

- [15] M. Sarich, F. Noe, and C. Schütte, "On the approximation quality of Markov state models," *Multiscale Modeling Simul.*, vol. 8, no. 4, pp. 1154–1177, 2010.
- [16] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, Nov. 2008.
- [17] H. J. Hartfiel, *Markov Set-Chains*, ser. Lecture Notes in Mathematics. Berlin, Germany: Springer-Verlag, 1998, vol. 1695.
- [18] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Cambridge, MA: Athena Scientific, 1996.
- [19] J. Cohen, "Random evolutions in discrete and continuous time," *Stoch. Processes Appl.*, vol. 9, no. 4, pp. 245–251, 1979.
- [20] M. Kepler, "Random evolutions processes induced by discrete time Markov Chains," *Port. Math.*, vol. 55, no. 4, pp. 391–400, 1998.
- [21] I. Kozine and L. Utkin, "Interval-valued finite Markov Chains," *Reliable Computing*, vol. 8, pp. 97–113, 2002.
- [22] D. P. Bertsekas, "Convergence of discretization procedures in dynamic programming," *IEEE Trans. Autom. Control*, vol. AC-20, no. 3, pp. 415–419, Jun. 1975.
- [23] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*. Berlin, Germany: Springer Verlag, 1993.

A Lower Bound for Distributed Averaging Algorithms on the Line Graph

Alex Olshevsky and John N. Tsitsiklis

Abstract—We derive lower bounds on the convergence speed of a widely used class of distributed averaging algorithms. In particular, we prove that any distributed averaging algorithm whose state consists of a single real number and whose (possibly nonlinear) update function satisfies a natural smoothness condition has a worst case running time of at least on the order of n^2 on a line network of n nodes. Our results suggest that increased memory or expansion of the state space is crucial for improving the running times of distributed averaging algorithms.

Index Terms—, Cooperative control, distributed averaging, load balancing.

I. INTRODUCTION

The goal of this technical note is to analyze the fundamental limitations of a class of distributed averaging algorithms. These algorithms are message-passing rules for a collection of agents (which may be sensors, nodes of a communication network, or UAVs), each beginning with a real number, to estimate the average of these numbers using only nearest neighbor communications. Such algorithms are interesting because a number of sophisticated network coordination tasks can be reduced to averaging (see [1], [2], [6], [8], [9], [13], [20], [23], [25]), and also because they can be designed to be robust to frequent failures of communication links.

Manuscript received April 01, 2010; revised November 02, 2010; accepted June 05, 2011. Date of publication June 16, 2011; date of current version November 02, 2011. This work was supported by the National Science Foundation (NSF) under grants ECCS-0701623 and CMMI-0856063. Recommended by Associate Editor Y. Hong.

A. Olshevsky is with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, 01527 USA (e-mail: aolshevs@princeton.edu).

J. N. Tsitsiklis is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (e-mail: jnt@mit.edu).

Digital Object Identifier 10.1109/TAC.2011.2159652

A variety of such algorithms are available (see [10], [12], [15]–[18], [21], [22], [24], [26]). However, many of these algorithms tend to suffer from a common disadvantage: even when no link failures occur, their convergence times do not scale well with the number of agents. Our aim in this technical note is to show that this is, in fact, unavoidable for a common class of such algorithms; namely, that any distributed averaging algorithm that uses a single scalar state variable at each agent and satisfies a natural "smoothness" condition will have this property, even if no link failures occur and the communication graph is always a simple "line graph."

We thus identify a basic limitation on convergence speed for a common class of averaging algorithms. The major implication of our result is that algorithms which overcome this limitation must be somewhat different from the majority of averaging algorithms designed thus far: they must either use increased memory, a larger state space, or non-smooth updates.

A. Background and Basic Definitions

Definition of Local Averaging Algorithms: Agents $1, \dots, n$ begin with real numbers $x_1(0), \dots, x_n(0)$ stored in memory. At each round $t = 0, 1, 2, \dots$, agent i broadcasts $x_i(t)$ to each of its neighbors in some undirected graph $G(t) = (\{1, \dots, n\}, E(t))$ (without any self-arcs), and then sets $x_i(t+1)$ to be some function of $x_i(t)$ and of the values $x_{i'}(t), x_{i''}(t), \dots$ it has just received from its own neighbors

$$x_i(t+1) = f_{i,G(t)}(x_i(t), x_{i'}(t), x_{i''}(t), \dots). \quad (1)$$

We require each $f_{i,G(t)}$ to be a differentiable function. Each agent uses the incoming messages $x_{i'}(t), x_{i''}(t), \dots$ as the arguments of $f_{i,G(t)}$ in some arbitrary order; we assume that this order does not change, i.e., if $G(t_1) = G(t_2)$, then the message coming from the same neighbor of agent i is mapped to the same argument of $f_{i,G(t)}$ for $t = t_1$ and $t = t_2$. It is desired that

$$\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{n} \sum_{i=1}^n x_i(0) \quad (2)$$

for every i , for every sequence of graphs $G(t)$ having the property that

$$\text{the graph } (\{1, \dots, n\}, \cup_{s \geq t} E(s)) \text{ is connected for every } t \quad (3)$$

and for every possible way for the agents to map incoming messages to arguments of $f_{i,G(t)}$.

In words, as the number of rounds t approaches infinity, iteration (1) must converge to the average of the numbers $x_1(0), \dots, x_n(0)$. Note that the agents have no control over the communication graph sequence $G(t)$, which is exogenously provided by "nature." However, as we stated previously, every element of the sequence $G(t)$ must be undirected: this corresponds to bidirectional models of communication between agents. Moreover, the sequence $G(t)$ must satisfy the mild connectivity condition of (3), which says that the network cannot become disconnected after a finite period.

Local averaging algorithms are useful tools for information fusion due to their efficient utilization of resources (each agent stores only a single number in memory) as well as their robustness properties (the sequence of graphs $G(t)$ is time-varying, and it only needs to satisfy the relatively weak connectivity condition in (3) for the convergence in (2) to hold). As far as the authors are aware, no other class of schemes for averaging (e.g., flooding, fusion along a spanning tree, etc) is known to produce similar results under the same assumptions.

Remark 1: As can be seen from the subscripts, the update function $f_{i,G(t)}$ is allowed to depend on the agent and on the graph. Some de-

pendence on the graph is unavoidable since in different graphs an agent may have a different number of neighbors, in which case nodes will receive a different number of messages, so that even the number of arguments of $f_{i,G(t)}$ will depend on $G(t)$. It is often practically desired that $f_{i,G(t)}$ depend only weakly on the graph, as the entire graph may be unknown to agent i . For example, we might require that $f_{i,G(t)}$ be completely determined by the degree of i in $G(t)$. However, since our focus is on what distributed algorithms *cannot* do, it does not hurt to assume the agents have unrealistically rich information; thus we will not assume any restrictions on how $f_{i,G(t)}$ depends on $G(t)$.

Remark 2: We require the functions $f_{i,G(t)}$ to be smooth, for the following reason. First, we need to exclude unnatural algorithms that encode vector information in the infinitely many bits of a single real number. Second, although we make the convenient technical assumption that agents can transmit and store real numbers, we must be aware that in practice agents will transmit and store a quantized version of $x_i(t)$. Thus, we are mostly interested in algorithms that are not disrupted much by quantization. For this reason, we must prohibit the agents from using *discontinuous* update functions $f_{i,G(t)}$. For technical reasons, we actually go a little further, and prohibit the agents from using *non-smooth* update functions $f_{i,G(t)}$.

B. Examples

In order to provide some context, let us mention just a few of the distributed averaging schemes that have been proposed in the literature.

- 1) The max-degree method [18] involves picking $\epsilon(t)$ with the property $\epsilon(t) \leq 1/(d(t) + 1)$, where $d(t)$ is the largest degree of any agent in $G(t)$, and updating by

$$x_i(t+1) = x_i(t) + \epsilon(t) \sum_{j \in N_i(t)} (x_j(t) - x_i(t)).$$

Here we use $N_i(t)$ to denote the set of neighbors of agent i in $G(t)$. In practice, a satisfactory $\epsilon(t)$ may not be known to all of the agents, because this requires some global information. However, in some cases a satisfactory choice for $\epsilon(t)$ may be available, for example when an a priori upper bound on $d(t)$ is known.

- 2) The Metropolis method [24] involves setting $\epsilon_{ij}(t)$ to satisfy $\epsilon_{ij}(t) \leq \min(1/(d_i(t) + 1), 1/(d_j(t) + 1))$, where $d_i(t), d_j(t)$ are the degrees of agents i and j in $G(t)$, and updating by

$$x_i(t+1) = x_i(t) + \sum_{j \in N_i(t)} \epsilon_{ij}(t) (x_j(t) - x_i(t)).$$

- 3) The load-balancing algorithm of [17] involves updating by

$$x_i(t+1) = x_i(t) + \sum_{j \in N_i(t)} a_{ij}(t) (x_j(t) - x_i(t))$$

where $a_{ij}(t)$ is determined by the following rule: each agent selects exactly two neighbors, the neighbor with the largest value above its own and with the smallest value below its own. If i, j have both selected each other, then $a_{ij}(t) = 1/3$; else $a_{ij}(t) = 0$. The intuition comes from load-balancing: agents think of $x_i(t)$ as load to be equalized among their neighbors; they try to offload on their lightest neighbor and take from their heaviest neighbor.

We remark that the above load-balancing algorithm is not a “local averaging algorithm” according to our definition because $x_i(t+1)$ does not depend only on $x_i(t)$ and its neighbors; for example, agents i and j may not match up because j has a neighbor k with $x_k(t) > x_j(t)$. By contrast, the max-degree and Metropolis algorithm are indeed “local averaging algorithms.”

For each of the above algorithms, it is known that (2) holds provided the connectivity condition in (3) holds. A proof of this fact for the load-balancing algorithm is implicit in [17], and for the others it follows from the results of [3], [5], [14].

C. Our Contribution

Our goal is to study the worst-case convergence time of local averaging schemes. We will show that these do not have good convergence times, even when the communication graph $G(t)$ is restricted to be the same line graph at every time t . We next give an informal statement of our result.

Let $x(t)$ be the vector in \mathbb{R}^n whose i th component is $x_i(t)$. We define the convergence time $\tau(n, \epsilon)$ of a local averaging algorithm on the sequence $G(t)$ as the time until the square root of the sum of squared errors

$$V(x(t)) = \sum_{i=1}^n \left(x_i(t) - \frac{1}{n} \sum_{j=1}^n x_j(0) \right)^2$$

permanently shrinks by a factor of ϵ , i.e., $V(x(t)) \leq \epsilon^2 V(x(0))$ for all $t \geq \tau(n, \epsilon)$, for all initial vectors $x(0)$; $\tau(n, \epsilon)$ is defined to be the smallest number with this property. We are interested in how $\tau(n, \epsilon)$ scales with n and ϵ .

Currently, the best available upper bound for the convergence time is obtained with the load-balancing algorithm; in [17] it was proven that for any time-varying graph sequence $G(t)$

$$\tau(n, \epsilon) \leq C n^2 B \log \frac{1}{\epsilon}$$

for some absolute constant¹ C , and a parameter B related to the connectivity of the graph sequence $G(t)$. We are primarily interested in whether it is possible to improve the scaling with n to below n^2 . Are there nonlinear update functions $f_{i,G(t)}$ which speed up the convergence time?

Our main result is that the answer to this question is negative within the class of local averaging algorithms, even when the graph sequence $G(t)$ is restricted to be the same line-graph for all times t . For local averaging algorithms in this setting, we prove that

$$\tau(n, \epsilon) \geq c n^2 \log \frac{1}{\epsilon}$$

for some absolute constant c .

II. FORMAL STATEMENT AND PROOF OF MAIN RESULT

We next state our main theorem. The theorem begins by specializing our definition of local averaging algorithms to the case of a fixed line graph, and states a lower bound on the convergence time in this setting.

We will use the notation $\mathbf{1}$ to denote the vector in \mathbb{R}^n whose entries are all ones, and $\mathbf{0}$ to denote the vector whose entries are all 0. The average of the entries of a vector y will be denoted by \bar{y} .

Theorem 1: Let f_1, f_n be two differentiable functions from \mathbb{R}^2 to \mathbb{R} , and let f_2, f_3, \dots, f_{n-1} be differentiable functions from \mathbb{R}^3 to \mathbb{R} . Consider the dynamical system

$$\begin{aligned} x_1(t+1) &= f_1(x_1(t), x_2(t)), \\ x_i(t+1) &= f_i(x_i(t), x_{i-1}(t), x_{i+1}(t)), \quad i = 2, \dots, n-1, \\ x_n(t+1) &= f_n(x_{n-1}(t), x_n(t)). \end{aligned} \quad (4)$$

¹By “absolute constant” we mean that C does not depend on the problem parameters n, B, ϵ .

Suppose that there exists a function $\tau(n, \epsilon)$ such that

$$\frac{\|x(t) - \overline{x(0)\mathbf{1}}\|_2}{\|x(0) - \overline{x(0)\mathbf{1}}\|_2} < \epsilon \quad (5)$$

for all n and $\epsilon > 0$, all $t \geq \tau(n, \epsilon)$, and all initial conditions $x_1(0), \dots, x_n(0)$ for which not all $x_i(0)$ are equal. Then

$$\tau(n, \epsilon) \geq \frac{n^2}{30} \log \frac{1}{\epsilon} \quad (6)$$

for all $\epsilon > 0$ and $n \geq 3$.

Remark 3: The dynamical system described in the theorem statement is simply what a local averaging algorithm looks like on a line graph. The functions f_1, f_n are the update functions at the left and right endpoints of the line (which have only a single neighbor), while the update functions f_2, f_3, \dots, f_{n-1} are the ones used by the middle agents (which have two neighbors).

Remark 4: Theorem 1 provides a lower bound on the worst-case performance of local averaging algorithms on the line graph. Naturally, it also provides a lower bound on the worst-case performance of local averaging algorithms on arbitrary graph sequences drawn from any class of time-varying graph sequences that includes the sequence equal to the line graph at each step.

Remark 5: Our lower bound is tight in the following sense: it is well-known that the max-degree method and the Metropolis method both achieve $O(n^2 \log(n/\epsilon))$ convergence time on the line graph (see [17] for a proof). Thus, up to a logarithmic factor, it is impossible to improve the conclusion of Theorem 1.

Remark 6: Fix some $n \geq 3$. A corollary of our theorem is that there are no local averaging algorithms which compute the average in finitely many steps and whose convergence time can be upper bounded on a ball around the origin. More precisely, there is no local averaging algorithm which, starting from initial conditions $x(0)$ in some ball around the origin, always results in $x(t) = \bar{x}\mathbf{1}$ for all times t larger than some T which is independent of the initial condition. We will sketch a proof of this after proving Theorem 1. By contrast, the existence of such algorithms in slightly different models of agent interactions was demonstrated in [7] and [19].

Proof of Theorem 1

We first briefly sketch the proof strategy. We will begin by noting that $\mathbf{0}$ must be an equilibrium of (4); then, we will argue that an upper bound on the convergence time of (4) would imply a similar convergence time bound on the linearization of (4) around the equilibrium of $\mathbf{0}$. This step will rely on the smoothness of the functions f_1, \dots, f_n to establish the relationship between the convergence time of (4) and its linearization. We will then apply a previous $\Omega(n^2)$ convergence time lower bound for linear schemes, proved by the authors in [21], to conclude the proof.

Let f (without a subscript) be the mapping from \mathbb{R}^n to itself that maps $x(t)$ to $x(t+1)$ according to (4). We assume henceforth that the mapping f satisfies the assumptions of Theorem 1. As an immediate consequence (cf. (5)), we obtain that $x(t)$ always converges to $x(0)\mathbf{1}$.

Lemma 1: $f(a\mathbf{1}) = a\mathbf{1}$, for any $a \in \mathbb{R}$.

Proof: Suppose that $x(0) = a\mathbf{1}$. Then, the initial average is a , so that

$$a\mathbf{1} = \lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} x(t+1) = \lim_{t \rightarrow \infty} f(x(t)).$$

We use the continuity of f to get

$$a\mathbf{1} = f(\lim_{t \rightarrow \infty} x(t)) = f(a\mathbf{1}).$$

For $i, j = 1, \dots, n$, we define $a_{ij} = \partial f_i(0)/\partial x_j$, and the matrix

$$A = f'(\mathbf{0}) = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \cdots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & a_{n,n-1} & a_{nn} \end{pmatrix}.$$

Lemma 2: For any integer $k \geq 1$

$$\lim_{x \rightarrow \mathbf{0}} \frac{\|f^k(x) - A^k x\|_2}{\|x\|_2} = 0$$

where f^k refers to the k -fold composition of f with itself.

Proof: The fact that $f(\mathbf{0}) = \mathbf{0}$ implies by the chain rule that the derivative of f^k at $x = \mathbf{0}$ is A^k . The above equation is a restatement of this fact. ■

Lemma 3: Suppose that $x^T \mathbf{1} = 0$. Then

$$\lim_{m \rightarrow \infty} A^m x = \mathbf{0}.$$

Proof: Let $k = \tau(n, 1/2)$. Since $x^T \mathbf{1} = 0$, (5) implies that $\|f^k(x)\|_2 \leq \|x\|_2/2$. By Lemma 2, there exists a ball \mathcal{B} around the origin such that for all $x \in \mathcal{B}$, with $x \neq \mathbf{0}$, we have

$$\frac{\|f^k(x) - A^k x\|_2}{\|x\|_2} \leq \frac{1}{4}.$$

Since we can scale x without affecting the assumptions or conclusions of the lemma we are trying to prove, we can assume that $x \in \mathcal{B}$. It follows that:

$$\begin{aligned} \frac{\|A^k x\|_2}{\|x\|_2} &= \frac{\|A^k x - f^k(x) + f^k(x)\|_2}{\|x\|_2} \\ &\leq \frac{1}{4} + \frac{\|f^k(x)\|_2}{\|x\|_2} \\ &\leq \frac{1}{4} + \frac{1}{2} \\ &\leq \frac{3}{4}. \end{aligned}$$

Since this inequality implies that $A^k x \in \mathcal{B}$, we can apply the same argument recursively to get

$$\lim_{m \rightarrow \infty} (A^k)^m x = \mathbf{0}$$

which implies the conclusion of the lemma. ■

Lemma 4: $A\mathbf{1} = \mathbf{1}$.

Proof: We have

$$A\mathbf{1} = \lim_{h \rightarrow 0} \frac{f(\mathbf{0} + h\mathbf{1}) - f(\mathbf{0})}{h} = \lim_{h \rightarrow 0} \frac{h\mathbf{1}}{h} = \mathbf{1}$$

where we used Lemma 1. ■

Lemma 5: For every vector $x \in \mathbb{R}^n$,

$$\lim_{k \rightarrow \infty} A^k x = \bar{x}\mathbf{1}$$

where $\bar{x} = (\sum_{i=1}^n x_i)/n$.

Proof: Every vector x can be written as

$$x = \bar{x}\mathbf{1} + y$$

where $y^T \mathbf{1} = 0$. Thus

$$\lim_{k \rightarrow \infty} A^k x = \lim_{k \rightarrow \infty} A^k (\bar{x} \mathbf{1} + y) = \bar{x} \mathbf{1} + \lim_{k \rightarrow \infty} A^k y = \bar{x} \mathbf{1}$$

where we used Lemmas 3 and 4. ■

Lemma 6: The matrix A has the following properties:

- 1) $a_{ij} = 0$ whenever $|i - j| > 1$.
- 2) The directed graph $G' = (\{1, \dots, n\}, E')$, with $E' = \{(i, j) \mid a_{ij} \neq 0\}$, is strongly connected.
- 3) $A \mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T A = \mathbf{1}^T$.
- 4) An eigenvalue of A of largest modulus has modulus 1.
- 5) A has an eigenvector v , with real eigenvalue $\lambda \in (1 - 6/n^2, 1)$, such that $v^T \mathbf{1} = 0$.

Proof:

- 1) True because of the definitions of f and A .
- 2) Suppose not. Then, there is a nonempty set $S \subset \{1, \dots, n\}$ with the property that $a_{ij} = 0$ whenever $i \in S$ and $j \in S^c$. Consider the vector x with $x_i = 0$ for $i \in S$, and $x_j = 1$ for $j \in S^c$. Clearly, $(1/n) \sum_i x_i > 0$, but $(A^k x)_i = 0$ for $i \in S$. This contradicts Lemma 5. We note that we cannot begin by assuming the graph the graph G' is undirected; however, this is true, and a consequence of this lemma, as we will remark below.
- 3) The first equality was already proven in Lemma 4. For the second, let $b = \mathbf{1}^T A$. Consider the vector

$$z = \lim_{k \rightarrow \infty} A^k e_i \tag{7}$$

where e_i is the i th unit vector. By Lemma 5

$$z = \frac{\mathbf{1}^T e_i}{n} \mathbf{1} = \frac{1}{n} \mathbf{1}.$$

On the other hand

$$\lim_{k \rightarrow \infty} A^k e_i = \lim_{k \rightarrow \infty} A^{k+1} e_i = \lim_{k \rightarrow \infty} A^k (A e_i).$$

Applying Lemma 5 again, we get

$$z = \frac{\mathbf{1}^T (A e_i)}{n} \mathbf{1} = \frac{b_i}{n} \mathbf{1}$$

where b_i is the i th component of b . We conclude that $b_i = 1$; since no assumption was made on i , this implies that $b = \mathbf{1}$, which is what we needed to show.

- 4) We already know that $A \mathbf{1} = \mathbf{1}$, so that an eigenvalue with modulus 1 exists. Now suppose there is an eigenvalue with larger modulus, that is, there is some vector $x \in \mathbb{C}^n$ such that $Ax = \lambda x$ and $|\lambda| > 1$. Then, $\lim_k \|A^k x\|_2 = \infty$. By writing $x = x_{\text{real}} + i x_{\text{imaginary}}$, we immediately have that $A^k x = A^k x_{\text{real}} + i A^k x_{\text{imaginary}}$. But by Lemma 5 both $A^k x_{\text{real}}$ and $A^k x_{\text{imaginary}}$ approach some finite multiple of $\mathbf{1}$ as $k \rightarrow \infty$, so $\|A^k x\|_2$ is bounded above. This is a contradiction.
- 5) The following fact is a combination of Theorems 4.1 and 6.1 in [21]: Consider an $n \times n$ matrix A such that $a_{ij} = 0$ whenever $|i - j| > 1$, and such that the graph with edge set $\{(i, j) \mid a_{ij} \neq 0\}$ is connected. Let $\lambda_1, \lambda_2, \dots$ be its eigenvalues in order of decreasing modulus. Suppose that $\lambda_1 = 1$, $A \mathbf{1} = \mathbf{1}$, and $\pi^T A = \pi^T$, for some vector π satisfying $\sum_i \pi_i = 1$, and $\pi_i \geq 1/(Cn)$ for some positive C and for all i . Then, A has a real eigenvalue in $(1 - 6C/n^2, 1)$. Furthermore, the corresponding right eigen-

²The reference [21] proves that an eigenvalue lies in $(1 - c_1 C/n^2, 1)$ for some absolute constant c_1 . By tracing through the proof, we find that we can take $c_1 = 6$.

vector is orthogonal to $\mathbf{1}$, since right-eigenvectors of a matrix are orthogonal to left-eigenvectors with different eigenvalues.

By parts 1–4, all the assumptions of the result from [21] are satisfied with $\pi = \mathbf{1}/n$ and $C = 1$, thus completing the proof of the lemma. ■

Remark 7: An alternative proof of part 5 is possible. One can argue that parts 1 and 3 force A to be symmetric, and that Lemma 5 implies that the elements a_{ij} must be nonnegative. Once these two facts are established, the results of [4] imply that an eigenvalue has to lie in $(1 - c/n^2, 1)$, for a certain absolute constant c .

Proof of Theorem 1: Let v be an eigenvector of A with the properties in part 5 of Lemma 6. Fix a positive integer k . For any $\epsilon^* > 0$, we can pick $x \neq 0$ to be a small enough multiple of v so that

$$\frac{\|f^k(x) - A^k x\|_2}{\|x\|_2} \leq \epsilon^*.$$

This is possible by Lemma 2. Then, we have

$$\frac{\|f^k(x)\|_2}{\|x\|_2} \geq \frac{\|A^k x\|_2}{\|x\|_2} - \epsilon^* \geq \left(1 - \frac{6}{n^2}\right)^k - \epsilon^*.$$

Using the orthogonality property $x^T \mathbf{1} = 0$, we have $\bar{x} = 0$, and

$$\sup_{\substack{y \in \mathbb{R}^n \\ y \perp \text{span}\{\mathbf{1}\}}} \frac{\|f^k(y) - \bar{y} \mathbf{1}\|_2}{\|y - \bar{y} \mathbf{1}\|_2} \geq \frac{\|f^k(x)\|_2}{\|x\|_2} \geq \left(1 - \frac{6}{n^2}\right)^k - \epsilon^*.$$

Now at time $k = \tau(n, \epsilon)$, the left-hand side is at most ϵ , by the definition of $\tau(n, \epsilon)$, so that

$$\epsilon \geq \left(1 - \frac{6}{n^2}\right)^{\tau(n, \epsilon)} - \epsilon^*$$

and letting $\epsilon^* \rightarrow 0$, we obtain

$$\epsilon \geq \left(1 - \frac{6}{n^2}\right)^{\tau(n, \epsilon)}.$$

Since $n \geq 3$, we have $1 - 6/n^2 \in (0, 1)$, and

$$\tau(n, \epsilon) \geq \frac{1}{\log\left(1 - \frac{6}{n^2}\right)} \log \epsilon.$$

Now using the bound $\log(1 - \alpha) \geq 5(\alpha - 1)$ for $\alpha \in [0, 2/3]$, we get

$$\tau(n, \epsilon) \geq \frac{n^2}{30} \log \frac{1}{\epsilon}.$$

Remark 8: We now sketch the proof of the claim we made earlier that a local averaging algorithm cannot average in finitely many steps and have a convergence time which is uniformly bounded over a ball around the origin. Fix $n \geq 3$. Suppose that for any $x(0)$ in some ball \mathcal{B} around the origin, a local averaging algorithm results in $x(t) = \bar{x} \mathbf{1}$ for all $t \geq T$, where T is independent of $x(0)$.

The proof of Theorem 1 shows that given any $k, \epsilon > 0$, one can pick a vector $v(\epsilon)$ so that if $x(0) = v(\epsilon)$ then $V(x(k))/V(x(0)) \geq \left((1 - 6/n^2)^k - \epsilon\right)^2$. Moreover, the vectors $v(\epsilon)$ can be chosen to be arbitrarily small. One simply picks $k = T$ and $\epsilon < (1 - 6/n^2)^k$ to get that $x(T)$ is not a multiple of $\mathbf{1}$; and furthermore, picking $v(\epsilon)$ small enough in norm so that it lies in \mathcal{B} results in a contradiction. ■

Remark 9: Theorem 1 gives a lower bound on how long we must wait for the 2-norm $\|x(t) - \bar{x}\mathbf{1}\|_2$ to shrink by a factor of ϵ . What if we replace the 2-norm with other norms, for example with the ∞ -norm? Since $\mathcal{B}_\infty(\mathbf{0}, r/\sqrt{n}) \subset \mathcal{B}_2(\mathbf{0}, r) \subset \mathcal{B}_\infty(\mathbf{0}, r)$, it follows that if the ∞ -norm shrinks by a factor of ϵ , then the 2-norm must shrink by at least $\sqrt{n}\epsilon$. Since ϵ only enters the lower bound of Theorem 1 logarithmically, the answer only changes by a factor of $\log n$ when passing to the ∞ -norm. Similarly, since for any $p > p' \geq 1$, $\mathcal{B}_p(\mathbf{0}, r/n^{1/p'-1/p}) \subset \mathcal{B}_{p'}(\mathbf{0}, r) \subset \mathcal{B}_p(\mathbf{0}, r)$, the same argument shows that, modulo some logarithmic factors, we can replace $p = 2$ with any other p -norm.

III. CONCLUSION

We have proved a lower bound on the convergence time of local averaging algorithms which scales quadratically in the number of agents. This lower bound holds even if all the communication graphs are equal to a fixed line graph. Our work points to a number of open questions.

- 1) Is it possible to loosen the definition of local averaging algorithms to encompass a wider class of algorithms? In particular, is it possible to weaken the requirement that each $f_{i,G(t)}$ be smooth, perhaps only to the requirement that it be piecewise-smooth or continuous, and still obtain a $\Omega(n^2)$ lower bound? What are the limitations of algorithms that allow the update functions $f_{i,G(t)}$ to depend explicitly on time, even if $G(t)$ does not change?
- 2) Does the worst-case convergence time change if we introduce some memory and allow $x_i(t+1)$ to depend on the last k sets of messages received by agent i ? Alternatively, there is the broader question of how much is there to be gained if every agent is allowed to keep track of extra variables. Some positive results in this direction were obtained in [11].
- 3) What if each node maintains a small number of update functions, and is allowed to choose which of them to apply based on the messages received? Our lower bound does not apply to such schemes, so it is an open question whether its possible to design practical algorithms along these lines with worst-case convergence time scaling better than n^2 .

REFERENCES

- [1] M. Alighanbari and J. P. How, "Unbiased Kalman consensus algorithm," in *Proc. Amer. Control Conf.*, Minneapolis, MN, Jun. 14–16, 2006, pp. 3519–3524.
- [2] L. Brunet, H. L. Choi, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 912–926, Aug. 2009.
- [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. (CDC-ECC'05)*, Seville, Spain, Dec. 2005, pp. 2996–3000.
- [4] S. Boyd, P. Diaconis, J. Sun, and L. Xiao, "Fastest mixing Markov chain on a path," *Amer. Math. Monthly*, vol. 113, no. 1, pp. 70–74, Jan. 2006.
- [5] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: Convergence rates, measurement delays, and asynchronous events," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 601–623, 2008.
- [6] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE J. Selected Areas Commun.*, vol. 26, no. 4, pp. 622–633, Apr. 2008.
- [7] J. Cortes, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.
- [8] C. Gao, J. Cortes, and F. Bullo, "Notes on averaging over acyclic digraphs and discrete coverage control," *Automatica*, vol. 44, no. 8, pp. 2120–2127, 2008.
- [9] N. Hayashi and T. Ushio, "Application of a consensus problem to fair multi-resource allocation in real-time systems," in *Proc. 47th IEEE Conf. Decision Control*, Cancun, Mexico, 2008, pp. 2450–2455.
- [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] K. Jung, D. Shah, and J. Shin, "Distributed averaging via lifted Markov chains," *Cornell Univ. Library* pp. 1–28, 2008 [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/0908/0908.4073v1.pdf
- [12] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.
- [13] Q. Li and D. Rus, "Global clock synchronization for sensor networks," in *Proc. 23rd Conf. IEEE Commun. Soc. (INFOCOM'04)*, Hong Kong, Mar. 2004, pp. 564–574.
- [14] S. Li and H. Wang, "Multi-Agent Coordination using Nearest-Neighbor Rules: Revisiting the Vicsek Model," Tech. Rep., 2004 [Online]. Available: <http://arxiv.org/abs/cs.MA/0407021>
- [15] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.
- [16] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.
- [17] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2506–2517, Nov. 2009.
- [18] 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.
- [19] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proc. Amer. Control Conf.*, New York, Jul. 2007, pp. 650–660.
- [20] M. Schwager, J.-J. Slotine, and D. Rus, "Consensus learning for distributed coverage control," in *Proc. Int. Conf. Robot. Autom.*, Pasadena, CA, May 2008, pp. 1042–1048.
- [21] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, 2009.
- [22] 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.
- [23] F. Wuhid, R. Stadler, and M. Dam, "Gossiping for threshold detection," in *Proc. 11th IFIP/IEEE Int. Conf. Symp. Integr. Netw. Manag.*, 2009, pp. 259–266.
- [24] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.
- [25] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. Inform. Processing Sensor Netw.*, Los Angeles, CA, Apr. 2005, pp. 63–70.
- [26] M. Zhu and S. Martinez, "On the convergence time of asynchronous distributed quantized averaging algorithms," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 386–390, Feb. 2011.