (y_d - y_{d+1})^2 \end{aligned}$$

where the last inequality follows from the fact that the union of the sets $D(t)$ is only missing those d for which $y_d = y_{d+1}$. ■

We next establish a bound on the variance decrease that plays a key role in our convergence analysis.

Lemma 9: Let Assumptions 1 and 6 hold, and suppose that $V(kB) > 0$. Then

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2n^2} \quad \text{for all } k.$$

Proof: Without loss of generality, we assume that the components of $x(kB)$ have been sorted in nonincreasing order. By Lemma 8, we have

$$V(kB) - V((k+1)B) \geq \frac{\eta}{2} \sum_{i=1}^{n-1} (x_i(kB) - x_{i+1}(kB))^2.$$

This implies that

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2} \frac{\sum_{i=1}^{n-1} (x_i(kB) - x_{i+1}(kB))^2}{\sum_{i=1}^{n} (x_i(kB) - \bar{x}(kB))^2}.$$

Observe that the right-hand side does not change when we add a constant to every $x_i(kB)$. We can therefore assume, without loss of generality, that $\bar{x}(kB) = 0$, so that

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2} \min_{\substack{x_1 \geq x_2 \geq \cdots \geq x_n \\ \sum_i x_i = 0}} \frac{\sum_{i=1}^{n-1} (x_i - x_{i+1})^2}{\sum_{i=1}^{n} x_i^2}.$$

Note that the right-hand side is unchanged if we multiply each x_i by the same constant. Therefore, we can assume, without loss of generality, that $\sum_{i=1}^n x_i^2 = 1$, so that

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2} \min_{\substack{x_1 \geq x_2 \geq \cdots \geq x_n \\ \sum_i x_i = 0, \sum_i x_i^2 = 1}} \sum_{i=1}^{n-1} (x_i - x_{i+1})^2. \quad (13)$$

The requirement $\sum_i x_i^2 = 1$ implies that the average value of x_i^2 is $1/n$, which implies that there exists some j such that $|x_j| \geq 1/\sqrt{n}$. Without loss of generality, let us suppose that this x_j is positive.[3]

[3]Otherwise, we can replace x with $-x$ and subsequently reorder to maintain the property that the components of x are in descending order. It can be seen that these operations do not affect the objective value.

The rest of the proof relies on a technique from [12] to provide a lower bound on the right-hand side of (13). Let

$$z_i = x_i - x_{i+1} \quad \text{for } i < n, \quad \text{and} \quad z_n = 0.$$

Note that $z_i \geq 0$ for all i and

$$\sum_{i=1}^{n} z_i = x_1 - x_n.$$

Since $x_j \geq 1/\sqrt{n}$ for some j, we have that $x_1 \geq 1/\sqrt{n}$; since $\sum_{i=1}^n x_i = 0$, it follows that at least one x_i is negative, and therefore $x_n < 0$. This gives us

$$\sum_{i=1}^{n} z_i \geq \frac{1}{\sqrt{n}}.$$

Combining with (13), we obtain

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2} \min_{z_i \geq 0, \sum_i z_i \geq 1/\sqrt{n}} \sum_{i=1}^{n} z_i^2.$$

The minimization problem on the right-hand side is a symmetric convex optimization problem, and therefore has a symmetric optimal solution, namely $z_i = 1/n^{1.5}$ for all i. This results in an optimal value of $1/n^2$. Therefore

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2n^2}$$

which is the desired result. ■

We are now ready for our main result, which establishes that the convergence time of the sequence of vectors $x(k)$ generated by (1) is of order $O(n^2 B/\eta)$.

Theorem 10: Let Assumptions 1 and 6 hold. Then, there exists an absolute constant[4] c such that we have

$$V(k) \leq \epsilon V(0) \quad \text{for all } k \geq c \left(\frac{n^2}{\eta}\right) B \log\left(\frac{1}{\epsilon}\right).$$

Proof: The result follows immediately from Lemma 9. ■

Recall that, according to Lemma 7, Assumption 2 implies Assumption 6. In view of this, the convergence time bound of Theorem 10 holds for any n and any sequence of weights satisfying Assumptions 1 and 2. In the next subsection, we show that this bound is tight when the stronger Assumption 2 holds.

C. Tightness

The next proposition shows that the convergence time bound of Theorem 10 is tight under Assumption 2.

Proposition 11: There exist constants c and n_0 with the following property. For any $n \geq n_0$, nonnegative integer B, $\eta < 1/2$, and $\epsilon < 1$, there exist a sequence of weight matrices $A(k)$ satisfying Assumptions 1 and 2, and an initial value $x(0)$ such that if $V(k)/V(0) \leq \epsilon$, then

$$k \geq c \frac{n^2}{\eta} B \log \frac{1}{\epsilon}.$$

Proof: Let P be the circulant shift operator defined by $Pe_i = e_{i+1}$, $Pe_n = e_1$, where e_i is a unit vector with the i-th

[4]We say c is an absolute constant when it does not depend on any of the parameters in the problem, in this case n, B, η, ϵ.

entry equal to 1, and all other entries equal to 0. Consider the symmetric circulant matrix defined by

$$A = (1-2\eta)I + \eta P + \eta P^{-1}.$$

Let $A(k) = A$, when k is a multiple of B, and $A(k) = I$ otherwise. Note that this sequence satisfies Assumptions 1 and 2.

The second largest eigenvalue of A is

$$\lambda_2(A) = 1 - 2\eta + 2\eta\cos\frac{2\pi}{n}$$

([9, eq. (3.7)]). Therefore, using the inequality $\cos x \geq 1 - x^2/2$

$$\lambda_2(A) \geq 1 - \frac{4\eta\pi^2}{n^2}.$$

For n large enough, the quantity on the right-hand side is nonnegative. Let the initial vector $x(0)$ be the eigenvector corresponding to $\lambda_2(A)$. Then

$$\frac{V(kB)}{V(0)} = \lambda_2(A)^{2k} \geq \left(1 - \frac{8\eta\pi^2}{n^2}\right)^k.$$

For the right-hand side to become less than ϵ, we need $k = \Omega((n^2/\eta)\log(1/\epsilon))$. This implies that for $V(k)/V(0)$ to become less than ϵ, we need $k = \Omega((n^2/\eta)B\log(1/\epsilon))$. ■

IV. Saving a Factor of n: Faster Averaging on Undirected Graphs

In the previous section, we have shown that a large class of averaging algorithms have $O(B(n^2/\eta)\log 1/\epsilon)$ convergence time. Moreover, we have shown that this bound is tight, in the sense that there exist matrices satisfying Assumptions 1 and 6 which converge in $\Omega(B(n^2/\eta)\log 1/\epsilon)$.

In this section, we consider decentralized ways of synthesizing the weights $a_{ij}(k)$ while satisfying Assumptions 1 and 6. Our focus is on improving convergence time bounds by constructing "good" schemes.

We assume that the communications of the nodes are governed by an exogenous sequence of graphs $\overline{G}(k) = (N, \overline{\mathcal{E}}(k))$ that provides strong connectivity over time periods of length B. This sequence of graphs constrains the matrices $A(k)$ that we can use; in particular, we require that $a_{ij}(k) = 0$ if $(j,i) \notin \overline{\mathcal{E}}(k)$. Naturally, we assume that $(i,i) \in \overline{\mathcal{E}}(k)$ for every i.

Several such decentralized protocols exist. For example, each node may assign

$$a_{ij}(k) = \epsilon, \quad \text{if } (j,i) \in \overline{\mathcal{E}}(k) \text{ and } i \neq j,$$
$$a_{ii}(k) = 1 - \epsilon \cdot \deg(i),$$

where $\deg(i)$ is the degree of i in $\overline{G}(k)$. If ϵ is small enough and the graph $\overline{G}(k)$ is undirected [i.e., $(i,j) \in \overline{\mathcal{E}}(k)$ if and only if $(j,i) \in \overline{\mathcal{E}}(k)$], this results in a nonnegative, doubly stochastic matrix (see [15]). However, if a node has $\Theta(n)$ neighbors, η will be of order $\Theta(1/n)$, resulting in $\Theta(n^3)$ convergence time. Moreover, this argument applies to all protocols in which nodes assign equal weights to all their neighbors; see [2] and [21] for more examples.

In this section, we examine whether it is possible to synthesize the weights $a_{ij}(k)$ in a decentralized manner, so that $a_{ij}(k) \geq \eta$ whenever $a_{ij}(k) \neq 0$, where η is a positive constant independent of n and B. We show that this is indeed possible, under the additional assumption that the graphs $\overline{G}(k)$ are undirected. Our algorithm is data-dependent, in that $a_{ij}(k)$ depends not only on the graph $\overline{G}(k)$, but also on the data vector $x(k)$. Furthermore, it is a decentralized 3-hop algorithm, in that $a_{ij}(k)$ depends only on the data at nodes within a distance of at most 3 from i. Our algorithm is such that the resulting sequences of vectors $x(k)$ and graphs $G(k) = (N, \mathcal{E}(k))$, with $\mathcal{E}(k) = \{(j,i) \mid a_{ij}(k) > 0\}$, satisfy Assumptions 1 and 6. Thus, a convergence time result can be obtained from Theorem 10.

A. The Algorithm

The algorithm we present here is a variation of an old *load balancing* algorithm (see [8] and Chapter 7.3 of [1]).[5]

At each step of the algorithm, each node offers some of its value to its neighbors, and accepts or rejects such offers from its neighbors. Once an offer from i to j, of size $\delta > 0$, has been accepted, the updates $x_i \leftarrow x_i - \delta$ and $x_j \leftarrow x_j + \delta$ are executed.

We next describe the formal steps the nodes execute at each time k. For clarity, we refer to the node executing the steps below as node C. Moreover, the instructions below sometimes refer to the neighbors of node C; this always means current neighbors at time k, when the step is being executed, as determined by the current graph $\overline{G}(k)$. We assume that at each time k, all nodes execute these steps in the order described below, while the graph remains unchanged.

Balancing Algorithm:

1. Node C broadcasts its current value x_C to all its neighbors.
2. Going through the values it just received from its neighbors, Node C finds the smallest value that is less than its own. Let D be a neighbor with this value. Node C makes an offer of $(x_C - x_D)/3$ to node D.
   If no neighbor of C has a value smaller than x_C, node C does nothing at this stage.
3. Node C goes through the incoming offers. It sends an acceptance to the sender of a largest offer, and a rejection to all the other senders. It updates the value of x_C by adding the value of the accepted offer.
   If node C did not receive any offers, it does nothing at this stage.
4. If an acceptance arrives for the offer made by node C, node C updates x_C by subtracting the value of the offer.

Note that the new value of each node is a linear combination of the values of its neighbors. Furthermore, the weights $a_{ij}(k)$ are completely determined by the data and the graph at most 3 hops from node i in $\overline{G}(k)$. This is true because in the course of execution of the above steps, each node makes at most three transmission to its neighbors, so the new value of node C cannot depend on information more than 3 hops away from C.

[5]This algorithm was also considered in [16], but in the absence of a result such as Theorem 10, a weaker convergence time bound was derived.

B. Performance Analysis

In the following theorem, we are able to remove a factor of n from the worst-case convergence time bounds of Theorem 10.

Theorem 12: Consider the balancing algorithm, and suppose that $\overline{G}(k) = (N, \overline{\mathcal{E}}(k))$ is a sequence of undirected graphs such that $(N, \overline{\mathcal{E}}(kB) \cup \overline{\mathcal{E}}(kB+1) \cup \cdots \cup \overline{\mathcal{E}}((k+1)B-1))$ is connected, for all integers k. There exists an absolute constant c such that we have

$$V(k) \leq \epsilon V(0) \quad \text{for all } k \geq cn^2 B \log\left(\frac{1}{\epsilon}\right).$$

Proof: Note that with this algorithm, the new value at some node i is a convex combination of the previous values of itself and its neighbors. Furthermore, the algorithm keeps the sum of the nodes' values constant, because every accepted offer involves an increase at the receiving node equal to the decrease at the offering node. These two properties imply that the algorithm can be written in the form

$$x(k+1) = A(k)x(k)$$

where $A(k)$ is a doubly stochastic matrix, determined by $\overline{G}(k)$ and $x(k)$. It can be seen that the diagonal entries of $A(k)$ are positive and, furthermore, all nonzero entries of $A(k)$ are larger than or equal to 1/3; thus, $\eta = 1/3$.

We claim that the algorithm [in particular, the sequence $\mathcal{E}(A(k))$] satisfies Assumption 6. Indeed, suppose that at time kB, the nodes are reordered so that the values $x_i(kB)$ are nonincreasing in i. Fix some $d \in \{1, \ldots, n-1\}$, and suppose that $x_d(kB) \neq x_{d+1}(kB)$. Let $S^+ = \{1, \ldots, d\}$ and $S^- = \{d+1, \ldots, n\}$.

Because of our assumptions on the graphs $\overline{G}(k)$, there will be a first time t in the interval $\{kB, \ldots, (k+1)B-1\}$, at which there is an edge in $\overline{\mathcal{E}}(t)$ between some $i^* \in S^+$ and $j^* \in S^-$. Note that between times kB and t, the two sets of nodes, S^+ and S^-, do not interact, which implies that $x_i(t) \geq x_d(kB)$, for $i \in S^+$, and $x_j(t) < x_d(kB)$, for $j \in S^-$.

At time t, node i^* sends an offer to a neighbor with the smallest value; let us denote that neighbor by k^*. Since $(i^*, j^*) \in \overline{\mathcal{E}}(t)$, we have $x_{k^*}(t) \leq x_{j^*}(t) < x_d(kB)$, which implies that $k^* \in S^-$. Node k^* will accept the largest offer it receives, which must come from a node with a value no smaller than $x_{i^*}(t)$, and therefore no smaller than $x_d(kB)$; hence the latter node belongs to S^+. It follows that $\mathcal{E}(A(t))$ contains an edge between k^* and some node in S^+, showing that Assumption 6 is satisfied.

The claimed result follows from Theorem 10, because we have shown that all of the assumptions in that theorem are satisfied with $\eta = 1/3$. ∎

V. Quantization Effects

In this section, we consider a quantized version of the update rule (1). This model is a good approximation for a network of nodes communicating through finite bandwidth channels, so that at each time instant, only a finite number of bits can be transmitted. We incorporate this constraint in our algorithm by assuming that each node, upon receiving the values of its neighbors, computes the convex combination $\sum_{j=1}^n a_{ij}(k)x_j(k)$ and quantizes it. This update rule also captures a constraint that each node can only store quantized values.

Unfortunately, under Assumptions 1 and 2, if the output of (1) is rounded to the nearest integer, the sequence $x(k)$ is not guaranteed to converge to consensus; see [11]. We therefore choose a quantization rule that rounds the values down, according to

$$x_i(k+1) = \left\lfloor \sum_{j=1}^n a_{ij}(k)x_j(k) \right\rfloor \tag{14}$$

where $\lfloor \cdot \rfloor$ represents rounding *down* to the nearest multiple of $1/Q$, and where Q is some positive integer.

We adopt the natural assumption that the initial values are already quantized.

Assumption 13: For all i, $x_i(0)$ is a multiple of $1/Q$.

For convenience we define

$$U = \max_i x_i(0), \quad L = \min_i x_i(0).$$

We use K to denote the total number of relevant quantization levels, i.e.,

$$K = (U - L)Q$$

which is an integer by Assumption 13.

A. Quantization Level Dependent Bound

We first present a convergence time bound that depends on the quantization level Q.

Proposition 14: Let Assumptions 1, 2, and 13 hold. Let $\{x(k)\}$ be generated by the update rule (14). If $k \geq nBK$, then all components of $x(k)$ are equal.

Proof: Consider the nodes whose initial value is U. There are at most n of them. As long as not all entries of $x(k)$ are equal, then every B iterations, at least one node must use a value strictly less than U in an update; such a node will have its value decreased to $U - 1/Q$ or less. It follows that after nB iterations, the largest node value will be at most $U - 1/Q$. Repeating this argument, we see that at most nBK iterations are possible before all the nodes have the same value. ∎

Although the above bound gives informative results for small K, it becomes weaker as Q (and, therefore, K) increases. On the other hand, as Q approaches infinity, the quantized system approaches the unquantized system; the availability of convergence time bounds for the unquantized system suggests that similar bounds should be possible for the quantized one. Indeed, in the next subsection, we adopt a notion of convergence time parallel to our notion of convergence time for the unquantized algorithm; as a result, we obtain a bound on the convergence time which is independent of the total number of quantization levels.

B. Quantization Level Independent Bound

We adopt a slightly different measure of convergence for the analysis of the quantized consensus algorithm. For any $x \in \Re^n$, we define $m(x) = \min_i x_i$ and

$$\underline{V}(x) = \sum_{i=1}^n (x_i - m(x))^2.$$

We will also use the simpler notation $m(k)$ and $\underline{V}(k)$ to denote $m(x(k))$ and $\underline{V}(x(k))$, respectively, where it is more convenient to do so. The function $\underline{V}$ will be our Lyapunov function for the analysis of the quantized consensus algorithm. The reason for not using our earlier Lyapunov function, V, is that for the quantized algorithm, V is not guaranteed to be monotonically nonincreasing in time. On the other hand, we have that $V(x) \leq \underline{V}(x) \leq 4nV(x)$ for any[6] $x \in \Re^n$. As a consequence, any convergence time bounds expressed in terms of $\underline{V}$ translate to essentially the same bounds expressed in terms of V, up to a logarithmic factor.

Before proceeding, we record an elementary fact which will allow us to relate the variance decrease $V(x) - V(y)$ to the decrease, $\underline{V}(x) - \underline{V}(y)$, of our new Lyapunov function. The proof involves simple algebra, and is therefore omitted.

Lemma 15: Let $u_1, \ldots, u_n$ and $w_1, \ldots, w_n$ be real numbers satisfying

$$\sum_{i=1}^{n} u_i = \sum_{i=1}^{n} w_i.$$

Then, the expression

$$f(z) = \sum_{i=1}^{n} (u_i - z)^2 - \sum_{i=1}^{n} (w_i - z)^2$$

is a constant, independent of the scalar z.

Our next lemma places a bound on the decrease of the Lyapunov function $\underline{V}(t)$ between times kB and $(k+1)B - 1$.

Lemma 16: Let Assumptions 1, 6, and 13 hold. Let $\{x(k)\}$ be generated by the update rule (14). Suppose that the components $x_i(kB)$ of the vector $x(kB)$ have been ordered from largest to smallest, with ties broken arbitrarily. Then, we have

$$\underline{V}(kB) - \underline{V}((k+1)B) \geq \frac{\eta}{2} \sum_{i=1}^{n-1} (x_i(kB) - x_{i+1}(kB))^2.$$

Proof: For all k, we view (14) as the composition of two operators

$$y(k) = A(k)x(k),$$

where $A(k)$ is a doubly stochastic matrix, and

$$x(k+1) = \lfloor y(k) \rfloor$$

where the quantization $\lfloor \cdot \rfloor$ is carried out componentwise.

We apply Lemma 15 with the identification $u_i = x_i(k)$, $w_i = y_i(k)$. Since multiplication by a doubly stochastic matrix preserves the mean, the condition $\sum_i u_i = \sum_i w_i$ is satisfied. By considering two different choices for the scalar z, namely, $z_1 = \bar{x}(k) = \bar{y}(k)$ and $z_2 = m(k)$, we obtain

$$V(x(k)) - V(y(k)) = \underline{V}(x(k)) - \sum_{i=1}^{n} (y_i(k) - m(k))^2. \quad (15)$$

Note that $x_i(k+1) - m(k) \leq y_i(k) - m(k)$. Therefore

$$\underline{V}(x(k)) - \sum_{i=1}^{n} (y_i(k) - m(k))^2 \leq \underline{V}(x(k)) - \sum_{i=1}^{n} (x_i(k+1) - m(k))^2. \quad (16)$$

Furthermore, note that since $x_i(k+1) \geq m(k+1) \geq m(k)$ for all i, we have that $x_i(k+1) - m(k+1) \leq x_i(k+1) - m(k)$. Therefore

$$\underline{V}(x(k)) - \sum_{i=1}^{n} (x_i(k+1) - m(k))^2 \leq \underline{V}(x(k)) - \underline{V}(x(k+1)). \quad (17)$$

By combining (15)–(17), we obtain

$$V(x(t)) - V(y(t)) \leq \underline{V}(x(t)) - \underline{V}(x(t+1)) \quad \text{for all } t.$$

Summing the preceding relations over $t = kB, \ldots, (k+1)B - 1$, we further obtain

$$\sum_{t=kB}^{(k+1)B-1} \Big(V(x(t)) - V(y(t)) \Big) \leq \underline{V}(x(kB)) - \underline{V}(x((k+1)B)).$$

To complete the proof, we provide a lower bound on the expression

$$\sum_{t=kB}^{(k+1)B-1} \Big(V(x(t)) - V(y(t)) \Big).$$

Since $y(t) = A(t)x(t)$ for all t, it follows from Lemma 4 that for any t:

$$V(x(t)) - V(y(t)) = \sum_{i<j} w_{ij}(t)(x_i(t) - x_j(t))^2$$

where $w_{ij}(t)$ is the (i,j)th entry of $A(t)^T A(t)$. Using this relation and following the same line of analysis used in the proof of Lemma 8 [where the relation $x_i(t) \geq y_{d_1}$ holds in view of the assumption that $x_i(kB)$ is a multiple of $1/Q$ for all $k \geq 0$, cf. Assumption 13] , we obtain the desired result. ■

The next theorem contains our main result on the convergence time of the quantized algorithm.

Theorem 17: Let Assumptions 1, 6, and 13 hold. Let $\{x(k)\}$ be generated by the update rule (14). Then, there exists an absolute constant c such that we have

$$\underline{V}(k) \leq \epsilon \underline{V}(0) \quad \text{for all } k \geq c \left(\frac{n^2}{\eta} \right) B \log \left(\frac{1}{\epsilon} \right).$$

[6]The first inequality follows because $\sum_i (x_i - z)^2$ is minimized when z is the mean of the vector x; to establish the second inequality, observe that it suffices to consider the case when the mean of x is 0 and $V(x) = 1$. In that case, the largest distance between m and any x_i is 2 by the triangle inequality, so $\underline{V}(x) \leq 4n$.

Proof: Let us assume that $\underline{V}(kB) > 0$. From Lemma 16, we have

$$\underline{V}(kB) - \underline{V}((k+1)B) \geq \frac{\eta}{2}\sum_{i=1}^{n-1}(x_i(kB) - x_{i+1}(kB))^2,$$

where the components $x_i(kB)$ are ordered from largest to smallest. Since $\underline{V}(kB) = \sum_{i=1}^{n}(x_i(kB) - x_n(kB))^2$, we have

$$\frac{\underline{V}(kB) - \underline{V}((k+1)B)}{\underline{V}(kB)} \geq \frac{\eta}{2}\frac{\sum_{i=1}^{n-1}(x_i(kB) - x_{i+1}(kB))^2}{\sum_{i=1}^{n}(x_i(kB) - x_n(kB))^2}.$$

Let $y_i = x_i(kB) - x_n(kB)$. Clearly, $y_i \geq 0$ for all i, and $y_n = 0$. Moreover, the monotonicity of $x_i(kB)$ implies the monotonicity of y_i

$$y_1 \geq y_2 \geq \cdots \geq y_n = 0.$$

Thus

$$\frac{\underline{V}(kB) - \underline{V}((k+1)B)}{\underline{V}(kB)} \geq \frac{\eta}{2}\min_{\substack{y_1 \geq y_2 \geq \cdots \geq y_n \\ y_n = 0}} \frac{\sum_{i=1}^{n-1}(y_i - y_{i+1})^2}{\sum_{i=1}^{n} y_i^2}.$$

Next, we simply repeat the steps of Lemma 9. We can assume without loss of generality that $\sum_{i=1}^{n} y_i^2 = 1$. Define $z_i = y_i - y_{i+1}$ for $i = 1, \ldots, n-1$ and $z_n = 0$. We have that z_i are all nonnegative and $\sum_i z_i = y_1 - y_n \geq 1/\sqrt{n}$. Therefore

$$\frac{\eta}{2}\min_{\substack{y_1 \geq y_2 \geq \cdots \geq y_n \\ \sum_i y_i^2 = 1}} \sum_{i=1}^{n-1}(y_i - y_{i+1})^2 \geq \frac{\eta}{2}\min_{z_i \geq 0, \sum_i z_i \geq 1/\sqrt{n}} \sum_{i=1}^{n} z_i^2.$$

The minimization problem on the right-hand side has an optimal value of at least $1/n^2$, and the desired result follows. ∎

C. Extensions and Modifications

In this subsection, we comment briefly on some corollaries of Theorem 17.

First, we note that the results of Section IV immediately carry over to the quantized case. Indeed, in Section IV, we showed how to pick the weights $a_{ij}(k)$ in a decentralized manner, based only on local information, so that Assumptions 1 and 6 are satisfied, with $\eta \geq 1/3$. When using a quantized version of the balancing algorithm, we once again manage to remove the factor of $1/\eta$ from our upper bound.

Proposition 18: For the quantized version of the balancing algorithm, and under the same assumptions as in Theorem 12, if $k \geq cn^2 B\log(1/\epsilon))$, then $\underline{V}(k) \leq \epsilon \underline{V}(0)$, where c is an absolute constant.

Second, we note that Theorem 17 can be used to obtain a bound on the time until the values of all nodes are equal. Indeed, we observe that in the presence of quantization, once the condition $\underline{V}(k) < 1/Q^2$ is satisfied, all components of $x(k)$ must be equal.

Proposition 19: Consider the quantized algorithm (14), and assume that Assumptions 1, 6, and 13 hold. If $k \geq c(n^2/\eta)B\big[\log Q + \log \underline{V}(0)\big]$, then all components of $x(k)$ are equal, where c is an absolute constant.

D. Tightness

We now show that the quantization-level independent bound in Theorem 17 is tight, even when the weaker Assumption 6 is replaced with the stronger Assumption 2.

Proposition 20: There exist absolute constant s c and n_0 with the following property. For any nonnegative integer $B, \eta < 1/2$, $\epsilon < 1$, and and $n \geq n_0$, there exist a sequence of weight matrices $A(k)$ satisfying Assumptions 1 and 2, and an initial value $x(0)$ satisfying Assumption 13, and a number quantization levels $Q(n)$ (depending on n) such that under the dynamics of (14), if $\underline{V}(k)/\underline{V}(0) \leq \epsilon$, then

$$k \geq c\,\frac{n^2}{\eta}\,B\log\frac{1}{\epsilon}.$$

Proof: We have demonstrated in Proposition 11 a similar result for the unquantized algorithm. Namely, we have shown that for n large enough and for any B, $\eta < 1/2$, and $\epsilon < 1$, there exists a weight sequence $a_{ij}(k)$ and an initial vector $x(0)$ such that the first time when $V(t) \leq \epsilon V(0)$ occurs after $\Omega((n^2/\eta)B\log(1/\epsilon))$ steps. Let T^* be this first time.

Consider the quantized algorithm under the exact same sequence $a_{ij}(k)$, initialized at $\lfloor x(0) \rfloor$. Let $\hat{x}_i(t)$ refer to the value of node i at time t in the quantized algorithm under this scenario, as opposed to $x_i(t)$ which denotes the value in the unquantized algorithm. Since quantization can only decrease a nodes value by at most $1/Q$ at each iteration, it is easy to show, by induction, that

$$x_i(t) \geq \hat{x}_i(t) \geq x_i(t) - \frac{t}{Q}.$$

We can pick Q large enough so that, for $t < T^*$, the vector $\hat{x}(t)$ is as close as desired to $x(t)$.

Therefore, for $t < T^*$ and for large enough Q, $\underline{V}(\hat{x}(t))/\underline{V}(\hat{x}(0))$ will be arbitrarily close to $\underline{V}(x(t))/\underline{V}(x(0))$. From the proof of Proposition 11, we see that $x(t)$ is always a scalar multiple of $x(0)$. Since $\underline{V}(x)/V(x)$ is invariant under multiplication by a constant, it follows that $\underline{V}(x(t))/\underline{V}(x(0)) = V(x(t))/V(x(0))$. Since this last quantity is above ϵ for $t < T^*$, it follows that provided Q is large enough, $\underline{V}(\hat{x}(t))/\underline{V}(\hat{x}(0))$ is also above ϵ for $t < T^*$. This proves the proposition. ∎

E. Quantization Error

Despite favorable convergence properties of our quantized averaging algorithm (14), the update rule does not preserve the average of the values at each iteration. Therefore, the common limit of the sequences $x_i(k)$, denoted by x_f, need not be equal to the exact average of the initial values. We next provide an upper bound on the error between x_f and the initial average, as a function of the number of quantization levels.

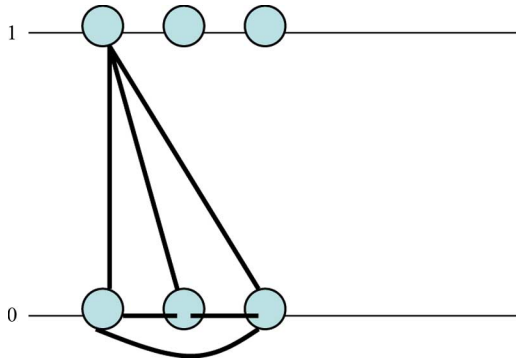

Fig. 2. Initial configuration. Each node takes the average value of its neighbors.

Proposition 21: There is an absolute constant c such that for the common limit x_f of the values $x_i(k)$ generated by the quantized algorithm (14), we have

$$\left| x_f - \frac{1}{n}\sum_{i=1}^{n} x_i(0) \right| \leq \frac{c}{Q}\,\frac{n^2}{\eta}\,B\log(Qn(U-L)).$$

Proof: By Proposition 19, after $O\Big((n^2/\eta)B\log(Q\underline{V}(x(0)))\Big)$ iterations, all nodes will have the same value. Since $\underline{V}(x(0))) \leq n(U-L)^2$ and the average decreases by at most $1/Q$ at each iteration, the result follows. ∎

Let us assume that the parameters B, η, and $U-L$ are fixed. Proposition 21 implies that as n increases, the number of bits used for each communication, which is proportional to $\log Q$, needs to grow only as $O(\log n)$ to make the error negligible. Furthermore, this is true even if the parameters B, $1/\eta$, and $U-L$ grow polynomially in n.

For a converse, it can be seen that $\Omega(\log n)$ bits are needed. Indeed, consider n nodes, with $n/2$ nodes initialized at 0, and $n/2$ nodes initialized at 1. Suppose that $Q < n/2$; we connect the nodes by forming a complete subgraph over all the nodes with value 0 and exactly one node with value 1; see Fig. 2 for an example with $n = 6$. Then, each node forms the average of its neighbors. This brings one of the nodes with an initial value of 1 down to 0, without raising the value of any other nodes. We can repeat this process, to bring all of the nodes with an initial value of 1 down to 0. Since the true average is $1/2$, the final result is $1/2$ away from the true average. Note now that Q can grow linearly with n, and still satisfy the inequality $Q < n/2$. Thus, the number of bits can grow as $\Omega(\log n)$, and yet, independent of n, the error remains $1/2$.

VI. Conclusion

We studied distributed algorithms for the averaging problem over networks with time-varying topology, with a focus on tight bounds on the convergence time of a general class of averaging algorithms. We first considered algorithms for the case where agents can exchange and store continuous values, and established tight convergence time bounds. We next studied averaging algorithms under the additional constraint that agents can only store and send quantized values. We showed that these algorithms guarantee convergence of the agents values to consensus within some error from the average of the initial values. We provided a bound on the error that highlights the dependence on the number of quantization levels.

Our paper is a contribution to the growing literature on distributed control of multi-agent systems. Quantization effects are an integral part of such systems but, with the exception of a few recent studies, have not attracted much attention in the vast literature on this subject. In this paper, we studied a quantization scheme that guarantees consensus at the expense of some error from the initial average value. We used this scheme to study the effects of the number of quantization levels on the convergence time of the algorithm and the distance from the true average.

The framework provided in this paper motivates a number of further research directions:

a) The algorithms studied in this paper assume that there is no delay in receiving the values of the other agents, which is a restrictive assumption in network settings. Understanding the convergence of averaging algorithms and implications of quantization in the presence of delays is an important topic for future research.
b) We studied a quantization scheme with favorable convergence properties, that is, rounding down to the nearest quantization level. Investigation of other quantization schemes and their impact on convergence time and error is left for future work.
c) The quantization algorithm we adopted implicitly assumes that the agents can carry out computations with continuous values, but can store and transmit only quantized values. Another interesting area for future work is to incorporate the additional constraint of finite precision computations into the quantization scheme.
d) Although our bounds are tight in the worst case over all graphs, they are not guaranteed to perform better on well-connected graphs as compared to sparse graphs with many potential bottlenecks. An interesting question is whether it is be possible to pick averaging algorithms that learn the graph and make optimal use of its information diffusion properties.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.
[2] P. A. Bliman and G. Ferrari-Trecate, "Average consensus problems in networks of agents with delayed communications," in *Proc. Joint 44th IEEE Conf. Decision Control Eur. Control Conf.*, 2005, pp. 7066–7071.
[3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision Control Eur. Control Conf.*, 2005, pp. 2996–3000.
[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. IEEE INFOCOM*, 2005, pp. 1653–1664.
[5] R. Carli, F. Fagnani, P. Frasca, T. Taylor, and S. Zampieri, "Average consensus on networks with transmission noise or quantization," in *Proc. Eur. Control Conf.*, 2007, pp. 4189–4189–4194.
[6] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, Communication Constraints in the State Agreement Problem Department of Mathematics - Politecnico di Torino, Torino, Italy, Int. Rep. 32, 2005.
[7] J. Cortes, "Analysis and design of distributed algorithms for chi-consensus," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 3363–3368.
[8] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel Distrib. Comput.*, vol. 7, no. 2, pp. 279–301, 1989.

[9] R. M. Gray, "Toeplitz and circulant matrices: A review," *Found. Trends Commun. Inform. Theory*, vol. 2, no. 3, pp. 155–239, 2006.
[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
[11] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 2728–2733.
[12] H. J. Landau and A. M. Odlyzko, "Bounds for the eigenvalues of certain stochastic matrices," *Linear Algebra Appl.*, vol. 38, pp. 5–15, 1981.
[13] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Computers*, vol. 55, no. 2, pp. 214–224, Feb. 2006.
[14] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.
[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[16] A. Olshevsky and J. N. Tsitsiklis, "Convergence rates in distributed consensus averaging," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 3387–3392.
[17] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
[18] J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., MIT, Cambridge, 1984.
[19] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.
[20] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, 1995.
[21] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.
[22] L. Xiao, S. Boyd, and S. J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.

Angelia Nedić received the B.S. degree in mathematics from the University of Montenegro, Podgorica, in 1987, the M.S. degree in mathematics from the University of Belgrade, Belgrade, Yugoslavia, in 1990, the Ph.D. degree in mathematics and mathematical physics from Moscow State University, Moscow, Russia, in 1994, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2002.

She has been at the BAE Systems Advanced Information Technology from 2002 to 2006. Since 2006, she has been a member of the faculty of the Department of Industrial and Enterprise Systems Engineering (IESE), University of Illinois at Urbana-Champaign. Her general interest is in optimization including fundamental theory, models, algorithms, and applications. Her current research interest is focused on large-scale convex optimization, distributed multi-agent optimization, and duality theory with applications in decentralized optimization.

Dr. Nedić received an NSF Faculty Early Career Development (CAREER) Award in Operations Research in 2008.

Alex Olshevsky received the B.S. degree in mathematics and the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, and the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, where he is currently pursuing the Ph.D. degree.

His research interests are in networks and control.

Asuman Ozdaglar (M'95) received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, in 1996, and the S.M. and the Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1998 and 2003, respectively.

Since 2003, she has been a member of the faculty of the Electrical Engineering and Computer Science Department, MIT, where she is currently the Class of 1943 Associate Professor. She is affiliated with the Laboratory for Information and Decision Systems and the Operations Research Center at MIT. She is the co-author of *Convex Analysis and Optimization* (Athena Scientific, 2003). Her research interests include optimization theory, with emphasis on nonlinear programming and convex analysis, game theory, with applications in communication, social, and economic networks, and distributed optimization and control.

Dr. Ozdaglar received the Microsoft Fellowship, the MIT Graduate Student Council Teaching Award, the NSF Career Award, and the 2008 Donald P. Eckman Award of the American Automatic Control Council.

John N. Tsitsiklis (F'99) received the B.S. degree in mathematics and the B.S., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1980, 1980, 1981, and 1984, respectively.

He is currently a Clarence J. Lebel Professor with the Department of Electrical Engineering, MIT. He has served as a Codirector of the MIT Operations Research Center from 2002 to 2005, and in the National Council on Research and Technology in Greece (2005–2007). His research interests are in systems, optimization, communications, control, and operations research. He has coauthored four books and more than a hundred journal papers in these areas.

Dr. Tsitsiklis received the Outstanding Paper Award from the IEEE Control Systems Society (1986), the M.I.T. Edgerton Faculty Achievement Award (1989), the Bodossakis Foundation Prize (1995), and the INFORMS/CSTS Prize (1997). He is a Fellow of INFORMS, and a member of the National Academy of Engineering. In 2008, he was conferred the title of Doctor honoris causa, from the Université catholique de Louvain.